Руководство FreeBSD

Аннотация

Добро пожаловать в FreeBSD! Это руководство охватывает установку и повседневное использование FreeBSD 14.2-RELEASE и 13.4-RELEASE. Эта книга является результатом совместной работы многих людей. Некоторые разделы могут быть устаревшими. Те, кто хочет помочь обновить и расширить этот документ, могут отправить электронное письмо на Список рассылки Проекта Документации FreeBSD.

Последняя версия этой книги доступна на веб-сайте FreeBSD. Предыдущие версии можно получить по адресу https://docs.FreeBSD.org/doc/. Книгу можно скачать в различных форматах и с различными опциями сжатия с сервера загрузки FreeBSD или с одного из многочисленных зеркальных сайтов. Поиск можно выполнить по этому руководству и другим документам на странице поиска.

Содержание

Предисловие . . . 10
- Целевая аудитория . . . 10
- Четвёртое издание . . . 10
- Третье издание . . . 10
- Второе издание (2004) . . . 11
- Первое издание (2001) . . . 11
- Структура этой книги . . . 12
- Договоренности, используемые в этой книге . . . 16
- Благодарности . . . 17

I: В начале . . . 19
- 1. Введение . . . 20
  - 1.1. Краткий обзор . . . 20
  - 1.2. Добро пожаловать во FreeBSD! . . . 20
  - 1.3. О Проекте FreeBSD . . . 23
- 2. Установка FreeBSD версий 9.X и более поздних . . . 29
  - 2.1. Краткий обзор . . . 29
  - 2.2. Аппаратные требования . . . 29
  - 2.3. Перед установкой . . . 30
  - 2.4. Начало установки . . . 35
  - 2.5. Введение в bsdinstall . . . 42
  - 2.6. Установка по сети . . . 47
  - 2.7. Выделение дискового пространства . . . 49
  - 2.8. Завершение установки . . . 58
  - 2.9. После установки . . . 61
  - 2.10. Решение проблем . . . 90
  - 2.11. Использование Live CD . . . 92
- 3. Основы FreeBSD . . . 93
  - 3.1. Краткий обзор . . . 93
  - 3.2. Виртуальные консоли и терминалы . . . 93
  - 3.3. Пользователи и основы управления учётными записями . . . 96
  - 3.4. Права доступа . . . 107
  - 3.5. Структура каталогов . . . 113
  - 3.6. Организация дисков . . . 115
  - 3.7. Монтирование и размонтирование файловых систем . . . 122
  - 3.8. Процессы и даемоны . . . 125
  - 3.9. Командные процессоры . . . 129
  - 3.10. Текстовые редакторы . . . 132
  - 3.11. Устройства и файлы устройств . . . 133

4. Установка приложений: порты и пакеты . . . 136
4.1. Обзор . . . 136
4.2. Обзор установки программного обеспечения . . . 136
4.3. Поиск программного обеспечения . . . 138
4.4. Использование бинарных пакетов . . . 140
4.5. Использование pkgng для управления бинарными пакетами . . . 143
4.6. Использование Коллекции Портов . . . 149
4.7. Действия после установки . . . 160
4.8. Обработка нерабочих портов . . . 161
5. Το Σύστημα X Window . . . 163
5.1. Обзор . . . 163
5.2. Основы X . . . 163
5.3. Установка X11 . . . 166
5.4. Конфигурация X11 . . . 167
5.5. Использование шрифтов в X11 . . . 174
5.6. Менеджеры экранов (Display Managers) X . . . 179
5.7. Графические оболочки . . . 182
II: Общие задачи . . . 187
6. Приложения для настольного компьютера . . . 188
6.1. Краткий обзор . . . 188
6.2. Браузеры . . . 189
6.3. Бизнес приложения . . . 192
6.4. Программы просмотра документов . . . 195
6.5. Финансовые программы . . . 197
6.6. Итоги . . . 199
7. Мультимедиа . . . 200
7.1. Краткий обзор . . . 200
7.2. Настройка звуковой карты . . . 201
7.3. Звук MP3 . . . 205
7.4. Воспроизведение видео . . . 208
7.5. Настройка ТВ тюнеров . . . 217
7.6. Сканеры . . . 219
8. Настройка ядра FreeBSD . . . 225
8.1. Краткий обзор . . . 225
8.2. Зачем собирать собственное ядро? . . . 225
8.3. Определение аппаратного обеспечения . . . 226
8.4. Драйвера, подсистемы и модули ядра . . . 227
8.5. Сборка и установка собственного ядра . . . 228
8.6. Конфигурационный файл . . . 231
8.7. Решение проблем . . . 248
9. Печать . . . 250

9.1. Краткий обзор . . . 250
9.2. Введение . . . 250
9.3. Основная настройка . . . 251
9.4. Расширенная настройка принтера . . . 267
9.5. Использование принтеров . . . 300
9.6. Альтернативы стандартному спулеру . . . 308
9.7. Выявление проблем . . . 309
10. Двоичная совместимость с Linux . . . 314
10.1. Краткий обзор . . . 314
10.2. Установка . . . 314
10.3. Установка Mathematica® . . . 318
10.4. Установка Maple™ . . . 320
10.5. Установка MATLAB® . . . 323
10.6. Установка Oracle® . . . 326
10.7. Установка SAP® R/3® . . . 329
10.8. Дополнительные сведения . . . 353
III: Системное администрирование . . . 355
11. Настройка и оптимизация . . . 356
11.1. Введение . . . 356
11.2. Начальное конфигурирование . . . 356
11.3. Основные настройки . . . 358
11.4. Настройка приложений . . . 358
11.5. Запуск сервисов . . . 359
11.6. Настройка утилиты `cron` . . . 362
11.7. Использование rc во FreeBSD 5.X и последующих версиях . . . 364
11.8. Настройка карт сетевых интерфейсов . . . 366
11.9. Настройка виртуальных серверов . . . 370
11.10. Файлы настройки . . . 371
11.11. Настройка с помощью sysctl . . . 376
11.12. Оптимизация дисков . . . 377
11.13. Изменение ограничений, накладываемых ядром . . . 381
11.14. Увеличение объема подкачки . . . 384
11.15. Управление питанием и ресурсами . . . 386
11.16. Использование и отладка FreeBSD ACPI . . . 387
12. Процесс загрузки FreeBSD . . . 395
12.1. Описание . . . 395
12.2. Проблема загрузки . . . 395
12.3. Менеджер загрузки и этапы загрузки . . . 396
12.4. Взаимодействие с ядром во время загрузки . . . 403
12.5. Хинты устройств . . . 404
12.6. Init: инициализация управления процессами . . . 405

12.7. Процесс остановки системы . . . 406
13. Безопасность . . . 408
13.1. Краткое описание . . . 408
13.2. Введение . . . 408
13.3. Защита FreeBSD . . . 410
13.4. DES, MD5, и шифрование . . . 419
13.5. Одноразовые пароли . . . 420
13.6. TCP Wrappers . . . 424
13.7. KerberosIV . . . 427
13.8. Kerberos5 . . . 435
13.9. OpenSSL . . . 444
13.10. VPN через IPsec . . . 447
13.11. OpenSSH . . . 460
13.12. Списки контроля доступа файловой системы (ACL) . . . 466
13.13. Мониторинг вопросов безопасности в ПО сторонних разработчиков . . . 468
13.14. Сообщения безопасности FreeBSD . . . 469
13.15. Учёт используемых ресурсов . . . 471
14. Принудительный контроль доступа (MAC) . . . 473
14.1. Краткий обзор . . . 473
14.2. Ключевые термины этой главы . . . 474
14.3. Описание MAC . . . 475
14.4. Метки MAC . . . 477
14.5. Настройка модулей . . . 482
14.6. Модуль MAC bsdextended . . . 483
14.7. Модуль MAC ifoff . . . 484
14.8. Модуль MAC portacl . . . 485
14.9. Политики MAC, использующие метки . . . 486
14.10. Модуль MAC partition . . . 488
14.11. Модуль многоуровневой безопасности MAC (MLS) . . . 489
14.12. Модуль MAC Biba . . . 490
14.13. Модуль MAC LOMAC . . . 492
14.14. Реализация защищенной среды с MAC . . . 492
14.15. Другой пример: Использование MAC для защиты веб сервера . . . 498
14.16. Решение проблем с инфраструктурой MAC . . . 500
15. Аудит событий безопасности . . . 502
15.1. Краткий обзор . . . 502
15.2. Ключевые понятия . . . 503
15.3. Настройка системы аудита . . . 503
15.4. Работа с журналами аудита . . . 508
16. Устройства хранения . . . 512
16.1. Краткий обзор . . . 512

16.2. Имена устройств . . . 512
16.3. Добавление дисков . . . 513
16.4. RAID . . . 516
16.5. USB устройства хранения . . . 521
16.6. Запись и использование оптических носителей (CD) . . . 523
16.7. Создание и использование оптических носителей (DVD) . . . 531
16.8. Дискеты . . . 537
16.9. Создание и использование архивных копий на магнитной ленте . . . 539
16.10. Создание резервных копий на дискетах . . . 542
16.11. Стратегии резервного копирования . . . 544
16.12. Основы технологии резервного копирования . . . 545
16.13. Сетевые файловые системы, файловые системы в памяти и с отображением в файл . . . 549
16.14. Мгновенные копии файловых систем . . . 554
16.15. Квотирование файловых систем . . . 555
16.16. Шифрование дисковых разделов . . . 560
16.17. Шифрование области подкачки . . . 568
17. GEOM: Модульная инфраструктура преобразования дисковых запросов . . . 570
17.1. Краткий обзор . . . 570
17.2. Введение в GEOM . . . 570
17.3. RAID0 - Создание дисковой последовательности (Striping) . . . 570
17.4. RAID1 - Зеркалирование (Mirroring) . . . 572
17.5. Сетевые устройства GEOM Gate . . . 576
17.6. Метки дисковых устройств . . . 577
17.7. Журналирование UFS средствами GEOM . . . 580
18. Поддержка файловых систем . . . 583
18.1. Краткий обзор . . . 583
18.2. Файловая система ZFS . . . 583
19. Локализация - использование и настройка i18n/L10n . . . 593
19.1. Краткий обзор . . . 593
19.2. Использование локализации . . . 593
19.3. Поиск приложений i18n . . . 600
19.4. Настройка локализации для некоторых языков . . . 600
20. Обновление системы и смена версии FreeBSD . . . 604
20.1. Краткий обзор . . . 604
20.2. Обновление FreeBSD . . . 604
20.3. Обновление документации . . . 613
20.4. Использование ветви разработки . . . 616
20.5. Синхронизация исходных текстов . . . 619
20.6. Пересборка мира . . . 619
20.7. Отслеживание исходных текстов для нескольких машин . . . 631

IV: Сетевые коммуникации . . . 633
21. Последовательные соединения . . . 634
21.1. Краткое описание . . . 634
21.2. Введение . . . 634
21.3. Терминалы . . . 639
21.4. Входящие соединения по модему . . . 644
21.5. Исходящие соединения по модему . . . 653
21.6. Настройка последовательной консоли . . . 657
22. PPP и SLIP . . . 667
22.1. Краткий обзор . . . 667
22.2. PPP уровня пользователя . . . 667
22.3. PPP уровня ядра . . . 681
22.4. Решение проблем с соединениями PPP . . . 689
22.5. Использование PPP через Ethernet (PPPoE) . . . 693
22.6. Использование PPP через ATM (PPPoA) . . . 695
22.7. Использование SLIP . . . 698
23. Электронная почта . . . 710
23.1. Краткий обзор . . . 710
23.2. Использование электронной почты . . . 710
23.3. Настройка sendmail . . . 713
23.4. Установка другой почтовой программы . . . 716
23.5. Поиск и устранение неисправностей . . . 718
23.6. Расширенное руководство . . . 722
23.7. Настройка почты только для отправки . . . 724
23.8. Использование почты с коммутируемым соединением . . . 725
23.9. SMTP аутентификация . . . 726
23.10. Почтовые программы пользователей . . . 728
23.11. Использование fetchmail . . . 736
23.12. Использование procmail . . . 737
24. Сетевые серверы . . . 739
24.1. Краткий обзор . . . 739
24.2. "Супер-сервер"inetd . . . 739
24.3. Network File System (NFS) . . . 744
24.4. Network Information System (NIS/YP) . . . 752
24.5. Автоматическая настройка сети (DHCP) . . . 771
24.6. Domain Name System (DNS) . . . 777
24.7. Apache HTTP сервер . . . 791
24.8. Файл сервер и печать для Microsoft® Windows® клиентов (Samba) . . . 797
24.9. Протокол передачи файлов (FTP) . . . 800
24.10. Синхронизация часов через NTP . . . 801
24.11. * Remote Host Logging with `syslogd` . . . 805

25. Межсетевые экраны . . . 806
25.1. Введение . . . 806
25.2. Принципы работы межсетевых экранов . . . 806
25.3. Пакеты межсетевых экранов . . . 807
25.4. Packet Filter (PF, межсетевой экран OpenBSD) и ALTQ . . . 808
25.5. * IPFILTER (IPF) . . . 811
25.6. IPFW . . . 820
26. Сложные вопросы работы в сети . . . 841
26.1. Краткий обзор . . . 841
26.2. Сетевые шлюзы и маршруты . . . 841
26.3. Беспроводные сети . . . 849
26.4. Bluetooth . . . 857
26.5. Мосты . . . 867
26.6. Работа с бездисковыми станциями . . . 870
26.7. ISDN . . . 877
26.8. Даемон преобразования сетевых адресов (natd) . . . 882
26.9. IP по параллельному порту (PLIP) . . . 886
26.10. IPv6 . . . 888
26.11. Асинхронный режим передачи (ATM) . . . 894
V: Приложения . . . 897
Приложение A: Получение FreeBSD . . . 898
A.1. Наборы CD и DVD . . . 898
A.2. FTP сайты . . . 898
A.3. Использование CTM . . . 905
A.4. Использование Subversion . . . 908
A.5. Использование rsync . . . 911
Приложение B: Библиография . . . 914
B.1. Книги и журналы, специализирующиеся на FreeBSD . . . 914
B.2. Руководства для пользователей . . . 915
B.3. Руководства для администраторов . . . 915
B.4. Руководства для программистов . . . 916
B.5. Внутренности операционной системы . . . 917
B.6. Безопасность . . . 917
B.7. Оборудование . . . 917
B.8. История UNIX® . . . 918
B.9. Прочие издания . . . 918
Приложение C: Ресурсы в интернет . . . 919
C.1. Списки рассылки . . . 919
C.2. Новостные группы Usenet . . . 938
C.3. Серверы World Wide Web . . . 939
C.4. Адреса Email . . . 942

Приложение D: PGP ключи 943
D.1. Офицеры 943

Предисловие

Целевая аудитория

Начинающим пользователям FreeBSD будет полезна первая часть этой книги, которая проводит читателя через процесс установки FreeBSD и ненавязчиво объясняет концепции и соглашения, на которых базируется UNIX®. Работа с этой частью требует несколько большего, чем желания исследовать и способности принятия новых концепций по мере их описания.

Если вы преодолеете все сложности, то вас будет ждать вторая, много большая часть Руководства, которая является всеобъемлющим справочником обо всех темах, которые могут интересовать администраторов FreeBSD. Некоторые из глав этой части могут рекомендовать вам предварительное изучение других материалов, о чём упоминается в кратком обзоре в начале каждой главы.

Список рекомендуемой дополнительной литературы вы можете найти в Библиографии.

Четвёртое издание

Актуальная версия Руководства является результатом усилий рабочей группы, которая занималась рецензированием и обновлением содержимого документа. Вот те ключевые изменения, появившиеся после выхода четвёртого издания Руководства:

- Выполнен перевод формата Руководства с Docbook на Hugo и AsciiDoctor
- Создан Портал документации FreeBSD.
- Добавлена глава Wayland, описывающая установку и настройку Wayland во FreeBSD.
- Существенно дополнена Библиография.

Третье издание

Текущая онлайновая версия Руководства является результатом совместных усилий многих сотен добровольных участников за последние 10 лет. Ниже перечисляются некоторые значимые изменения, появившиеся после публикации двухтомного третьего издания в 2004 году:

- Добавлена глава WINE, описывающая порядок запуска приложений Windows® во FreeBSD.
- Добавлена глава DTrace, содержащая информацию о мощном инструменте анализа производительности DTrace.
- Добавлена глава Другие файловые системы, содержащая информацию о поддержке прочих файловых систем во FreeBSD, таких как ZFS от компании Sun™.
- Добавлена глава Аудит Событий Безопасности, посвящённая новым возможностям аудита во FreeBSD и описывающая его использование.

- Добавлена глава Виртуализация с информацией об установке FreeBSD в виртуализированном программном окружении.
- Добавлена глава Установка FreeBSD, описывающая установку FreeBSD при помощи новой установочной утилиты bsdinstall.

Второе издание (2004)

Третье Издание являлось кульминацией более чем двух лет работы отдельных членов проекта документации FreeBSD. Печатная версия выросла до размера, потребовавшего публикации в виде двух отдельных томов. Вот основные изменения в новой редакции:

- Глава Настройка и оптимизация была расширена новой информацией о стандарте ACPI управления электропитанием и ресурсами, системной утилите `cron` и дополнительных параметрах оптимизации ядра.
- Глава Безопасность была дополнена новой информацией о виртуальных частных сетях (VPN), списках контроля доступа файловой системы (ACL) и бюллетенях безопасности.
- Глава Принудительный контроль доступа (MAC) является новой главой этого Издания. Она описывает, что такое MAC и как этот механизм может быть использован для защиты системы FreeBSD.
- Глава Устройства хранения была расширена новой информацией об устройствах хранения USB, образах файловой системы, квотах файловой системы, файловых системах в файлах и в сети, а также зашифрованных дисковых разделах.
- К главе PPP и SLIP был добавлен раздел об устранении неполадок.
- Глава Электронная почта была дополнена новой информацией об использовании альтернативных транспортных агентов, аутентификации SMTP, UUCP, fetchmail, procmail и другими темами повышенной сложности.
- Глава Сетевые серверы появилась в этой редакции. В неё включена информация о настройке Apache HTTP Server и ftpd, а также настройке сервера Samba для работы с клиентами Microsoft® Windows®. Некоторые разделы были перемещены сюда из главы Сложные вопросы работы в сети.
- Глава Сложные вопросы работы в сети была расширена новой информацией об использовании устройств Bluetooth® во FreeBSD, настройке беспроводных сетей, и сетях Asynchronous Transfer Mode (ATM).
- Был добавлен глоссарий, объединяющий информацию о технических терминах, используемых в книге.
- Множество эстетических улучшений были внесены в таблицы и иллюстрации этой книги.

Первое издание (2001)

Вторая редакция стала кульминацией более чем двухлетней работы членов Проекта документации FreeBSD. Нижеследующий список перечисляет все значительные изменения, внесённые в эту редакцию:

- Был добавлен полный указатель тем.
- Все ASCII-иллюстрации были заменены на графические.
- Был добавлен стандартный краткий обзор к каждому разделу для того, чтобы читатель мог получить представление о содержании раздела и о том, что необходимо знать для его изучения.
- Содержимое было логически реорганизовано на три части: "В Начале", "Системное администрирование" и "Приложения".
- Глава Основы FreeBSD была расширена и содержит дополнительную информацию о процессах, даемонах и сигналах.
- Глава Установка приложений была расширена и содержит дополнительную информации об управлении бинарными пакетами.
- Глава X Window System была полностью переписана и обращает больше внимания на современные технологии для рабочего стола, такие, как KDE и GNOME на XFree86™ 4.X.
- Глава Процесс загрузки FreeBSD была расширена.
- Глава Устройства хранения была составлена из того, что раньше было двумя различными главами: "Диски" и "Резервное копирование". Мы считаем, что данные темы будут проще и более полно описаны как одна глава. Был добавлен раздел о программном и аппаратном RAID.
- Глава Последовательные соединения была полностью реорганизована и обновлена для FreeBSD 4.X/5.X.
- Глава PPP и SLIP была существенно обновлена.
- Было добавлено множество новых разделов в главу Сложные вопросы работы в сети.
- Глава Электронная почта была расширена, теперь она включает больше информации о настройке sendmail.
- Глава Двоичная совместимость с Linux была дополнена включением информации об установке Oracle® и SAP® R/3®.
- Следующие новые темы были рассмотрены в этой, второй, редакции:
  - Настройка и оптимизация.
  - Мультимедиа.

Структура этой книги

Эта книга разбита на пять частей. В первой части, *В начале*, рассматривается установка и основные навыки использования FreeBSD. Предполагается, что читатель освоит эти разделы последовательно, возможно пропуская разделы, в которых обсуждаются уже знакомые для него темы. Вторая часть, *Общие задачи*, рассказывает о некоторых наиболее часто используемых возможностях FreeBSD. Этот раздел и все последующие могут быть прочитаны не по порядку. Каждая глава начинается с краткого обзора, который описывает, о чём говорится в ней и что читатель должен будет знать для прочтения этой главы. Это сделано для того, чтобы случайно встретивший этот материал читатель мог найти разделы, которые его интересуют. В третьей части, *Системное администрирование*, рассмотрены

вопросы администрирования. В четвертой части, *Сетевые коммуникации*, охвачены темы, связанные с серверами и сетью. Пятая часть содержит приложения и справочную информацию.

Введение

Знакомит пользователя с FreeBSD. Рассказывает об истории проекта FreeBSD, его задачах и модели разработки.

Установка FreeBSD

Проводит пользователя через весь процесс установки FreeBSD 9.x и более поздних версий с использованием sysinstall.

Основы FreeBSD

Рассказывает об основных командах и функциональности операционной системы FreeBSD. Если вы знакомы с Linux® или другой UNIX®-подобной операционной системой, возможно, вы пропустите эту главу.

Установка приложений

Рассказывает о процессе установки программного обеспечения сторонних производителей с использованием "Коллекции Портов FreeBSD" и стандартных бинарных пакетов.

X Window System

Описывает X Window System вообще и использование X11 под управлением FreeBSD в частности. Также описывает популярные окружения рабочего стола, такие как KDE и GNOME.

Wayland

Описывает сервер дисплеев Wayland в целом и использование Wayland во FreeBSD в частности. Также рассказывает о популярных композитных менеджерах, таких как Wayfire, Hikari и Sway.

Приложения для настольного компьютера

Перечисляет некоторые популярные приложения для рабочей станции, такие как веб-браузеры и офисные пакеты и описывает процесс их установки на FreeBSD.

Мультимедиа

Показывает, как настроить поддержку воспроизведения звука и видео на вашей системе. Также описывает некоторые примеры приложений для воспроизведения звука и видео.

Настройка ядра FreeBSD

Объясняет, почему вам может понадобиться перенастроить ядро и детально описывает процесс настройки, сборки и установки нового ядра.

Печать

Рассказывает об управлении принтерами в FreeBSD, включая информацию об титульных страницах, учёте использования принтеров и первоначальной настройке.

Двоичная совместимость с Linux®

Описывает возможности Linux®-совместимости во FreeBSD. Также содержит подробные инструкции по установке многих популярных приложений для Linux®, таких как Oracle® и Mathematica®.

WINE

Описывает WINE и содержит подробные инструкции по установке. Также рассказывает о том, как работает WINE, как установить вспомогательный модуль для GUI, как запускать приложения Windows® во FreeBSD, и содержит другие советы и решения.

Настройка и оптимизация

Описывает всевозможные параметры настройки FreeBSD, которые может использовать системный администратор для оптимальной настройки системы. Также описывает различные конфигурационные файлы, используемые в FreeBSD и расположение этих файлов на диске.

Процесс загрузки FreeBSD

Рассказывает о процессе загрузки FreeBSD и объясняет, как управлять этим процессом при помощи различных настроек.

Безопасность

Описывает множество различных утилит, которые помогут вам поддерживать FreeBSD в безопасном, надёжном состоянии, включая Kerberos, IPsec и OpenSSH.

Изоляторы

Описывает технологию изоляции и преимущества изоляторов перед традиционной технологией chroot, поддерживаемой во FreeBSD.

Принудительный контроль доступа (MAC)

Описывает что такое принудительный контроль доступа (Mandatory Access Control, MAC) и как этот механизм может быть использован для защиты системы FreeBSD.

Аудит Событий Безопасности

Описывает, что представляет из себя Аудит Событий FreeBSD, как его можно установить и настроить, а также то, как можно анализировать или отслеживать аудиторские следы.

Устройства хранения

Описывает как управлять накопителями информации и файловыми системами в FreeBSD, включая физические диски, массивы RAID, оптические и ленточные носители, диски в оперативной памяти и сетевые файловые системы.

GEOM

Рассказывает о подсистеме GEOM в FreeBSD и описывает различные поддерживаемые уровни RAID.

Платформа хранения OpenZFS

Описывает платформу хранения OpenZFS и содержит руководство по её быстрому запуску, а также информацию о сложных вопросах эксплуатации OpenZFS под

управлением FreeBSD.

Поддержка файловых систем

Рассматривает поддержку дополнительных файловых систем во FreeBSD, таких как ext2, ext3 и ext4.

Виртуализация

Описывает возможности систем виртуализации и то, как они могут использоваться с FreeBSD.

Локализация - использование и настройка i18n/L10n

Описывает использование FreeBSD с языками, отличными от английского. Рассказывает о локализации на уровне системы и отдельных приложений.

Обновление системы и смена версии FreeBSD

Объясняет различия между FreeBSD-STABLE, FreeBSD-CURRENT и FreeBSD-RELEASE. Рассказывает, кому из пользователей будет полезно отслеживать версию системы в разработке и вкратце описывает этот процесс. Описывает методы, которык могут быть применены пользователями для обновления их систем до самой последнего безопасного релиза.

DTrace

Описывает порядок настройки и использования инструмента DTrace компании Sun™ во FreeBSD. Динамическая трассировка может помочь в локализации проблем с производительностью за счёт выполнения анализа системы в реальном режиме времени.

USB Device Mode / USB OTG

Описывает использование USB Device Mode и USB On The Go (USB OTG) во FreeBSD.

PPP и SLIP

Описывает использование PPP для соединения с удалёнными системами во FreeBSD.

Электронная почта

Описывает использование различных компонентов почтового сервера и более углублённо рассматривает простые вопросы конфигурации для наиболее популярного программного обеспечения почтовых серверов: sendmail.

Сетевые серверы

Предоставляет детальные инструкции и примеры файлов настройки для использования компьютера с FreeBSD в качестве файлового сервера (NFS), сервера доменных имен (DNS), сервера сетевой информационной системы (NIS), или сервера точного времени (ntpd).

Межсетевые экраны

Описывает принципы, на которых основаны программные брандмауэры, и содержит детали конфигурирования различных брандмауэров, доступных в FreeBSD.

Сложные вопросы работы в сети

Рассматривает множество вопросов работы с сетью, включая совместный доступ компьютеров вашей локальной сети к интернет, расширенные вопросы маршрутизации, беспроводные соединения, Bluetooth®, ATM, IPv6 и многое другое.

Получение FreeBSD

Перечисляет различные источники, из которых можно получить FreeBSD на CDROM или DVD, равно как и различные сайты в интернет, с которых можно скачать и установить FreeBSD.

Библиография

Эта книга касается многих различных тем, которые могут сподвигнуть вас на более детальное изучение. Библиография перечисляет множество отличных книг, упоминаемых в тексте.

Ресурсы в Интернет

Описывает множество форумов, доступных для пользователей FreeBSD, где можно задать вопросы и поучаствовать в технических обсуждениях FreeBSD.

Ключи PGP

Содержит ключи PGP некоторых разработчиков FreeBSD.

Договоренности, используемые в этой книге

Для того чтобы обеспечить целостность и простоту чтения текста в данной книге, мы применяем некоторые договорённости.

Типографические договорённости

Наклонный шрифт

Наклонный шрифт используется для имен файлов, адресов в интернет (URL), выделенного текста и первого применения технических терминов.

`Моноширинный шрифт`

`Моноширинных шрифт` используется для сообщений об ошибках, команд, имен пользователей, названий групп, названий устройств, переменных и фрагментов кода.

Полужирный шрифт

Полужирный шрифт используется для обозначения приложений, команд и комбинаций клавиш.

Пользовательский ввод

Клавиши представляются в виде **полужирного текста** для того, чтобы выделяться среди остального текста. Комбинации клавиш, которые должны вводиться одновременно, разделяются символом `+`, например:

`Ctrl` + `Alt` + `Del`

Это будет означать, что пользователь должен нажать клавиши `Ctrl`, `Alt` и `Del` одновременно.

Комбинации клавиш, которые должны вводиться последовательно, разделяются запятыми, например:

`Ctrl` + `X`, `Ctrl` + `S`

Это будет означать, что пользователь должен нажать `Ctrl` и `X` одновременно, после чего одновременно нажать `Ctrl` и `S`.

Примеры

Примеры, которые начинаются с C:\>, обозначают команды MS-DOS®. Если не указано обратного, эти команды могут вводиться из окна "Командная строка" в современных системах Microsoft® Windows®.

```
C:\> tools\fdimage floppies\kern.flp A:
```

Примеры, которые начинаются с # обозначают команды, которые должны быть запущены с правами суперпользователя в FreeBSD. Вы можете войти в систему как пользователь `root` для того, чтобы ввести эти команды или войти в систему обычным пользователем и использовать su(1) для того, чтобы получить привилегии суперпользователя.

```
# dd if=kern.flp of=/dev/fd0
```

Примеры, начинающиеся с %, указывают, что команда должна быть исполнена с правами обычного пользователя. Если не указано обратного, используется синтаксис C-shell для установки переменных окружения и других команд.

```
% top
```

Благодарности

Книга, которую вы держите в руках являет собой результат труда многих сотен людей по всему миру. Не имеет значения, присылали ли они исправления опечаток или предоставляли целые главы, их труд был полезен.

Несколько компаний поддерживали разработку этого документа, оплачивая авторам их труд, оплачивая публикацию и т.д. В частности, BSDi (в последствии приобретённая компанией Wind River Systems) оплачивала труд по улучшению этой книги участникам Проекта Документации FreeBSD, что в итоге сделало возможным выпуск первой печатной версии в марте 2000 года (ISBN 1-57176-241-8). Впоследствии компания Wind River Systems оплатила работу нескольких авторов по улучшению генерации книги в удобном для печати виде и добавлению нескольких глав. Кульминация этой работы являла собой публикацию второй печатной версии в ноябре 2001 года (ISBN 1-57176-303-1). В 2003-2004 годах FreeBSD Mall, Inc заплатила нескольким контрибьюторам за улучшение Handbook при подготовке к

третьей редакции. Третье печатное издание было разделено на два тома. Оба тома были опубликованы и получили названия The FreeBSD Handbook 3rd Edition Volume 1:User Guide (ISBN 1-57176-327-9) и The FreeBSD Handbook 3rd Edition Volume 2: Administrators Guide (ISBN 1-57176-328-7).

Часть I: В начале

Эта часть Руководства Пользователя FreeBSD предназначена для пользователей и администраторов - новичков в FreeBSD. Эти главы:

- Введут вас в FreeBSD.
- Проведут вас по процессу установки FreeBSD.
- Обучат вас некоторым основам UNIX®.
- Покажут вам как устанавливать программные пакеты не входящие в стандартную поставку FreeBSD.
- Введут вас в X Window, оконную систему для UNIX®, и опишут как настроить графическое окружение и сделать вашу работу более продуктивной.

Мы попытались сократить множество ссылок в тексте до минимума для того, чтоб вы могли прочитать этот раздел Руководства с начала до конца с минимумом перелистываний страниц.

Глава 1. Введение

1.1. Краткий обзор

Мы благодарим вас за интерес к FreeBSD! Следующая глава расскажет о некоторых аспектах проекта FreeBSD, таких как история, цели, модель разработки, и прочее.

Из этой главы вы узнаете:

- Какое отношение имеет FreeBSD к другим операционным системам.
- Историю проекта FreeBSD.
- Цели проекта FreeBSD.
- Основы модели разработки FreeBSD с открытыми исходными текстами.
- И, конечно, откуда появилось имя "FreeBSD".

1.2. Добро пожаловать во FreeBSD!

FreeBSD - это основанная на 4.4BSD-Lite операционная система для компьютеров Intel (x86 и Itanium®), AMD64, Alpha™ и Sun UltraSPARC®. Ведется работа по портированию и на другие архитектуры. Вы можете также прочесть об истории FreeBSD, или о текущем релизе. Если вы заинтересованы в помощи проекту (кодом, аппаратным обеспечением, деньгами), прочтите статью Помощь FreeBSD.

1.2.1. Что может FreeBSD?

FreeBSD имеет заслуживающие внимания возможности. Некоторые из них:

- *Вытесняющая многозадачность* с динамическим регулированием приоритетов, позволяющая плавно и справедливо распределить ресурсы компьютера между приложениями и пользователями, даже при тяжелейших нагрузках.
- *Многопользовательская поддержка*, которая позволяет множеству людей использовать FreeBSD совместно для различных задач. Это значит, например, что системная периферия, такая как принтеры и ленточные устройства, правильно разделяется всеми пользователями в системе или сети, и что пользователям или группам пользователей могут быть установлены лимиты каждого ресурса, защищая критические системные ресурсы от перегрузок.
- Мощный *TCP/IP-стек* с поддержкой промышленных стандартов, таких как SLIP, PPP, NFS, DHCP и NIS. Это означает, что FreeBSD может легко взаимодействовать с другими системами, а также работать сервером масштаба предприятия, предоставляя жизненно важные функции, такие как NFS (удалённый доступ к файлам) и услуги электронной почты, или представить вашу организацию в Интернете, обеспечивая работу служб WWW, FTP, маршрутизацию и функции межсетевого экрана (брандмауэра).
- *Защита памяти* гарантирует, что приложения (или пользователи) не смогут чинить препятствия друг другу. Фатальная ошибка в выполнении одного приложения не

скажется на работоспособности всей системы.

- FreeBSD *32-разрядная* операционная система (*64-разрядная* на Alpha, Itanium®, AMD64, и UltraSPARC®) и изначально создавалась именно такой.
- Промышленный стандарт *X Window System* (X11R6) предоставляет графический интерфейс пользователя (GUI) для большинства VGA карт и мониторов, и поставляется с полными исходными текстами.
- *Двоичная совместимость* с большинством программ, созданных для Linux, SCO, SVR4, BSDI и NetBSD.
- Тысячи *готовых к использованию* приложений доступны из коллекций *портов* и *пакетов* FreeBSD. Зачем искать что-то в сети, когда вы можете найти всё прямо здесь?
- Тысячи других *легко адаптируемых* приложений доступны в Интернете. FreeBSD совместима по исходным текстам с большинством популярных коммерческих UNIX®-систем и, таким образом, большинство приложений требуют лишь небольших изменений для сборки (или не требуют вообще).
- *Виртуальная память* с поддержкой сброса неиспользуемых страниц по требованию и "объединение виртуальной памяти и буферного кэша" спроектированы так, чтобы максимально эффективно удовлетворить приложения с огромными аппетитами к памяти и, в то же время, сохранить интерактивность для остальных пользователей.
- Поддержка *симметричной многопроцессорности* (SMP) для машин с несколькими процессорами.
- Полный комплект инструментов для разработчика: *C*, *C++* и *Fortran*. Множество дополнительных языков программирования для исследований и разработки также доступны из коллекций портов и пакетов.
- Доступность *исходных текстов* всей системы означает, что вы имеете максимальный контроль над операционной средой. Зачем выбирать закрытые решения и уповать на милость производителя, когда вы можете получить по-настоящему открытую систему?
- Обширная *online-документация*.
- *И многое-многое другое!*

FreeBSD основана на 4.4BSD-Lite от Computer Systems Research Group (CSRG) Калифорнийского Университета, Беркли, и продолжает славную традицию разработки BSD-систем. В дополнении к прекрасной работе, предоставленной CSRG, Проект FreeBSD тратит многие тысячи часов для тонкой настройки системы для максимальной производительности и надёжности в условиях максимально приближенным к "боевым". Когда большинство коммерческих гигантов только пытаются достичь такого уровня возможностей, производительности и надежности операционных систем для ПК, FreeBSD может предложить все это прямо *сейчас*!

Применение FreeBSD в действительности ограничено только вашим воображением. От разработки программного обеспечения до автоматизации производства, от складского учета до дистанционной коррекции азимутов спутниковых антенн; если задачи можно решить с помощью коммерческих UNIX®-систем, скорее всего, они решаемы и с помощью FreeBSD! FreeBSD также существенно выигрывает за счет буквально тысяч высококачественных приложений, разработанных исследовательскими центрами и

университетами во всём мире, и доступных за минимальную цену или даже бесплатно. Коммерческие приложения также доступны, и их с каждым днем становится всё больше.

Поскольку исходные тексты FreeBSD общедоступны, система может быть оптимизирована в почти невероятной степени для специальных приложений или проектов, а это, обычно, невозможно при использовании операционных систем от большинства коммерческих производителей. Вот несколько примеров того, как сейчас используется FreeBSD:

- *Интернет-службы:* мощнейший TCP/IP стек делает FreeBSD идеальной платформой для большинства Интернет-приложений, таких как:
  - FTP-серверы
  - Серверы World Wide Web (как стандартные, так и защищённые [SSL])
  - Межсетевые экраны (firewalls) и шлюзы NAT ("IP-маскарадинг")
  - Серверы электронной почты
  - Серверы новостей или дискуссионных групп USENET
  - и многое другое…

  Вы можете начать своё знакомство с FreeBSD, используя недорогой ПК класса 386, а впоследствии увеличить её мощь до сервера масштаба предприятия с четырьмя процессорами Xeon и RAID контроллером.

- *Образование:* Вы студент и ваше образование связано с компьютерами или другими инженерными дисциплинами? Нет лучшего пути начать изучение операционных систем, архитектуры компьютера и работы в сети, чем освоить FreeBSD. Количество свободно доступных пакетов САПР, математических и графических пакетов также делают её чрезвычайно полезной для тех, кто использует компьютер как инструмент для выполнения *другой* работы!
- *Исследования:* За счёт доступности исходных текстов для всей системы, FreeBSD - превосходная платформа как для изучения операционных систем и исследований в других областях компьютерных наук. Свободная природа FreeBSD позволяет удалённым группам сотрудничать, обмениваться идеями и совместными разработками, не беспокоясь о наличии специальных лицензий или ограничений на то, что может обсуждаться в открытых форумах.
- *Работа в сети:* Нужен новый маршрутизатор? Сервер имён (DNS)? Межсетевой экран, защищающий от проникновения извне в вашу сеть? FreeBSD может превратить давно списанный и пылящийся в углу 386-й или 486-й ПК в мощный маршрутизатор с возможностью фильтрации пакетов.
- *Рабочая станция X Window:* FreeBSD прекрасный выбор, если вам нужен недорогой X-терминал, использующий свободно распространяемый сервер X11. В отличие от X-терминала, на FreeBSD можно запускать множество приложений локально, если требуется, таким образом перенеся часть нагрузки с центрального сервера. FreeBSD может быть загружена "на бездисковой станции", что делает рабочую станцию ещё дешевле и проще в администрировании.
- *Разработка программного обеспечения:* Базовая поставка FreeBSD распространяется с полным набором инструментов для разработки, включая знаменитые компилятор GNU

C/C++ и отладчик.

FreeBSD доступна как в исходных текстах, так и в двоичном виде на CDROM, DVD и через анонимный доступ к FTP. Подробнее о том, как получить FreeBSD, см. в Получение FreeBSD.

1.2.2. Кто использует FreeBSD?

FreeBSD используется в качестве платформы на некоторых крупнейших сайтах в интернет, включая:

- Yahoo!
- Apache
- Blue Mountain Arts
- Pair Networks
- Sony Japan
- Netcraft
- Weathernews
- Supervalu
- TELEHOUSE America
- Sophos Anti-Virus
- JMA Wired

и на многих других.

1.3. О Проекте FreeBSD

В следующей части рассказывается о том, что из себя представляет проект, включая краткую историю, цели проекта и модель разработки проекта.

1.3.1. Краткая история FreeBSD

Проект FreeBSD возник в первой половине 1993 года, частично как результат развития "Неофициального комплекта исправлений к 386BSD (patchkit)", последними 3-мя координаторами этого проекта: Nate Williams, Rod Grimes и мною.

Нашей главной задачей было зафиксировать промежуточное состояние проекта 386BSD, чтобы исправить множество проблем, которые механизм patchkit (набор исправлений) не мог решить. Некоторые из вас, возможно, помнят раннее рабочее название этого проекта: "386BSD 0.5" или "386BSD Interim".

386BSD была операционной системой Билла Джолица, которая на тот момент сильно страдала от почти годичного пренебрежения к ней автора. Так как patchkit разрастался, его поддержание становилось более неудобным день от дня, мы пришли к единодушному соглашению, что нужно что-то делать, и решили помочь Биллу, предоставив этот промежуточный "очистительный" снимок состояния системы. Эти планы были грубо

оборваны, когда Билл внезапно решил прекратить поддержку проекта без каких-либо ясных комментариев, что должно быть сделано.

Нам потребовалось немного времени, чтобы прийти к решению продолжать следовать той же цели, даже без поддержки Билла, и мы приняли имя "FreeBSD", придуманное Дэвидом Гринмэном. Наши начальные цели были определены после консультаций с пользователями существовавшей системы, и как только стало понятно, что проект на пути к тому, чтобы стать реальностью, я связался с компанией Walnut Creek CDROM и поделился идеями о путях последующего улучшения каналов распространения FreeBSD для множества пользователей без доступа к Internet. Компания Walnut Creek CDROM не только поддержала идею распространения FreeBSD на CD, но ещё и предоставила проекту компьютер для работы и быстрый доступ к Интернету. Без почти беспрецедентной веры Walnut Creek CDROM в этот, в то время, полностью неизвестный проект, вряд ли FreeBSD зашла бы так далеко и так быстро, как сегодня.

Первым дистрибутивом, распространяемым как на CDROM, так и в сети, стала FreeBSD 1.0, выпущенная в декабре 1993 года. Эта версия была выполнена на основе ленты 4.3BSD-Lite ("Net/2") из Калифорнийского Университета в Беркли, с многочисленными добавлениями из проекта 386BSD и Фонда Свободного Программного Обеспечения. Это был довольно внушительный успех для первой попытки, и мы закрепили его с выходом FreeBSD 1.1 RELEASE в мае 1994 года.

В это же время, на горизонте сгустились тучи в связи с назревающим скандалом между Novell и Калифорнийским Университетом, Беркли. Это был вялотекущий судебный процесс о легальности версии Net/2 из Беркли. По условиям достигнутого соглашения, Калифорнийский Университет признавал, что большие куски Net/2 были "унаследованным" кодом, права на который принадлежат компании Novell, которая, в свою очередь, приобрела эти права ранее у AT&T. Взамен Беркли получил "благословение" Novell на то, что версия 4.4BSD-Lite после её выхода будет объявлена полностью "свободной", а всем пользователям Net/2 будет настоятельно рекомендовано перейти на неё. Это также касалось FreeBSD, и проекту было дано время до конца июля 1994 года для прекращения распространения его продукта, базирующегося на Net/2. На этих условиях проекту было разрешено выпустить последний релиз до окончания срока, и это была FreeBSD 1.1.5.1.

Тогда проект FreeBSD приступил к сложнейшей задаче буквально пересоздания с нуля на основе абсолютно новой и довольно неполной системы 4.4BSD-Lite. Версии "Lite" были в прямом смысле light (лёгкими) отчасти потому, что группа CSRG удалила большие куски кода, необходимого для создания реально загружающейся системы (по причине различных лицензионных требований), и фактически порт 4.4BSD для платформы Intel был очень неполным. Проекту потребовалось время почти до ноября 1994 года для того, чтобы выполнить этот переход, и на этом этапе FreeBSD 2.0 была опубликована в сети и на CDROM (в конце декабря). Несмотря на множество "острых углов" в этой версии, она пользовалась значительным успехом и была продолжена более устойчивой и простой в установке FreeBSD 2.0.5, выпущенной в июне 1995 года.

Мы выпустили FreeBSD 2.1.5 в августе 1996, и она стала достаточно популярной среди ISP и в коммерческой среде, чтобы выпустить еще один релиз из ветви 2.1-STABLE. Это была FreeBSD 2.1.7.1, вышедшая в феврале 1997 и завершившая главную ветвь разработки 2.1-STABLE. Сейчас в режиме поддержки, в эту ветвь (RELENG_2_1_0) вносятся только

улучшения защиты и другие критически важные исправления.

FreeBSD 2.2 была ответвлена от основной линии разработки ("-CURRENT") в ноябре 1996 как ветвь RELENG_2_2, а первая полная версия (2.2.1) появилась в апреле 1997. Последующие версии ветви 2.2 появлялись летом и в конце 1997 года, а последняя версия (2.2.8) вышла в ноябре 1998. Первая официальная версия 3.0 была подготовлена к выходу в октябре 1998, завершив развитие ветви 2.2

Третье ветвление произошло 20 января 1999 года: появились ветви 4.0-CURRENT и 3.X-STABLE. Из ветви 3.X-STABLE были получены: 3.1 - 15 февраля 1999, 3.2 - 15 мая 1999, 3.3 - 16 сентября 1999, 3.4 - 20 декабря 1999, 3.5 - 24 июня 2000, за которым последовал через несколько дней немного обновленный релиз 3.5.1, содержащий несколько исправлений в области защиты Kerberos. Это был последний релиз из ветви 3.X.

Другое ветвление было выполнено 13 марта 2000 года, в результате чего появилась ветвь 4.X-STABLE. Из этой ветви было выпущено несколько релизов: 4.0-RELEASE был представлен в марте 2000 года, а последний 4.11-RELEASE был выпущен в январе 2005 года.

Долгожданный 5.0-RELEASE был анонсирован 19 января 2003 года. Он стал кульминацией приблизительно трех лет работы, с этого релиза начался курс FreeBSD на расширенную поддержку мультипроцессорности и потоков в приложениях, а также появилась поддержка платформ UltraSPARC® и `ia64`. За этим релизом последовал релиз 5.1 в июне 2003 года. Последним релизом 5.X из ветви -CURRENT стал 5.2.1-RELEASE, представленный в феврале 2004.

Ветвь RELENG_5 была создана в августе 2004, затем последовал выпуск релиза 5.3-RELEASE, который открыл серию релизов из ветви 5-STABLE. Самый последний релиз 11.2-RELEASE был выпущен June 28, 2018. Из ветви RELENG_5 релизы больше выпускаться не будут.

Очередная ветвь, RELENG_6, была создана в июле 2005 года. 6.0-RELEASE, первый релиз из этой ветви, был выпущен в ноябре 2005 года. Последний из релизов ветви RELENG_6, 12.0-RELEASE, был выпущен December 11, 2018. Из ветви RELENG_6 будут выпускаться еще релизы.

На данный момент, долговременные разработки и проекты продолжаются в ветке 7.X-CURRENT, и по ходу разработки будут доступны снэпшот-релизы 7.X на CDROM (и, конечно же, в сети), постоянно выкладываемые на сервер снэпшотов как промежуточные результаты.

1.3.2. Цели Проекта FreeBSD

Целью Проекта FreeBSD является предоставление программного обеспечения, которое может быть использовано для любых целей и без дополнительных ограничений. Многие из нас внесли значительный вклад в код (и проект) и совершенно не против получать за это иногда финансовую компенсацию, но мы определенно не собираемся ее требовать. Мы верим, что первая и основная наша "миссия" это предоставление кода для всех, кому он необходим, и для любых целей, так чтобы этот код становился всё более распространённым и предоставлял самые широкие возможности. Это, я верю, является одной из наиболее фундаментальных целей Свободного Программного Обеспечения, и мы с энтузиазмом

поддерживаем её.

Тот код в нашем дереве исходных текстов, который попадает под Стандартную Общественную Лицензию GNU (GPL) или Стандартную Общественную Лицензию Ограниченного Применения GNU (LGPL), предоставляется с дополнительными условиями, хотя они обеспечивают только возможность доступа, а не его ограничение. По причине дополнительных сложностей, которые могут появится при коммерческом использовании GPL-продуктов, мы предпочитаем ПО, предоставленное под более свободной лицензией BSD, когда это возможно.

1.3.3. Модель Разработки FreeBSD

Разработка FreeBSD - это очень открытый и гибкий процесс. FreeBSD в буквальном смысле создана из кода, предоставленного сотнями людей со всего мира, в чем вы можете убедится, взглянув на список этих людей. Инфраструктура разработки FreeBSD позволяет этим сотням разработчиков сотрудничать с помощью Интернета. Мы постоянно ищем новых разработчиков и новые идеи, и те, кто заинтересован в более тесном взаимодействии и хочет принять участие в проекте, должны просто связаться с нами в рассылке freebsd-hackers. Для тех, кто желает уведомить других пользователей FreeBSD об основных направлениях работы, доступен Список рассылки анонсов FreeBSD.

Для независимой работы или тесного сотрудничества, полезно знать о проекте и процессе разработки FreeBSD следующее:

CVS-репозиторий

Главное дерево исходных текстов FreeBSD поддерживается с помощью CVS (Concurrent Versions System), свободно доступной системой контроля исходных текстов, которая поставляется вместе с FreeBSD. Основной CVS репозиторий располагается на компьютере, находящемся в городе Санта Клара, Калифорния (США), откуда и распространяется на множество зеркал по всему миру. Дерево CVS, содержащее ветви -CURRENT и -STABLE, может быть легко скопировано на ваш локальный компьютер. Дополнительную информацию о том, как это сделать, можно найти в разделе Синхронизация дерева исходных текстов.

Список коммиттеров

Коммиттеры - это люди, которые имеют доступ на *запись* к главному дереву CVS, и имеют право вносить изменения в главное дерево исходных текстов FreeBSD (термин "коммиттер" появился от названия команды cvs(1) `commit`, которая используется для внесения изменений в CVS-репозиторий). Лучший способ предоставить ваши соображения на рассмотрение коммиттеров - использовать команду send-pr(1). Если что-то произошло с системой, вы можете достучаться до них посылкой письма по адресу Список рассылки коммиттеров FreeBSD.

Core-группа FreeBSD

Core-группа FreeBSD могла бы быть эквивалентом Совета Директоров, если бы Проект FreeBSD был компанией. Главная задача Core-группы - гарантировать, что проект в целом в хорошем состоянии и движется в правильном направлении. Приглашение постоянных и ответственных разработчиков присоединиться к группе коммиттеров - одна из

функций Core-группы, так же, как и приглашение новых членов в Core-группу по мере того, как другие уходят. Нынешний состав команды был выбран из рядов коммиттеров путем общего голосования в июле 2006 года. Выборы проходят каждые 2 года.

Некоторые члены Core-группы имеют особые области ответственности, то есть, они являются ответственными за работу отдельной большой части системы. Полный список разработчиков FreeBSD и областей их ответственности можно найти в Списке участников.

> Большинство членов Core-группы - волонтёры, и не получают никакой финансовой выгоды от участия в проекте, поэтому вы не должны рассматривать возложенную на них "ответственность" как "гарантированную поддержку". Аналогия с "советом директоров" не очень точна и, вероятно, гораздо правильнее будет сказать, что это люди, которые посвятили себя FreeBSD, хотя и достойны лучшей участи!

Внешняя помощь

Последней, но однозначно не менее значимой, и наибольшей группой разработчиков являются сами пользователи, которые предоставляют комментарии и исправления ошибок нам на почти постоянной основе. Основной путь участвовать в не централизованной разработке - это подписка на Список рассылки FreeBSD, посвящённый техническим дискуссиям, где обсуждаются подобные вещи. Обратитесь к Ресурсы в интернет за дальнейшей информацией о различных списках рассылки FreeBSD.

Список участников проекта FreeBSD очень длинный и постоянно растет, так почему бы вам не присоединится к нему, предоставив что-нибудь проекту FreeBSD сегодня?

Предоставление кода - не единственный способ помочь проекту; более полный список того, что необходимо сделать, можно найти на Web-сайте проекта FreeBSD.

Вообще говоря, наша модель разработки организована как "нечеткий набор концентрированных колец". Централизованная модель разработана для удобства *пользователей* FreeBSD, которые получают простую систему контроля за одной центральной базой кода, и позволяет не оставить за бортом проекта потенциальных помощников! Мы желаем предоставить стабильную операционную систему с большим количеством согласованных прикладных программ, которые пользователи смогут легко установить и использовать - наша модель очень хорошо подходит для решения этой задачи.

Всё, что мы просим от желающих присоединится к нам как разработчики, - хотя бы часть той преданности постоянному успеху FreeBSD, которой отличаются нынешние разработчики!

1.3.4. Текущая версия FreeBSD

FreeBSD - это свободно доступная, с полными исходными текстами, основанная на 4.4BSD-Lite версия для компьютерных систем, основанных на Intel i386™, i486™, Pentium®, Pentium® Pro, Celeron®, Pentium® II, Pentium® III, Pentium® 4 (или совместимыми), Xeon™, DEC Alpha™ и Sun UltraSPARC®. В основном она базируется на программном обеспечении от группы CSRG, U.C. Berkley, с некоторым дополнениями из NetBSD, OpenBSD, 386BSD и Free Software

Foundation.

С момента выпуска FreeBSD версии 2.0 в конце 1994 года, производительность, возможности и стабильность FreeBSD существенно возросли. Самое большое изменение - это полное обновление системы виртуальной памяти с объединением виртуальной памяти и буферного кэша файловой системы, что не только увеличивает производительность, но и уменьшает количество используемой FreeBSD памяти, делая 5 Мбайтовую конфигурацию более приемлемым минимумом. Другие улучшения включают полную поддержку клиента и сервера NIS, поддержку транзакций TCP, поддержку "дозвона по запросу" в PPP, встроенную поддержку DHCP, улучшенную подсистеме SCSI, поддержку адаптеров ISDN, ATM, FDDI, Fast и Gigabit Ethernet (1000 Мбит), улучшенную поддержку новейших контролеров Adaptec и многие тысячи исправленных ошибок.

В дополнение к базовой системе, FreeBSD предоставляет коллекцию портированого ПО, включающую тысячи популярных программ. На момент подготовки этого документа в ней было более 36000 портов! В коллекцию входят множество программ от http-серверов до игр, языков программирования, текстовых редакторов и всего прочего. Полная Коллекция Портов требует приблизительно 3 GB дискового пространства, потому что порт представляет собой "изменения" оригинальных исходных текстов. Это сильно упрощает нам процесс обновления портов и существенно уменьшает объём занимаемого дискового пространства по сравнению со старой (1.0) Коллекцией Портов. Для того, чтобы скомпилировать и установить программу, необходимо всего лишь перейти в каталог порта программы, набрать `make install` и дать системе сделать все остальное. Полные исходные тексты для каждого порта, который вы устанавливаете, загружаются автоматически с CDROM или локального FTP-сервера, поэтому вам нужно только дисковое пространство для сборки необходимых портов. Почти каждый порт предоставляется также как скомпилированный "пакет", который может быть установлен с помощью простой команды (`pkg_add`) теми, кто предпочитает не компилировать порты из исходных текстов. Дополнительная информация о пакетах и портах находится в Установка приложений. порты и пакеты.

Множество дополнительных документов, которые могут пригодиться в процессе установки и использования FreeBSD, находятся в каталоге /usr/shared/doc на любой машине, работающей под управлением современной версии FreeBSD. Вы можете просматривать локально установленные документы с помощью любого браузера, поддерживающего HTML, используя следующие ссылки:

Руководство FreeBSD

/usr/shared/doc/ru_RU.KOI8-R/books/handbook/index.html

FreeBSD FAQ (Часто задаваемые вопросы)

/usr/shared/doc/ru_RU.KOI8-R/books/faq/index.html

Вы также можете просмотреть основные (и наиболее часто обновляемые) копии на http://www.FreeBSD.org/ru/.

Глава 2. Установка FreeBSD версий 9.X и более поздних

2.1. Краткий обзор

FreeBSD поставляется с простой в использовании текстовой программой установки. FreeBSD 9.0-RELEASE и более поздние укомплектованы установщиком, называемым bsdinstall, в то время как в релизах, предшествующих FreeBSD 9.0-RELEASE, для установки используется sysinstall. В этом разделе описана работа с программой bsdinstall. Работа с установщиком sysinstall описана в Установка FreeBSD версий 8.X и более ранних.

После прочтения этого раздела вы будете знать:

- Как создавать установочные носители для FreeBSD.
- Разбиение и именование разделов жестких дисков во FreeBSD.
- Как запустить bsdinstall.
- Вопросы, задаваемые утилитой bsdinstall, что они значат и как на них отвечать.

Перед прочтением этого раздела вам необходимо:

- Прочитать список поддерживаемого оборудования, который прилагается к устанавливаемой вами версии FreeBSD, а также убедиться, что ваше оборудование поддерживается.

В общем, эти инструкции по установке написаны для машин архитектуры i386™ ("PC-совместимая"). Там, где это необходимо, будут даны указания для других платформ. Между установщиком и этим документом могут быть незначительные различия, поэтому используйте эту главу как общее руководство, а не как точную пошаговую инструкцию.

2.2. Аппаратные требования

2.2.1. Минимальная конфигурация

Минимальная аппаратная конфигурация, достаточная для установки FreeBSD, зависит от версии FreeBSD и от аппаратной архитектуры.

Краткое изложение этой информации дано в следующих разделах. В зависимости от способа установки FreeBSD вам также может потребоваться поддерживаемый привод CDROM, а в некоторых случаях - сетевой адаптер. Об этом будет сказано в Подготовка установочного носителя информации.

2.2.1.1. FreeBSD/i386

Для FreeBSD/i386 необходим 486 процессор или выше, а также - как минимум 64 МБ ОЗУ. Для

самой минимальной установки потребуется не менее 1.1 ГБ свободного места на жестком диске.

Для устаревших компьютеров более эффективным способом повышения производительности является увеличение объема ОЗУ и объема жесткого диска, нежели установка более быстродействующего процессора.

2.2.1.2. FreeBSD/amd64

Существует два класса процессоров, на которых может работать FreeBSD/amd64. К первому принадлежат процессоры AMD64, включая AMD Athlon™64, AMD Athlon™64-FX, AMD Opteron™ и более новые.

Ко второму классу процессоров, на которых работает FreeBSD/amd64, принадлежат процессоры архитектуры Intel® EM64T. Перечень процессоров включает следующие семейства: Intel® Core™ 2 Duo, Quad, Extreme, семейства Intel® Xeon™ 3000, 5000 и 7000, а также Intel® Core™ i3, i5 и i7.

Если ваш компьютер построен на чипсете nVidia nForce3 Pro-150, то вам *необходимо* отключить IO APIC в BIOS. Если для этого нет опции в BIOS, отключите ACPI в операционной системе. В чипсете Pro-150 содержатся ошибки, для которых пока не существует исправлений.

2.2.1.3. FreeBSD/powerpc Apple® Macintosh®

Поддерживаются все американские системы Apple® Macintosh® с встроенным USB. Для многопроцессорных машин есть поддержка SMP.

Ядро (32-бит) может адресовать лишь первые 2 ГБ ОЗУ. На Blue & White PowerMac G3 не поддерживается FireWire®.

2.2.1.4. FreeBSD/sparc64

Поддерживаемые FreeBSD/sparc64 системы перечислены в проекте FreeBSD/sparc64.

Для FreeBSD/sparc64 требуется отдельный жесткий диск. На данный момент нет возможности разделять диск с другой операционной системой.

2.2.2. Поддерживаемое оборудование

Архитектуры и устройства, поддерживаемые каждым релизом FreeBSD, перечислены в файле Hardware Notes. Файл, как правило, называется HARDWARE.TXT, и располагается в корневом каталоге установочного носителя. Также копии списка поддерживаемого оборудования находятся на странице Release Information веб сайта FreeBSD.

2.3. Перед установкой

2.3.1. Сделайте резервные копии данных

Сделайте резервные копии всех важных данных с того компьютера, на который планируется установка FreeBSD. Проверьте пригодность резервных копий до начала установки. Перед внесением изменений на диск инсталлятор FreeBSD запросит подтверждение, но как только изменения будут внесены, то отменить их уже будет невозможно.

2.3.2. Решите куда установить FreeBSD

Если FreeBSD будет единственной установленной операционной системой, и она будет занимать весь жесткий диск, то можете смело пропустить этот раздел. Но если FreeBSD будет разделять диск с другими операционными системами, то во время установки вам понадобится понимание принципов разбиения дисков.

2.3.2.1. Разделы диска для FreeBSD/i386 и FreeBSD/amd64

Весь объем жестких дисков может быть разделен на множество частей. Эти части называются *разделами*.

Есть два способа деления диска на разделы. Традиционный способ - *Master Boot Record* (MBR) - хранит таблицу разделов, вмещающую до четырех *первичных разделов*. (Так сложилось исторически, что во FreeBSD эти разделы называются *слайсами*.) Возможны ситуации, в которых четыре раздела недостаточно, поэтому один из первичных разделов может быть превращен в *расширенный раздел*. Внутри расширенного раздела может быть создано несколько *логических разделов*. Результирующая структура выглядит немного неуклюже, но такова она есть.

Создание *Таблицы Разделов GUID* (GUID Partition Table, GPT) - это более новый и простой способ деления диска. Также новый способ (GPT) по сравнению с традиционным способом разбиения (MBR) гораздо более гибкий. Распространённые реализации GPT позволяют создавать до 128 разделов на одном диске, тем самым исключая необходимость создания неудобных сущностей наподобие логических дисков.

Некоторые старые операционные системы, например Windows® XP, не совместимы со схемой GPT. Если на один диск необходимо установить FreeBSD совместно с такой операционной системой, то следует воспользоваться схемой MBR.

Стандартному загрузчику FreeBSD необходим первичный раздел (MBR) или GPT раздел. (Обратитесь к Процесс загрузки FreeBSD за более подробной информацией о процессе загрузки FreeBSD.) Если все первичные или GPT разделы уже задействованы, то для FreeBSD один из них необходимо будет освободить.

Минимальная установка FreeBSD занимает ни много ни мало - 1 ГБ дискового пространства. Однако, это *очень* минимальная установка, практически не оставляющая свободного места. Более реалистичным минимумом является 3 ГБ без графической подсистемы, а если будет использоваться графическая подсистема, то 5 ГБ или более. Свободное пространство также потребуется приложениям от третьих лиц.

Для создания разделов существует разнообразие свободно распространяемых и коммерческих утилит. GParted Live это свободно распространяемый загрузочный дистрибутив, в который включен редактор разделов GParted. Также GParted включен в многие другие дистрибутивы Live CD от Linux.

Утилиты для создания разделов могут повредить ваши данные. Поэтому сделайте полную резервную копию и проверьте её целостность перед модификацией разделов диска.

Определенные трудности составляет изменение размеров разделов Microsoft® Vista. В таких случаях может пригодиться установочный CDROM от самой Microsoft® Vista.

Пример 1. Использование существующего раздела

Компьютер с ОС Windows® имеет жесткий диск размером 40 ГБ, диск разбит на два раздела по 20 ГБ. Windows® именует их дисками C: и D:. На диске C: данными занято 10 ГБ, а на диске D: - 5 ГБ.

Перемещение данных с диска D: на диск C: освобождает второй раздел для установки FreeBSD.

Пример 2. Уменьшение размера существующего раздела

Компьютер с ОС Windows® имеет жесткий диск размером 40 ГБ, на котором создан один большой раздел, занимающий весь жесткий диск. Windows® именует этот раздел диском C:. На этом разделе данные занимают 15 ГБ. Конечная цель - отвести для Windows® раздел размером 20 ГБ, а второй раздел размером 20 ГБ задействовать для установки FreeBSD.

Подобное перераспределение можно выполнить одним из двух способов:

1. Сделайте резервную копию данных вашей Windows®. Далее, переустановите Windows®, создав во время инсталляции раздел размером 20 ГБ.
2. Используйте утилиту редактирования разделов (наподобие GParted) для уменьшения раздела Windows®, а в освободившемся пространстве создайте новый раздел для установки FreeBSD.

Разделы диска, содержащие разные операционные системы, делают возможной загрузку по выбору одной из имеющихся операционных систем. Альтернативный способ, позволяющий загружать несколько операционных систем в одно и то же время, описан в разделе, называемом virtualization.

2.3.3. Соберите информацию о сетевых настройках

Некоторым вариантам установки FreeBSD для загрузки файлов необходимо наличие соединения с сетью. Инсталлятор запросит информацию о подключении для настройки соединения с сетью через интерфейс Ethernet (через кабельный модем или к модем DSL с

интерфейсом Ethernet).

Для автоматического конфигурирования сетевых интерфейсов часто применяется протокол *DHCP*. Если в подключаемой сети сервис DHCP отсутствует, информацию о подключении к необходимо взять у системного администратора или провайдера Интернет.

1. IP адрес
2. Маска подсети
3. IP адрес шлюза по умолчанию
4. Доменное имя локальной сети
5. IP адрес DNS сервера/серверов

2.3.4. Проверьте сведения об обнаруженных ошибках FreeBSD

Хотя проект FreeBSD борется за то, чтобы каждый релиз FreeBSD был настолько стабильным, насколько это возможно, ошибки порой вкрадываются в процесс разработки. В очень редких случаях эти ошибки влияют на процесс установки. Как только эти проблемы обнаруживаются и исправляются, их описание попадает в сообщения об ошибках FreeBSD, находящиеся на сайте FreeBSD. Проверьте сообщения об ошибках перед установкой и убедитесь, что отсутствуют проблемы, которые могут затронуть установку.

Информация о всех релизах, включая сообщения об ошибках каждого релиза, может быть найдена на странице информации о релизах веб сайта FreeBSD.

2.3.5. Подготовка установочного носителя информации

Установка FreeBSD начинается с загрузки компьютера с установочного носителя, будь то CD, DVD или USB флеш-накопитель. Инсталлятор - это не та программа, которую можно запустить из другой операционной системы.

В дополнение к стандартному установочному носителю, который содержит копии всех установочных файлов FreeBSD, также существует вариант, предназначенный исключительно для загрузки и называемый *bootonly*. Установочный носитель bootonly не содержит копий инсталляционных файлов, а загружает их из сети во время установки. Поэтому образ bootonly CD гораздо меньше объемом, а также при его использовании загружаются лишь необходимые файлы, тем самым уменьшается нагрузка на сетевое соединение.

Копии образов установочных носителей находятся на веб сайте FreeBSD. Также, в каталоге с файлами установочных образов находится файл CHECKSUM.SHA256, который понадобится вам для проверки целостности скачанного файла образа. Проверка целостности файла образа производится сравнением *контрольных сумм*. Для подсчета последних FreeBSD предоставляет sha256(1), другие операционные системы также располагают подобными программами. Сравните полученную контрольную сумму с одной из CHECKSUM.SHA256. Контрольные суммы должны совпасть полностью. Несовпадение контрольных сумм значит, что файл поврежден и к использованию не пригоден.

Если у вас уже имеется копия FreeBSD на CDROM, DVD, или USB флеш-накопителе, то нижеследующий текст можно опустить.

CD- и DVD-образы FreeBSD являются загрузочными. Для установки необходим один из них. Запишите образ на CD или DVD диск при помощи программы для записи CD, которая есть в вашей текущей операционной системе. Во FreeBSD запись дисков осуществляется утилитой cdrecord(1) из комплекта sysutils/cdrtools Коллекции Портов.

Для создания загрузочного флеш-накопителя выполните следующие шаги:

1. Получение образа для флеш-накопителя

   Образы для флеш-накопителя для FreeBSD 9.0-RELEASE и более поздних могут быть скачаны с каталога ISO-IMAGES/ по адресу `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/arch/arch/ISO-IMAGES/version/FreeBSD-version-RELEASE-arch-memstick.img`. Замените *arch* и *version* соответственно на архитектуру и номер версии которую вы планируете установить. Например, образы для флеш-накопителей FreeBSD/i386 9.0-RELEASE находятся на ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/i386/ISO-IMAGES/9.0/FreeBSD-9.0-RELEASE-i386-memstick.img.

   Для FreeBSD 8.*X* и более ранних версий используется иной путь каталогов. Детали загрузки и установки FreeBSD 8.*X* и более ранних версий описаны в Установка FreeBSD версий 8.X и более ранних.

   Имя образа для флеш-накопителя имеет суффикс .img. Каталог ISO-IMAGES/ содержит определённое количество разных образов, и выбор конкретного образа зависит от устанавливаемой версии FreeBSD, а в некоторых случаях - и от аппаратного обеспечения.

   Перед продолжением *сделайте резервную копию* данных с флеш-накопителя, так как следующая процедура *уничтожит* их.

2. Запись образа на флеш-накопитель

   Procedure: Использование FreeBSD для записи образа

   В нижеследующем примере показано использование /dev/da0 в качестве устройства, на которое производится запись. Удостоверьтесь в том, что целевое устройство выбрано верно, иначе вы можете повредить существующие данные.

   a. Запись образа при помощи dd(1)

      Файл .img не является обычным файлом. Это *образ* всего содержимого флеш-накопителя. Этот файл *не может* быть просто скопированным подобно обычному файлу, он должен быть записан непосредственно на целевое

устройство при помощи dd(1):

```
# dd if=FreeBSD-9.0-RELEASE-i386-memstick.img of=/dev/da0 bs=64k
```

Procedure: Использование Windows® для записи образа

> Удостоверьтесь в правильности выбора буквы диска, указываемой как целевое устройство, иначе вы перезапишете и повредите существующие данные.

a. Получение Image Writer для Windows®

   Image Writer для Windows® - это свободно распространяемое приложение, при помощи которого можно корректно записать образ на флеш-накопитель. Скачайте его с https://launchpad.net/win32-image-writer/ и сохраните в любую директорию.

b. Запись образа при помощи Image Writer

   Кликните дважды на иконке Win32DiskImager для запуска приложения. Удостоверьтесь, что буква диска, отображаемая в боксе `Device`, соответствует устройству флеш-накопителя. Кликните на иконке с папкой и выберите образ, который будет записан на флеш-накопитель. Нажмите кнопку **[Save]** для подтверждения выбора имени файла. Проверьте, что всё верно, а также что нет открытых директорий с флеш-накопителя в других окнах. Когда всё готово, нажмите кнопку **[Write]** для записи образа на флеш-накопитель.

Установка с дискет более не поддерживается.

Теперь вы готовы начать установку FreeBSD.

2.4. Начало установки

По умолчанию, установщик не изменяет данные на ваших дисках до тех пор, пока вы не увидите следующее сообщение:

```
Your changes will now be written to disk.  If you
have chosen to overwrite existing data, it will
be PERMANENTLY ERASED. Are you sure you want to
commit your changes?
```

Установка может быть прервана в любой момент до появления этого

> предупреждения, при этом содержимое дисков изменено не будет. Если вы обеспокоены тем, что что-то было настроено неверно, то вы можете просто выключить компьютер до этого сообщения, при этом никаких повреждений существующих данных не произойдет.

2.4.1. Загрузка

2.4.1.1. Загрузка на i386™ и amd64

1. Если вы подготовили "загрузочный" USB-накопитель, как описано в Подготовка установочного носителя информации, то вставьте его в USB гнездо перед включением компьютера.

   Если вы загружаетесь с CDROM, то вам необходимо будет включить компьютер и при первой возможности вставить CD диск.

2. Настройте вашу машину на загрузку с CDROM или с USB, в зависимости от того, какое устройство используется для установки. Настройки BIOS позволяют выбрать конкретное загрузочное устройство. Большинство систем также предоставляют возможность выбрать загрузочное устройство во время запуска, часто эта возможность активируется по нажатию клавиши `F10`, `F11`, `F12` или `Escape`.

3. Если ваш компьютер загружается как обычно и запускает существующую операционную систему, то:

   a. Диск не был вставлен заблаговременно. Оставьте его в приводе и попробуйте перезагрузить ваш компьютер.

   b. Ранее внесенные изменения в BIOS не сработали. Попробуйте повторить шаг настройки BIOS пока не получите необходимый порядок загрузки.

   c. Ваш нынешний BIOS не поддерживает загрузку с имеющегося загрузочного накопителя. В этом случае можно использовать Plop Boot Manager для загрузки более старых машин с CD или USB.

4. FreeBSD начнет загружаться. Если вы загружаетесь с CDROM, вы увидите поток сообщений, подобный следующему (информация о версиях опущена):

```
Booting from CD-ROM...
645MB medium detected
CD Loader 1.2

Building the boot loader arguments
Looking up /BOOT/LOADER... Found
Relocating the loader and the BTX
Starting the BTX loader

BTX loader 1.00 BTX version is 1.02
Consoles: internal video/keyboard
BIOS CD is cd0
BIOS drive C: is disk0
```

```
BIOS drive D: is disk1
BIOS 636kB/261056kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0x64daa0 data=0xa4e80+0xa9e40 syms
=[0x4+0x6cac0+0x4+0x88e9d]
\
```

5. Отображается меню загрузчика FreeBSD:

Рисунок 1. Меню загрузчика FreeBSD

Выждите десять секунд или нажмите `Enter`.

2.4.1.2. Загрузка Macintosh® PowerPC®

На большинстве машин удерживание клавиши `C` на клавиатуре во время начальной загрузки активирует загрузку с CD. Иначе, удерживайте `Command` + `Option` + `O` + `F`, или `Windows` + `Alt` + `O` + `F` на не-Apple® клавиатурах. На приглашение `0 >` введите

```
boot cd:,\ppc\loader cd:0
```

Для Xserves без клавиатур, ознакомьтесь с загрузкой в Open Firmware, которая описана на сайте поддержки Apple®.

2.4.1.3. Загрузка sparc64

Большинство систем sparc64 настроены на автоматическую загрузку с жесткого диска. Для того, чтобы установить FreeBSD, вам потребуется выполнить загрузку по сети или с CDROM,

что подразумевает получение доступа к PROM (OpenFirmware).

Для того, чтобы получить доступ к PROM, перегрузите систему и дождитесь появления загрузочных сообщений. Вид сообщений зависит от модели машины, но должен выглядеть подобно следующему:

```
Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
Copyright 1998-2001 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.2, 128 MB memory installed, Serial 51090132.
Ethernet address 0:3:ba:b:92:d4, Host ID: 830b92d4.
```

Если ваша система продолжает загружаться с жесткого диска, то чтобы получить приглашение PROM вам необходимо нажать на клавиатуре `L1` + `A` или `Stop` + `A`, или же послать сигнал `BREAK` через последовательную консоль (используя, например, `~#` в tip(1) или cu(1)). Приглашение выглядит подобно следующему:

```
ok      ①
ok {0} ②
```

① Приглашение, отображающееся на системах с одним центральным процессором.

② Приглашение, отображающееся на многопроцессорных (SMP) системах, цифра указывает на количество активных центральных процессоров.

На этом этапе вставьте CDROM в привод и наберите `boot cdrom` в приглашении PROM.

2.4.2. Просмотр результата определения устройств (device probe)

Выводимые на экран во время начальной загрузки системы последние пару сотен строк сохраняются, и при необходимости могут быть просмотрены.

Чтобы просмотреть содержимое буфера, нажмите `Scroll Lock`. Это включит режим буфера прокрутки. Далее, для просмотра сохраненных сообщений вы можете использовать клавиши навигации или клавиши `PageUp` и `PageDown`. Чтобы выйти из режима просмотра буфера нажмите еще раз `Scroll Lock`.

Включите прокрутку экранного буфера и просмотрите сообщения, которые были вытеснены с экрана во время определения устройств ядром. Вы увидите текст, подобный к Типичный вывод сообщений определения устройств, однако его содержимое будет отличаться в зависимости от комплекта устройств, установленных в ваш компьютер.

Типичный вывод сообщений определения устройств

```
Copyright (c) 1992-2011 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 9.0-RELEASE #0 r225473M: Sun Sep 11 16:07:30 BST 2011
    root@psi:/usr/obj/usr/src/sys/GENERIC amd64
```

```
CPU: Intel(R) Core(TM)2 Duo CPU     T9400  @ 2.53GHz (2527.05-MHz K8-class CPU)
  Origin = "GenuineIntel"  Id = 0x10676  Family = 6  Model = 17  Stepping = 6
  Features
=0xbfebfbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,CL
FLUSH,DTS,ACPI,MMX,FXSR,SSE,SSE2,SS,HTT,TM,PBE>
  Features2
=0x8e3fd<SSE3,DTES64,MON,DS_CPL,VMX,SMX,EST,TM2,SSSE3,CX16,xTPR,PDCM,SSE4.1>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
  TSC: P-state invariant, performance statistics
real memory  = 3221225472 (3072 MB)
avail memory = 2926649344 (2791 MB)
Event timer "LAPIC" quality 400
ACPI APIC Table: <TOSHIB A0064   >
FreeBSD/SMP: Multiprocessor System Detected: 2 CPUs
FreeBSD/SMP: 1 package(s) x 2 core(s)
 cpu0 (BSP): APIC ID:  0
 cpu1 (AP): APIC ID:  1
ioapic0: Changing APIC ID to 1
ioapic0 <Version 2.0> irqs 0-23 on motherboard
kbd1 at kbdmux0
acpi0: <TOSHIB A0064> on motherboard
acpi0: Power Button (fixed)
acpi0: reservation of 0, a0000 (3) failed
acpi0: reservation of 100000, b6690000 (3) failed
Timecounter "ACPI-safe" frequency 3579545 Hz quality 850
acpi_timer0: <24-bit timer at 3.579545MHz> port 0xd808-0xd80b on acpi0
cpu0: <ACPI CPU> on acpi0
ACPI Warning: Incorrect checksum in table [ASF!] - 0xFE, should be 0x9A
(20110527/tbutils-282)
cpu1: <ACPI CPU> on acpi0
pcib0: <ACPI Host-PCI bridge> port 0xcf8-0xcff on acpi0
pci0: <ACPI PCI bus> on pcib0
vgapci0: <VGA-compatible display> port 0xcff8-0xcfff mem 0xff400000-
0xff7fffff,0xe0000000-0xefffffff irq 16 at device 2.0 on pci0
agp0: <Intel GM45 SVGA controller> on vgapci0
agp0: aperture size is 256M, detected 131068k stolen memory
vgapci1: <VGA-compatible display> mem 0xffc00000-0xffcfffff at device 2.1 on pci0
pci0: <simple comms> at device 3.0 (no driver attached)
em0: <Intel(R) PRO/1000 Network Connection 7.2.3> port 0xcf80-0xcf9f mem 0xff9c0000-
0xff9dffff,0xff9fe000-0xff9fefff irq 20 at device 25.0 on pci0
em0: Using an MSI interrupt
em0: Ethernet address: 00:1c:7e:6a:ca:b0
uhci0: <Intel 82801I (ICH9) USB controller> port 0xcf60-0xcf7f irq 16 at device 26.0
on pci0
usbus0: <Intel 82801I (ICH9) USB controller> on uhci0
uhci1: <Intel 82801I (ICH9) USB controller> port 0xcf40-0xcf5f irq 21 at device 26.1
on pci0
usbus1: <Intel 82801I (ICH9) USB controller> on uhci1
uhci2: <Intel 82801I (ICH9) USB controller> port 0xcf20-0xcf3f irq 19 at device 26.2
on pci0
```

```
usbus2: <Intel 82801I (ICH9) USB controller> on uhci2
ehci0: <Intel 82801I (ICH9) USB 2.0 controller> mem 0xff9ff800-0xff9ffbff irq 19 at
device 26.7 on pci0
usbus3: EHCI version 1.0
usbus3: <Intel 82801I (ICH9) USB 2.0 controller> on ehci0
hdac0: <Intel 82801I High Definition Audio Controller> mem 0xff9f8000-0xff9fbfff irq
22 at device 27.0 on pci0
pcib1: <ACPI PCI-PCI bridge> irq 17 at device 28.0 on pci0
pci1: <ACPI PCI bus> on pcib1
iwn0: <Intel(R) WiFi Link 5100> mem 0xff8fe000-0xff8fffff irq 16 at device 0.0 on pci1
pcib2: <ACPI PCI-PCI bridge> irq 16 at device 28.1 on pci0
pci2: <ACPI PCI bus> on pcib2
pcib3: <ACPI PCI-PCI bridge> irq 18 at device 28.2 on pci0
pci4: <ACPI PCI bus> on pcib3
pcib4: <ACPI PCI-PCI bridge> at device 30.0 on pci0
pci5: <ACPI PCI bus> on pcib4
cbb0: <RF5C476 PCI-CardBus Bridge> at device 11.0 on pci5
cardbus0: <CardBus bus> on cbb0
pccard0: <16-bit PCCard bus> on cbb0
isab0: <PCI-ISA bridge> at device 31.0 on pci0
isa0: <ISA bus> on isab0
ahci0: <Intel ICH9M AHCI SATA controller> port 0x8f58-0x8f5f,0x8f54-0x8f57,0x8f48-
0x8f4f,0x8f44-0x8f47,0x8f20-0x8f3f mem 0xff9fd800-0xff9fdfff irq 19 at device 31.2 on
pci0
ahci0: AHCI v1.20 with 4 3Gbps ports, Port Multiplier not supported
ahcich0: <AHCI channel> at channel 0 on ahci0
ahcich1: <AHCI channel> at channel 1 on ahci0
ahcich2: <AHCI channel> at channel 4 on ahci0
acpi_lid0: <Control Method Lid Switch> on acpi0
battery0: <ACPI Control Method Battery> on acpi0
acpi_button0: <Power Button> on acpi0
acpi_acad0: <AC Adapter> on acpi0
acpi_toshiba0: <Toshiba HCI Extras> on acpi0
acpi_tz0: <Thermal Zone> on acpi0
attimer0: <AT timer> port 0x40-0x43 irq 0 on acpi0
Timecounter "i8254" frequency 1193182 Hz quality 0
Event timer "i8254" frequency 1193182 Hz quality 100
atkbdc0: <Keyboard controller (i8042)> port 0x60,0x64 irq 1 on acpi0
atkbd0: <AT Keyboard> irq 1 on atkbdc0
kbd0 at atkbd0
atkbd0: [GIANT-LOCKED]
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: model GlidePoint, device ID 0
atrtc0: <AT realtime clock> port 0x70-0x71 irq 8 on acpi0
Event timer "RTC" frequency 32768 Hz quality 0
hpet0: <High Precision Event Timer> iomem 0xfed00000-0xfed003ff on acpi0
Timecounter "HPET" frequency 14318180 Hz quality 950
Event timer "HPET" frequency 14318180 Hz quality 450
Event timer "HPET1" frequency 14318180 Hz quality 440
Event timer "HPET2" frequency 14318180 Hz quality 440
```

```
Event timer "HPET3" frequency 14318180 Hz quality 440
uart0: <16550 or compatible> port 0x3f8-0x3ff irq 4 flags 0x10 on acpi0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
ppc0: cannot reserve I/O port range
est0: <Enhanced SpeedStep Frequency Control> on cpu0
p4tcc0: <CPU Frequency Thermal Control> on cpu0
est1: <Enhanced SpeedStep Frequency Control> on cpu1
p4tcc1: <CPU Frequency Thermal Control> on cpu1
Timecounters tick every 1.000 msec
hdac0: HDA Codec #0: Realtek ALC268
hdac0: HDA Codec #1: Lucent/Agere Systems (Unknown)
pcm0: <HDA Realtek ALC268 PCM #0 Analog> at cad 0 nid 1 on hdac0
pcm1: <HDA Realtek ALC268 PCM #1 Analog> at cad 0 nid 1 on hdac0
usbus0: 12Mbps Full Speed USB v1.0
usbus1: 12Mbps Full Speed USB v1.0
usbus2: 12Mbps Full Speed USB v1.0
usbus3: 480Mbps High Speed USB v2.0
ugen0.1: <Intel> at usbus0
uhub0: <Intel UHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus0
ugen1.1: <Intel> at usbus1
uhub1: <Intel UHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus1
ugen2.1: <Intel> at usbus2
uhub2: <Intel UHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus2
ugen3.1: <Intel> at usbus3
uhub3: <Intel EHCI root HUB, class 9/0, rev 2.00/1.00, addr 1> on usbus3
uhub0: 2 ports with 2 removable, self powered
uhub1: 2 ports with 2 removable, self powered
uhub2: 2 ports with 2 removable, self powered
uhub3: 6 ports with 6 removable, self powered
ugen2.2: <vendor 0x0b97> at usbus2
uhub8: <vendor 0x0b97 product 0x7761, class 9/0, rev 1.10/1.10, addr 2> on usbus2
ugen1.2: <Microsoft> at usbus1
ada0 at ahcich0 bus 0 scbus1 target 0 lun 0
ada0: <Hitachi HTS543225L9SA00 FBEOC43C> ATA-8 SATA 1.x device
ada0: 150.000MB/s transfers (SATA 1.x, UDMA6, PIO 8192bytes)
ada0: Command Queueing enabled
ada0: 238475MB (488397168 512 byte sectors: 16H 63S/T 16383C)
ada0: Previously was known as ad4
ums0: <Microsoft Microsoft 3-Button Mouse with IntelliEyeTM, class 0/0, rev 1.10/3.00,
addr 2> on usbus1
SMP: AP CPU #1 Launched!
cd0 at ahcich1 bus 0 scbus2 target 0 lun 0
cd0: <TEAC DV-W28S-RT 7.0C> Removable CD-ROM SCSI-0 device
cd0: 150.000MB/s transfers (SATA 1.x, ums0: 3 buttons and [XYZ] coordinates ID=0
UDMA2, ATAPI 12bytes, PIO 8192bytes)
cd0: cd present [1 x 2048 byte records]
ugen0.2: <Microsoft> at usbus0
ukbd0: <Microsoft Natural Ergonomic Keyboard 4000, class 0/0, rev 2.00/1.73, addr 2>
on usbus0
```

```
kbd2 at ukbd0
uhid0: <Microsoft Natural Ergonomic Keyboard 4000, class 0/0, rev 2.00/1.73, addr 2>
on usbus0
Trying to mount root from cd9660:/dev/iso9660/FREEBSD_INSTALL [ro]...
```

Внимательно просмотрите вывод определения устройств и убедитесь, что FreeBSD обнаружила все ожидаемые вами устройства. Если устройство не было найдено, то оно не будет упомянуто в выводе. Модули ядра позволяют вам добавить поддержку устройств, драйвера которых отсутствуют в ядре GENERIC.

После процедуры определения устройств вы увидите Выбор вариантов работы установочного носителя. Установочный носитель может использоваться одним из трёх способов: для установки FreeBSD, как Live CD, или просто для доступа к оболочке FreeBSD. Используйте клавиши навигации для выбора опции, а `Enter` - для подтверждения выбора.

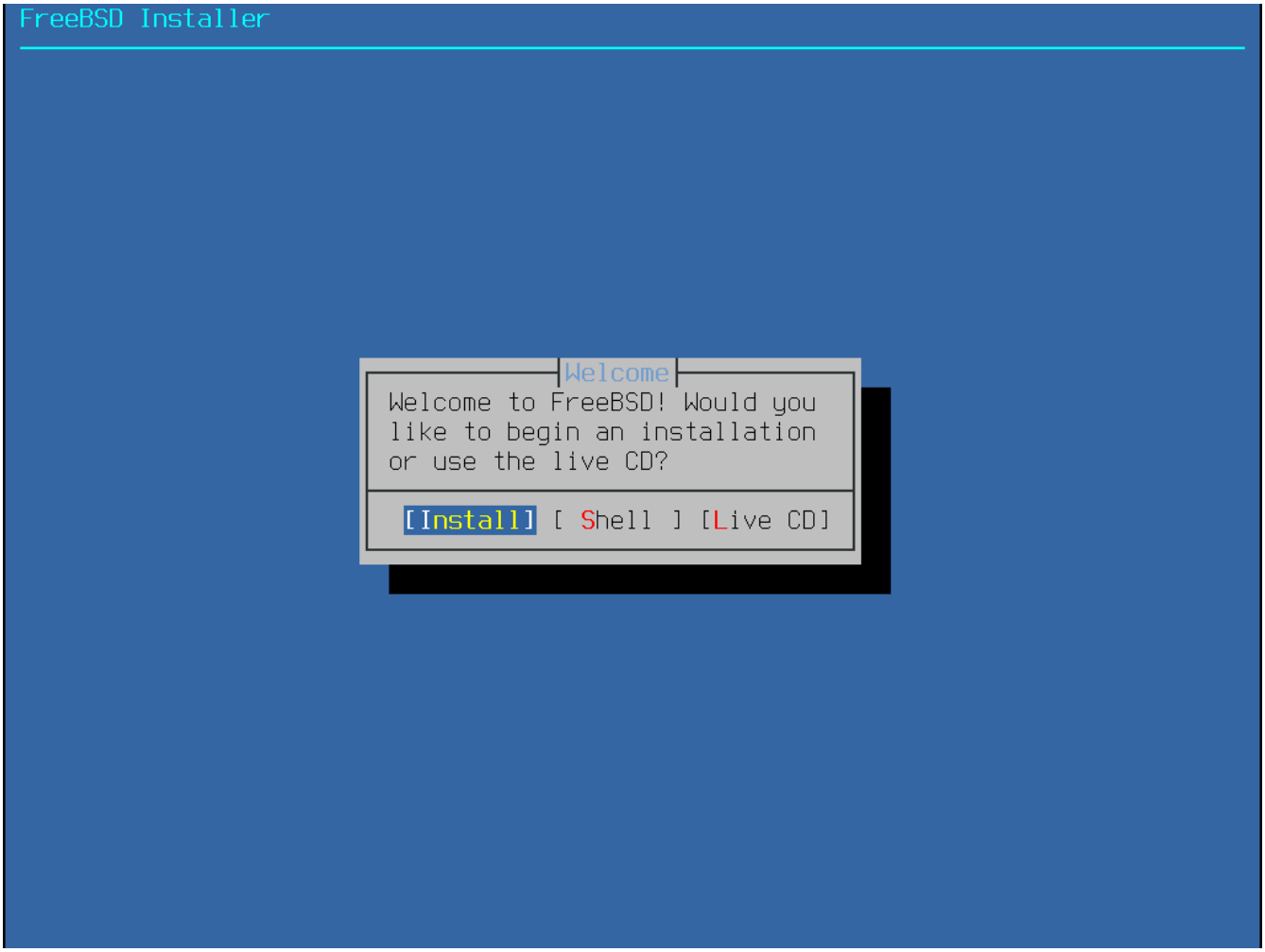


Рисунок 2. Выбор вариантов работы установочного носителя

Выбор опции **[Install]** вызовет программу-установщик.

2.5. Введение в bsdinstall

bsdinstall это текстовая программа для установки FreeBSD, созданная Nathan Whitehorn <nwhitehorn@FreeBSD.org> и представленная в 2011 году для FreeBSD 9.0.

В комплекте с PC-BSD есть программа pc-sysinstall от Kris Moore <kmoore@FreeBSD.org>, которая также может использоваться для установки FreeBSD. Несмотря на то, что эту программу путают с bsdinstall, обе они между собой никак не связаны.

Система меню bsdinstall контролируется клавишами навигации, а также `Enter`, `Tab`, `Space` и другими.

2.5.1. Выбор раскладки клавиатуры (Keymap)

В зависимости от используемой системной консоли, bsdinstall может предложить выбрать отличную от настроенной по умолчанию раскладку клавиатуры.

Рисунок 3. Выбор раскладки клавиатуры

Если нажата кнопка **[YES]**, отобразится следующее меню выбора раскладки клавиатуры. Иначе, это меню выбора отображено не будет, а будет использоваться раскладка клавиатуры по умолчанию.

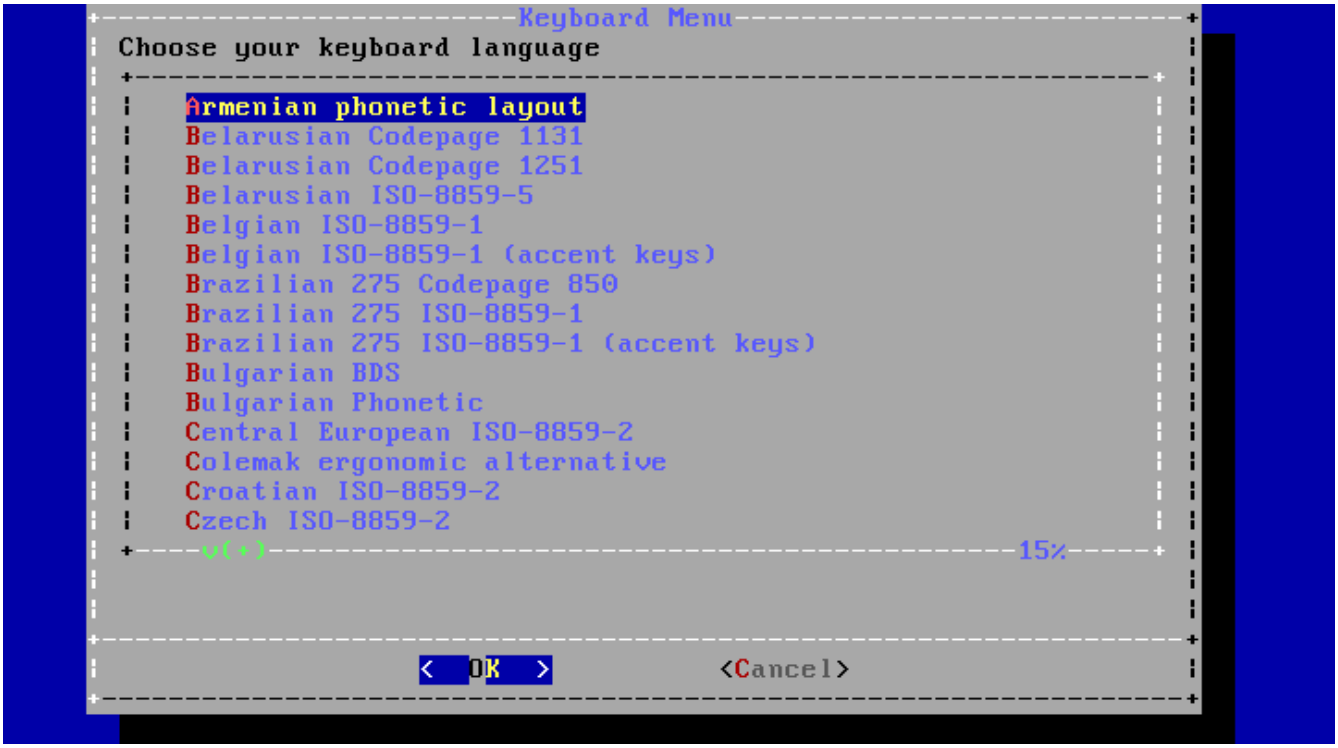


Рисунок 4. Меню выбора раскладки клавиатуры

Используя клавиши навигации и клавишу `Enter` выберите раскладку, которая наиболее близко соответствует клавиатуре, подключенной к системе.

> Нажатие `Esc` приведет к выбору раскладки по умолчанию. Выбор опции United States of America ISO-8859-1 тоже является безопасным в том случае, если возникают трудности с определением раскладки.

2.5.2. Установка имени хоста

Далее, bsdinstall предложит указать имя хоста для устанавливаемой системы.

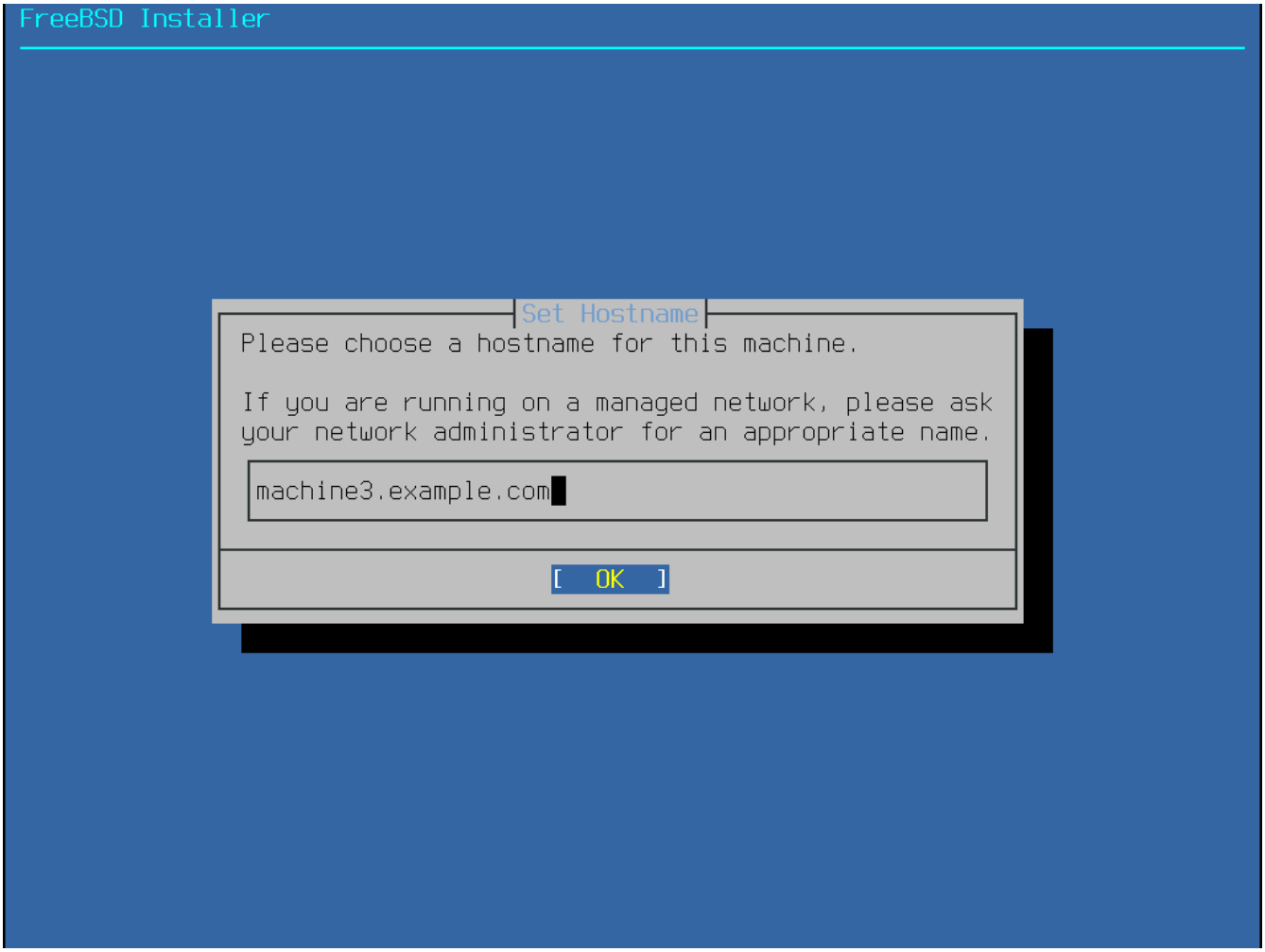


Рисунок 5. Установка имени хоста

Вводимое имя хоста должно быть полным (fully-qualified), например: `machine3.example.com`.

2.5.3. Выбор устанавливаемых компонентов

Далее, bsdinstall предложит выбрать дополнительные компоненты для установки.

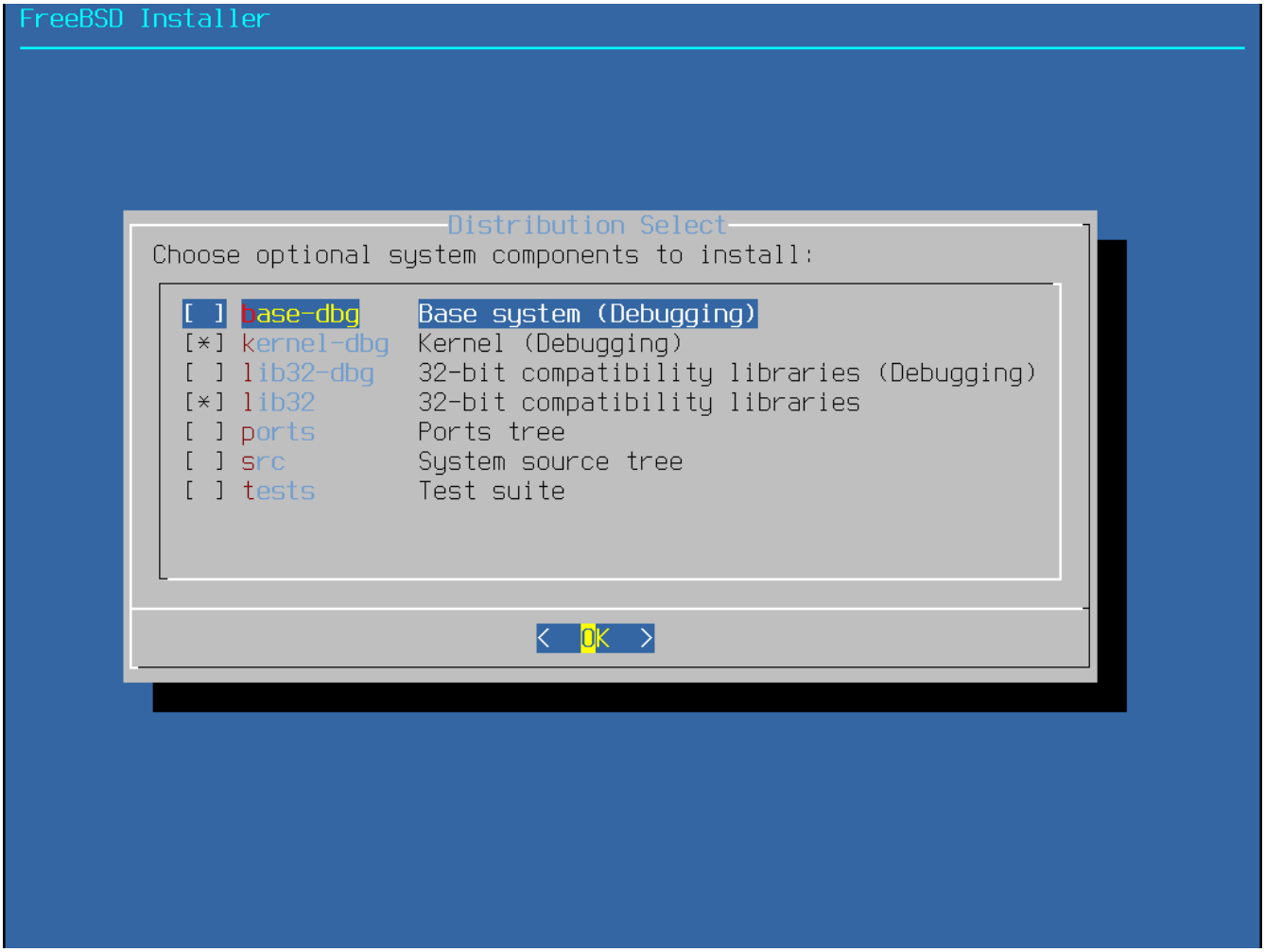


Рисунок 6. Выбор устанавливаемых компонентов

Определение перечня компонентов для установки в наибольшей мере зависит от планируемого использования системы и от количества доступного дискового пространства. Ядро и набор утилит FreeBSD (вместе называемые "базовой системой") устанавливаются всегда.

В зависимости от типа установки, некоторые из следующих компонентов могут не появляться.

Дополнительные компоненты

- `doc` - Дополнительная документация, преимущественно исторического характера. Документация, предоставляемая Проектом Документирования FreeBSD может быть установлена позже.
- `games` - Несколько традиционных игр BSD, в том числе fortune, rot13, и другие.
- `lib32` - Библиотеки совместимости для запуска 32-битных приложений на 64-битных версиях FreeBSD.
- `ports` - Коллекция Портов FreeBSD.

  Коллекция Портов - это простой и удобный способ установки программ. Она не содержит исходных кодов, необходимых для компиляции приложений. Коллекция Портов - это множество файлов, при помощи которого автоматизируется загрузка, компиляция и установка программных пакетов сторонних разработчиков. В Установка приложений.

порты и пакеты описано, как использовать коллекцию портов.

Программа установки не проверяет наличие свободного места. Поэтому выбирайте эту опцию лишь тогда, когда имеется достаточно свободного места на жестком диске. Что касается FreeBSD 9.0, Коллекция Портов занимает около 3 GB дискового пространства. Учтите, что для более новых версий FreeBSD занимаемое Коллекцией Портов дисковое пространство будет расти.

- `src` - Исходный код системы.

  FreeBSD распространяется с полным исходным кодом как для ядра, так и для программ базовой системы. Для большинства приложений исходный код системы не нужен, однако он может потребоваться при построении некоторых программ, распространяемых в виде исходных кодов (например, драйверов или модулей ядра), или для разработки FreeBSD.

  Полное дерево исходных кодов требует 1 ГБ дискового пространства, пересборка всей системы FreeBSD требует дополнительно 5 ГБ пространства.

2.6. Установка по сети

Установочный носитель *bootonly* не содержит копий установочных файлов. В случае использования такого носителя необходимые файлы должны быть получены загрузкой из сети.

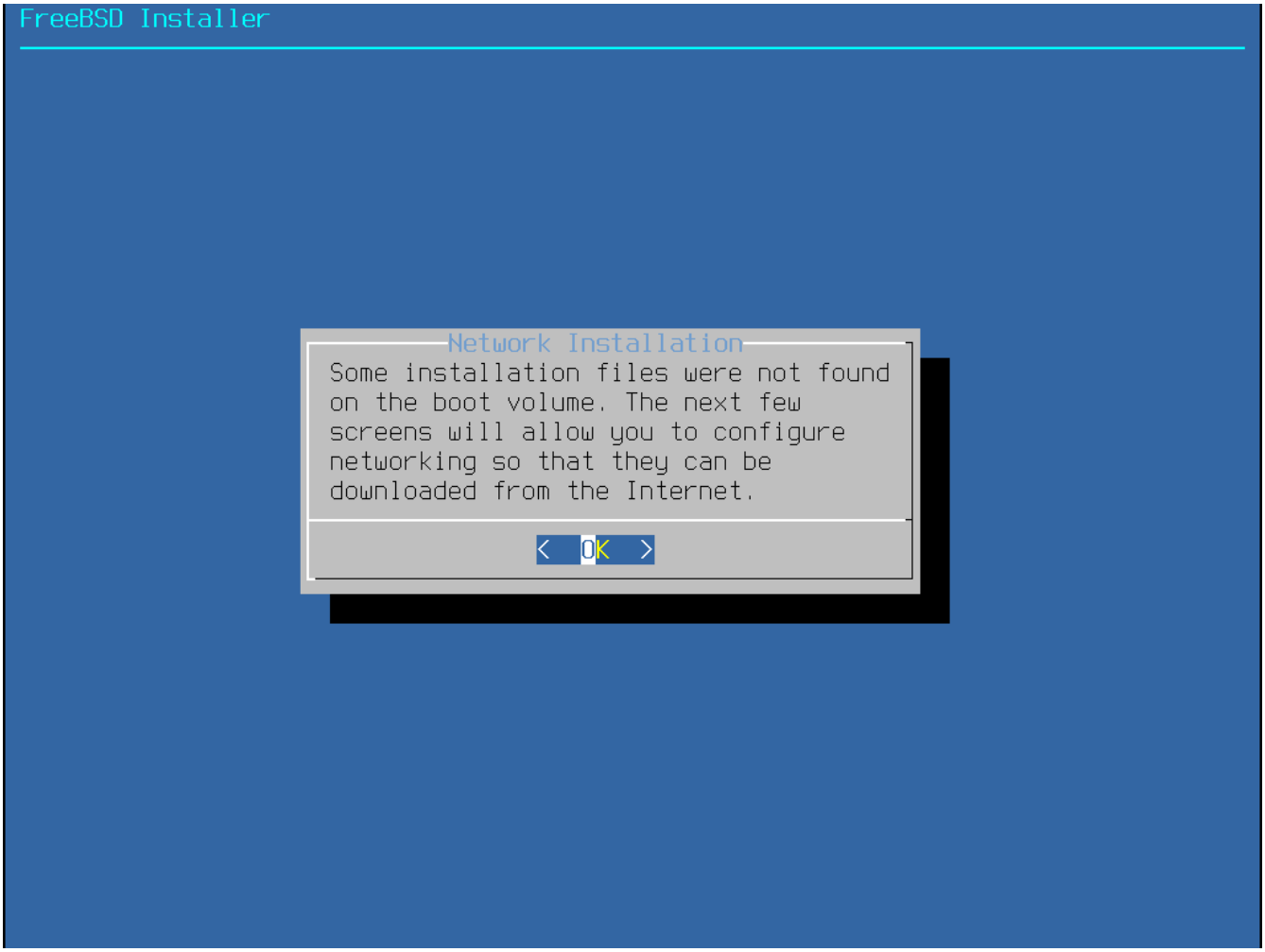


Рисунок 7. Установка по сети

После настройки сетевого соединения, которая детально описана в Настройка сетевых интерфейсов, выбирается зеркало сайта. Зеркала сайта содержат копии файлов FreeBSD. Выберите зеркало, размещенное в том регионе мира, что и компьютер, на который устанавливается FreeBSD. Если зеркало расположено ближе к целевому компьютеру, то файлы могут быть получены быстрее, тем самым уменьшится время установки.

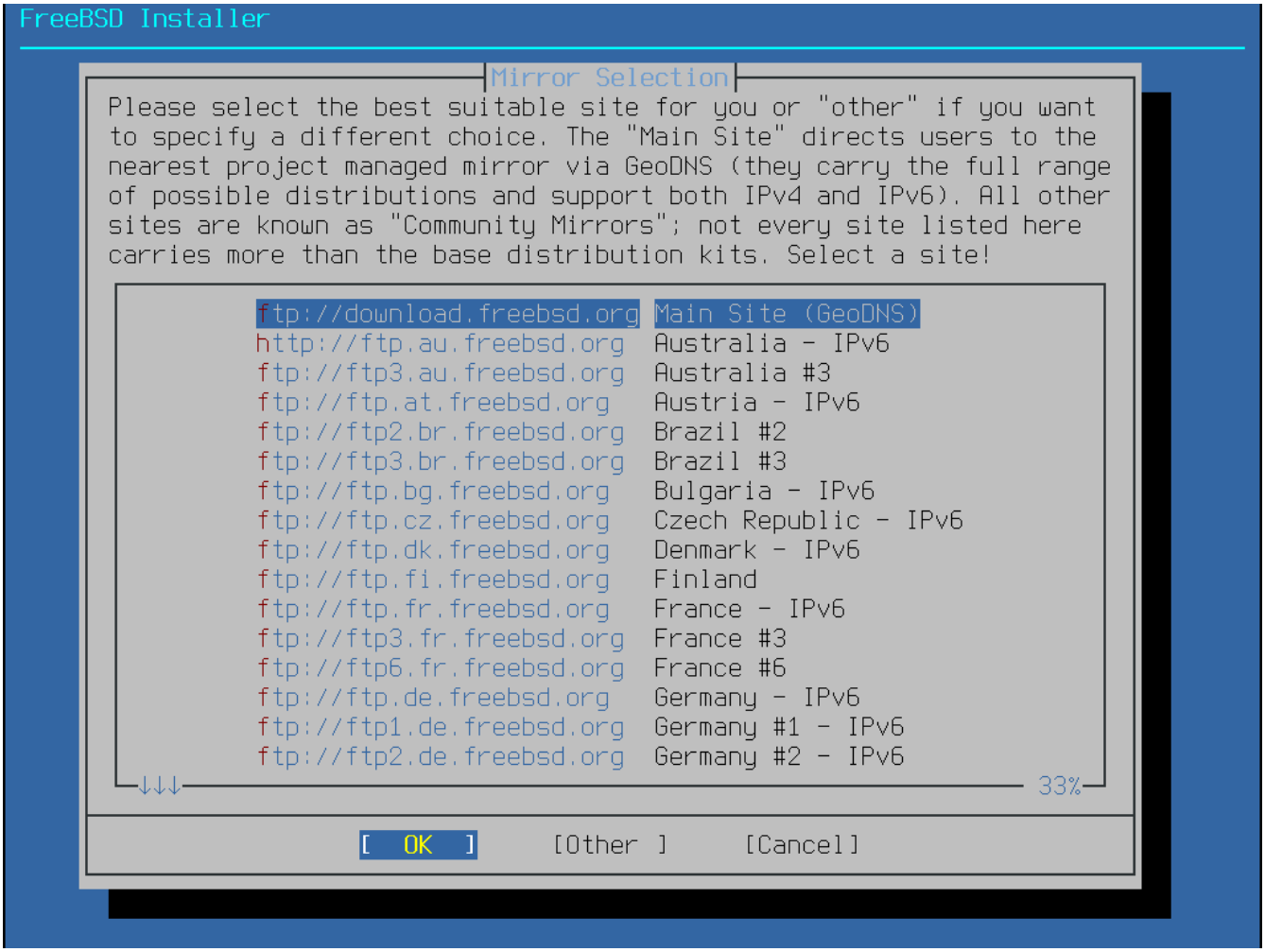


Рисунок 8. Выбор зеркала сайта

Дальнейший сценарий одинаков для всех способов установки.

2.7. Выделение дискового пространства

Есть три способа осуществить разбиение дискового пространства для FreeBSD. *Шаблонное* (*guided*) разбиение автоматически настраивает разделы диска, *ручное* (*manual*) разбиение позволяет опытным пользователям создавать разделы согласно своим требованиям. И наконец, есть возможность вызвать командный интерпретатор, в котором можно будет непосредственно запускать утилиты наподобие gpart(8), fdisk(8) и bsdlabel(8).

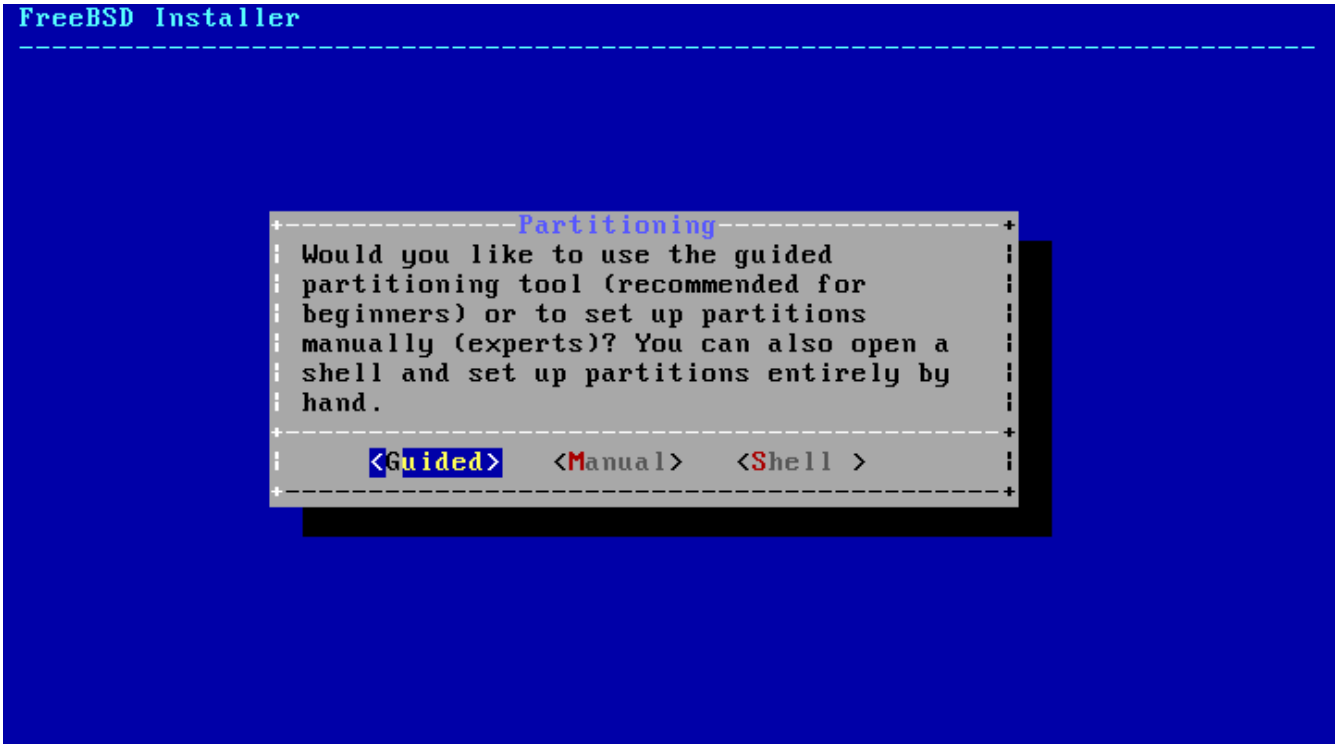


Рисунок 9. Выбор способа разбиения: шаблонное (guided) или ручное (manual)

2.7.1. Шаблонное (guided) разбиение

Если в системе есть несколько дисков, то выберите один, на который будет устанавливаться FreeBSD.

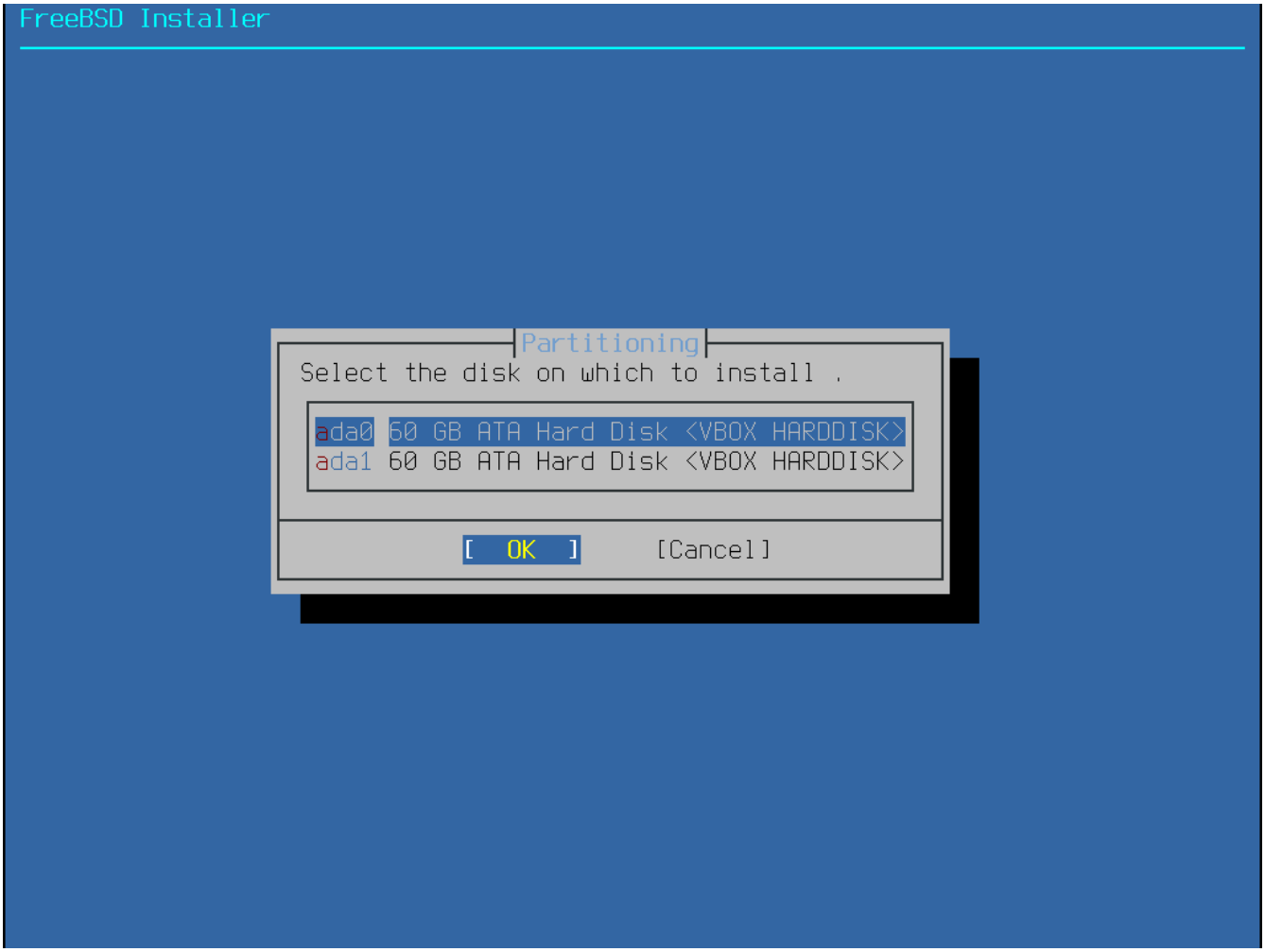


Рисунок 10. Выбор из множества дисков

Для FreeBSD может быть выделен весь диск или только его часть. Если выбирается **[Entire Disk]**, то создается стандартное разбиение, занимающее весь диск. Выбрав **[Partition]**, вы получите создание разделов в неиспользуемой области диска.

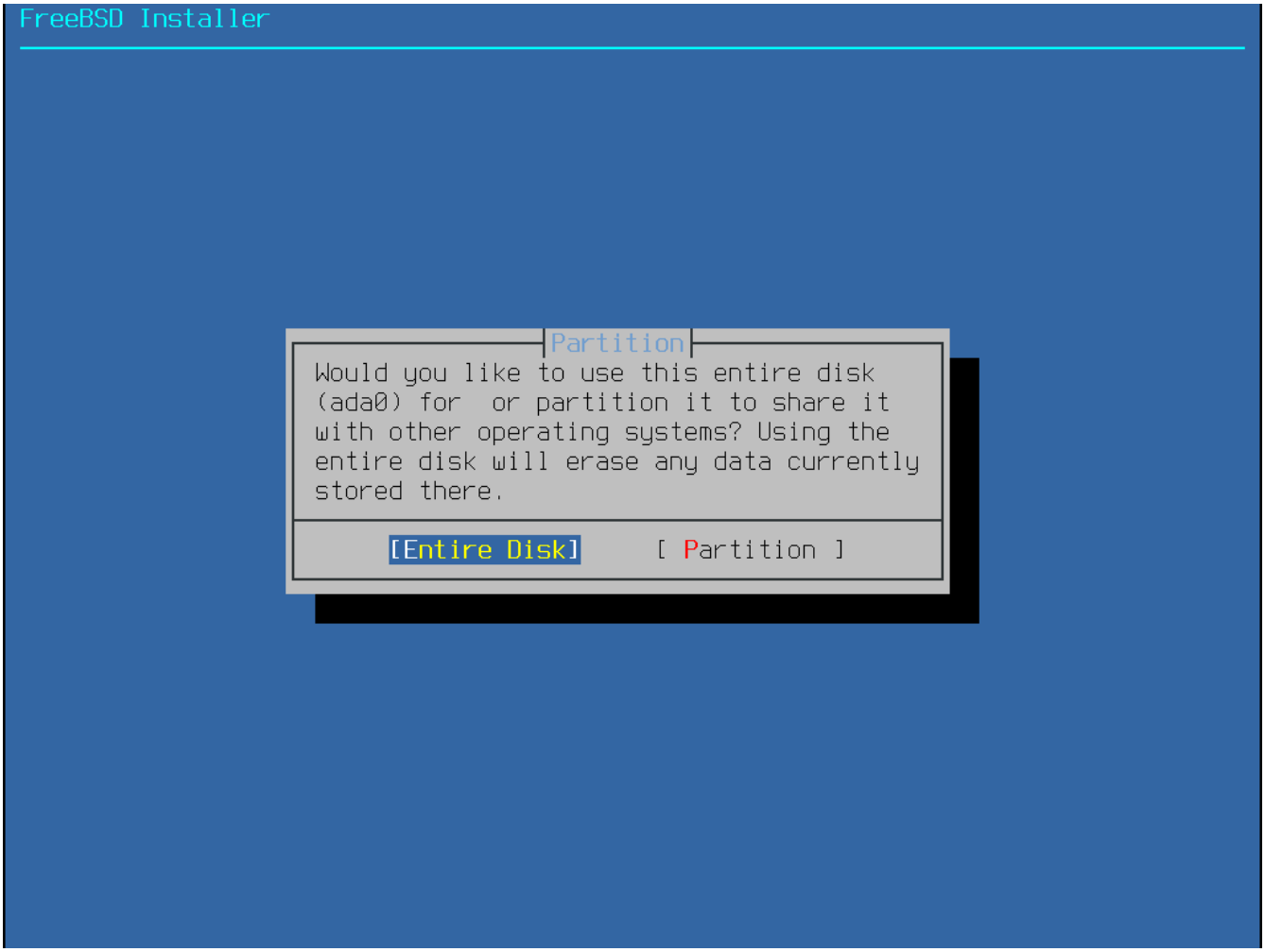


Рисунок 11. Выбор всего диска или раздела

По завершении разбиения дискового пространства внимательно просмотрите результат. Если была допущена ошибка, то вам предоставляется возможность либо вернуть конфигурацию к исходному состоянию нажав **[Revert]**, либо выполнить автоматическое переразбиение выбрав **[Auto]**. Также разделы могут быть созданы, изменены или удалены вручную. Если результат разбиения корректен, выберите **[Finish]** для продолжения установки.

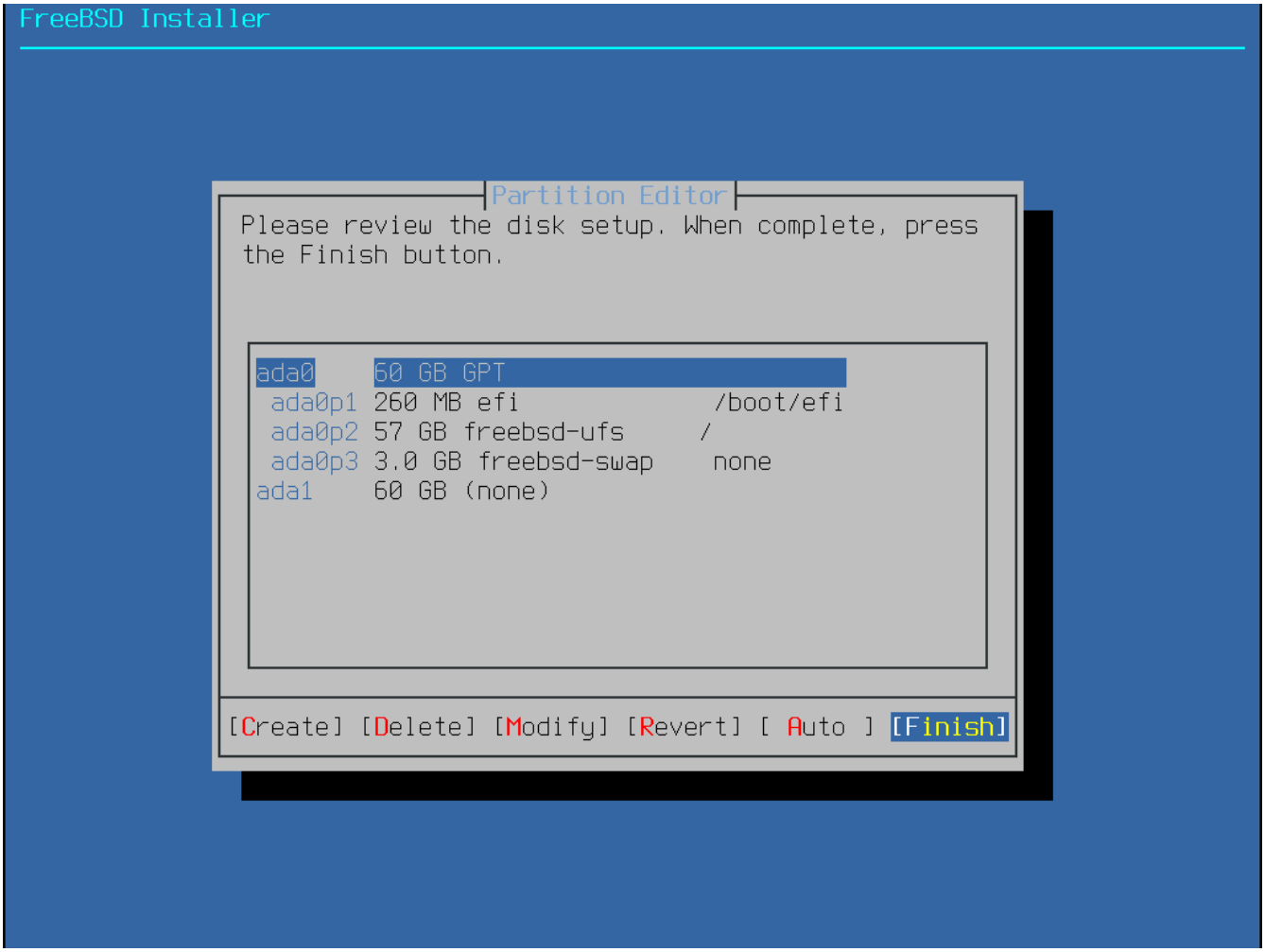


Рисунок 12. Просмотр созданных разделов

2.7.2. Ручное (manual) разбиение

Ручное разбиение начинается с редактора разделов.

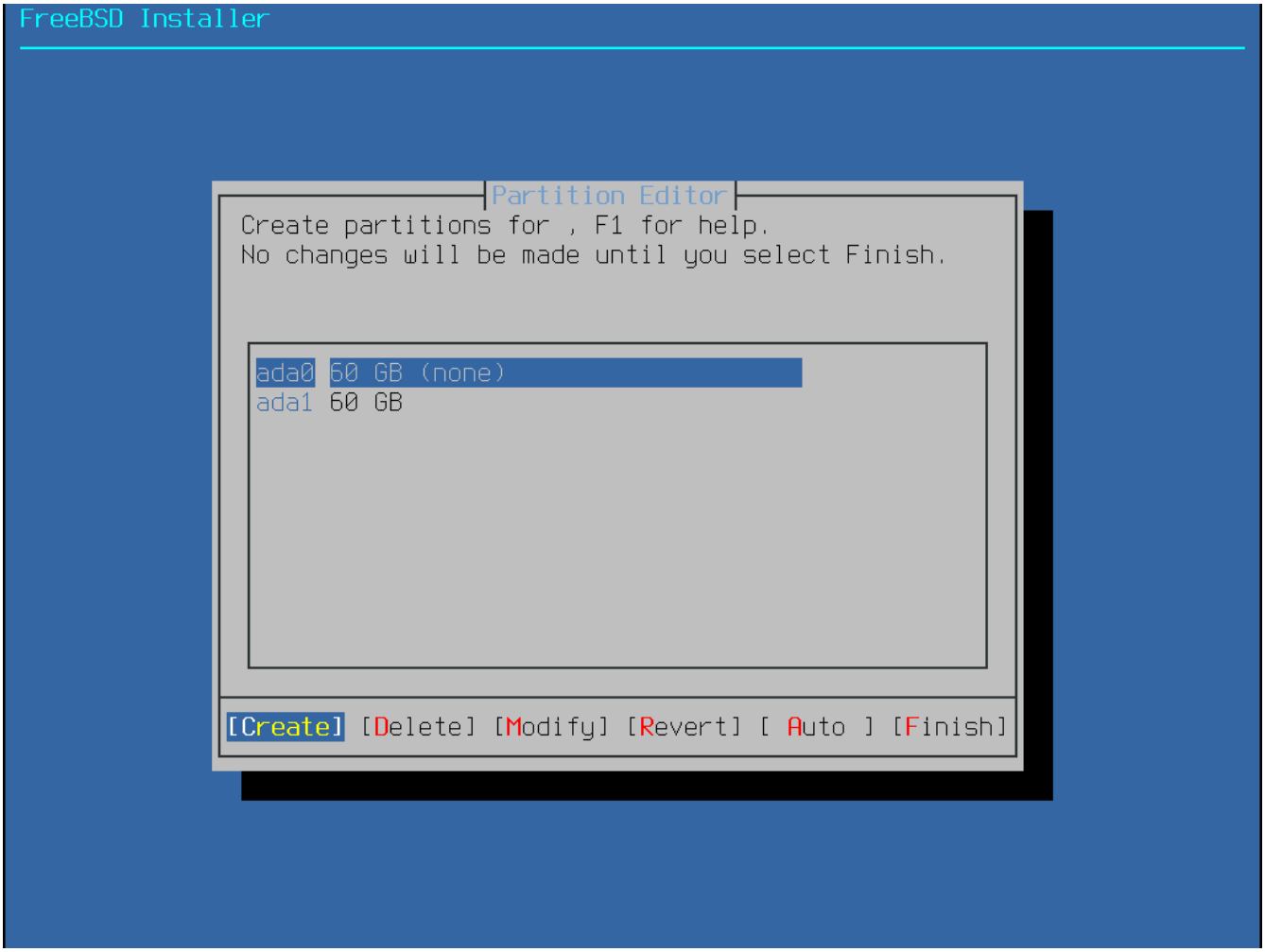


Рисунок 13. Ручное создание разделов

Перемещение подсвечивания на имя устройства (в этом примере - ada0) и выбор **[Create]** приведет вас к меню с перечнем *схем разбиения*.

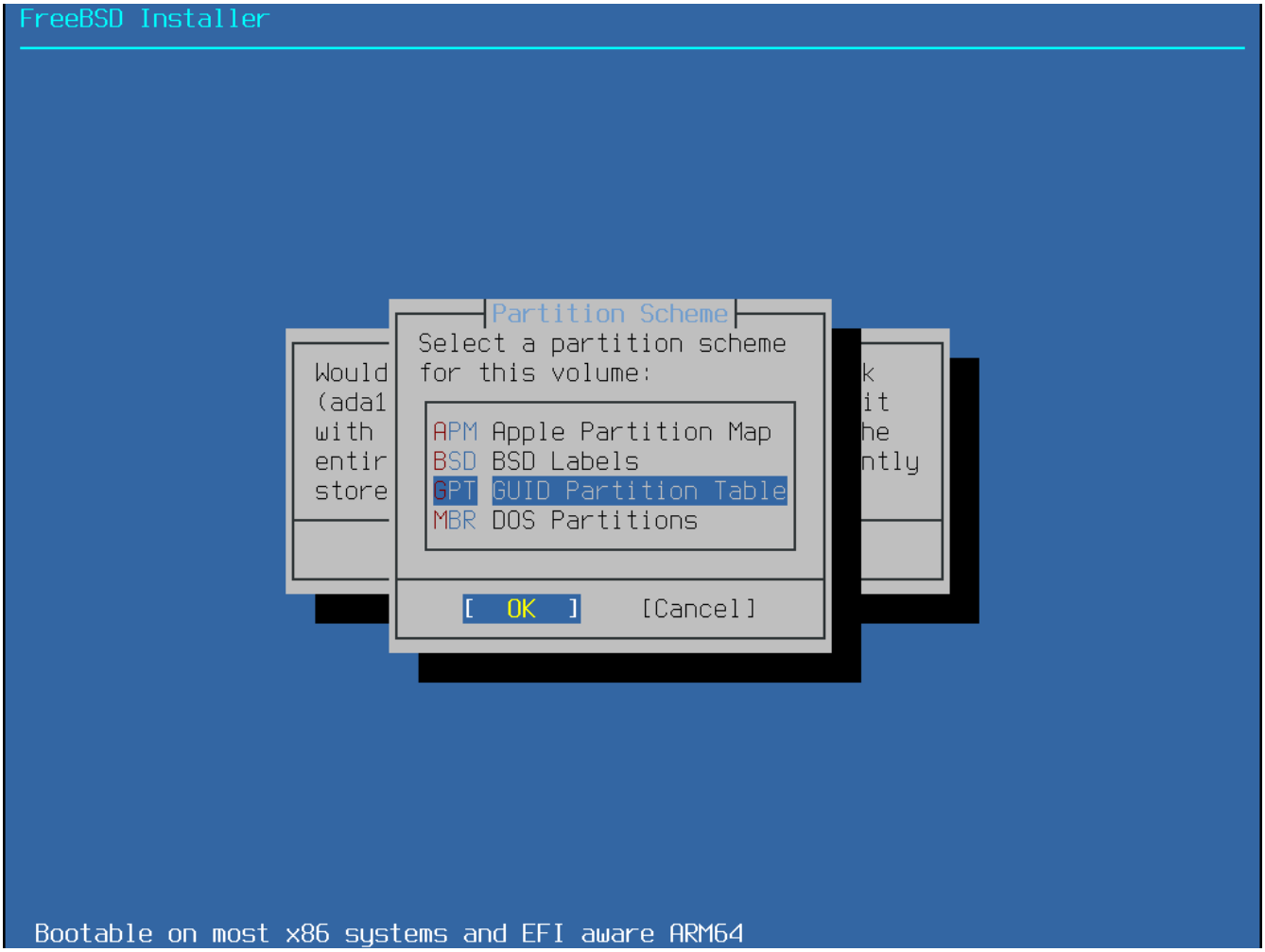


Рисунок 14. Выбор схемы разбиения

Как правило, схема GPT является наиболее подходящей для PC-совместимых компьютеров. Для более старых операционных систем, которые несовместимы с GPT, может потребоваться разбиение MBR. Остальные схемы разбиения в общем используются для нераспространенных или старых компьютерных систем.

Таблица 1. Схемы разбиения

Аббревиатура	Описание
APM	Apple Partition Map, используемая на PowerPC® Macintosh®.
BSD	Метки BSD без MBR, иногда называемые "dangerously dedicated mode". За подробностями обратитесь к bsdlabel(8).
GPT	Таблица разделов GUID.
MBR	Master Boot Record.
PC98	Разновидность MBR, используемая компьютерами NEC PC-98.
VTOC8	Volume Table Of Contents, используемая компьютерами Sun SPARC64 и UltraSPARC.

После того, как схема разбиения определена, повторный выбор **[Create]** приводит к

созданию новых разделов диска.

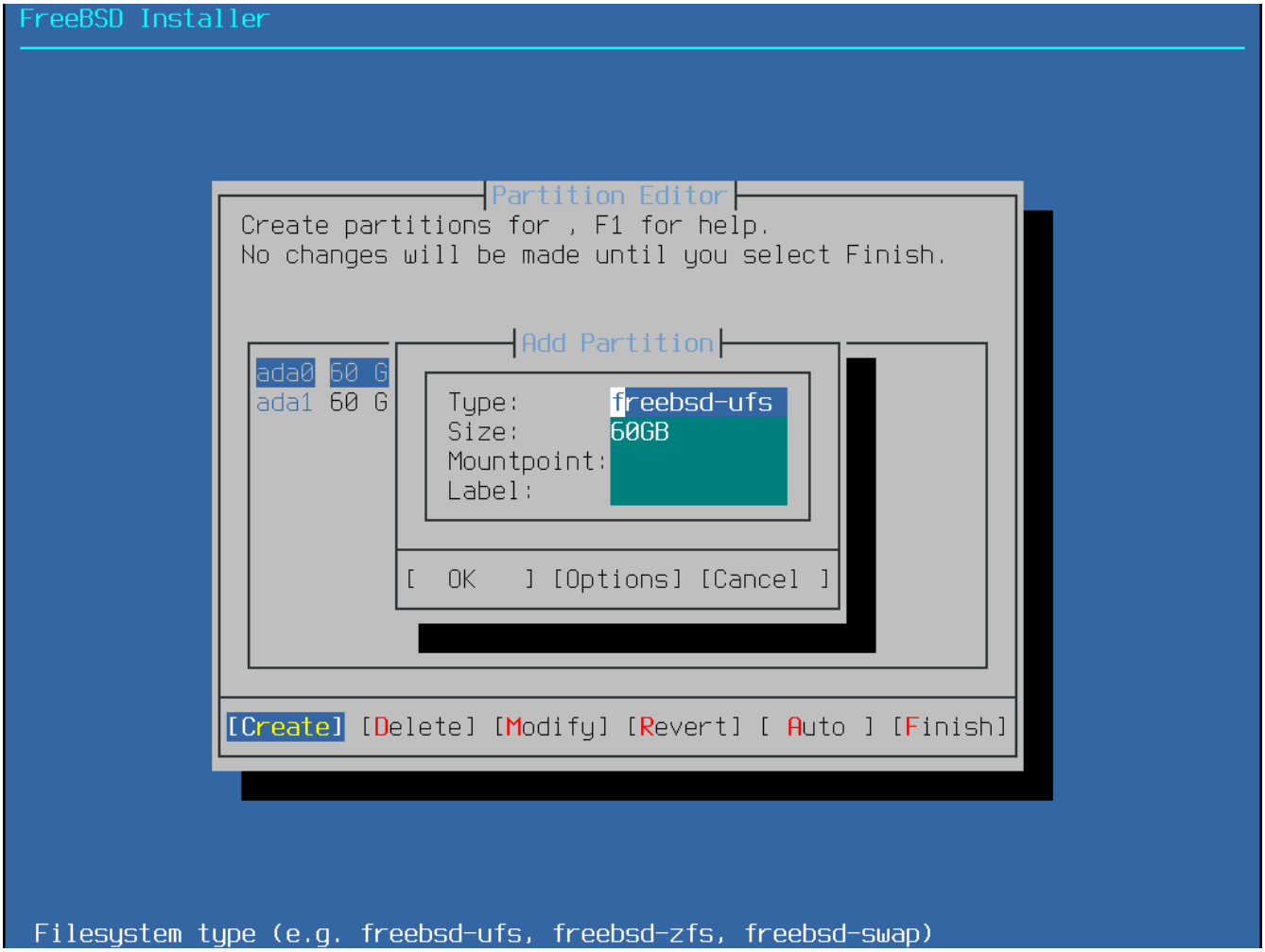


Рисунок 15. Создание нового раздела

Стандартная установка FreeBSD со схемой GPT создаст как минимум три раздела:

Стандартные GPT разделы FreeBSD

- `freebsd-boot` - загрузочный код FreeBSD.
- `freebsd-ufs` - файловая система UFS FreeBSD.
- `freebsd-swap` - FreeBSD область подкачки.

Также необходимо упомянуть, что для разделов, которые будут содержать файловую систему ZFS FreeBSD следует задействовать тип раздела `freebsd-zfs`. Обратитесь к Файловая система ZFS. Сведения об имеющихся в наличии типах разделов GPT содержатся в gpart(8).

Разумеется, возможно создание большего количества разделов с файловыми системами, и некоторые пользователи предпочитают выделять отдельные разделы для таких файловых систем, как /, /var, /tmp, и /usr. Иллюстрация подобного разбиения приведена в Создание традиционного разбиения под файловые системы..

При указании размеров допускается использование общепринятых аббревиатур, таких как *K* для килобайт, *M* для мегабайт, или *G* для гигабайт.

Должное выравнивание секторов обеспечивает наилучшую

производительность, а создание разделов с размерами, кратными 4 Кбайт, помогает обеспечить правильное выравнивание как на дисках с размером сектора 512 байт, так и на устройствах с размером сектора 4 Кбайт. В общем, задание размеров, кратных 1 Мбайт или 1 Гбайт - это наиболее простой способ выполнить выравнивание начал разделов на позицию, кратную 4 Кбайт. Исключение: на данный момент размер раздела *freebsd-boot* не должен превышать 512 Кбайт из-за ограничений загрузочного кода.

В случае, если раздел будет содержать файловую систему, ей потребуется точка монтирования. Если планируется создать единственный раздел UFS, то точка монтирования должна быть /.

Также будет запрошена *метка*. Метка - это имя, присвоенное разделу. Имя устройства или его номер может измениться если устройство будет подключено к другому контроллеру или порту, а метка раздела останется неизменной. Ссылки на метки вместо имён устройств и номеров разделов в файлах типа /etc/fstab делают систему более толерантной к замене оборудования. Метки GPT появляются после подключения диска в каталоге /dev/gpt/. У других схем разбиения есть свои особенности поддержки меток, и их метки располагаются в других подкаталогах каталога /dev/.

Во избежание конфликтов имен меток используйте уникальные имена для каждой файловой системы. Несколько букв, взятых от имени компьютера, его назначения или размещения может быть добавлено к метке. Например, корневому разделу UFS для компьютера в лаборатории можно присвоить метку `labroot` или `rootfs-lab`.

Пример 3. Создание традиционного разбиения под файловые системы.

Для традиционного разбиения, в котором каталоги /, /var, /tmp и /usr представляют собой отдельные файловые системы на их собственных разделах, создайте схему разбиения GPT, потом создайте разделы, как это указано ниже. Показанные размеры разделов являются типичными для жесткого диска размером 20Гб. Если диск большего размера, то будет уместным отвести больше места для раздела подкачки или для раздела с файловой системой /var. Задействованные в этом примере метки имеют префикс `ex`, от слова "example", вам же рекомендуется использовать другие уникальные имена меток.

По умолчанию, загрузчик gptboot FreeBSD ожидает, что первый найденный раздел UFS будет корневым разделом (/).

Тип раздела	Размер	Точка монтирования	Метка
`freebsd-boot`	`512K`		
`freebsd-ufs`	`2G`	/	`exrootfs`
`freebsd-swap`	`4G`		`exswap`
`freebsd-ufs`	`2G`	/var	`exvarfs`

Тип раздела	Размер	Точка монтирования	Метка
freebsd-ufs	1G	/tmp	extmpfs
freebsd-ufs	соглашайтесь со значением по умолчанию (оставшаяся часть объема диска)	/usr	exusrfs

Для продолжения установки по завершении создания необходимых разделов выберите **[Finish]**.

2.8. Завершение установки

Следующий шаг - ваш последний шанс прервать установку и предотвратить изменение данных на жестком диске.

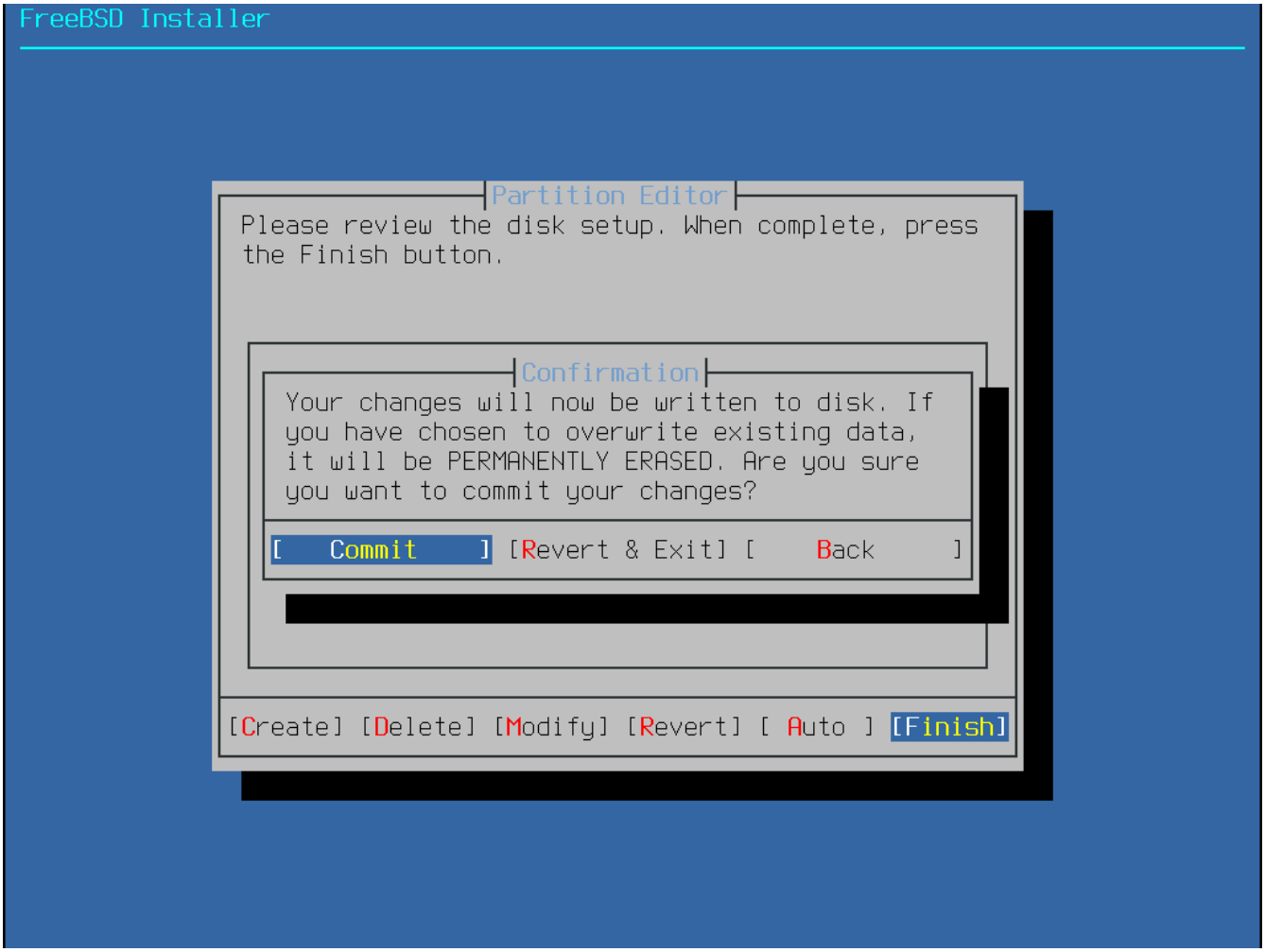


Рисунок 16. Заключительное подтверждение

Для продолжения выберите **[Commit]**. Если необходимо внести изменения, для возвращения к редактору разделов нажмите **[Back]**. Выбор **[Revert & Exit]** дает возможность выйти из установщика без внесения изменений на жесткий диск.

Продолжительность установки варьируется в зависимости от выбранного дистрибутива, способа установки и быстродействия компьютера. Далее последует очередь сообщений, информирующих о ходе установки.

Первым делом установщик запишет информацию о разделах на диск и отформатирует разделы посредством `newfs`.

Если выполняется установка по сети, то bsdinstall продолжит загрузку необходимых файлов дистрибутива.

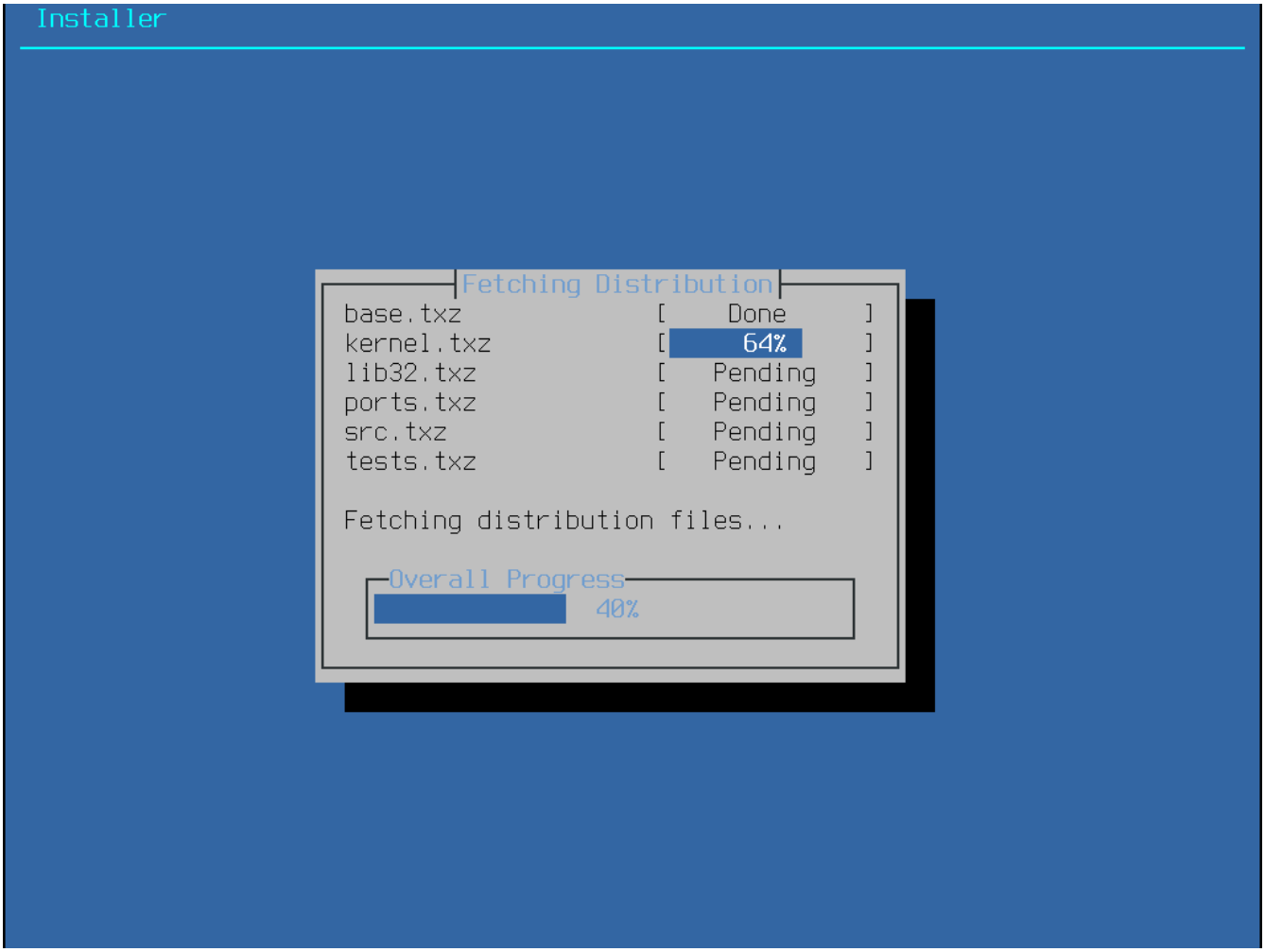


Рисунок 17. Загрузка файлов дистрибутива

Далее последует проверка целостности файлов дистрибутива, чтобы удостовериться, что они не были повреждены во время загрузки или чтения с установочного носителя.

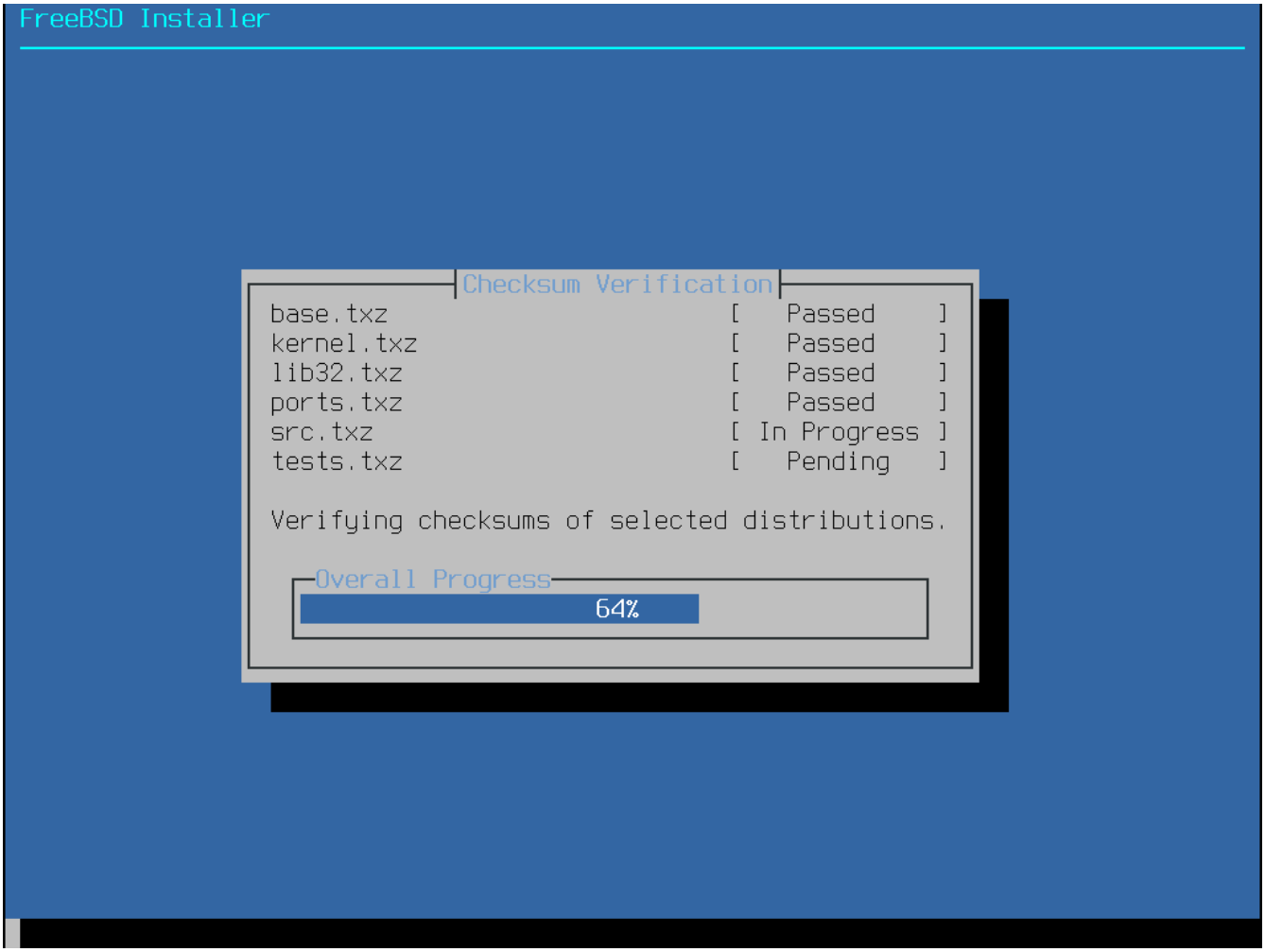


Рисунок 18. Проверка файлов дистрибутива

И в заключение, проверенные файлы распаковываются на диск.

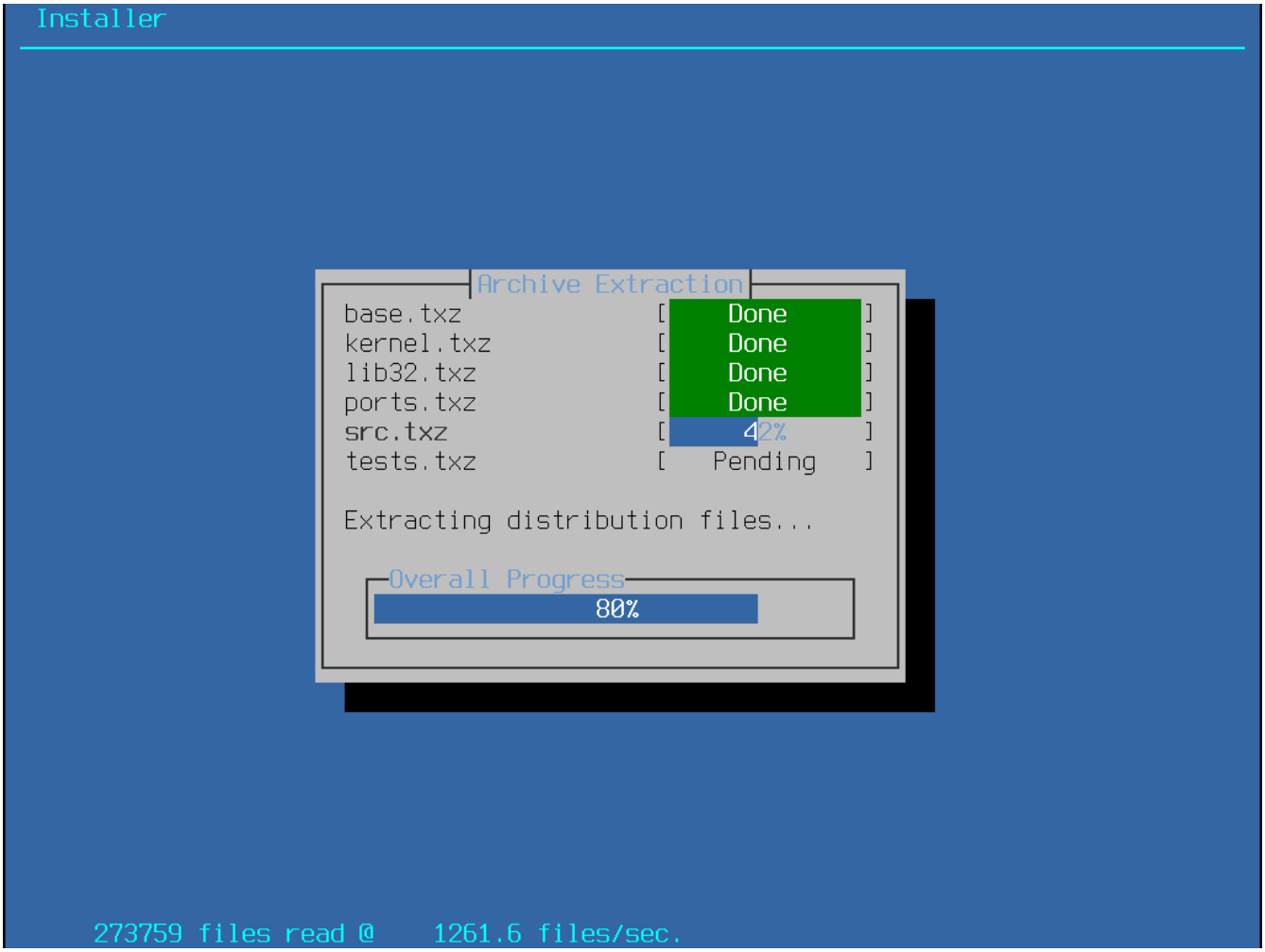


Рисунок 19. Извлечение файлов дистрибутива

Как только запрошенные файлы дистрибутива распакуются, bsdinstall приступит к выполнению послеустановочных конфигурационных задач (смотрите После установки).

2.9. После установки

После успешной установки FreeBSD последуют меню настройки различных опций. Настройки опций могут быть изменены путем повторного входа в соответствующие разделы финального конфигурационного меню перед загрузкой в свежеустановленную систему FreeBSD.

2.9.1. Установка пароля пользователя `root`

Установка пароля пользователя `root` - обязательна. Заметьте, что во время ввода пароля набираемые символы не отображаются на экране. После ввода будет запрошен повторный ввод пароля. Это помогает предотвратить опечатки при наборе.

```
FreeBSD Installer
=========================

Please select a password for the system management account (root):
Typed characters will not be visible.
Changing local password for root
New Password:
Retype New Password:
```

Рисунок 20. Установка пароля пользователя `root`

Настройки опций продолжатся после успешной установки пароля.

2.9.2. Настройка сетевых интерфейсов

Настройка сетевых интерфейсов будет опущена в случае, если она уже была выполнена как часть подготовки при установке *bootonly*.

Далее будет отображен перечень всех сетевых интерфейсов, найденных на компьютере. Выберите тот, который планируете настроить.

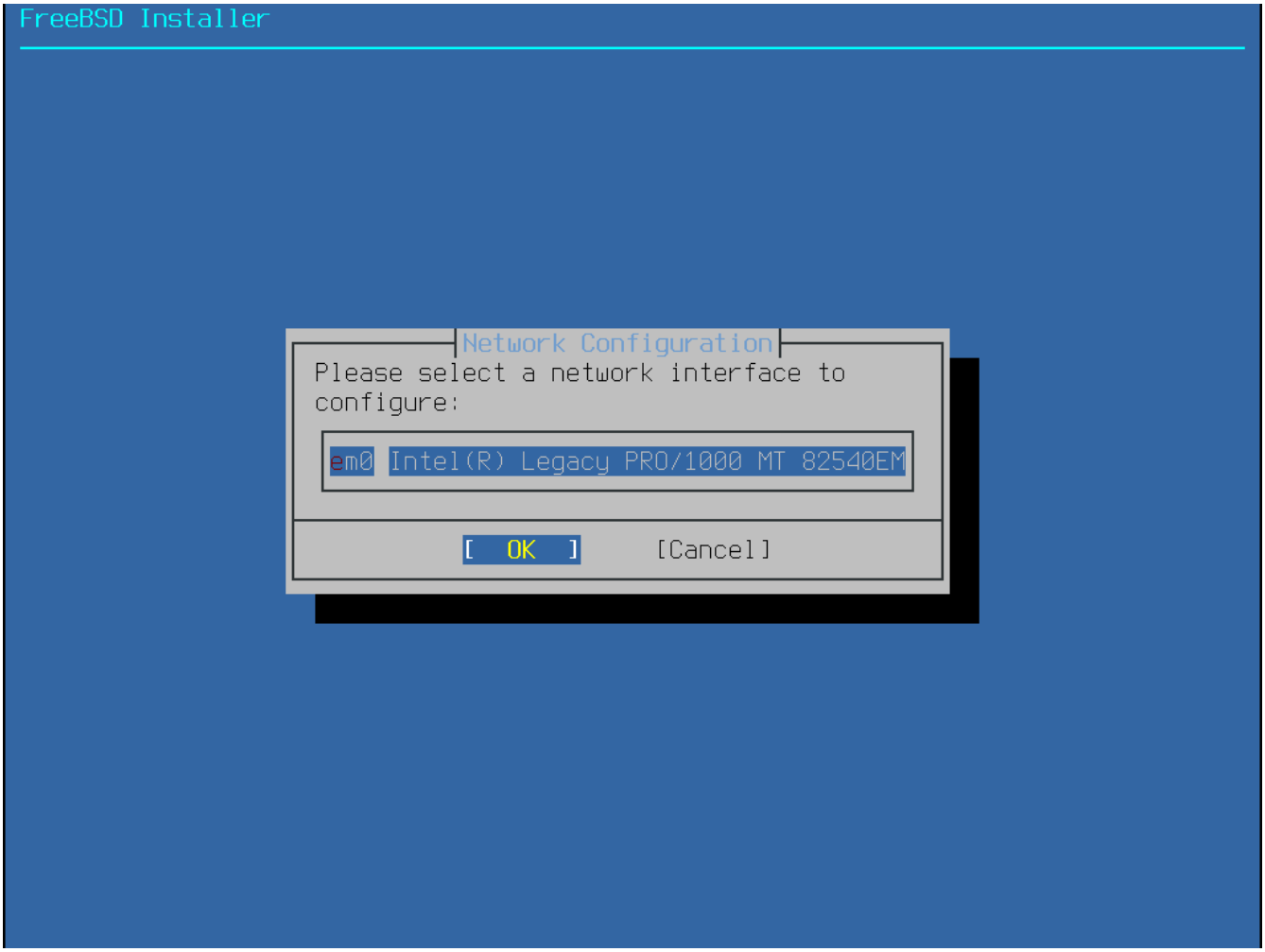


Рисунок 21. Выберите сетевой интерфейс

2.9.2.1. Настройка беспроводного сетевого интерфейса

Если выбран беспроводной сетевой интерфейс, то для подключения к сети потребуется ввести параметры сетевой идентификации и безопасности.

Беспроводные сети распознаются по так называемому Service Set Identifier, или SSID. SSID - это краткое уникальное имя, присваиваемое каждой сети.

Большинство беспроводных сетей шифруют передаваемые данные чтобы защитить их от неавторизированного прослушивания. Настоятельно рекомендуется применять стандарт WPA2. Более старые стандарты, например WEP, не обеспечивают достаточного уровня безопасности.

Первым делом, при подключении к беспроводной сети необходимо выполнить поиск беспроводных точек доступа.

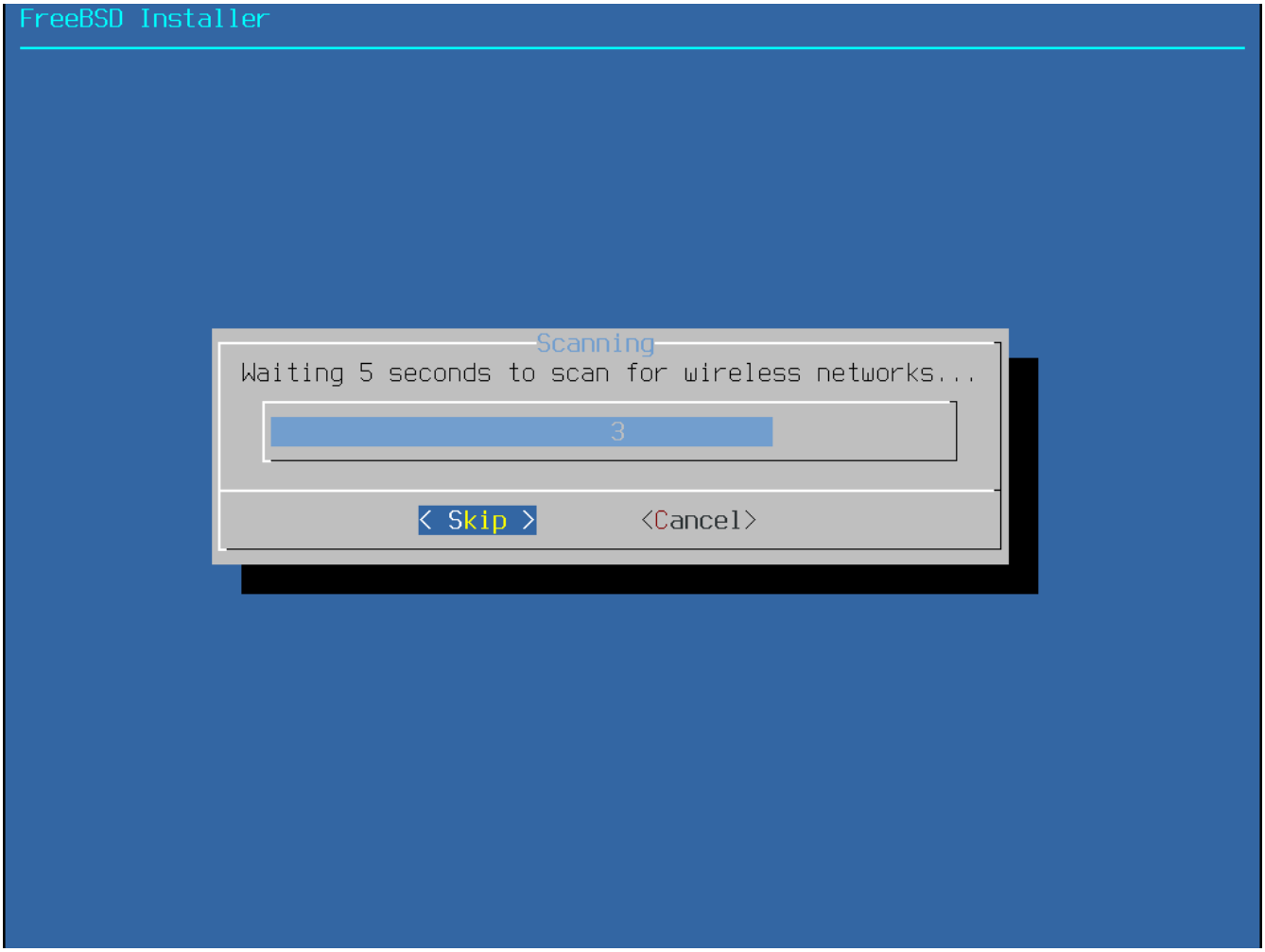


Рисунок 22. Поиск беспроводных точек доступа

Список найденных сетей будет содержать несколько SSID с описанием типов шифрования, поддерживаемых обнаруженными беспроводными сетями. Если искомый SSID не появляется в списке, то запустите сканирование повторно, выбрав **[Rescan]**. Если искомая сеть снова не появится в списке, проверьте соединение с антенной или попробуйте разместить компьютер ближе к точке доступа. Запускайте повторный поиск после каждого вашего действия.

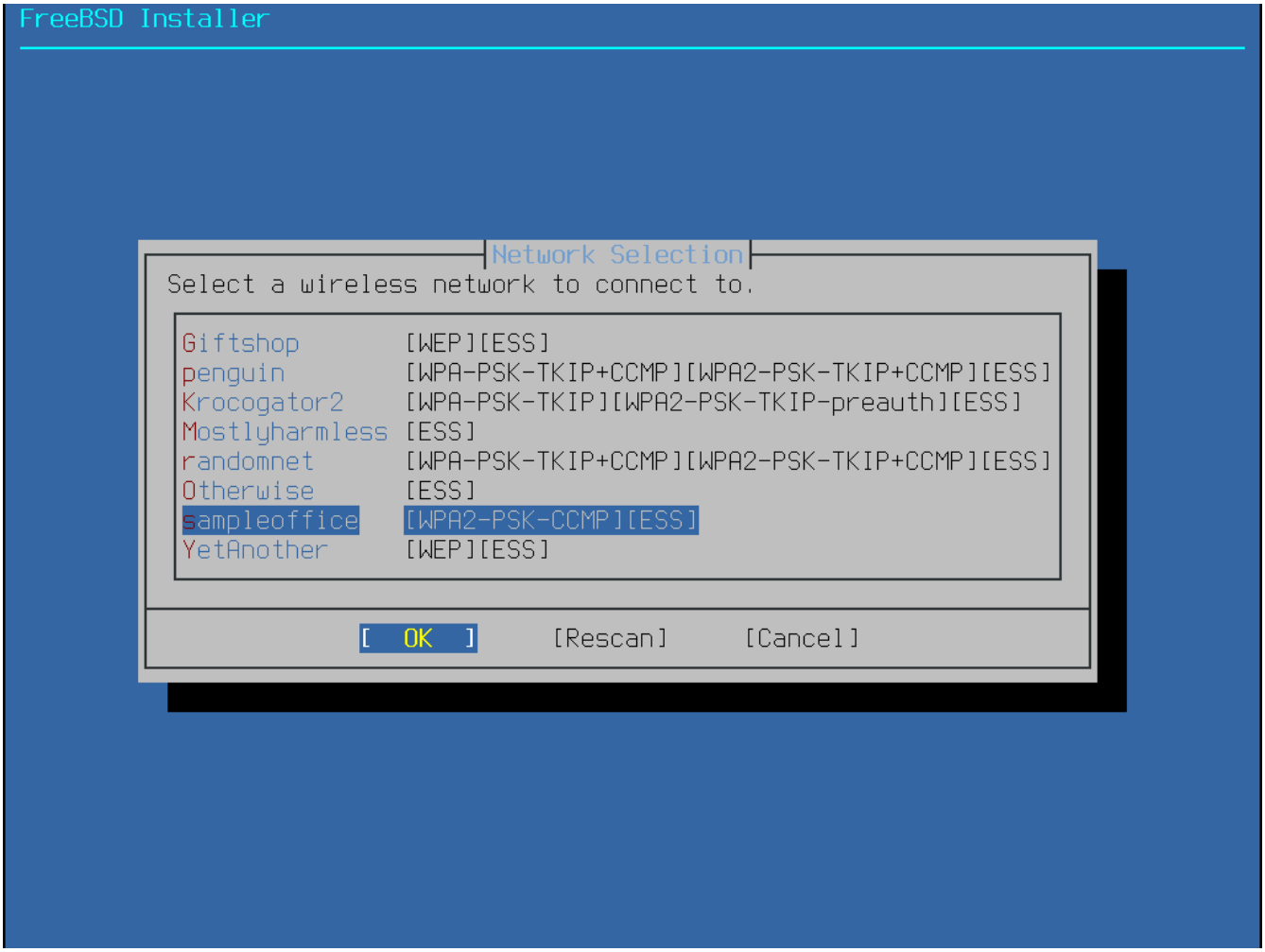


Рисунок 23. Выбор беспроводной сети

После выбора сети потребуется ввести дополнительную информацию о соединении. Для WPA2 потребуется пароль (также известный как Pre-Shared Key или PSK). В целях безопасности набираемые в поле ввода пароля символы на экране отображаются звездочками.

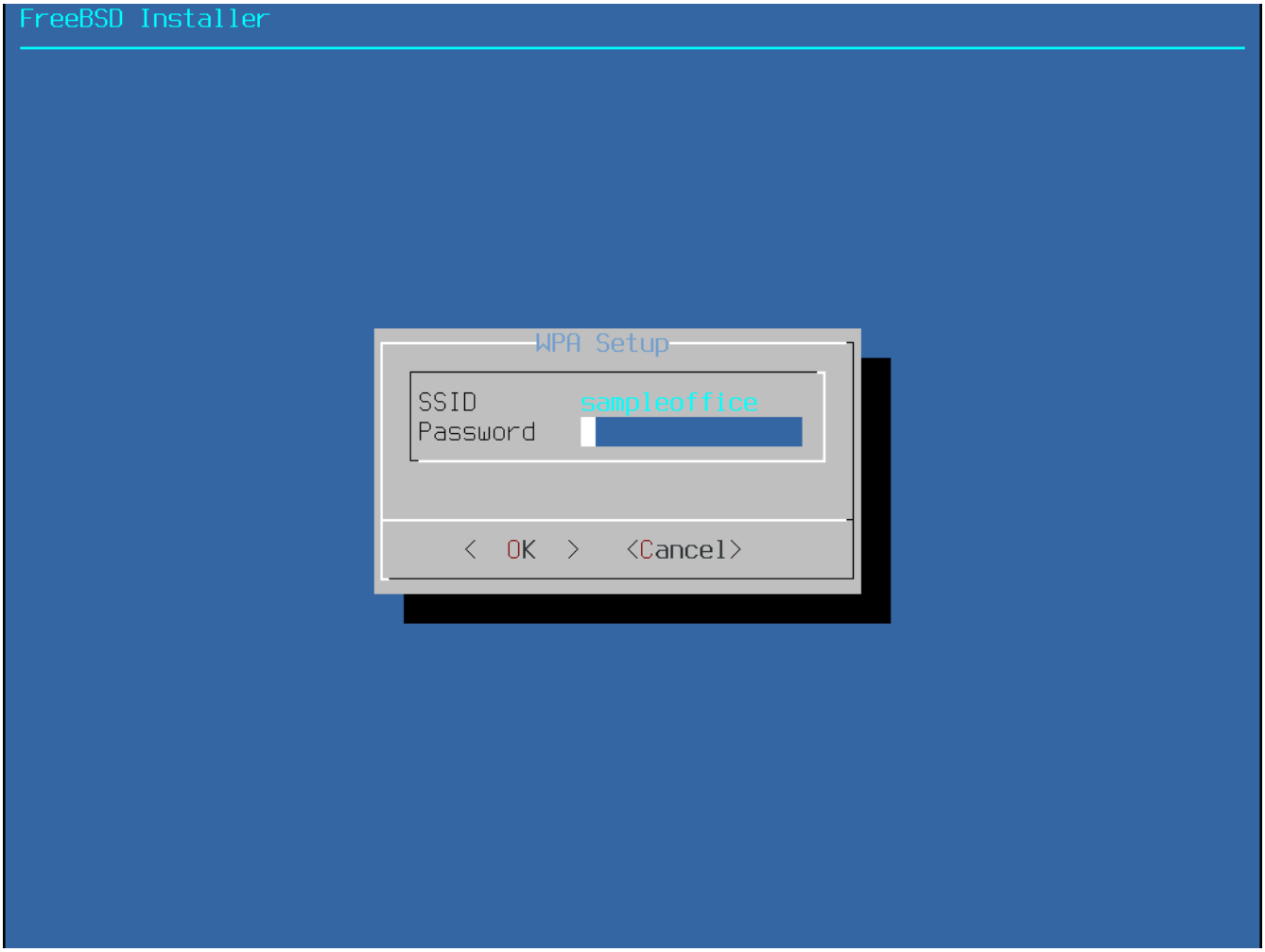


Рисунок 24. Настройка WPA2

После выбора беспроводной сети и ввода сопутствующих параметров безопасности последует настройка сетевых протоколов.

2.9.2.2. Настройка сетевых протоколов: IPv4

Определитесь, есть ли необходимость в подключении к сети IPv4. Это наиболее распространённый сетевой протокол.

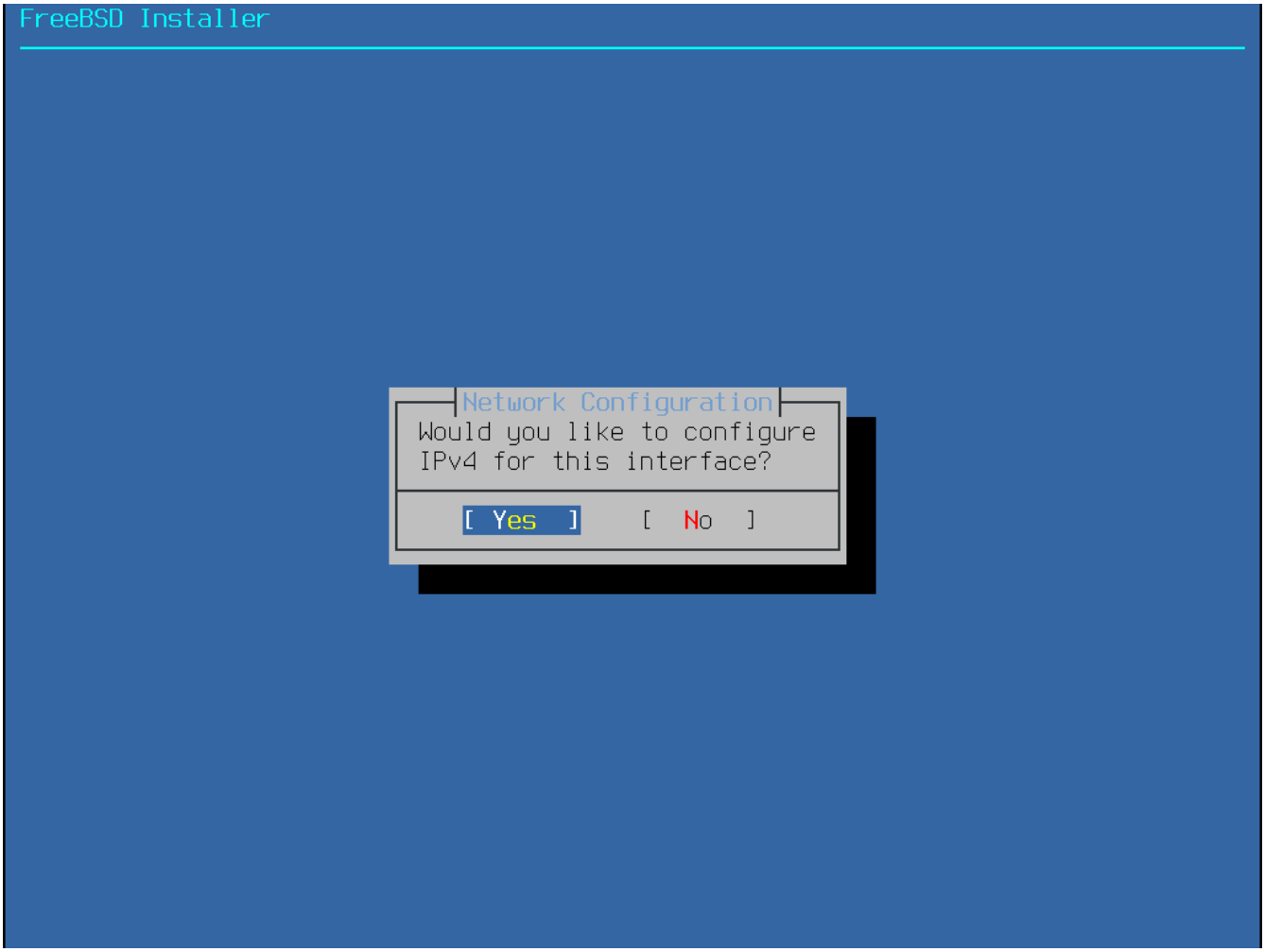


Рисунок 25. Выберите настройку протокола IPv4

Существует два способа настройки протокола IPv4 на сетевом интерфейсе. Сервис *DHCP* автоматически установит корректную конфигурацию сетевого интерфейса, и это - предпочтительный способ настройки. *Статическая* конфигурация требует ручного ввода настроек протокола IPv4.

> Не пытайтесь ввести произвольные данные, они работать не будут. Получите перечисленную в Соберите информацию о сетевых настройках информацию у сетевого администратора или поставщика услуг Интернет.

2.9.2.2.1. Настройка протокола IPv4 на сетевом интерфейсе посредством DHCP

Если в сети есть сервис DHCP, то для автоматического конфигурирования сетевого интерфейса выберите **[Yes]**.

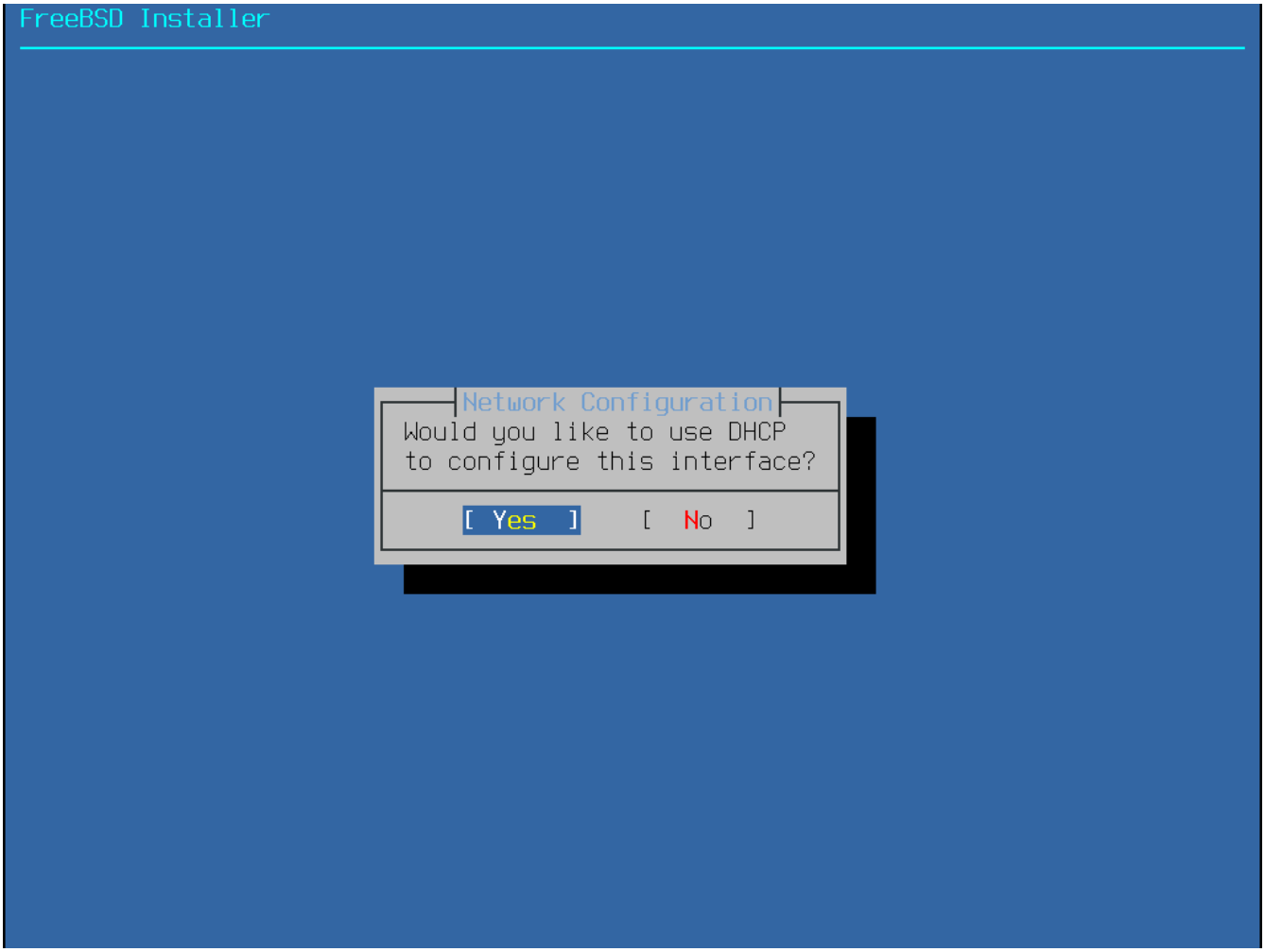


Рисунок 26. Выбор настройки протокола IPv4 посредством DHCP

2.9.2.2.2. Статическая настройка протокола IPv4 на сетевом интерфейсе

Статическая настройка сетевого интерфейса требует ввода некоторой информации о подключении IPv4.

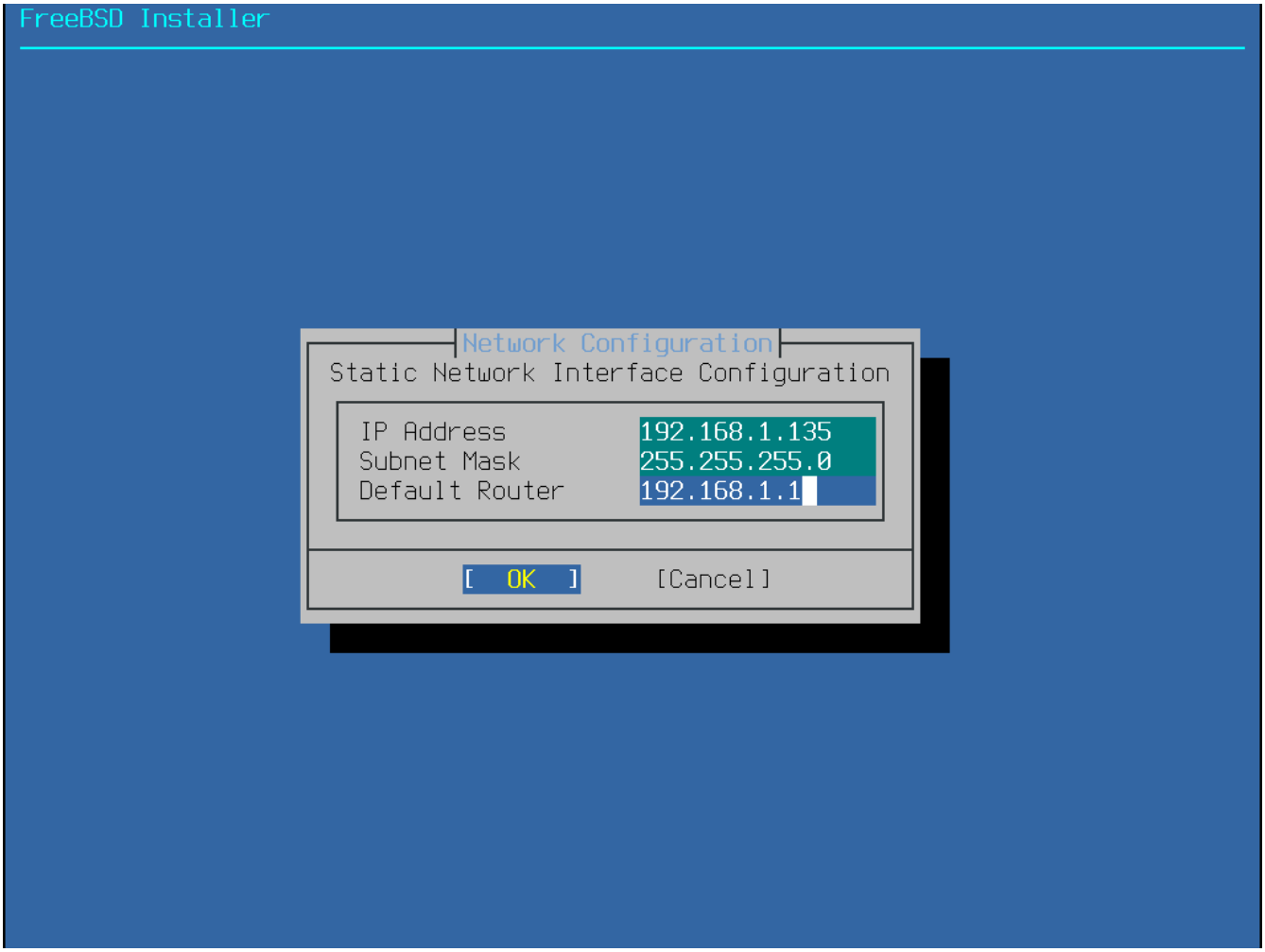


Рисунок 27. Статическая настройка IPv4 на сетевом интерфейсе

- `IP Address` - адрес IPv4, который должен быть установлен на этом компьютере. Этот адрес должен быть уникальным и не должен использоваться другим оборудованием в локальной сети.
- `Subnet Mask` - маска, используемая в локальной сети. Часто маска имеет значение `255.255.255.0`.
- `Default Router` - IP адрес маршрутизатора для этого подключения. Обычно этот адрес установлен на маршрутизаторе или ином сетевом оборудовании, которое соединяет локальную сеть с сетью Интернет. Также известен, как *шлюз по умолчанию (default gateway)*.

2.9.2.3. Настройка сетевых протоколов: IPv6

IPv6 это более новый сетевой протокол. Если есть необходимость и возможность подключения к сети IPv6, выберите в этом меню **[Yes]**.

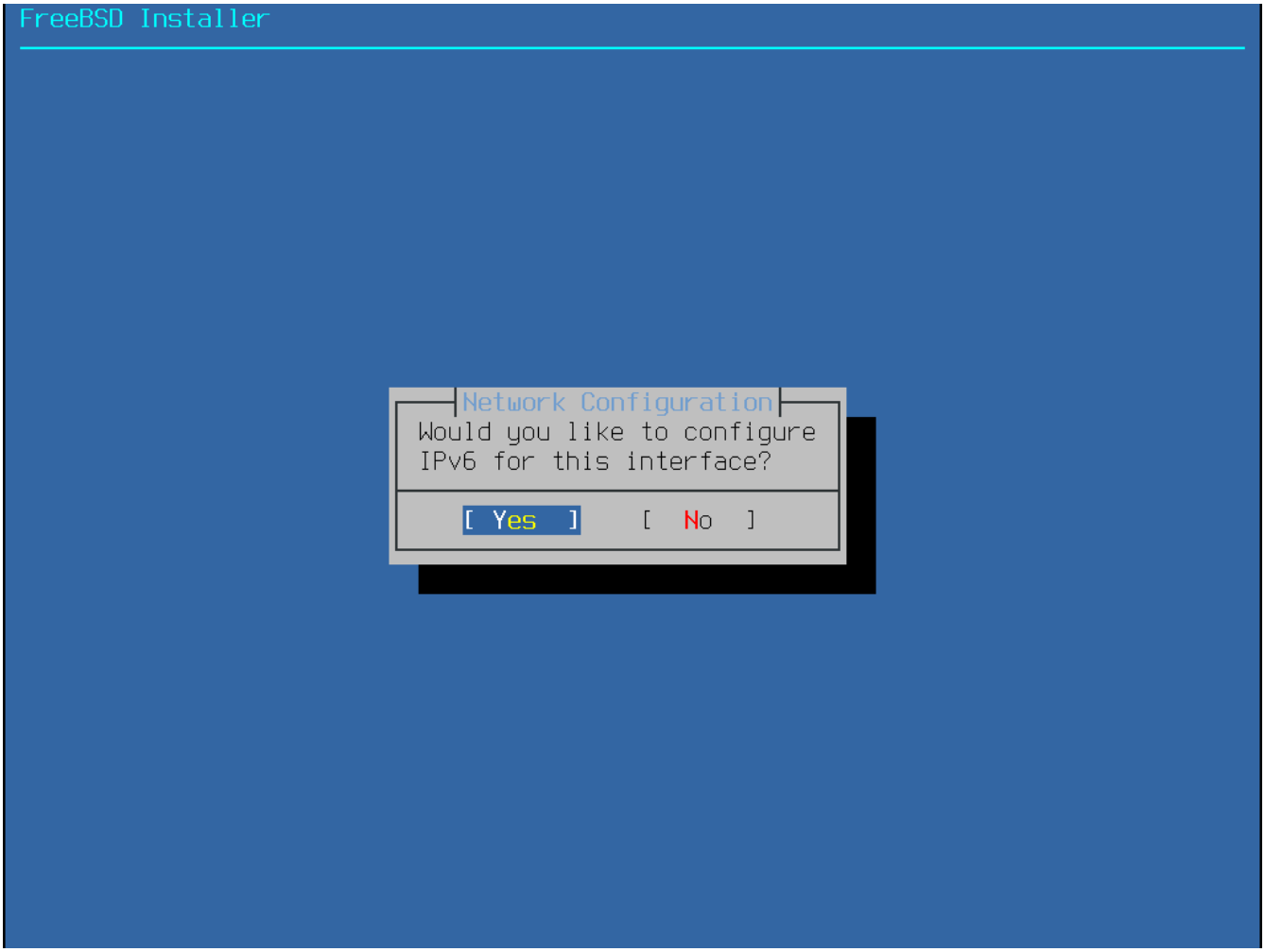


Рисунок 28. Выберите настройку протокола IPv6 на сетевом интерфейсе

Для протокола IPv6 также возможны два способа настройки сетевого интерфейса. *SLAAC* или *StateLess Address AutoConfiguration* автоматически установит корректные настройки сетевого интерфейса. *Статическая* конфигурация требует ручного ввода настроек протокола IPv6.

2.9.2.3.1. IPv6 SLAAC

SLAAC позволяет сетевому элементу запросить у локального маршрутизатора необходимую для автоматической настройки информацию. За подробностями обратитесь к RFC4862.

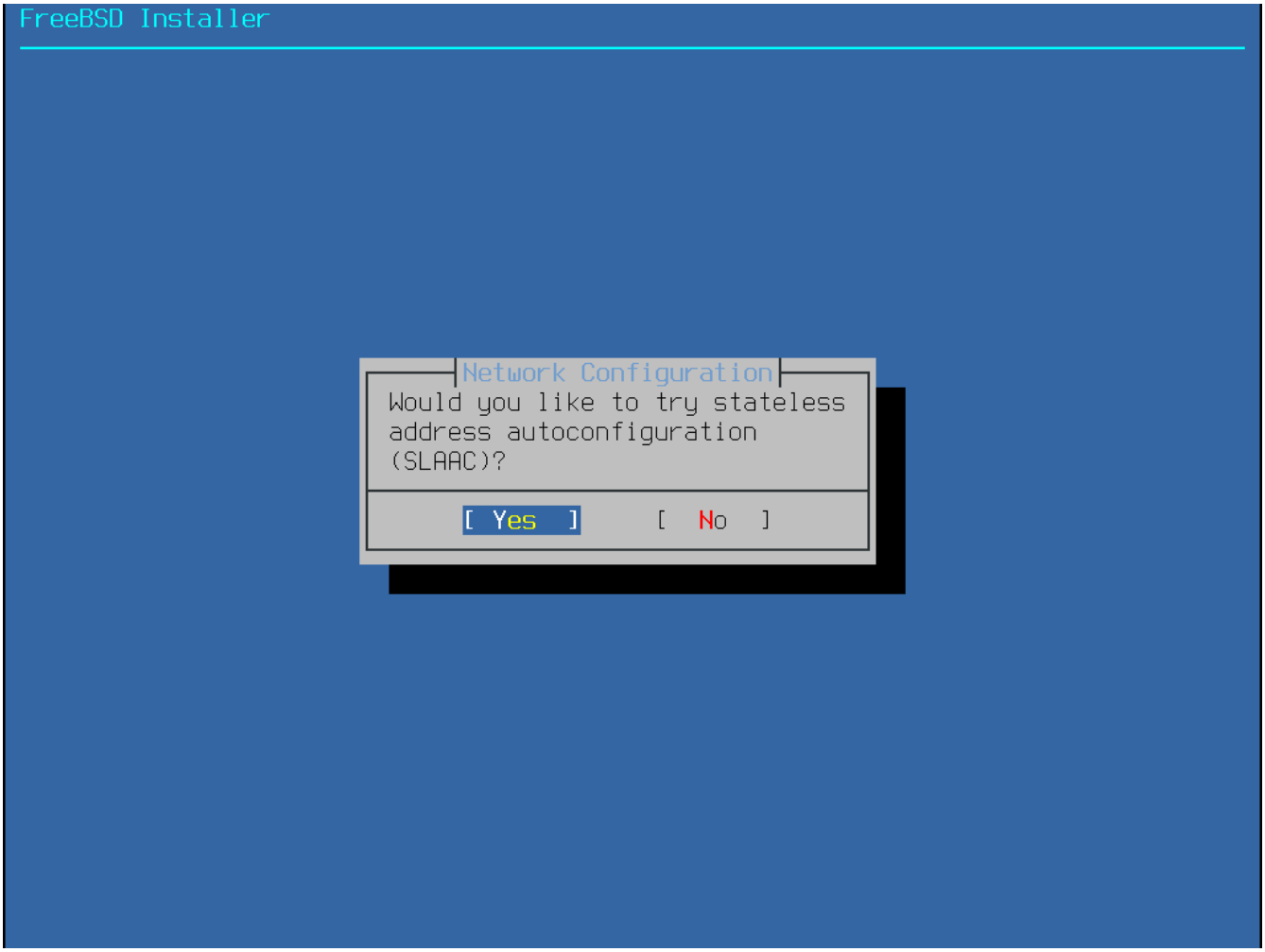


Рисунок 29. Выберите настройку протокола IPv6 посредством SLAAC

2.9.2.3.2. Статическая настройка протокола IPv6 на сетевом интерфейсе

Статическая настройка сетевого интерфейса требует ручного ввода информации о IPv6 подключении.

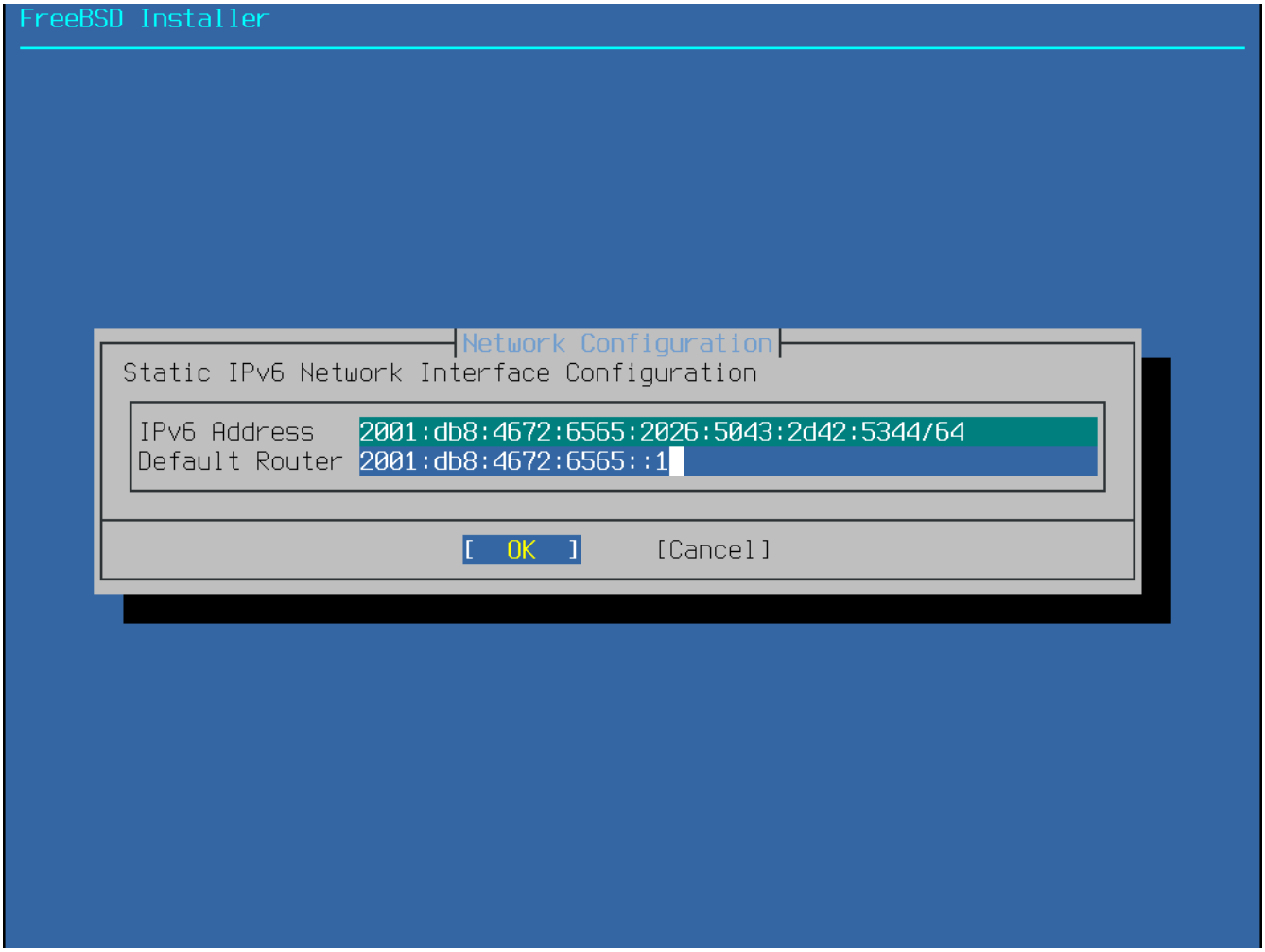


Рисунок 30. Статическая настройка протокола IPv6

- `IPv6 Address` - вводимый вручную IP адрес, который присвоен этому компьютеру. Этот адрес должен быть уникальным и не должен быть занят другим оборудованием в локальной сети.
- `Default Router` - IPv6 адрес маршрутизатора для этой сети. Обычно, это адрес маршрутизатора или другого сетевого оборудования, которое соединяет локальную сеть с сетью Интернет. Также известен как *шлюз по умолчанию*.

2.9.2.4. Настройка Резолвера DNS

Domain Name System (или *DNS*) Резолвер выполняет преобразования имен хостов в сетевые адреса, а также преобразования сетевых адресов в имена хостов. Если для автоматического конфигурирования сетевого интерфейса использовался DHCP или SLAAC, то информация о Резолвере может уже присутствовать в системе. Иначе, впишите в поле Search имя локального домена. DNS #1 и DNS #2 - это IP адреса локальных серверов DNS. По крайней мере один сервер должен быть указан.

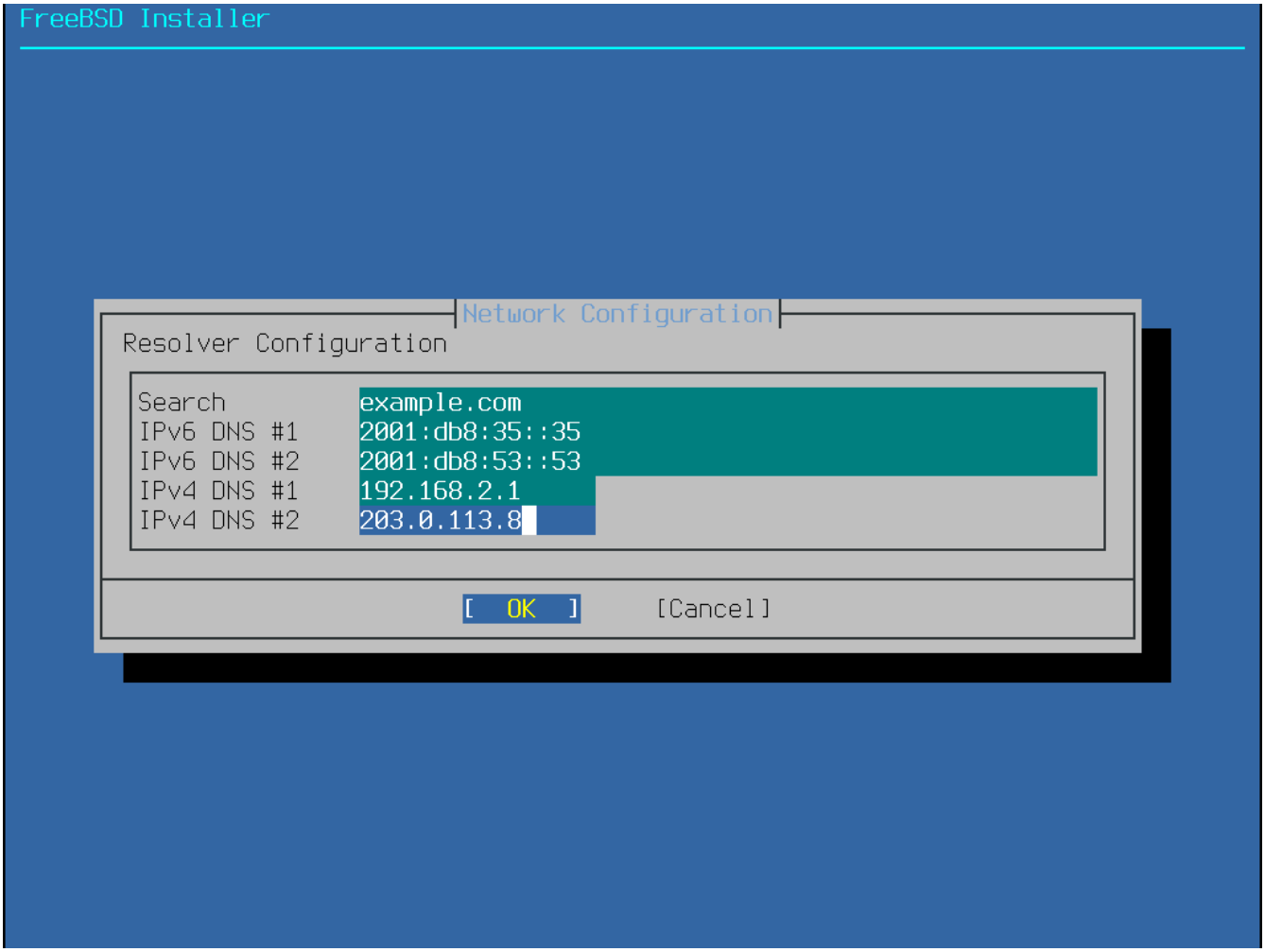


Рисунок 31. Конфигурирование Резолвера DNS

2.9.3. Установка часового пояса

Установка часового пояса для вашей машины позволит ей автоматически корректировать время согласно местным законам и правильно выполнять остальные зависимые от часового пояса функции.

Данный пример верен для машины, находящейся в восточном часовом поясе Соединенных Штатов. Разумеется, ваши настройки должны соответствовать вашему географическому местоположению.

Рисунок 32. Выбор местного времени или времени UTC

Выберите **[Yes]** или **[No]** согласно тому, как настроены часы вашего компьютера, далее нажмите `Enter`. Если вы не знаете какое значение выбрать, UTC или местное, то нажмите **[No]** для того, чтобы выбрать наиболее распространённую конфигурацию - местное время.

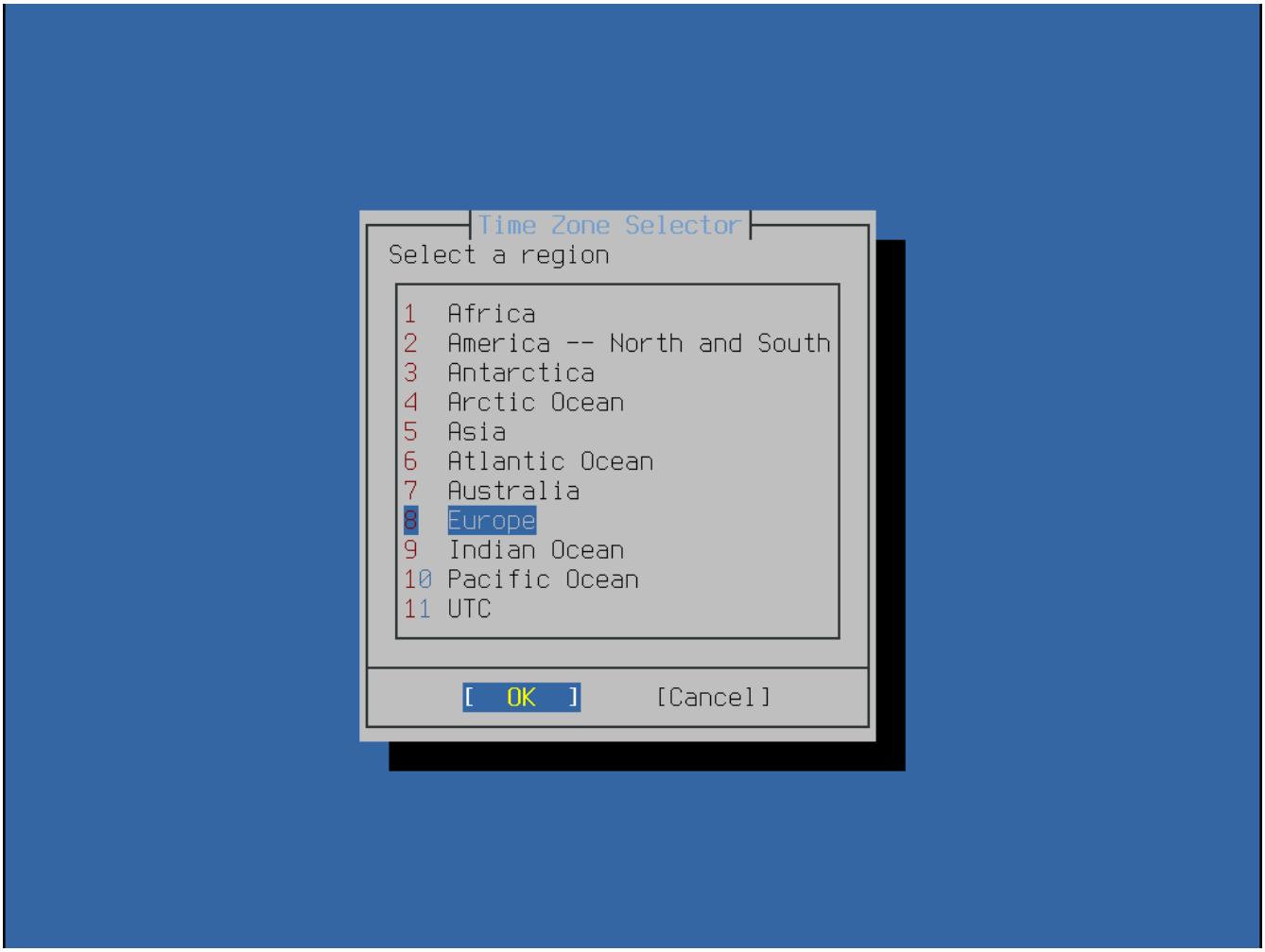


Рисунок 33. Выберите регион

Соответствующий регион выбирается при помощи клавиш навигации и подтверждается нажатием клавиши `Enter`.

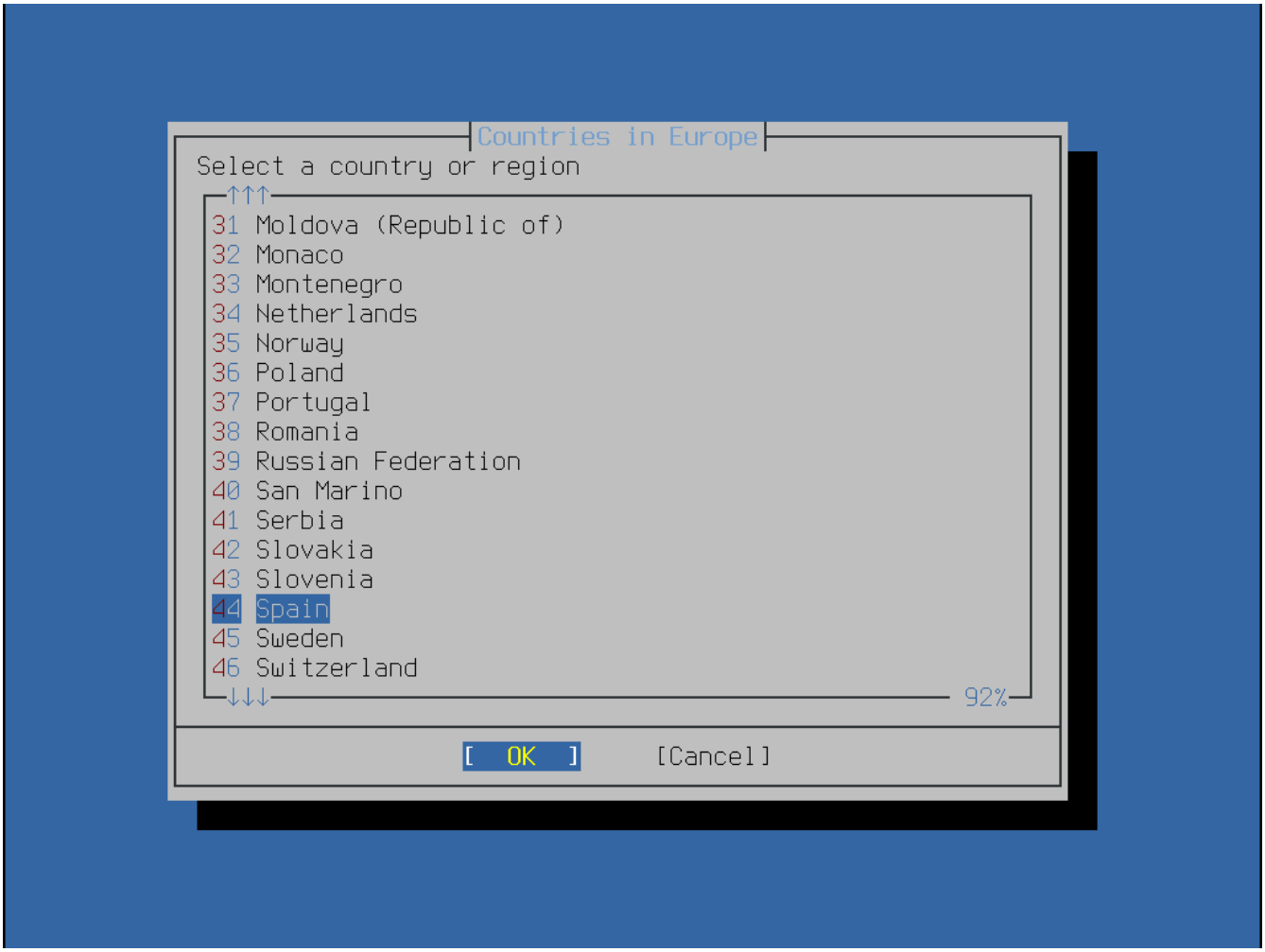


Рисунок 34. Выберите страну

Выберите необходимую страну при помощи клавиш навигации и подтвердите выбор клавишей `Enter`.

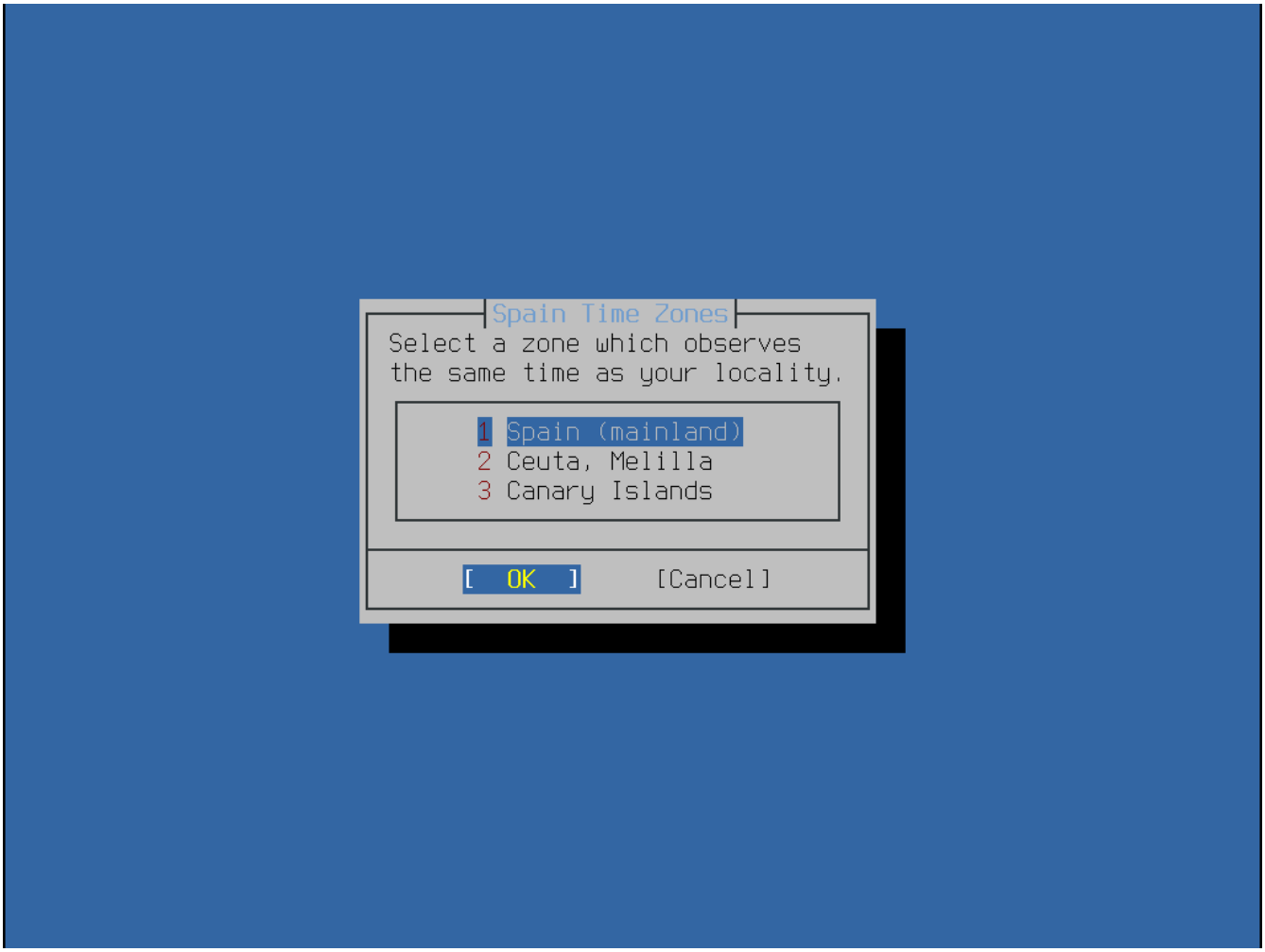


Рисунок 35. Выберите часовой пояс

Соответствующий часовой пояс выбирается клавишами навигации и подтверждается нажатием клавиши `Enter`.

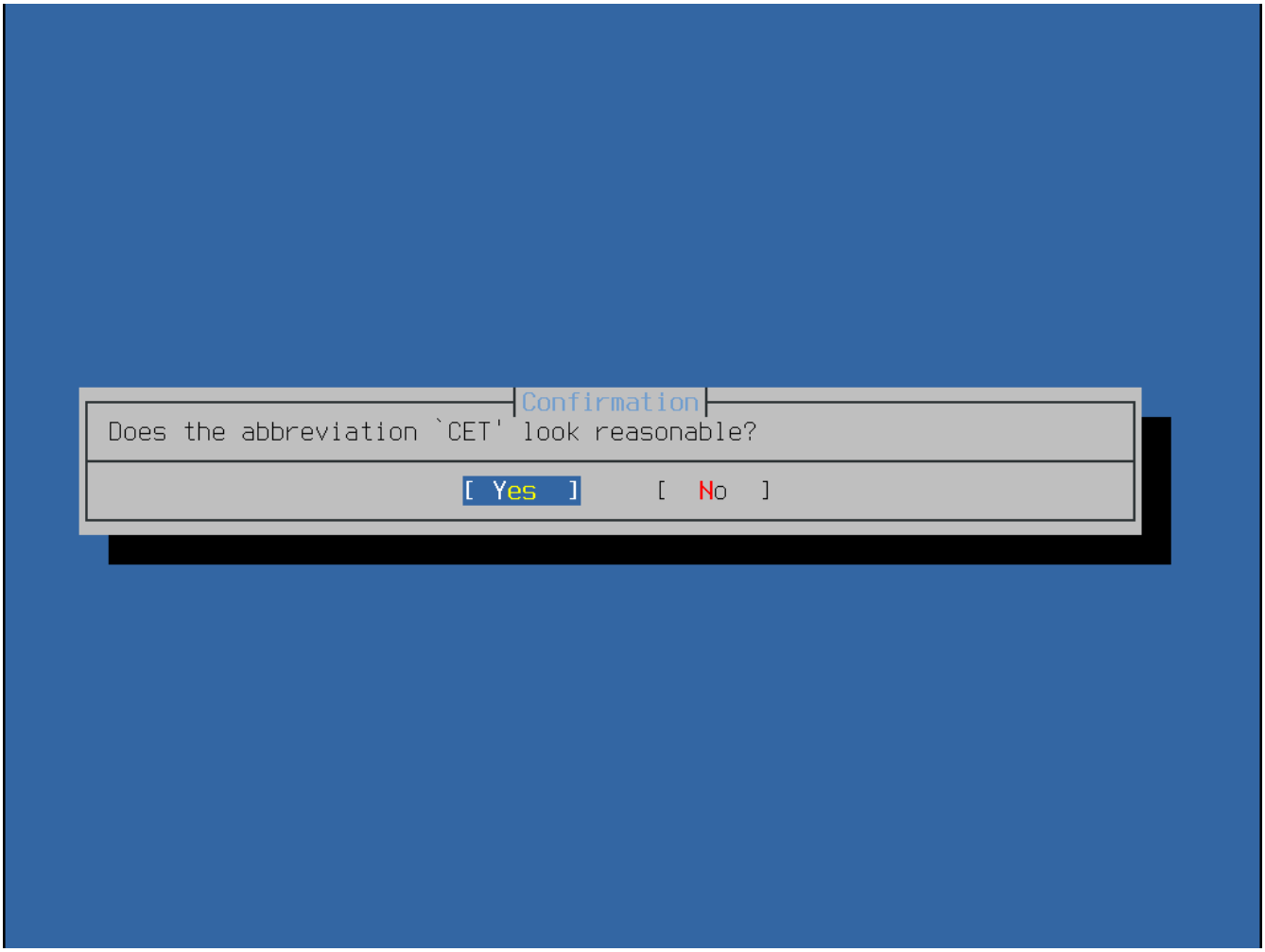


Рисунок 36. Подтверждение выбора часового пояса

Подтвердите, что аббревиатура для часового пояса является приемлемой. Если данная опция настроена верно, то нажмите клавишу `Enter` для продолжения послеустановочного конфигурирования.

2.9.4. Активирование дополнительных сетевых сервисов

На данном этапе установщик предлагает отметить дополнительные сетевые сервисы, которые будут запускаться при загрузке системы. Все нижеследующие сервисы не являются обязательными.

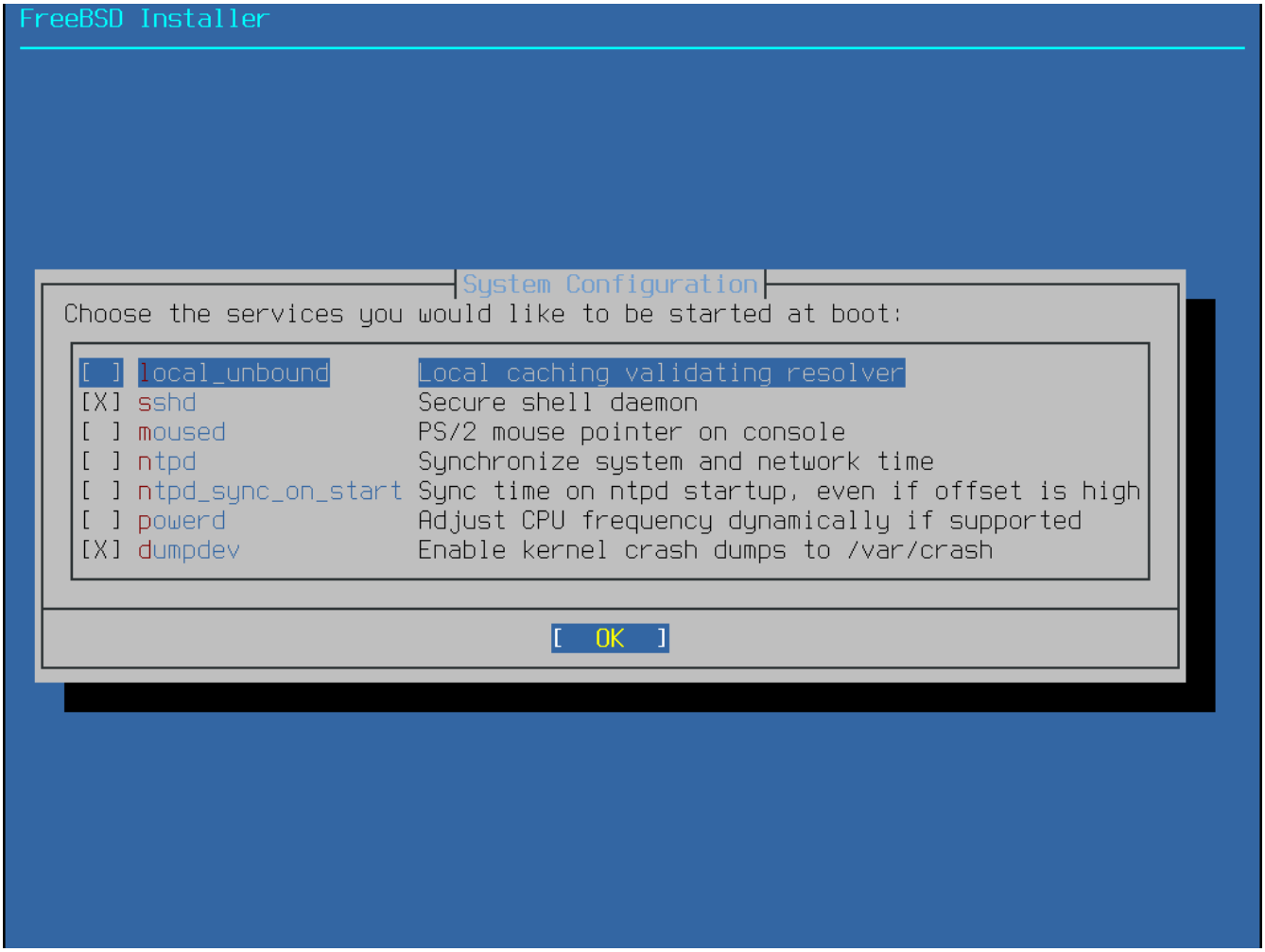


Рисунок 37. Выбор дополнительных активируемых сервисов

Дополнительные сервисы

- `sshd` - Secure Shell (SSH) демон для безопасного удаленного доступа.
- `moused` - Обеспечивает использование мыши в системной консоли.
- `ntpd` - Network Time Protocol (NTP) демон для автоматической синхронизации времени.
- `powerd` - Системная утилита для контроля потребляемой мощности и профилей энергосбережения.

2.9.5. Разрешение сохранения аварийных дампов

Далее, bsdinstall запросит, будет ли разрешено создание аварийных дампов (crash dump) на целевой системе. Сохранение аварийных дампов может быть весьма полезным при поиске неполадок в системе, поэтому пользователям рекомендуется при всякой возможности включать сохранение аварийных дампов. Выберите **[Yes]** для разрешения сохранения аварийных дампов или **[No]** для отмены их сохранения и продолжения послеустановочной настройки.

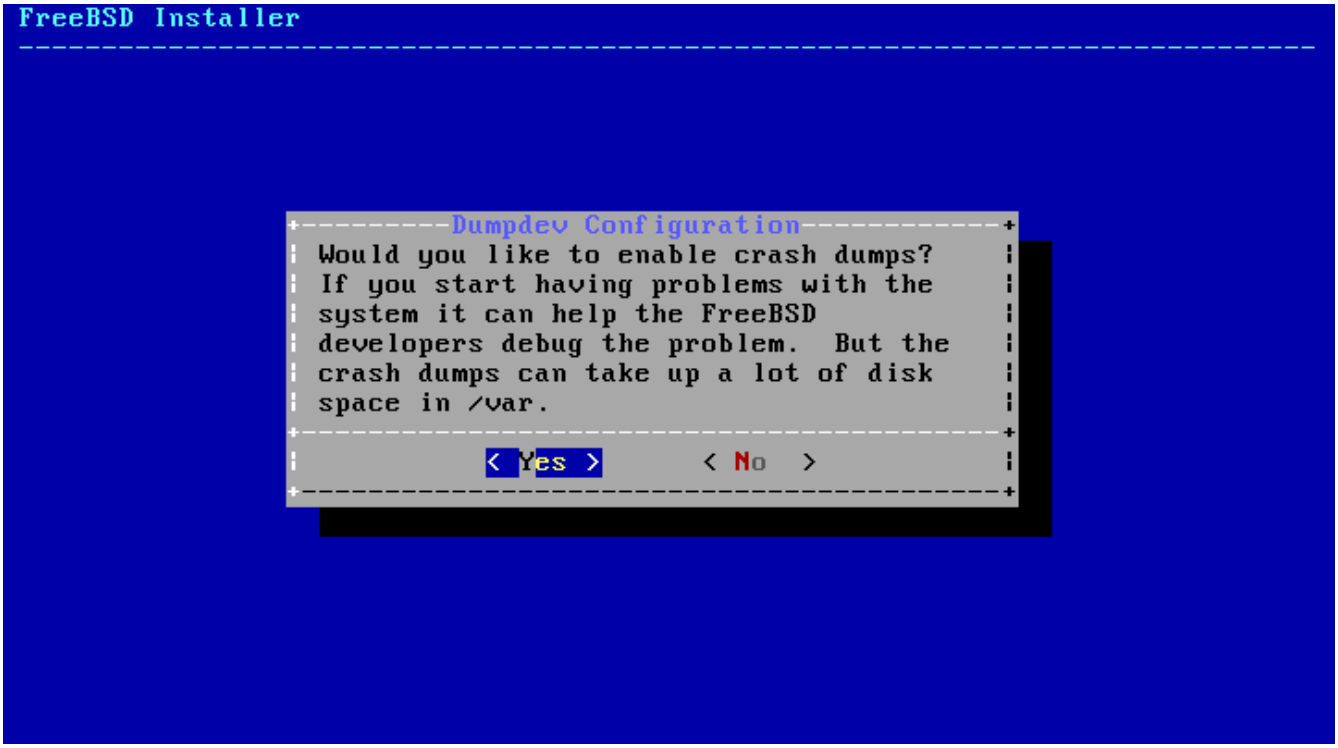


Рисунок 38. Разрешение сохранения аварийных дампов

2.9.6. Добавление пользователей

Добавление хотя бы одного пользователя в процессе установки позволит эксплуатировать систему исключая необходимость входа под учетной записью `root`. Работа в системе с правами пользователя `root` особенна тем, что по существу нет ограничений или защиты от действий пользователя. Вход под обычным пользователем является более благоразумным и безопасным.

Для добавления новых пользователей выберите **[Yes]**.

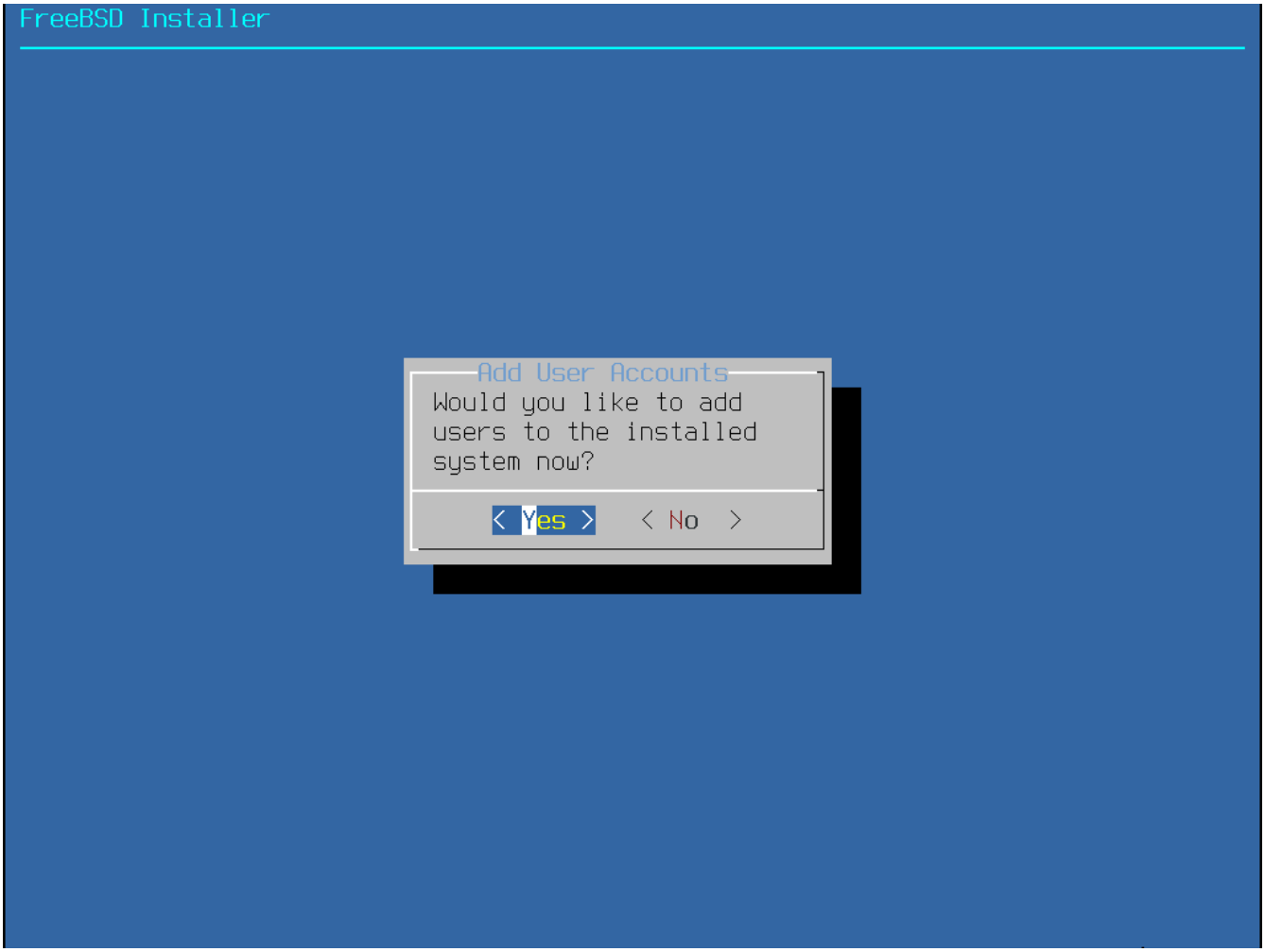


Рисунок 39. Добавление пользовательских учетных записей

Введите информацию о новом пользователе.

```
FreeBSD Installer
========================
Add Users

Username: imani
Full name: imani
Uid (Leave empty for default):
Login group [imani]:
Login group is imani. Invite imani into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh nologin) [sh]:
Home directory [/home/imani]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
```

Рисунок 40. Ввод информации о пользователе

Информация о пользователе

- `Username` - Имя, которое будет набирать пользователь для входа в систему. Часто оно формируется из объединенных вместе первой буквы имени и фамилии.
- `Full name` - Полное имя пользователя.
- `Uid` - Идентификатор пользователя. Обычно это поле не заполняется, система сама присвоит ему значение.
- `Login group` - Имя группы для этого пользователя. Обычно это поле также не заполняется, система поставит значение по умолчанию.
- `Invite user into other groups?` - Перечень групп, в которые будет внесен пользователь.
- `Login class` - Обычно оставляется пустым для принятия значения по умолчанию.
- `Shell` - Интерактивная оболочка для этого пользователя. В данном примере была выбрана оболочка csh(1).
- `Home directory` - Домашний каталог пользователя. Как правило, значение по умолчанию является корректным.
- `Home directory permissions` - Права на домашний каталог пользователя. Значение по умолчанию является корректным в большинстве случаев.
- `Use password-based authentication?` - Обычно "yes".
- `Use an empty password?` - Обычно "no".

- `Use a random password?` - Обычно "no".
- `Enter password` - Пароль для этого пользователя. Набираемые символы не отображаются на экране.
- `Enter password again` - Пароль необходимо ввести еще раз (для сверки).
- `Lock out the account after creation?` - Обычно "no".

После заполнения необходимых полей будет отображен итог и система переспросит, корректны ли введённые данные. Если во время ввода информации была допущена ошибка, то необходимо ответить `no` и ввести данные еще раз. Если вас всё устраивает, выберите `yes` для создания новой учетной записи пользователя.

```
FreeBSD Installer
=========================
Add Users

Username: imani
Full name: imani
Uid (Leave empty for default):
Login group [imani]:
Login group is imani. Invite imani into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh nologin) [sh]:
Home directory [/home/imani]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : imani
Password   : *****
Full Name  : imani
Uid        : 1001
Class      :
Groups     : imani wheel
Home       : /home/imani
Home Mode  :
Shell      : /bin/sh
Locked     : no
OK? (yes/no) [yes]:
adduser: INFO: Successfully added (imani) to the user database.
Add another user? (yes/no) [no]:
```

Рисунок 41. Заполненная форма ввода информации о новом пользователе

Ответьте `yes` на вопрос "Add another user?" если необходимо добавить другие учетные записи. Для завершения добавления пользователей и продолжения послеустановочной настройки выберите `no`.

За более детальной информацией об управлении учетными записями обратитесь к Пользователи и основы управления учетными записями.

2.9.7. Завершение настройки

После того, как установка и конфигурирование завершены, вам предоставляется заключительная возможность подкорректировать настройки.

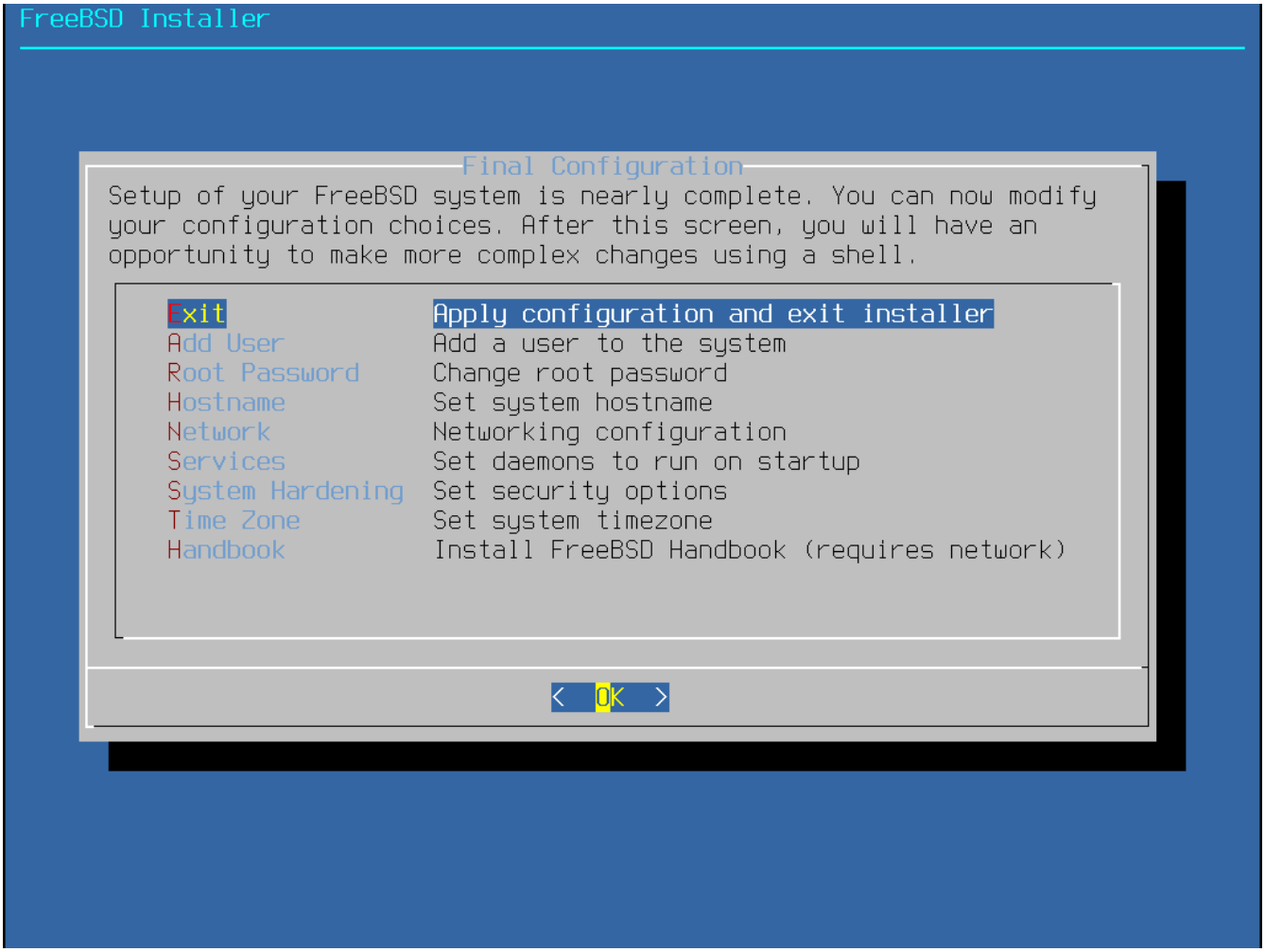


Рисунок 42. Финальное конфигурационное меню

Используйте это меню для внесения любых изменений или для выполнения дополнительного конфигурирования перед завершением установки.

Опции финального конфигурационного меню

- `Add User` - Описано в Добавление пользователей.
- `Root Password` - Описано в Установка пароля пользователя `root`.
- `Hostname` - Описано в Установка имени хоста.
- `Network` - Описано в Настройка сетевых интерфейсов.
- `Services` - Описано в Активирование дополнительных сетевых сервисов.
- `Time Zone` - Описано в Установка часового пояса.
- `Handbook` - Загрузка и установка Руководства FreeBSD (которое вы в данный момент читаете).

По завершении настройки для выхода из финального конфигурационного меню выберите **[Exit]**.

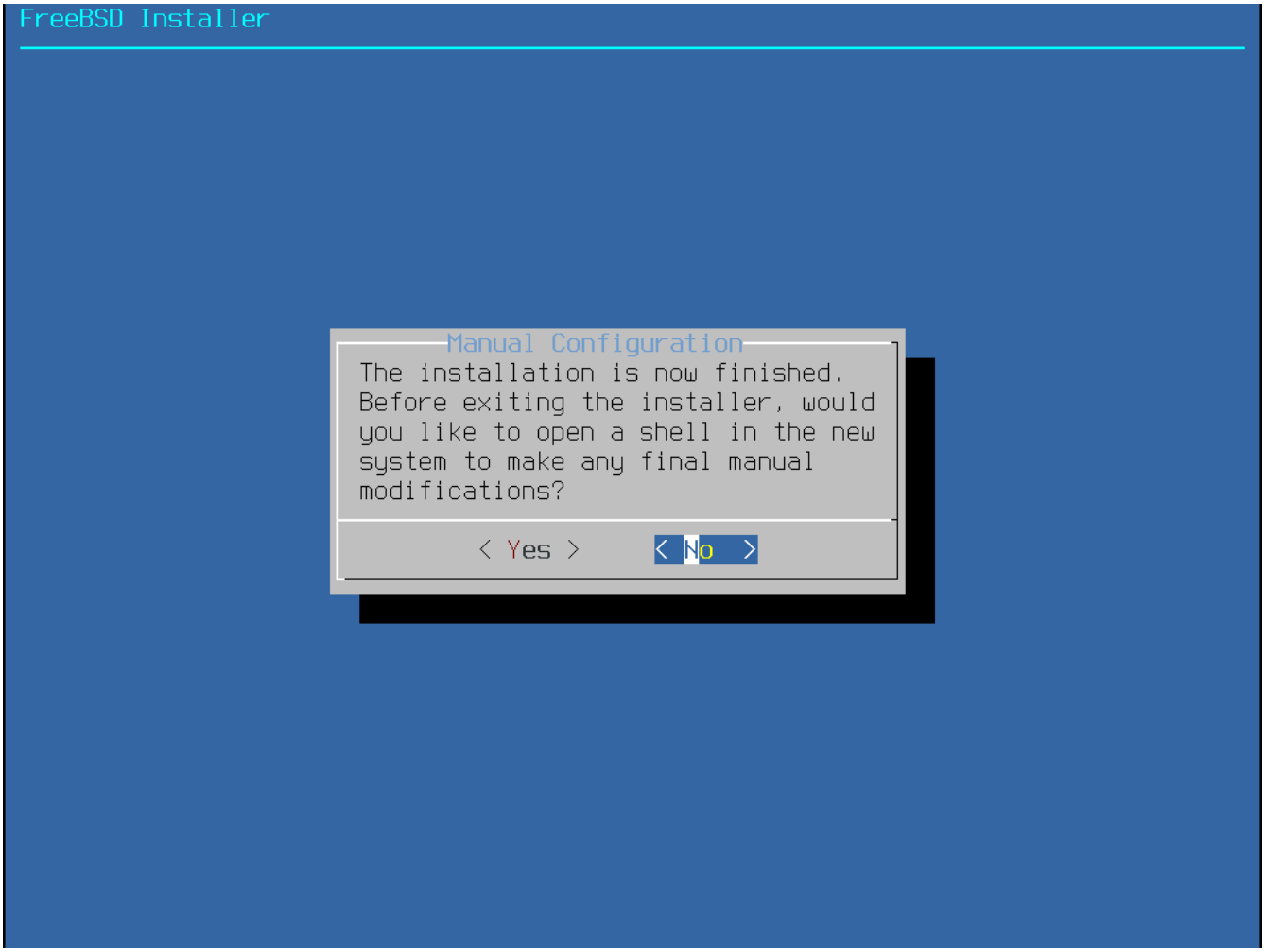


Рисунок 43. Ручная настройка

bsdinstall уточнит, есть ли какие настройки, которые необходимо выполнить до перезагрузки в свежеустановленную систему. Для входа в командный интерпретатор новой системы выберите **[Yes]**, для перехода к последнему шагу установки нажмите **[No]**.

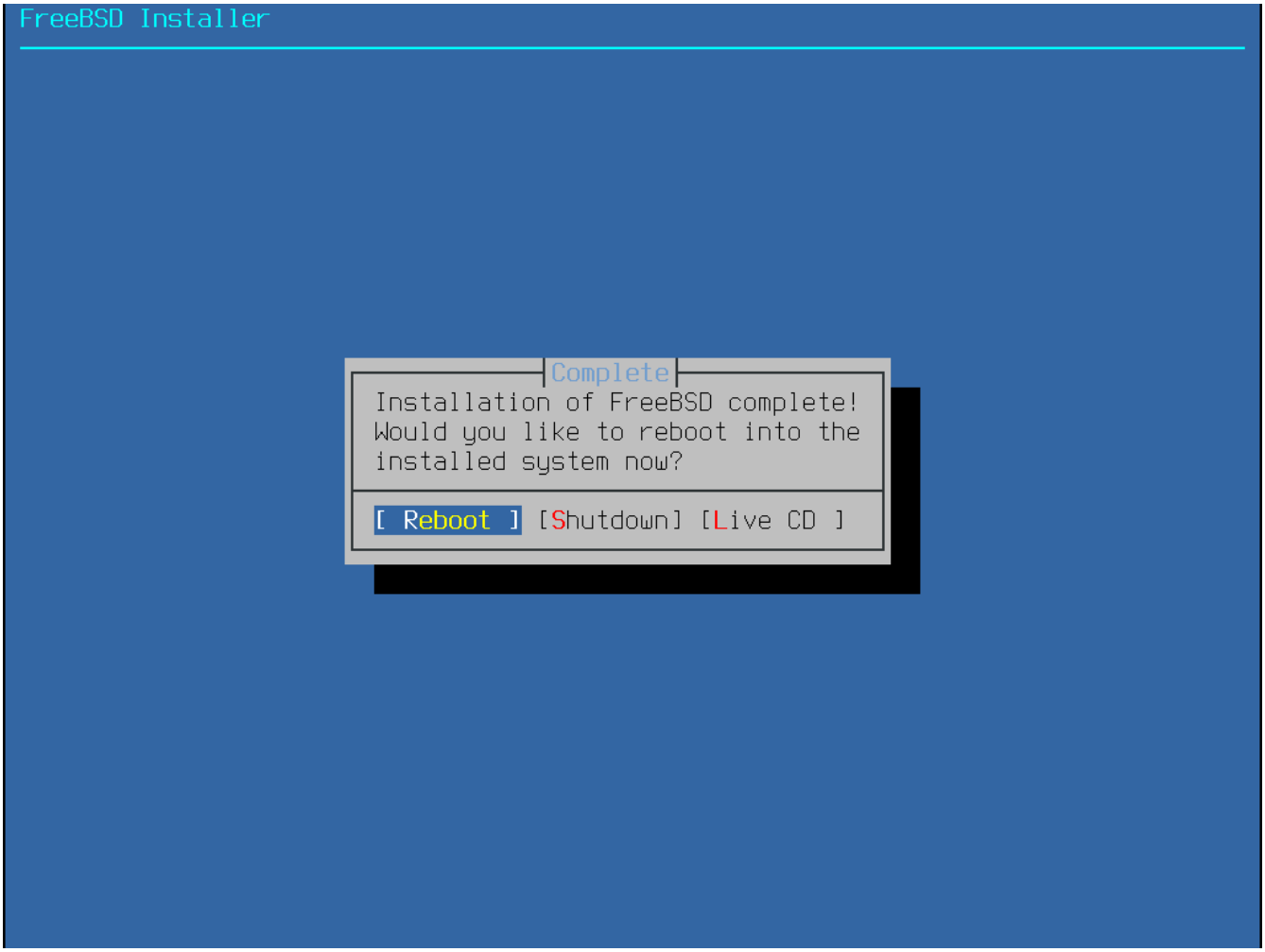


Рисунок 44. Завершение установки

Если необходимо дальнейшее конфигурирование или особая установка, то выбор **[Live CD]** загрузит установочный носитель в режим Live CD.

После того, как установка завершена, для перезагрузки компьютера и запуска новой системы FreeBSD выберите **[Reboot]**. Не забудьте извлечь установочный CD, DVD или USB-накопитель, иначе компьютер может снова с него загрузиться.

2.9.8. Загрузка и завершение работы FreeBSD

2.9.8.1. (FreeBSD/i386 Booting) Загрузка FreeBSD/i386

Во время загрузки FreeBSD отображается множество информационных сообщений. Большинство из них вытеснится за пределы экрана; это нормально. По завершении загрузки системы будет отображено приглашение ко входу (login prompt). Сообщения, которые переместились за пределы экрана, могут быть просмотрены: при нажатии `Scroll-Lock` включается режим *буфера прокрутки*. Клавиши `PgUp`, `PgDn`, а также клавиши навигации могут быть задействованы для прокручивания буфера. Повторное нажатие `Scroll-Lock` разблокирует дисплей и вернет его в нормальный режим.

На приглашение `login:` введите добавленное во время установки имя пользователя, в этом примере - `asample`. За исключением случаев крайней необходимости избегайте входа под учетной записью `root`.

Упомянутый выше буфер прокрутки ограничен в размере, поэтому в него могут умещаться не все сообщения. После входа в систему большинство из них можно просмотреть подав команду `dmesg | less` из командной строки. Для возврата к командной строке после просмотра сообщений нажмите `q`.

Типичные сообщения загрузки (информация о версиях опущена):

```
Copyright (c) 1992-2011 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.

    root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC amd64
CPU: Intel(R) Core(TM)2 Duo CPU     E8400  @ 3.00GHz (3007.77-MHz K8-class CPU)
  Origin = "GenuineIntel"  Id = 0x10676  Family = 6  Model = 17  Stepping = 6
  Features
=0x783fbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,MMX
,FXSR,SSE,SSE2>
  Features2=0x209<SSE3,MON,SSSE3>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
real memory  = 536805376 (511 MB)
avail memory = 491819008 (469 MB)
Event timer "LAPIC" quality 400
ACPI APIC Table: <VBOX   VBOXAPIC>
ioapic0: Changing APIC ID to 1
ioapic0 <Version 1.1> irqs 0-23 on motherboard
kbd1 at kbdmux0
acpi0: <VBOX VBOXXSDT> on motherboard
acpi0: Power Button (fixed)
acpi0: Sleep Button (fixed)
Timecounter "ACPI-fast" frequency 3579545 Hz quality 900
acpi_timer0: <32-bit timer at 3.579545MHz> port 0x4008-0x400b on acpi0
cpu0: <ACPI CPU> on acpi0
pcib0: <ACPI Host-PCI bridge> port 0xcf8-0xcff on acpi0
pci0: <ACPI PCI bus> on pcib0
isab0: <PCI-ISA bridge> at device 1.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <Intel PIIX4 UDMA33 controller> port 0x1f0-0x1f7,0x3f6,0x170-
0x177,0x376,0xd000-0xd00f at device 1.1 on pci0
ata0: <ATA channel 0> on atapci0
ata1: <ATA channel 1> on atapci0
vgapci0: <VGA-compatible display> mem 0xe0000000-0xe0ffffff irq 18 at device 2.0 on
pci0
em0: <Intel(R) PRO/1000 Legacy Network Connection 1.0.3> port 0xd010-0xd017 mem
0xf0000000-0xf001ffff irq 19 at device 3.0 on pci0
em0: Ethernet address: 08:00:27:9f:e0:92
pci0: <base peripheral> at device 4.0 (no driver attached)
pcm0: <Intel ICH (82801AA)> port 0xd100-0xd1ff,0xd200-0xd23f irq 21 at device 5.0 on
pci0
```

```
pcm0: <SigmaTel STAC9700/83/84 AC97 Codec>
ohci0: <OHCI (generic) USB controller> mem 0xf0804000-0xf0804fff irq 22 at device 6.0
on pci0
usbus0: <OHCI (generic) USB controller> on ohci0
pci0: <bridge> at device 7.0 (no driver attached)
acpi_acad0: <AC Adapter> on acpi0
atkbdc0: <Keyboard controller (i8042)> port 0x60,0x64 irq 1 on acpi0
atkbd0: <AT Keyboard> irq 1 on atkbdc0
kbd0 at atkbd0
atkbd0: [GIANT-LOCKED]
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: model IntelliMouse Explorer, device ID 4
attimer0: <AT timer> port 0x40-0x43,0x50-0x53 on acpi0
Timecounter "i8254" frequency 1193182 Hz quality 0
Event timer "i8254" frequency 1193182 Hz quality 100
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
atrtc0: <AT realtime clock> at port 0x70 irq 8 on isa0
Event timer "RTC" frequency 32768 Hz quality 0
ppc0: cannot reserve I/O port range
Timecounters tick every 10.000 msec
pcm0: measured ac97 link rate at 485193 Hz
em0: link state changed to UP
usbus0: 12Mbps Full Speed USB v1.0
ugen0.1: <Apple> at usbus0
uhub0: <Apple OHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus0
cd0 at ata1 bus 0 scbus1 target 0 lun 0
cd0: <VBOX CD-ROM 1.0> Removable CD-ROM SCSI-0 device
cd0: 33.300MB/s transfers (UDMA2, ATAPI 12bytes, PIO 65534bytes)
cd0: Attempt to query device size failed: NOT READY, Medium not present
ada0 at ata0 bus 0 scbus0 target 0 lun 0
ada0: <VBOX HARDDISK 1.0> ATA-6 device
ada0: 33.300MB/s transfers (UDMA2, PIO 65536bytes)
ada0: 12546MB (25694208 512 byte sectors: 16H 63S/T 16383C)
ada0: Previously was known as ad0
Timecounter "TSC" frequency 3007772192 Hz quality 800
Root mount waiting for: usbus0
uhub0: 8 ports with 8 removable, self powered
Trying to mount root from ufs:/dev/ada0p2 [rw]...
Setting hostuuid: 1848d7bf-e6a4-4ed4-b782-bd3f1685d551.
Setting hostid: 0xa03479b2.
Entropy harvesting: interrupts ethernet point_to_point kickstart.
Starting file system checks:
/dev/ada0p2: FILE SYSTEM CLEAN; SKIPPING CHECKS
/dev/ada0p2: clean, 2620402 free (714 frags, 327461 blocks, 0.0% fragmentation)
Mounting local file systems:.
vboxguest0 port 0xd020-0xd03f mem 0xf0400000-0xf07fffff,0xf0800000-0xf0803fff irq 20
at device 4.0 on pci0
vboxguest: loaded successfully
```

```
Setting hostname: machine3.example.com.
Starting Network: lo0 em0.
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:9f:e0:92
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
Starting devd.
Starting Network: usbus0.
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 192.168.1.142 -- renewal in 43200 seconds.
add net ::ffff:0.0.0.0: gateway ::1
add net ::0.0.0.0: gateway ::1
add net fe80::: gateway ::1
add net ff02::: gateway ::1
ELF ldconfig path: /lib /usr/lib /usr/lib/compat /usr/local/lib
32-bit compatibility ldconfig path: /usr/lib32
Creating and/or trimming log files.
Starting syslogd.
No core dumps found.
Clearing /tmp (X related).
Updating motd:.
Configuring syscons: blanktime.
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
10:a0:f5:af:93:ae:a3:1a:b2:bb:3c:35:d9:5a:b3:f3 root@machine3.example.com
The key's randomart image is:
+--[RSA1 1024]----+
|    o..          |
|   o . .         |
|  .   o          |
|       o         |
|    o   S        |
|   + + o         |
|o . + *          |
|o+ ..+ .         |
|==o..o+E         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
```

```
The key fingerprint is:
7e:1c:ce:dc:8a:3a:18:13:5b:34:b5:cf:d9:d1:47:b2 root@machine3.example.com
The key's randomart image is:
+--[ DSA 1024]----+
|       ..     . .|
|      o  .   . + |
|     . ..   . E .|
|    . .  o o . . |
|     +  S = .    |
|    +  . = o     |
|     +  . * .    |
|    . .  o .     |
|      .o. .      |
+-----------------+
Starting sshd.
Starting cron.
Starting background file system checks in 60 seconds.

Thu Oct  6 19:15:31 MDT 2011

FreeBSD/amd64 (machine3.example.com) (ttyv0)

login:
```

На медленных машинах генерирование ключей RSA и DSA может занять ощутимое время. Это происходит лишь при первой загрузке новой системы, и лишь в случае, когда sshd настроен на автоматический запуск. Последующие загрузки будут проходить быстрее.

По умолчанию во FreeBSD не устанавливается никаких графических оболочек, однако в наличии они имеются. За более подробной информацией обратитесь к X Window System.

2.9.9. Завершение работы FreeBSD

Корректное завершение работы компьютера с FreeBSD помогает защитить от повреждений не только данные, но даже и аппаратное обеспечение. Не стоит просто выключать питание. Если вы входите в группу `wheel`, то станьте суперпользователем набрав в командной строке команду `su` и введя пароль пользователя `root`. Или же, войдите в систему как `root` и наберите команду `shutdown -p now`. Система корректно завершит работу и выключится.

Комбинация клавиш `Ctrl` + `Alt` + `Del` может быть задействована для перезагрузки системы, однако во время нормальной работы пользоваться ею не рекомендуется.

2.10. Решение проблем

Нижеследующий раздел описывает часто встречающиеся и сообщенные пользователями проблемы, возникающие в ходе установки.

2.10.1. Что делать, если что-то идет не так

По причине различных ограничений архитектуры PC, определение периферийных устройств (device probing) не может быть достоверным на все 100%, однако, есть несколько шагов, которые вы можете предпринять, если определение завершится неудачно.

Просмотрите Информацию об оборудовании (Hardware Notes) для вашей версии FreeBSD чтобы убедиться, что ваше оборудование поддерживается.

Если ваше оборудование поддерживается, а зависания или другие проблемы продолжаются, то вам необходимо будет построить собственное ядро. Это позволит вам добавить поддержку устройств, которые отсутствуют в ядре GENERIC. Ядро на установочных дисках сконфигурировано исходя из предположения, что большинство устройств находятся в настройках по умолчанию касательно прерываний, адресов ввода/вывода, каналов DMA. Если ваше оборудование было перенастроено, то вам скорее всего необходимо будет отредактировать конфигурационный файл ядра и пересобрать его, чтобы сообщить FreeBSD о настройках, отличных от предполагаемых.

Также возможны случаи, когда процедура определения (probe) для отсутствующего устройства приводит к сбою процедуры определения для другого устройства, присутствующего в аппаратной конфигурации. В этом случае необходимо отключить процедуру (процедуры) определения для конфликтующего драйвера (драйверов).

Некоторое количество проблем с установкой может быть устранено или уменьшено путем обновления встроенного программного обеспечения различных аппаратных компонентов, особенно - материнской платы. Встроенное программное обеспечение материнской платы обычно называется BIOS. У большинства производителей материнских плат и компьютеров есть Web-сайты, содержащие как информацию об обновлениях, так и сами обновления.

В общем, производители не рекомендуют обновлять BIOS материнской платы, если на то нет веских причин, например, таких как появление критически важного обновления. Процесс обновления *может* потерпеть неудачу, тем самым оставив BIOS поврежденным, а компьютер - нерабочим.

2.10.2. Решение проблем: вопросы и ответы

2.10.2.1. Моя система зависает во время загрузки на этапе определения устройств (probing), или она ведет себя странно во время установки.

Касательно платформ i386, amd64 и ia64: если во время загрузки была обнаружена система ACPI, то FreeBSD повсеместно использует её для конфигурирования оборудования. К сожалению, до сих пор существуют неполадки как в драйвере ACPI, так и среди материнских плат и их BIOS. ACPI может быть отключена путём установки значения переменной `hint.acpi.0.disabled` на третьем этапе загрузки:

```
set hint.acpi.0.disabled="1"
```

Это значение сбрасывается каждый раз при загрузке системы, поэтому строку `hint.acpi.0.disabled="1"` необходимо добавить в файл /boot/loader.conf. Информация о загрузчике приведена в Описание.

2.11. Использование Live CD

FreeBSD Live CD находится на том же CD диске, что и установочная программа. Это удобно для тех пользователей, которые всё еще размышляют о пригодности для них ОС FreeBSD и желают проверить некоторые функциональные возможности до начала установки.

При работе с Live CD следует учесть следующее:

- Для получения доступа к системе необходимо осуществить аутентификацию. Допустимое имя пользователя - `root`, пароль - пустой.
- Так как система работает непосредственно с CD, производительность будет заметно ниже чем у системы, установленной на жесткий диск.
- Live CD предоставляет в распоряжение командную строку, а не графический интерфейс.

Глава 3. Основы FreeBSD

3.1. Краткий обзор

Эта глава посвящена основным командам и функциональности операционной системы FreeBSD. Большая часть нижеизложенного материала применима к любой UNIX®-подобной операционной системе. Новые пользователи FreeBSD призываются к внимательному чтению всей главы.

Прочитав эту главу, вы будете знать:

- Как использовать и настраивать виртуальные консоли.
- Как создавать пользователей и группы пользователей во FreeBSD и управлять ими.
- Как работают права доступа на файлы в UNIX® и файловые флаги во FreeBSD.
- Иерархию каталогов FreeBSD.
- Организацию дисков FreeBSD.
- Как монтировать и размонтировать файловые системы.
- Что такое процессы, даемоны и сигналы.
- Что такое командный процессор и как изменить используемые по умолчанию параметры рабочего окружения.
- Как пользоваться стандартными текстовыми редакторами.
- Что такое устройства и файлы устройств.
- Как пользоваться справочным руководством для получения дополнительной информации.

3.2. Виртуальные консоли и терминалы

Если только FreeBSD не была настроена на автоматический запуск графической среды при загрузке, система будет запускаться в режиме запроса ввода имени учётной записи в командной строке, как в этом примере:

```
login:
```

Первая строка содержит определённую информацию о системе. `amd64` указывает на то, что FreeBSD работает в 64-разрядной системе x86. `pc3.example.org` является именем хоста, а `ttyv0` указывает на то, что это "системная консоль". Вторая строка является приглашением к входу в систему.

Поскольку FreeBSD является многопользовательской системой, она должна каким-то образом отличать различных пользователей. Это достигается за счёт того, что каждый пользователь перед получением доступа к программам системы должен в эту систему войти. Каждый пользователь имеет уникальное "имя пользователя" и персональный

"пароль".

Для входа в системную консоль наберите имя пользователя, которое было настроено во время установки системы, описанной в разделе Добавление пользователей, и нажмите `Enter`. Затем введите пароль, соответствующий этому имени пользователя и нажмите `Enter`. Пароль *не отображается* по соображениям безопасности.

После ввода корректного пароля будет выдано сообщение дня (MOTD, Message Of The Day), за которым последует приглашение командной строки. В зависимости от командного процессора, который был выбран при создании пользователя, таким приглашением будет символ `#`, `$` или `%`. Это приглашение указывает на то, что теперь пользователь вошёл в системную консоль FreeBSD и может попытаться использовать имеющиеся команды.

3.2.1. Виртуальные консоли

Хотя системная консоль может использоваться для взаимодействия с системой, пользователь, работающий в режиме командной строки за клавиатурой системы FreeBSD, как правило, будет входить в систему через виртуальную консоль. Это так, потому что по умолчанию выдача системных сообщений настроена на их отображение на системной консоли. Эти сообщения будут выдаваться поверх команд или файлов, с коорым работает пользователь, что мешает сосредоточиться на текущей работе.

По умолчанию FreeBSD настроена так, что она предоставляет несколько виртуальных консолей для ввода команд. Каждая виртуальная консоль обладает собственным приглашением к входу в систему и командным процессором, а переключение между виртуальными консолями выполняется легко. В итоге это равнозначно предоставлению нескольких одновременно открытых окон в графической среде, но в режиме командной строки.

Для переключения между виртуальными консолями во FreeBSD зарезервированы комбинации клавиш от `Alt` + `F1` до `Alt` + `F8`. Используйте `Alt` + `F1` для переключения на системную консоль (`ttyv0`), `Alt` + `F2` для доступа к первой виртуальной консоли (`ttyv1`), `Alt` + `F3` для доступа ко второй виртуальной консоли (`ttyv2`) и так далее. При использовании Xorg в качестве графической консоли для переключения на текстовую виртуальную консоль используется комбинация `Ctrl` + `Alt` + `F1`.

При переключении от одной консоли к другой FreeBSD берёт на себя управление изображением на экране. В результате создаётся видимость наличия множества виртуальных экранов и клавиатур, которые могут быть использованы при наборе команд для их запуска во FreeBSD. Программы, запущенные на одной из виртуальных консолей, не прекращают своей работы, когда пользователь переключается на другую виртуальную консоль.

Обратитесь к kbdcontrol(1), vidcontrol(1), atkbd(4), syscons(4) и vt(4) для получения дополнительных технических описаний консоли FreeBSD и драйверов её клавиатуры.

Во FreeBSD количество доступных виртуальных консолей настраивается в следующем разделе файла `/etc/ttys`.

```
# name  getty                type    status  comments
#
ttyv0   "/usr/libexec/getty Pc"         xterm   on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv2   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv3   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv4   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv5   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv6   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv7   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

Для отключения какой-либо виртуальной консоли поместите символ комментария (`#`) в начале строки, соответствующей этой виртуальной консоли. К примеру, для уменьшения количества доступных виртуальных консолей с восьми до четырёх поместите `#` в начале последних четырёх строк, представляющих виртуальные консоли с `ttyv5` по `ttyv8`. *Не комментируйте* строку системной консоли `ttyv0`. Заметьте, что последняя виртуальная консоль (`ttyv8`) используется для доступа к графическому окружению, если был установлен пакет Xorg, как это описано в главе X Window System.

+ За детальным описанием каждой колонки этого файла и доступных параметров виртуальных консолей обратитесь к ttys(5).

3.2.2. Однопользовательский режим

В загрузочном меню FreeBSD имеется пункт, который называется "Boot Single User". При его выборе система загрузится в специальном режиме, который называется "однопользовательским". Этот режим обычно используется для восстановления работоспособности системы, которая не загружается, или для сброса пароля пользователя `root`, когда он неизвестен. В однопользовательском режиме недоступны сетевые функции и дополнительные виртуальные консоли. При всём при этом имеется полный доступ к системе с полномочиями пользователя `root`, и по умолчанию пароль пользователя `root` не требуется. По этим причинам для загрузки в таком режиме требуется физический доступ к клавиатуре, а решение о том, кто имеет физический доступ к клавиатуре, стоит рассматривать в контексте обеспечения безопасности системы FreeBSD.

+ Настройки, управляющие однопользовательским режимом, находятся в следующем разделе файла `/etc/ttys`:

```
# name  getty   type    status  comments
#
# Если консоль помечена как "insecure", то init будет запрашивать пароль
# пользователя root при переходе в однопользовательский режим
console none    unknown off secure
```

По умолчанию столбец status установлен в значение `secure`. Это предполагает, что

физический доступ к клавиатуре либо не важен, либо контролируется политикой физической безопасности. Если изменить этот параметр на `insecure`, то это будет означать, что само по себе окружение не является безопасным, так как любой может получить доступ к клавиатуре. Когда в этой строке значение меняется на `insecure`, то FreeBSD будет запрашивать пароль пользователя `root` в случае, когда пользователь выберет загрузку в однопользовательском режиме.

Будьте осторожны при изменении этого параметра на `insecure`*!* Если пароль пользователя `root` забыт, то загрузка в однопользовательском режиме всё ещё будет возможна, но может быть затруднена для тех, кто незнаком с процессом загрузки FreeBSD.

3.2.3. Изменение видеорежимов консоли

Установленный по умолчанию режим видео для консоли FreeBSD может быть изменён до значения 1024x768, 1280x1024 или любого другого, который поддерживается вашим монитором и видеокартой. Для использования другого видеорежима загрузите модуль `VESA`:

```
# kldload vesa
```

Для определения того, какие видеорежимы поддерживаются вашим оборудованием, воспользуйтесь vidcontrol(1). Чтобы получить список поддерживаемых видеорежимов, выполните следующую команду:

```
# vidcontrol -i mode
```

Выдача этой команды отобразит список видеорежимов, поддерживаемых оборудованием. Для выбора нового видеорежима укажите его при помощи vidcontrol(1), работая как пользователь `root`.

```
# vidcontrol MODE_279
```

Если новый видеорежим приемлем, то его можно определить постоянно устанавливаемым при загрузке, добавив его в `/etc/rc.conf`:

```
allscreens_flags="MODE_279"
```

3.3. Пользователи и основы управления учётными записями

FreeBSD позволяет одновременное использование компьютера множеством пользователей. В то время, как в одно и то же время только один пользователь может сидеть перед экраном и использовать клавиатуру, войти в систему по сети может любое количество

пользователей. Для того, чтобы использовать систему, каждый пользователь должен иметь собственную учётную запись.

В этом разделе описывается вот что:

- Различные виды пользовательских учётных записей в системе FreeBSD.
- Как добавлять, удалять и изменять учётные записи пользователей.
- Как устанавливать ограничения для управления ресурсами, к которым имеют доступ пользователи и группы пользователей.
- Как создавать группы и добавлять пользователей в качестве их членов.

3.3.1. Виды учётных записей

Так как все доступы к системе FreeBSD осуществляются через учётные записи, и все процессы запускаются пользователями, то управление пользователями и учётными записями является важным вопросом.

Существуют три основных вида учётных записей: системные учётные записи, пользовательские учётные записи и учётная запись суперпользователя.

3.3.1.1. Системные учётные записи

Системные учётные записи используются для запуска таких сервисов, как DNS, электронная почта и веб-серверы. Причиной этого является безопасность; если бы все сервисы работали с полномочиями суперпользователя, то они могли бы действовать без ограничений.

Примерами системных учётных записей являются `daemon`, `operator`, `bind`, `news` и `www`.

`nobody` является обобщённой непривилегированной системной учётной записью. Несмотря на это, чем больше сервисов используют `nobody`, тем больше файлов и процессов становятся связанными с этим пользователем, и, соответственно, тем болеепривилегированным становится этот пользователь.

3.3.1.2. Учётные записи пользователей

Учётные записи пользователей назначаются реальным людям и используются для входа в систему и её использования. Каждый человек, имеющий доступ к системе, должен иметь уникальную пользовательскую учётную запись. Это позволяет администратору понимать, кто и что делает, а также предотвращать затирание настроек одних пользователей другими пользователями.

Каждый пользователь может настроить свою собственную рабочую среду, приспособленную к использованию системы, при помощи выбора оболочки, используемой по умолчанию, редактора, привязок комбинаций клавиш и настроек языкового окружения.

С каждой пользовательской учётной записью в системе FreeBSD связана определенная информация:

Имя пользователя

Имя пользователя вводится в строке приглашения `login:`. У каждого пользователя обязательно должно быть уникальное имя пользователя. Существует определённый набор правил для создания допустимых имен пользователей, документированных в passwd(5). Для того, чтобы поддерживать обратную совместимость с приложениями, рекомендуется использовать имена пользователей, состоящие из восьми или меньшего количества символов в нижнем регистре.

Пароль

У каждой учётной записи имеется связанный с ней пароль.

Идентификатор пользователя (UID)

Идентификатор пользователя (UID) является числом, используемым для однозначной идентификации этого пользователя в системе FreeBSD. Команды, позволяющие указывать имя пользователя, сначала будут преобразовывать его в UID. Рекомендуется использовать UID меньше 65535, так как более высокие значения могут вызвать проблемы совместимости с некоторым программным обеспечением.

Идентификатор группы (GID)

Идентификатор группы (GID) является числом, используемым для однозначной идентификации основной группы, которой принадлежит пользователь. Группы являются механизмом контроля доступа к ресурсам на основе GID пользователя, а не его UID. Это может значительно уменьшить размер некоторых конфигурационных файлов, а также позволяет пользователям быть членами более чем одной группы. Рекомендуется использовать значения GID, не превышающие 65535, так как превышение может вызвать сбои некоторого программного обеспечения.

Класс доступа (login class)

Классы доступа являются расширением механизма групп, дающим дополнительную гибкость при адаптации системы к различным пользователям. Классы доступа описываются в разделе Настройка классов доступа.

Периодичность смены пароля

По умолчанию пароли не устаревают. Однако устаревание пароля может быть включено по отдельности у каждого пользователя, принуждая всех или некоторых пользователей менять свои пароли по истечении определённого периода времени.

Время действия учётной записи

По умолчанию во FreeBSD время действия учётных записей не ограничено. При создании учётных записей с ограниченным сроком действия, например, учётных записей студентов учебного заведения, укажите дату истечения действия учетной записи при помощи pw(8). После наступления указанной даты учётная запись не может быть использована для входа в систему, хотя каталоги и файлы, соответствующие этой учётной записи, не исчезнут.

Полное имя пользователя

Имя пользователя является уникальным идентификатором учётной записи для FreeBSD, однако оно не обязательно соответствует реальному имени пользователя. Подобно

комментариям, эта информация может содержать пробелы, заглавные буквы и иметь длину более 8 символов.

Домашний каталог

Домашний каталог является полным путём к некоторому каталогу в системе. Это начальный каталог пользователя после его входа в систему. По общепринятому соглашению все домашние каталоги пользователей размещаются в `/home/username` или `/usr/home/username`. Пользователи хранят свои личные файлы и подкаталоги в своих домашних каталогах.

Оболочка пользователя

Оболочка обеспечивает стандартное пользовательское окружение для взаимодействия с системой. Существует множество различных видов оболочек, и у опытных пользователей будут свои предпочтения, которые могут быть отражены в настройках их учётных записей.

3.3.1.3. Учётная запись суперпользователя

Учётная запись суперпользователя, обычно называемая `root`, используется для управления системой без ограничения полномочий. По этой причине она не должна использоваться для таких повседневных задач, как отправка и получение почты, общий анализ системы или программирование.

В отличие от других пользовательских учётных записей, может работать без ограничений, и некорректное использование учётной записи суперпользователя может привести к грандиозным авариям. Учётные записи пользователей не способны уничтожить систему по ошибке, поэтому рекомендуется входить в систему как пользователь, и переключаться в режим суперпользователя только в случае, когда запускаемой команде требуются дополнительные полномочия.

Всегда проверяйте и перепроверяйте любые команды, запускаемые учётной записью суперпользователя, поскольку любой дополнительный пробел или отсутствующий символ может привести к безвозвратной потере данных.

Имеется несколько способов получения полномочий суперпользователя. Хотя можно входить как пользователь `root`, это крайне не рекомендуется.

Для того, чтобы стать суперпользователем, вместо этого используйте su(1). Если при запуске этой команды указывается параметр `-`, то пользователь также получит настройки окружения учётной записи `root`. Пользователь, запускающий эту команду, должен входить в группу `wheel`, в противном случае команда не будет выполнена. Пользователь также должен знать пароль для учётной записи пользователя `root`.

В этом примере пользователь становится суперпользователем только для запуска команды `make install`, так как этот шаг требует полномочий суперпользователя. После завершения работы команды пользователь набирает `exit` для того, чтобы выйтих из учётной записи суперпользователя и возвратиться к полномочиям собственной пользовательской учётной записи.

Пример 4. Установка программы с полномочиями суперпользователя

```
% configure
% make
% su -
Password:
# make install
# exit
%
```

Стандартный механизм su(1) хорошо работает для отдельных инсталляций или небольших сетей с единственным системным администратором. Альтернативой является установка пакета или порта security/sudo. Данное программное обеспечение обеспечивает протоколирование активностей и позволяет администратору настраивать, какие команды и какими пользователями могут запускаться с полномочиями суперпользователя.

3.3.2. Управление учётными записями

FreeBSD предоставляет набор различных команд для управления учётными записями пользователей. Наиболее часто используемые команды перечислены в разделе Инструменты для управления учётными записями, атем даны некоторые примеры их использования. Обратитесь к страницам Справочника каждой утилиты для получения дополнительной информации и примеров использования.

Таблица 2. Инструменты для управления учётными записями

Команда	Краткое описание
adduser(8)	Рекомендуемое приложение командной строки для добавления новых пользователей.
rmuser(8)	Рекомендуемое приложение командной строки для удаления пользователей.
chpass(1)	Гибкий инструмент для изменения информации в базе данных пользователей.
passwd(1)	Инструмент командной строки для изменения паролей пользователей.
pw(8)	Мощный и гибкий инструмент для изменения всех параметров пользовательских учётных записей.
bsdconfig(8)	Инструмент настройки системы с поддержкой управления учётными записями.

3.3.2.1. Добавление пользователя

Рекомендуемой программой для добавления новых пользователей является adduser(8). При добавлении нового пользователя эта программа автоматически обновляет `/etc/passwd` и `/etc/group`. Она также создаёт домашний каталог для этого пользователя, копирует

стандартные конфигурационные файлы из `/usr/share/skel` и опционально может отправить новому пользователю по электронной почте приветственное сообщение. Эта утилита должна запускаться с полномочиями суперпользователя.

Утилита adduser(8) является интерактивной и следует шагам создания новой учётной записи пользователя. Как показано в разделе Добавление пользователя во FreeBSD, вводите запрашиваемую информацию либо нажимайте `Return` для принятия значения по умолчанию, отображаемого в квадратных скобках. В этом примере пользователь был приглашён в группу `wheel`, позволяющей работать с полномочиями суперпользователя при помощи su(1). По окончании утилита выдаст запрос на создание ещё одного пользователя или завершение работы.

Пример 5. Добавление пользователя во FreeBSD

```
# adduser
```

Выводимый текст должен быть похож на следующее:

```
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
```

Так как при наборе пароля он не отображается, будьте внимательны к опечаткам при его вводе при создании пользовательской учётной записи.

3.3.2.2. Удаление пользователя

Для полного удаления пользователя из системы запустите rmuser(8) с полномочиями суперпользователя. Эта команда выполняет следующие шаги:

1. Удаляет пользовательскую информацию в crontab(1), если она имеется.
2. Удаляет все задания at(1), принадлежащие этому пользователю.
3. Отправляет сигнал SIGKILL всем процессам, владельцем которых является данный пользователь.
4. Удаляет пользователя из локального файла паролей системы.
5. Удаляет домашний каталог пользователя (если владельцем этого каталога является данный пользователь), включая обработку символических ссылок в маршруте к реальному домашнему каталогу.
6. Удаляет файлы входящей почты, принадлежащие этому пользователю, из `/var/mail`.
7. Удаляет все файлы, владельцем которых является этот пользователь, из /tmp`, `/var/tmp` и `/var/tmp/vi.recover`.
8. Удаляет имя пользователя из всех групп, в которых он состоит, в `/etc/group`. Если какая-то группа становится пустой, а её название совпадает с именем пользователя, то эта группа удаляется; эта группа соответствует уникальной пользовательской группе, создаваемой adduser(8).
9. Удаляет все очереди сообщений, сегменты совместно используемой памяти и семафоры, владельцем которых является данный пользователь.

rmuser(8) не может использоваться для удаления учётной записи суперпользователя, поскольку это практически гарантированно означает массовые сбои системы.

По умолчанию используется интерактивный режим, как показано в следующем примере.

Пример 6. Интерактивное удаление учётной записи с помощью `rmuser`

```
# rmuser jru
```

Выводимый текст должен быть похож на следующее:

```
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Removing user (jru): mailspool home passwd.
```

3.3.2.3. Изменение информации о пользователе

Любой пользователь может воспользоваться chpass(1) для изменения своего используемого по умолчанию командного процессора и персональной информации, связанной с его пользовательской учётной записью. Суперпользователь может использовать данную утилиту для изменения дополнительной информации об учётной записи любого пользователя.

При запуске без указания параметров, за исключением необязательного имени пользователя, chpass(1) отображает редактор, содержащий информацию пользователя. Когда пользователь завершает работу с редактором, база данных пользователей обновляется новой информацией.

Если утилита запущена не с полномочиями суперпользователя, то после выхода из редактора она запросит пароль пользователя.

В разделе Использование `chpass` с полномочиями суперпользователя суперпользователь набрал `chpass jru` и просматривал поля, которые могут быть изменены для данного пользователя. Если вместо этого эта команда будет запущена пользователем `jru`, то будут отображены и доступны для редактирования только шесть последних полей. Это показано в разделе Использование `chpass` с полномочиями обычного пользователя.

Пример 7. Использование `chpass` с полномочиями суперпользователя

```
# chpass
```

Выводимый текст должен быть похож на следующее:

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Пример 8. Использование chpass с полномочиями обычного пользователя

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Команды chfn(1) и chsh(1) являются ссылками на chpass(1), так же, как и ypchpass(1), ypchfn(1) и ypchsh(1). Так как NIS поддерживается автоматически, указание yp перед командой не обязательно. Настройка NIS рассматривается в Сетевые серверы.

3.3.2.4. Изменение пароля пользователя

Любой пользователь может легко сменить собственный пароль при помощи passwd(1). Для предотвращения случайных или неавторизованных изменений перед установкой нового пароля эта команду будут запрашивать первоначальный пароль пользователя:

Пример 9. Изменение собственного пароля

```
% passwd
```

Выводимый текст должен быть похож на следующее:

```
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Суперпользователь может сменить пароль любого пользователя, указав имя пользователя при запуске passwd(1). В тех случаях, когда эта утилита работает с полномочиями суперпользователя, она не будет запрашивать текущий пароль пользователя. Это позволяет менять пароль в случае, когда пользователь не может вспомнить первоначальный пароль.

Пример 10. Изменение пароля другого пользователя суперпользователем

```
# passwd jru
```

Выводимый текст должен быть похож на следующее:

```
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Как и в случае с chpass(1), yppasswd(1) является ссылкой на passwd(1), так что NIS работает с любой из команд.

3.3.2.5. Создание, удаление, изменение и просмотр пользователей и групп в системе

Утилита pw(8) может создавать, удалять, изменять и отображать пользователей и группы. Она функционирует как пользовательский интерфейс к системным файлам пользователей и групп. pw(8) обладает очень мощным набором параметров командной строки, что делает её подходящей для использования в скриптах командного процессора, однако новым пользователям она может показаться более сложной, чем другие команды, представленные в этом разделе.

3.3.3. Управление группами

Группа представляет собой список пользователей. Группа идентифицируется по её имени и GID. Во FreeBSD для определения того, что имеет право делать процесс, ядро использует UID процесса и список групп, которым он принадлежит. В большинстве случаев GID пользователя или процесса соотносится с первой группой из списка.

Соответствия имени группы и GID перечислены в `/etc/group`. Это обычный текстовый файл с четырьмя полями, разделёнными двоеточиями. Первое поле соответствует имени группы, второе является зашифрованным паролем, третье содержит GID, а четвёртое является списком членов группы, разделённых запятыми. За полным описанием синтаксиса обратитесь к group(5).

Суперпользователь может изменять `/etc/group` при помощи какого-либо текстового редактора, однако предпочтительным способом является редактирование файла с группами при помощи утилиты vigr(8), так как она может отловить некоторые распространённые ошибки.

В качестве альтернативы для добавления и редактирования групп можно использовать pw(8). Например, для добавления группы с именем `teamtwo` и последующей проверки её существования:

При использовании группы operator следует соблюдать осторожность, так как при этом могут быть ненамеренно даны полномочия этой группы, близкие к полномочиям суперпользователя, включая, но не ограничиваясь, правами на завершение работы системы и её перезапуск, а также на доступ ко всем объектам в `/dev`.

Пример 11. Добавление группы с использованием pw(8)

```
# pw groupadd teamtwo
# pw groupshow teamtwo
```

Выводимый текст должен быть похож на следующее:

```
teamtwo:*:1100:
```

В этом примере `1100` является GID группы `teamtwo`. На данный момент в `teamtwo` нет участников. Эта команда добавит `jru` в группу `teamtwo` в качестве участника.

Пример 12. Добавление учётных записей пользователей в новую группу при помощи pw(8)

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
```

Выводимый текст должен быть похож на следующее:

```
teamtwo:*:1100:jru
```

Аргументом к параметру `-M` является разделённый запятыми список пользователей, которых нужно добавить в новую (пустую) группу или заменить участников существующей группы. С точки зрения пользователя такое членство в группе отличается от основной группы, указанной в файле паролей (и является дополнительным к ней). Это значит, что пользователь не будет отображаться как участник группы при использовании параметра `groupshow` с pw(8), но будет отображаться при запросе информации с помощью id(1) или аналогичного инструмента. При использовании pw(8) для добавления пользователя в группу она работает только с `/etc/group` и не пытается считывать дополнительные сведения из `/etc/passwd`.

Пример 13. Добавление нового участника в группу с помощью pw(8)

```
# pw groupmod teamtwo -m db
# pw groupshow teamtwo
```

Выводимый текст должен быть похож на следующее:

```
teamtwo:*:1100:jru,db
```

В этом примере аргументом к параметру `-m` является разделённый запятыми список пользователей, которые должны быть добавлены в группу. В отличие от предыдущего примера, эти пользователи присоединяются к группе, а не замещают существующих в группе пользователей.

Пример 14. Использование id(1) для определения принадлежности к группе

```
% id jru
```

Выводимый текст должен быть похож на следующее:

```
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

В этом примере `jru` является членом групп `jru` и `teamtwo`.

За дополнительной информацией об этой команде и о формате `/etc/group` обратитесь к pw(8) и group(5).

3.4. Права доступа

Во FreeBSD с каждым файлом и каталогом связан набор прав доступа, для просмотра и изменения которых доступны несколько утилит. Понимание того, как работают эти права доступа, необходимо для обеспечения того, чтобы пользователи могли получать доступ к файлам, которые им нужны, и не могли некорректно обращаться к файлам, используемым операционной системой или владельцами которых является другие пользователи.

В этом разделе описываются традиционные полномочия UNIX®, используемые во FreeBSD. Для более тонкого управления доступом в файловой системе обратитесь к разделу Списки управления доступом.

В UNIX® базовые права доступа назначаются с использованием трёх типов доступа: чтение, запись и исполнение. Эти типы доступа используются для определения доступа к файлу для владельца файла, для группы и для прочих пользователей (всех остальных). Полномочия на чтение, запись и исполнение могут быть представлены в виде букв `r`, `w` и `x`. Также они могут быть представлены в виде двоичных чисел, так как каждое полномочие либо включено, либо выключено (`0`). При представлении в виде числа порядок прочтения всегда имеет вид `rwx`, где `r` имеет значение `4`, `w` имеет значение `2` и `x` равно `1`.

В Таблице 4.1 сведены возможные цифровые и символьные комбинации. В столбце "Список файлов каталога" символ `-` используется для отражения отсутствующего права доступа.

Таблица 3. Права доступа UNIX®

Значение	Права доступа	Список файлов каталога
0	Ничего не разрешено	---

Значение	Права доступа	Список файлов каталога
1	Нельзя читать и писать, разрешено исполнять	`--x`
2	Нельзя читать и исполнять, разрешено писать	`-w-`
3	Нельзя читать, разрешено писать и исполнять	`-wx`
4	Разрешено читать, нельзя писать и исполнять	`r--`
5	Разрешено читать и исполнять, нельзя писать	`r-x`
6	Разрешено читать и писать, нельзя исполнять	`rw-`
7	Разрешено все	`rwx`

Используйте параметр `-l` с командой ls(1) для получения подробного списка содержимого каталога, включающего столбец с информацией о полномочиях на файл для владельца, группы и всех остальных. Например, `ls -l` в произвольно выбранном каталоге может выдать следующее:

```
% ls -l
```

Выводимый текст должен быть похож на следующее:

```
total 530
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
```

В строке, соответствующей `myfile`, первый (самый левый) символ в первом столбце указывает на то, обычный ли это файл, каталог, специальное символьное устройство, сокет или какое-то другое специальное псевдофайловое устройство. В данном примере `-` указывает на то, что это обычный файл. Следующие три символа (в данном примере это `rw-`) определяют полномочия владельца файла. Последующие три символа, `r--`, определяют полномочия группы, которой принадлежит файл. Последние три символа, `r--`, определяют полномочия для всего остального мира. Дефис означает, что полномочия отсутствуют. В этом примере полномочия установлены таким образом, что владелец может выполнять операции чтения и записи в файл, группа может читать файл, а весь остальной мир может только читать файл. В соответствии с таблицей выше, полномочия для этого файлы могли бы быть представлены как `644`, где каждая цифра представляет три части полномочий на файл.

Как система управляет полномочиями на устройства? Во FreeBSD большинство устройств

представлено в виде файлов, которые программы могут открывать, читать и записывать в них данные. Эти специальные файлы устройств размещаются в каталоге `/dev`.

Каталоги обрабатываются так же, как и файлы. У них также имеются полномочия на чтение, запись и выполнение. Бит исполнимости для каталога имеет несколько другой, отличающийся от файлов, смысл. Когда каталог помечен как исполняемый, это означает, что в него можно перейти с помощью команды cd(1). Это также означает, что можно получить доступ к файлам в данном каталоге с учётом полномочий, установленных для этих файлов.

Для того, чтобы получить список файлов в каталоге, на него должны быть установлены полномочия на чтение. Для того, чтобы удалить из каталога какой-либо файл, имя которого известно, необходимо иметь полномочия на запись *и* исполнение каталога, содержащего соответствующий файл.

Существуют и другие права доступа, но они как правило используются в особых случаях, например, setuid-бит на выполняемые файлы и sticky-бит на каталоги. За дополнительной информацией о файловых полномочиях и о том, как их устанавливать, обратитесь к chmod(1).

3.4.1. Символическое обозначение полномочий

Символическое обозначение полномочий использует буквы вместо восьмеричных значений для назначения прав на файлы или каталоги. Символическое обозначение использует формат (кто) (действие) (полномочия), при этом доступны следующие значения:

Опция	Буква	Значение
(кто)	u	Пользователь (User)
(кто)	g	Группа (Group)
(кто)	o	Другие (Other)
(кто)	a	Все (All, "world")
(действие)	+	Добавление прав
(действие)	-	Удаление прав
(действие)	=	Явная установка прав
(права)	r	Чтение (Read)
(права)	w	Запись (Write)
(права)	x	Выполнение (Execute)
(права)	t	Sticky бит
(права)	s	SUID или SGID

Эти значения используются с командой chmod(1), но с буквами вместо цифр. Например, следующая команда блокирует доступ к *FILE* как для членов группы, соответствующей *FILE*, так и для всех прочих пользователей:

```
% chmod go= FILE
```

Для изменения более чем одного набора прав можно применить список значений, разделённых запятыми. Например, следующая команда удаляет права группы и "всех остальных" на запись в *FILE* и добавляет права на выполнение любым пользователям:

```
% chmod go-w,a+x FILE
```

3.4.2. Флаги файлов в FreeBSD

Кроме прав доступа к файлам, FreeBSD поддерживает использование "файловых флагов". Эти флаги привносят дополнительный уровень защиты и контроля над файлами, но не каталогами. При помощи этих флагов даже пользователь `root` может быть ограничен в удалении или изменении файлов.

Файловые флаги изменяются при помощи chflags(1). К примеру, для установки системного флага неудаляемости на файл `file1`, выполните следующую команду:

```
# chflags sunlink file1
```

Чтобы отключить системный флаг неудаляемости, укажите "no" перед `sunlink`:

```
# chflags nosunlink file1
```

Чтобы просмотреть флаги какого-либо файла, используйте команду ls(1) с параметрами `-lo`:

```
# ls -lo file1
```

```
-rw-r--r--  1 trhodes  trhodes  sunlnk 0 Mar  1 05:54 file1
```

Некоторые файловые флаги могут быть установлены или сняты только пользователем `root`. В остальных случаях флаги файла может устанавливать его владелец. Обратитесь к chflags(1) и chflags(2) для получения дополнительной информации.

3.4.3. Права доступа setuid, setgid и sticky

В дополнение к рассмотренным выше правам доступа и флагам файлов необходимо также упомянуть еще три вида прав доступа, о которых должны знать все системные администраторы. Это полномочия `setuid`, `setgid` и `sticky`.

Эти биты играют важную роль в определённых моментах работы UNIX®, так как они предоставляют функциональность, расширяющую права обычного пользователя. Чтобы

понять, как они работают, необходимо отметить различия между реальным идентификатором пользователя (UID) и действующим идентификатором пользователя (effective UID, EUID).

Реальный UID - это идентификатор пользователя, запустившего процесс на выполнение. Действующий UID (EUID) - это идентификатор пользователя, с которым на самом деле выполняется процесс. Например, утилита passwd(1) во время смены пароля пользователем запускается с реальным ID пользователя. Однако для того, чтобы актуализировать базу данных паролей, команда работает с действующим ID пользователя `root`. Это позволяет пользователям изменять их пароли и не наблюдать ошибку `Permission Denied`.

Полномочие setuid может быть задано в символьном виде добавлением права доступа `s` для пользователя, как в следующем примере:

```
# chmod u+s suidexample.sh
```

Полномочие setuid также можно задать, добавив число четыре (4) перед численным представлением набора полномочий, как показано в следующем примере:

```
# chmod 4755 suidexample.sh
```

Теперь права доступа на `suidexample.sh` выглядят подобно следующему:

```
-rwsr-xr-x   1 trhodes  trhodes    63 Aug 29 06:36 suidexample.sh
```

Заметьте, что `s` теперь является частью набора полномочий, относящихся к владельцу файла, и заменяет бит выполнимости. Это позволяет работать утилитам, которым требуется повышенный уровень полномочий, таким как passwd(1).

Указание параметра `nosuid` при запуске команды mount(8) приводит к тому, что такие программы перестают работать без выдачи предупреждений пользователям. Указанная возможность не является абсолютно надёжно работающей, так как обработчик `nosuid` может её обойти.

Чтобы увидеть, как это работает, откройте два терминала. В одном из них наберите `passwd`, работая как обычный пользователь. Пока утилита ждёт ввода нового пароля, просмотрите таблицу процессов и обратите внимание на информацию о пользователе процесса man.passwd[1].

В терминале А:

```
Changing local password for trhodes
Old Password:
```

В терминале Б:

```
# ps aux | grep passwd
```

```
trhodes  5232  0.0  0.2  3420  1608   0  R+    2:10AM   0:00.00 grep passwd
root     5211  0.0  0.2  3620  1724   2  I+    2:09AM   0:00.01 passwd
```

Хотя man.passwd[1] запущена от обычного пользователя, она использует действующий UID пользователя `root`.

Полномочие `setgid` выполняет ту же функцию, что и `setuid`; отличие заключается в том, что изменяются настройки прав для группы. Когда выполняются приложение или утилита с этой настройкой, то им назначаются полномочия на основании группы, владеющей файлом, а не пользователя, запустившего процесс.

Чтобы установить на какой-либо файл полномочие `setgid` в символическом виде, добавьте право доступа для группы при помощи chmod(1):

```
# chmod g+s sgidexample.sh
```

Альтернативным способом является выполнение команды chmod(1) с добавленным в начале числом два (2):

```
# chmod 2755 sgidexample.sh
```

В следующей выдаче обратите внимание на наличие `s` в перечне прав доступа для группы:

```
-rwxr-sr-x   1 trhodes  trhodes    44 Aug 31 01:49 sgidexample.sh
```

В этих примерах, несмотря на то, что сценарий оболочки является исполняемым файлом, он не будет выполняться с другим действующим идентификатором пользователя (EUID). Так происходит потому, что сценариям командного интерпретатора недоступен системный вызов setuid(2).

Позволяя повышать права доступа, биты полномочий `setuid` и `setgid` могут снижать безопасность системы. Третье специальное полномочие, `sticky bit`, может усиливать безопасность системы.

`Sticky bit`, будучи установленным на каталог, позволяет производить удаление файла только его владельцем. Это полезно для предотвращения удаления файлов в общедоступных каталогах, таких как `/tmp`, пользователями, которые не являются владельцами файлов. Чтобы использовать это полномочие, добавьте файлу режим `t`:

```
# chmod +t /tmp
```

Альтернативным способов является добавление единицы (1) перед набором прав доступа:

```
# chmod 1777 /tmp
```

Полномочие `sticky bit` будет отображаться как `t` в самом конце набора прав доступа:

```
# ls -al / | grep tmp
```

```
drwxrwxrwt  10 root  wheel         512 Aug 31 01:49 tmp
```

3.5. Структура каталогов

Структура каталогов FreeBSD является фундаментальным вопросом в достижении общего понимания устройства всей системы. Самым важным понятием является, несомненно, корневой каталог, или "/". Этот каталог является самым первым, монтируемым на этапе загрузки и содержащим базовую систему, необходимую для подготовки операционной системы к работе в многопользовательском режиме. Корневой каталог также содержит точки монтирования для других файловых систем, которые монтируются во время перехода к функционированию в многопользовательском режиме.

Точкой монтирования называется каталог, находящийся в родительской (обычно - корневой) файловой системе, к которому может быть подсоединена другая файловая система. Более глубоко это описывается в разделе Организация дисков. К стандартным точкам монтирования относятся `/usr/`, `/var/`, `/tmp/`, `/mnt/` и `/cdrom/`. Эти каталоги обычно перечислены как отдельные записи в файле `/etc/fstab`. Этот файл является таблицей с различными файловыми системами и точками монтирования, которая считывается системой. Большинство файловых систем в `/etc/fstab` монтируются во время загрузки автоматически из скрипта rc(8), если только в соответствующей записи для них не указано `noauto`. Более подробную информацию можно найти в разделе Файл fstab.

Полное описание иерархии файловой системы есть в hier(7). Таблица ниже содержит краткое описание наиболее часто упоминаемых каталогов.

Каталог	Описание
`/`	Корневой каталог файловой системы.
`/bin/`	Основные утилиты, необходимые для работы как в однопользовательском, так и в многопользовательском режимах.
`/boot/`	Программы и конфигурационные файлы, необходимые для нормальной загрузки операционной системы.

`/boot/defaults/`	Конфигурационные файлы, используемые в процессе загрузки операционной системы, со стандарными настройками. Обратитесь к loader.conf(5) для получения более подробной информации.
`/dev/`	Специальные файлы устройств, управляемые при помощи devfs(5)
`/etc/`	Основные конфигурационные файлы системы и скрипты.
`/etc/defaults/`	Конфигурационные файлы системы со стандартными настройками. Обратитесь к rc(8) для получения более подробной информации.
`/etc/periodic/`	Файлы сценариев, выполняемые ежедневно, еженедельно и ежемесячно при помощи cron(8). Обратитесь к periodic(8) для получения более подробной информации.
`/lib/`	Критически важные системные библиотеки, необходимые для выполнимых файлов в `/bin` и `/sbin`
`/libexec/`	Критически важные системные файлы
`/media/`	Содержит подкаталоги для использования в качестве точек монтирования для сменных носителей, таких как CD, накопители USB и гибкие диски
`/mnt/`	Пустой каталог, часто используемый системными администраторами как временная точка монтирования.
`/net/`	Автоматически монтируемые совместно используемые ресурсы NFS; обратитесь к auto_master(5)
`/proc/`	Файловая система процессов. Обратитесь к procfs(5) и mount_procfs(8) для получения более подробной информации.
`/rescue/`	Статически скомпилированные программы для восстановления после сбоев, как описано в rescue(8).
`/root/`	Домашний каталог пользователя `root`.
`/sbin/`	Системные утилиты и утилиты администрирования, необходимые для работы как в однопользовательском, так и в многопользовательском режимах.
`/tmp/`	Временные файлы, которые обычно *не* сохраняются при перезапуске системы. Размещаемая в оперативной памяти файловая система часто монтируется в `/tmp`. Это может быть автоматизировано с помощью переменных, относящихся к tmpmfs, в rc.conf(5) или с помощью записи в `/etc/fstab`; обратитесь к mdmfs(8) для получения более подробной информации.
`/usr/`	Основной набор пользовательских утилит и приложений.
`/usr/bin/`	Пользовательские утилиты и приложения общего назначения.
`/usr/include/`	Стандартные заголовочные файлы для языка C.
`/usr/lib/`	Архивные библиотеки.
`/usr/libdata/`	Файлы данных для различных утилит.

/usr/libexec/	Системные даемоны и системные утилиты, вызываемые другими программами.
/usr/local/	Локальные исполнимые файлы и библиотеки. Также используется в качестве стандартного целевого каталога в рамках инструментария портов FreeBSD. Внутри /usr/local общая структура каталогов должна следовать принципам, отражённым в hier(7) для /usr. Исключениями являются каталоги man, который расположен непосредственно в /usr/local, а не в /usr/local/share, и документация портов, которая располагается в share/doc/port.
/usr/ports/	Коллекция портов FreeBSD (опционально).
/usr/sbin/	Системные даемоны и системные утилиты, запускаемые пользователями.
/usr/share/	Файлы, не зависящие от архитектуры.
/usr/src/	Исходные тексты BSD и/или локальных программ.
/var/	Файлы журналов общего назначения, временные, перемещаемые файлы и файлы очередей печати.
/var/log/	Файлы различных системных журналов.
/var/tmp/	Временные файлы, которые обычно сохраняются после перезапуска системы.

3.6. Организация дисков

Наименьшей единицей, которую FreeBSD использует для поиска файлов, является имя файла. Имена файлов чувствительны к регистру, поэтому readme.txt и README.TXT являются двумя отдельными файлами. FreeBSD не использует расширение файла для определения того, является ли файл программой, документом или какой-то иной формой данных.

Файлы хранятся в каталогах. Каталоги могут не содержать файлов, либо могут содержать многие сотни файлов. Каталог также может содержать другие каталоги, что позволяет иметь иерархию вложенных друг в друга каталогов для организации данных.

Обращение к файлам и каталогам осуществляется указанием имени файла или каталога, дополняемого прямым слэшем /, за которым при необходимости могут следовать имена других каталогов. К примеру, если каталог foo содержит каталог bar, который содержит файл readme.txt, то полным именем, или *путём* файла является foo/bar/readme.txt. Заметьте, что это отличается от Windows®, в которой для отделения имён файлов и каталогов используется \. FreeBSD не использует символьных или каких-либо других именований устройств в пути. К примеру, набирать c:\foo\bar\readme.txt во FreeBSD не имеет смысла.

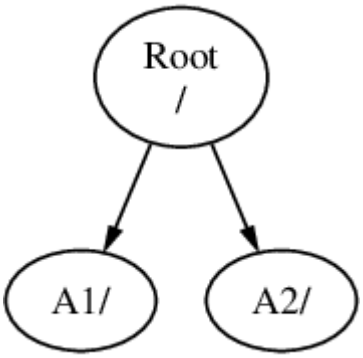
3.6.1. Файловые системы

Каталоги и файлы хранятся в файловой системе. Каждая файловая система на самом верхнем уровне содержит ровно один каталог, называемый *корневым каталогом* этой файловой системы. Этот корневой каталог может содержать другие каталоги. Одна из

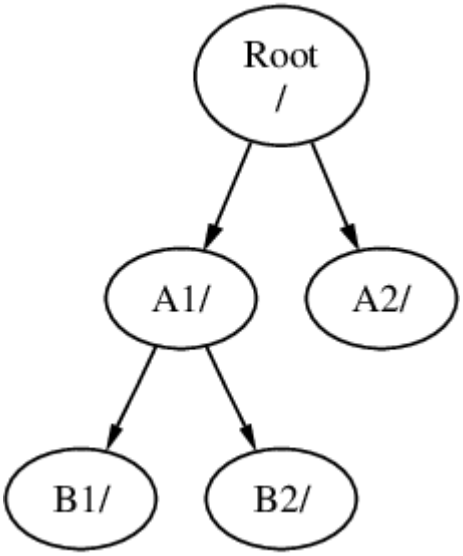
файловых систем назначается *корневой файловой системой*, или /. Любая другая файловая система *монтируется* в корневой файловой системе. Вне зависимости от того, сколько дисков присутствует в системе FreeBSD, каждый каталог выглядит как часть одного диска.

Рассмотрим три файловых системы, называющиеся A, B, и C. Каждая файловая система имеет один корневой каталог, в котором содержатся два других каталоги с именами A1, A2 (и, соответственно, B1, B2 и C1, C2).

Назовём A корневой файловой системой. Если для просмотра содержимого этого каталога использовать команду ls(1), то она покажет два подкаталога, A1 и A2. Дерево каталогов выглядит вот так:

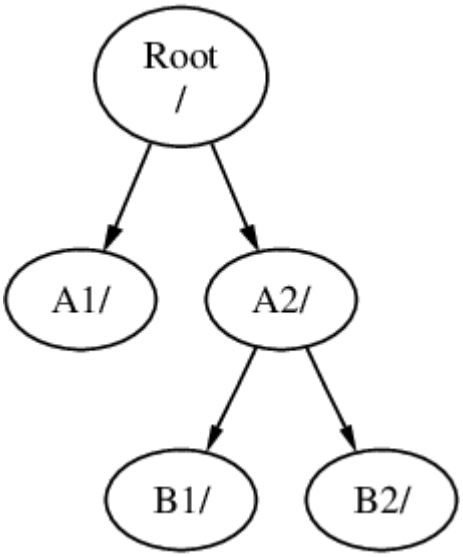


Файловая система должна быть смонтирована в каталог другой файловой системы. При монтировании файловой системы B в каталог A1, корневой каталог B заменяет A1, а каталоги в B отображаются в соответствии с этим:



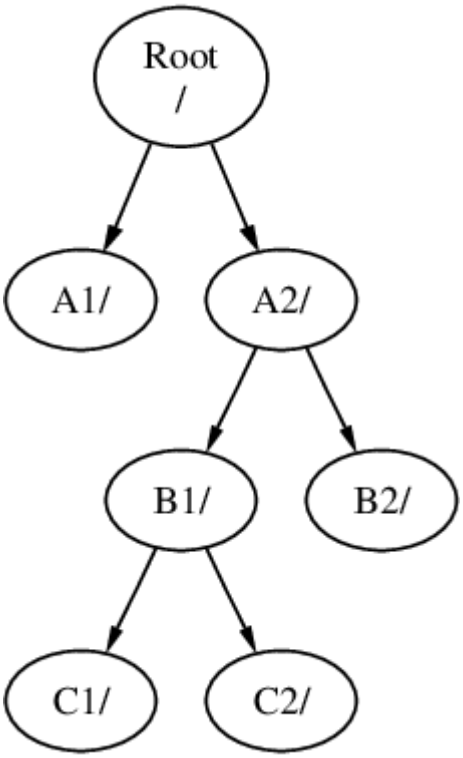
При необходимости к любым файлам, находящимся в каталогах B1 или B2, можно обратиться по маршруту /A1/B1 или /A1/B2. Все файлы, находившиеся в /A1, временно скрыты. Они появятся снова, если B будет *размонтирована* с A.

Если B была смонтирована в A2, диаграмма будет выглядеть так:

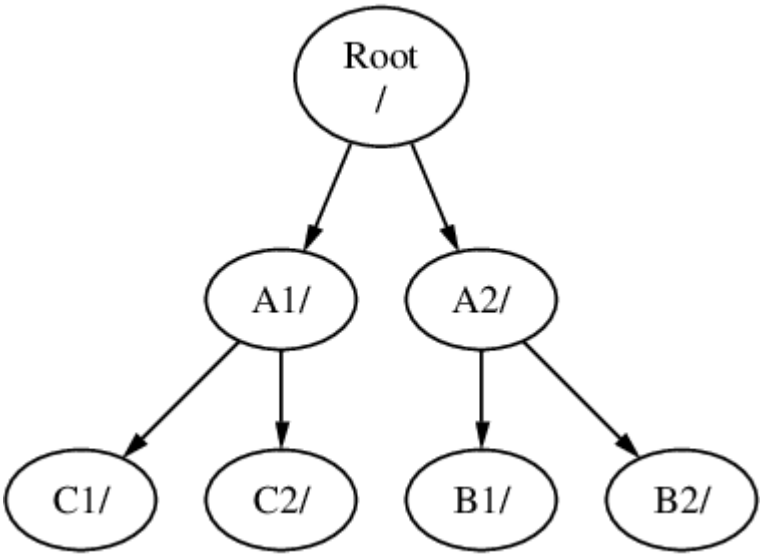


а пути будут `/A2/B1` и `/A2/B2`, соответственно.

Файловые системы могут быть смонтированы одна на другую. Продолжая предыдущий пример, файловая система `C` может быть смонтирована на каталог `B1` файловой системы `B` в таком порядке:



Или `C` может быть смонтирована прямо на файловую систему `A`, на каталог `A1`:



Вполне возможно иметь одну большую корневую файловую систему и не иметь потребности в создании других. У такого подхода есть несколько недостатков и одно преимущество.

Преимущества нескольких файловых систем

- Различные файловые системы могут иметь различные *параметры монтирования*. Например, корневая файловая система может быть смонтирована в режиме доступности только для чтения, что делает невозможным случайное удаление или редактирование какого-то критически важного файла. Отделение файловых систем, доступных пользователям для записи, таких как `/home`, от других файловых систем позволяет монтировать их с параметром *nosuid*. Этот параметр отменяет действие битов *suid*/ *guid* на исполняемых файлах в этой файловой системе, что потенциально повышает безопасность.
- FreeBSD автоматически оптимизирует расположение файлов на файловой системе в зависимости от того, как файловая система используется. Файловая система, содержащая множество мелких часто записываемых файлов, будет иметь оптимизацию, отличную от таковой для файловой системы, содержащей несколько больших файлов. На одной большой файловой системе эта оптимизация не работает.
- Файловые системы FreeBSD устойчивы к отключению электропитания. Тем не менее, потеря питания в критический момент все же может повредить структуру файловой системы. Разделение данных на несколько файловых систем повышает шансы, что система все-таки будет работать и делает более легким восстановление с резервной копии.

Преимущество одной файловой системы

- Размер файловых систем фиксирован. Если вы создаете файловую систему при установке FreeBSD и задаете определенный размер, позднее вы можете обнаружить что нужен раздел большего размера. Это не так легко сделать без резервного копирования, создания файловых систем нового размера и последующего восстановления сохраненных данных.

Во FreeBSD имеется команда growfs(8), которая позволяет увеличивать

размер файловой системы на лету, устраняя это ограничение. Файловая системы может быть расширена только на свободное пространство раздела, в котором она находится. Если имеется пространство за границей раздела, то раздел может быть увеличен при помощи gpart(8). Если раздел является последним на виртуальном диске, и диск был увеличен, то и раздел может быть увеличен.

3.6.2. Дисковые разделы

Файловые системы размещаются в *разделах*. Диски разбиваются на разделы в соответствии с одной из схем разбиения на разделы; обратитесь к разделу Разметка в неавтоматизированном режиме. Более новой схемой является GPT; более старые компьютеры, применяющие BIOS, используют MBR. GPT поддерживает разбиение диска на разделы, имеющие определённый размер, смещение и тип. Она поддерживает большое количество разделов и типов разделов, и рекомендуется к использованию во всех случаях, где это возможно. Разделы GPT используют имя диска с суффиксом, при этом суффикс `p1` обозначает первый раздел, `p2` второй раздел и так далее. Однако MBR поддерживает лишь небольшое количество разделов. Разделы MBR во FreeBSD называют `слайсами`. Слайсы могут быть использованы для разных операционных систем. Слайсы FreeBSD, в свою очередь, разбиваются на разделы при помощи меток BSD (обратитесь к bsdlabel(8)).

Номера слайсов начинаются с 1, следуют за именем устройства и предваряются `s`. Таким образом, "da0*s1*" является первым слайсом первого накопителя SCSI. На диске может быть только четыре физических слайса, но внутри физических слайсов подходящего типа могут размещаться логические слайсы. Эти расширенные слайсы нумеруются начиная с 5, так что "ada0*s5*" является первым расширенным слайсом на первом диске SATA. Эти устройства используются файловыми системами, занимающими весь слайс.

Каждый раздел GPT или BSD может содержать только одну файловую систему, и это значит, что файловые системы описываются либо при помощи их типичных точек монтирования в иерархии файловой системы, либо по имени раздела, в котором они размещены.

FreeBSD также использует дисковое пространство для *раздела подкачки*, обеспечивающего работу *виртуальной памяти*. Это позволяет вашему компьютеру работать так, как если бы у него было больше памяти, чем есть на самом деле. Когда у FreeBSD кончается память, она перемещает часть данных, не используемых в данный момент, в раздел подкачки и возвращает их обратно (перемещая в подкачку что-то другое), когда они нужны. Это явление называется *подкачкой*.

Для некоторых разделов BSD существуют определённые связанные с ними соглашения.

Раздел	Соглашение
`a`	Как правило, содержит корневую файловую систему.
`b`	Как правило, содержит пространство подкачки.

c	В обычном случае имеет такой же размер, что и окружающий слайс. Это позволяет утилитам, которым нужно обрабатывать весь слайс, таким, как сканер плохих блоков, использовать раздел c. В обычном случае создавать файловую систему в этом разделе не требуется.
d	Раздел d создавался для специальных целей, хотя сейчас они не актуальны и d может быть задействован как обычный раздел.

Слайсы и "опасно размеченные" физические устройства содержат разделы BSD, обозначаемые буквами от a до h. Эта буква добавляется к имени устройства, и, таким образом, "da0*a*" является разделом a на первом устройстве da, являющемся "эксклюзивно выделенным". "ada1s3*e*" является пятым разделом третьего слайса второго диска SATA.

Наконец, каждый диск идентифицирован. Имя диска начинается с кода, обозначающего тип диска, затем идет номер диска. В отличие от разделов и слайсов, нумерация дисков начинается с 0. Часто встречающиеся коды перечислены в разделе Имена дисковых устройств.

Для ссылки на раздел внутри слайса указывайте имя диска, s, номер слайса, а затем букву раздела. Примеры показаны в Примеры именований диска. В обозначение разделов GPT включается имя диска, p, а затем номер раздела.

Концептуальная модель диска отражает концептуальную модель рабиения диска с использованием слайсов MBR. Концептуальная модель диска показывает концептуальную модель диска, которая должна помочь прояснить ситуацию.

При установке FreeBSD с использованием MBR настройте дисковые слайсы и создайте разделы внутри слайса для использования во FreeBSD. При использовании GPT, настройте разделы для каждой файловой системы. В любом случае создайте файловую систему или раздел подкачки в каждом разделе, а также решите, где будет монтироваться каждая из файловых систем. Обратитесь к gpart(8) для получения информации о работе с разделами.

Таблица 4. Имена дисковых устройств

Тип дискового устройства	Имя дискового устройства
Приводы жёстких дисков SATA и IDE	ada
Приводы жёстких дисков SCSI и устройства хранения USB	da
Устройства хранения с интерфейсом NVMe	nvd или nda
Приводы CD-ROM с интерфейсом SATA и IDE	cd
Приводы CD-ROM с интерфейсом SCSI	cd
Приводы гибких дисков	fd
Ленточные накопители с интерфейсом SCSI	sa

Тип дискового устройства	Имя дискового устройства
Устройства хранения RAID	Например, `aacd` для Adaptec AdvancedRAID, `mlxd` и `mlyd` для Mylex, `amrd` для AMI MegaRAID, `idad` для Compaq Smart RAID и `twed` для 3ware RAID.

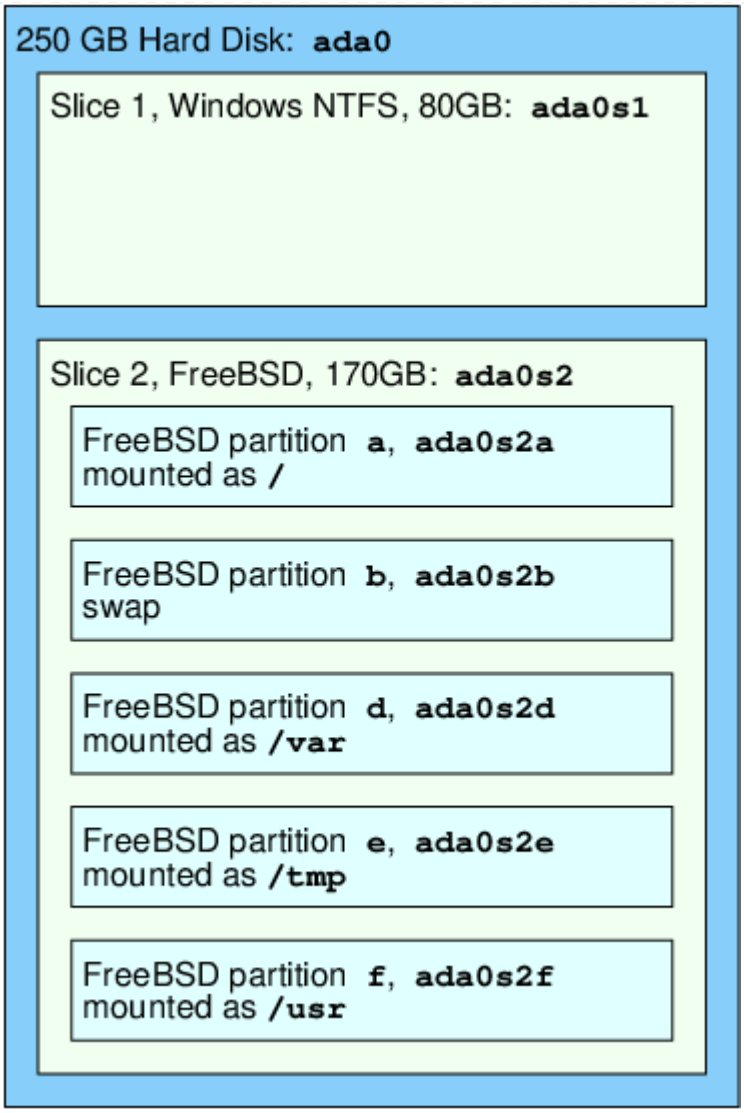
Пример 15. Примеры именований диска, слайса и раздела

Имя	Значение
`ada0s1a`	Первый раздел (`a`) на первом слайсе (`s1`) первого диска SATA (`ada0`).
`da1s2e`	Пятый раздел (`e`) на втором слайсе (`s2`) второго SCSI диска (`da1`).

Пример 16. Концептуальная модель диска

Эта диаграмма изображает первый подключенный к системе диск SATA с точки зрения FreeBSD. Предположим, что объём диска составляет 250 ГБ, и он содержит слайс размером 80 ГБ и слайс размером 170 ГБ (разделы MS-DOS®). Первый слайс содержит файловую систему Windows® NTFS, `C:`, а второй слайс содержит установленную FreeBSD. В этом примере установленная FreeBSD имеет четыре раздела с данными и раздел подкачки.

Каждый из четырёх разделов содержит файловую систему. Раздел `a` будет используется для корневой файловой системы, `d` для `/var/`, `e` для `/tmp/` и `f` для `/usr/`. Раздел, обозначенный буквой `c`, соответствует всему слайсу и поэтому не используется как обычный раздел.



3.7. Монтирование и размонтирование файловых систем

Файловая система лучше всего представима в виде дерева, которое, если можно так выразиться, растёт из `/`. Каталоги `/dev`, `/usr` и прочие, находящиеся в корневом каталоге, являются ветвями и, в свою очередь, могут иметь собственные ветви, такие как `/usr/local`, и так далее.

Имеются разные основания для того, чтобы размещать некоторые из этих каталогов в отдельных файловых системах. `/var` содержит каталоги `log/`, `spool/` и разные виды временных файлов и, таким образом, может полностью заполнить свободное дисковое пространство. Заполнение корневой файловой системы нежелательно, поэтому часто предпочитают отделить `/var` от `/`.

Другой частой причиной для размещения определённых веток каталогов в отдельных файловых системах является их расположение на отдельных физических дисках, отдельных виртуальных дисках, например, в Network File System, описанной в отдельном разделе, или на CDROM.

3.7.1. Файл fstab

В процессе загрузки (Процесс загрузки FreeBSD) файловые системы, перечисленные в `/etc/fstab`, монтируются автоматически, за исключением тех, для которых указан параметр `noauto`. Этот файл содержит записи в следующем формате:

```
устройство  /точка-монтирования тип файловой системы   опции   частота дампов  passno
```

`устройство`

Имя существующего устройства, соответствующее описанному в разделе Имена дисковых устройств.

`точка монтирования`

Существующий каталог, предназначенный для монтирования файловой системы.

`тип файловой системы`

Тип файловой системы, который передается программе mount(8). По умолчанию FreeBSD использует `ufs`.

`опции`

Либо `rw` для монтирования файловой системы в режиме чтения и записи, либо `ro` для файловых систем, доступных только в режиме чтения, за которыми могут следовать и другие нужные параметры. Довольно часто используется опция `noauto`, чтобы не монтировать автоматически файловые системы в процессе загрузки. Остальные параметры перечислены в mount(8).

`частота дампов`

Используется утилитой dump(8) для указания файловых систем, с которых требуется снимать копии. При отсутствии этого параметра он принимает нулевое значение.

`passno`

Определяет порядок, в котором файловые системы формата UFS должны проверяться при помощи fsck(8) после перезагрузки. Для файловых систем, которые должны быть пропущены при проверке, параметр `passno` должен быть установлен в нулевое значение. Корневая файловая система должна проверяться в первую очередь и иметь значение `passno`, установленное равным единице. Для других файловых систем значение этого параметра должно превышать единицу. Если две и более файловые системы имеют одинаковое значение `passno`, то fsck(8) будет пытаться проверять файловые системы параллельно, если это возможно.

Обратитесь к fstab(5) для получения дополнительной информацией о формате файла `/etc/fstab` и его параметрах.

3.7.2. Использование mount(8)

Файловые системы монтируются при помощи mount(8). Самый простой формат имеет следующий вид:

```
# mount __device__ __mountpoint__
```

Файловая система, включённая в `/etc/fstab`, также может быть смонтирована с указанием только лишь точки монтирования.

Эта команда имеет много параметров, описанных в mount(8). Самые часто используемые параметры таковы:

Параметры монтирования

`-a`

Смонтировать все файловые системы, перечисленные в файле `/etc/fstab`, за исключением тех, что помечены как "noauto", исключены параметром `-t` или уже смонтированы.

`-d`

Сделать все, кроме самого системного вызова mount. Эта опция полезна вместе с флагом `-v` для определения того, что на самом деле пытается сделать mount(8).

`-f`

Принудительное монтирование непроверенного раздела (опасно) или аннулирование полномочий на операции записи данных при понижении статуса монтирования файловой системы с доступной на чтение-запись на доступной только для операций чтения.

`-r`

Монтировать файловую систему в режиме только для чтения. Идентично использованию параметра `-o ro`.

`-t` ***fstype***

Смонтировать указанный тип файловой системы или, в случае указания `-a`, монтировать файловые системы только данного типа. По умолчанию применяется тип файловой системы "ufs".

`-u`

Обновить опции монтирования для файловой системы.

`-v`

Выдавать более подробную информацию.

`-w`

Монтировать файловую систему в режиме "чтение-запись".

Следующие значения могут быть переданы в качестве аргументов `-o` в виде списка значений, разделённых запятыми:

nosuid

Игнорировать setuid и setgid биты на файловой системе (еще одна полезная опция для повышения безопасности системы).

3.7.3. Использование umount(8)

Для размонтирования файловой системы используйте umount(8). Эта команда принимает один параметр, который может соответствовать точке монтирования, имени устройства либо принимать значение `-a` или `-A`.

Во всех вариантах принимается параметр `-f` для принудительного размонтирования и `-v` для выдачи подробной информации. Имейте в виду, что применение `-f` в целом не рекомендутся, так как может привести к аварийному завершению работы компьютера или повредить данные в файловой системе.

Для размонтирования всех смонтированных файловых систем или только тех типов файловых систем, что перечислены после параметра `-t`, воспользуйтесь параметрами `-a` и `-A`. Заметьте, что при использовании `-A` попытка размонтирования корневой файловой системы не предпринимается.

3.8. Процессы и даемоны

FreeBSD является многозадачной операционной системой. Каждая программа, выполняющаяся в некоторый выбранный момент времени, называется *процессом*. Каждая запускаемая команда порождает хотя бы один новый процесс, и имеется определённое количество системных процессов, которые запускает FreeBSD.

Каждый процесс идентифицируется уникальным номером, называемым *идентификатором процесса* (*process ID*) или *PID*. Подобно файлам, у каждого процесса имеется один владелец и группа, при этом полномочия владельца и группы используются для определения того, какие файлы и устройства могут быть открыты процессом. У большинства процессов также есть родительский процесс, которых их запустил. Например, командная оболочка является процессом, и любая команда, запущенная из командной оболочки, является процессом, для которого командная оболочка является родительским процессом. Исключением из этого правила является специальный процесс, который называется init(8), который всегда является первым процессом, запускаемым во время загрузки, и который всегда имеет PID, равный `1`.

Некоторые программы спроектированы не для того, чтобы работать в режиме ввода команд пользователя, и отключаются от терминала при первой возможности. К примеру, веб-сервер отвечает на веб-запросы, а не на команды пользователя. Другим примером такого типа приложений являются почтовые серверы. Программы такого типа известны под названием *даемоны*. Понятие даемона пришло из греческой мифологии и обозначает сущность, которая не является ни хорошей, ни плохой, и которая невидимо выполняет полезные дела. Это объясняет тот факт, что талисманом BSD является дружелюбно выглядящий даемон в кедах и с вилами.

Имеется соглашение, по которому программы, обычно работающие в режиме даемона,

именуются с "d" в конце названия. К примеру, BIND означает Berkeley Internet Name Domain, но на самом деле исполняемой программой является `named`. Программой веб-сервера Apache является `httpd`, а демоном очереди принтера является `lpd`. Это всего лишь соглашение об именовании. К примеру, основной почтовый демон для приложения Sendmail называется `sendmail`, а не `maild`.

3.8.1. Просмотр процессов

Для просмотра процессов, работающих в системе, воспользуйтесь ps(1) или top(1). Для выдачи статичного списка выполняемых в данный момент процессов, их PID, объёма используемой ими памяти и команды, которой они были запущены, используйте ps(1). Для отображения всех выполняющихся процессов и обновления этого списка каждые несколько секунд в целях интерактивного наблюдения за тем, что делает компьютер, используйте top(1).

По умолчанию ps(1) показывает пользователю только те команды, которые запущены пользователем и владельцем которых он является. К примеру:

```
% ps
```

Выводимый текст должен быть похож на следующее:

```
 PID TT  STAT    TIME COMMAND
8203  0  Ss   0:00.59 /bin/csh
8895  0  R+   0:00.00 ps
```

Выдача команды ps(1) организована в несколько столбцов. Столбец `PID` отображает идентификатор процесса. PID назначаются начиная с 1 и увеличиваются до 99999, а затем отсчёт начинается с начала. Однако PID не назначается повторно, если он уже используется. Столбец `TT` показывает терминал (tty), на котором выполняется программа, а `STAT` показывает состояние программы. `TIME` соответствует количеству времени, которое программа выполняется на центральном процессоре. Обычно это не то же самое время, что прошло с момента запуска программы, поскольку большинство программ проводят много времени в ожидании некоторого события, прежде чем занять время процессора. Наконец, `COMMAND` содержит команду, которая использовалась для запуска программы.

Имеется множество различных опций для изменения выводимой информации. Один из наиболее полезных наборов опций это `auxww`, при этом `a` отображает информацию о всех запущенных процессах всех пользователей, `u` показывает имя и объём используемой памяти пользователя, владеющего процессом, `x` отображает информацию о процессах-демонах, а `ww` указывает ps(1) на отображение всей командной строки для каждого процесса, вместо её обрезания в случае, если она слишком длинная, чтобы уместиться на экран.

Вывод top(1) выглядит похожим образом:

```
% top
```

Выводимый текст должен быть похож на следующее:

```
last pid:  9609;  load averages:  0.56,  0.45,  0.36              up 0+00:20:03
10:21:46
107 processes: 2 running, 104 sleeping, 1 zombie
CPU:  6.2% user,  0.1% nice,  8.2% system,  0.4% interrupt, 85.1% idle
Mem: 541M Active, 450M Inact, 1333M Wired, 4064K Cache, 1498M Free
ARC: 992M Total, 377M MFU, 589M MRU, 250K Anon, 5280K Header, 21M Other
Swap: 2048M Total, 2048M Free

  PID USERNAME    THR PRI NICE   SIZE    RES STATE   C   TIME   WCPU COMMAND
  557 root          1 -21  r31   136M 42296K select  0   2:20  9.96% Xorg
 8198 dru           2  52    0   449M 82736K select  3   0:08  5.96% kdeinit4
 8311 dru          27  30    0  1150M   187M uwait   1   1:37  0.98% firefox
  431 root          1  20    0 14268K  1728K select  0   0:06  0.98% moused
 9551 dru           1  21    0 16600K  2660K CPU3    3   0:01  0.98% top
 2357 dru           4  37    0   718M   141M select  0   0:21  0.00% kdeinit4
 8705 dru           4  35    0   480M    98M select  2   0:20  0.00% kdeinit4
 8076 dru           6  20    0   552M   113M uwait   0   0:12  0.00% soffice.bin
 2623 root          1  30   10 12088K  1636K select  3   0:09  0.00% powerd
 2338 dru           1  20    0   440M 84532K select  1   0:06  0.00% kwin
 1427 dru           5  22    0   605M 86412K select  1   0:05  0.00% kdeinit4
```

Вывод разбит на два раздела. Заголовок (первые пять или шесть строк) показывает PID последнего запущенного процесса, среднее значение загрузки системы (которое показывает насколько система занята), время работы системы с последней перезагрузки и текущее время. Остальные числа в заголовке относятся к количеству работающих процессов, объёму использования оперативной памяти и пространства подкачки, а также количеству времени, проводимого системой в различных состояниях центрального процессора. Если был загружен модуль файловой системы ZFS, то строка `ARC` содержит информацию о том, какой объём данных был считан из кэша оперативной памяти, а не с диска.

Под заголовком размещены несколько столбцов, содержащих информацию, похожую на результат работы ps(1), такую как PID, имя пользователя, объём времени ЦПУ и команда, которая запустила процесс. По умолчанию top(1) показывает также объём памяти, занятой процессом. Эта информация разделена на два столбца: один для суммарного объёма и один для занимаемого. Суммарный объём соответствует тому, что требовался приложению, а занимаемый соответствует объёму, фактически используемому сейчас.

top(1) автоматически обновляет экран каждые две секунды. Другое значение этого временного интервала может быть задано при помощи параметра `-s`.

3.8.2. Прекращение процессов

Одним из способов взаимодействия с любым работающим процессом или демоном

является отправка ему *сигнала* при помощи команды kill(1). Имеется множество различных сигналов; некоторые из них имеют специальное значение, тогда как другие описаны в документации приложения. Пользователь может посылать какой-либо сигнал только тем процессам, владельцем которых он является, а отправка сигнала процессу какого-то другого пользователя приведёт к ошибке запрета доступа. Исключением является пользователь `root`, который может отправлять сигналы чьим угодно процессам.

Операционная система также может отправлять сигнал процессу. Если приложение написано некорректно и пытается обратиться к области памяти, к которой оно не должно обращаться, FreeBSD посылает процессу сигнал "Segmentation Violation" (`SIGSEGV`). Если приложение было написано с учётом использования системного вызова alarm(3) для получения уведомления по истечении определённого периода времени, то ему будет отправлен сигнал "Alarm" (`SIGALRM`).

Для остановки процесса могут использоваться два сигнала: `SIGTERM` и `SIGKILL`. `SIGTERM` является вежливым способом завершить процесс, так как процесс может считать сигнал, закрыть какие-либо протоколирующие файлы, которые он мог открыть, и завершить то, что он делал до завершения работы. В некоторых случаях процесс может даже игнорировать `SIGTERM`, если выполняет задачу, которая не может быть прервана.

`SIGKILL` не может быть проигнорирован процессом. Отправка процессу `SIGKILL` обычно остановит этот процесс тотчас же.

> Существует несколько задач, которые не могут быть прерваны. К примеру, если процесс пытается выполнить чтение файла, находящегося на другом компьютере в сети, а другой компьютер недоступен, то такой процесс называют "непрерываемым". В конце концов время процесса истечёт, обычно после двух минут ожидания. Как только такой таймаут случится, процесс будет прекращён.

Другими часто используемыми сигналами являются `SIGHUP`, `SIGUSR1` и `SIGUSR2`. Так как эти сигналы являются сигналами общего назначения, различные приложения будут реагировать на них по-разному.

Например, после внесения изменений в конфигурационный файл веб-сервера ему нужно указать на повторное считывание настроек. Перезапуск `httpd` привёл бы к краткосрочной недоступности веб-сервера. Вместо этого отправьте даемону сигнал `SIGHUP`. Имейте в виду, что разные даемоны будут вести себя по-разному, поэтому обратитесь к документации по даемону для определения того, достигнет ли `SIGHUP` желаемых результатов.

> Прекращение случайного процесса в системе является плохой затеей. В частности, init(8), чей PID равен 1, является особенным процессом. Выполнение `/bin/kill -s KILL 1` является быстрым и нерекомендуемым способом завершить работу системы. *Всегда* дважды проверяйте параметры запуска kill(1) *перед* тем, как нажать на `Return`.

3.9. Командные процессоры

Командный процессор (или оболочка) предоставляет интерфейс командной строки для взаимодействия с операционной системой. Командный процессор получает команды из канала ввода и исполняет их. Многие командные интерпретаторы имеют встроенные функции, помогающие выполнять такие повседневной задачи, как управление файлами, включая их массовую обработку, редактирование командной строки, работа с макрокомандами и переменными окружения. FreeBSD поставляется вместе с несколькими командными процессорами, среди которых Bourne Shell (sh(1)) и расширенная версия C-shell (tcsh(1)), Другие командные процессоры, такие как `zsh` и `bash`, доступны в Коллекции портов FreeBSD.

Выбор используемого интерпретатора командной строки на самом деле является делом вкуса. Программист на языке C может чувствовать себя более комфортно с C-подобной оболочкой типа tcsh(1). Пользователь Linux® может предпочесть `bash`. Каждый командный процессор имеет свои уникальные особенности, которые могут работать, а могут и не работать в рабочем окружении, которое предпочитает пользователь, и именно поэтому имеются варианты выбора используемой оболочки.

Одной из распространённых функций оболочки командной строки является дополнение частичного имени файла до полного. После того, как пользователь набирает несколько первых символов команды или имени файла и нажимает клавишу `Tab`, командный процессор дополняет имя команды или файла до полного наименования. Рассмотрим случай с двумя файлами, которые называются `foobar` и `football`. Для удаления `foobar` пользователь может набрать `rm foo` и нажать `Tab` для формирования полного имени файла.

Однако оболочка только лишь отобразит `rm foo`. Сформировать полное имя файла невозможно, так как и `foobar`, и `football` начинаются с `foo`. Некоторые командные процессоры издадут звуковой сигнал или отобразят все варианты, если имеются совпадения более чем у одного имени. Тогда пользователь должен набрать дополнительные символы для идентификации желаемого имени файла. Набор `t` и повторное нажатие `Tab` достаточно для того, чтобы командный процессор определил желаемый файл и дополнил остаток его имени.

Дополнительные возможности при работе с интерпретатором дает использование переменных окружения. Переменные окружения представляют собой пары переменная/значение, сохраняемые в рабочем окружении интерпретатора. Это рабочее окружение может быть прочитано любой программой, запущенной из командного интерпретатора, и, таким образом, содержит много информации для настройки приложения. Часто используемые переменные окружения содержит список часто используемых переменных окружения и их значений. Заметьте, что имена переменных окружения всегда пишутся заглавными буквами.

Таблица 5. Часто используемые переменные окружения

Переменная	Описание
`USER`	Имя текущего пользователя.

PATH	Каталоги, разделенные двоеточием, для поиска исполняемых файлов.
DISPLAY	Сетевое имя дисплея Xorg для подключения при его доступности.
SHELL	Текущий командный интерпретатор.
TERM	Тип терминала пользователя. Используется, чтобы узнать возможности терминала.
TERMCAP	Список escape-последовательностей для управления различными функциями терминала.
OSTYPE	Вид операционной системы.
MACHTYPE	Процессорная архитектура системы.
EDITOR	Выбранный пользователем текстовый редактор.
PAGER	Предпочитаемая пользователем утилита для постраничного просмотра текста.
MANPATH	Каталоги, разделенные двоеточием, для поиска файлов системного справочника.

Порядок установки значения переменной окружения различна для разных оболочек. В командных процессорах tcsh(1) и csh(1) для задания переменных окружения используется setenv. В интерпретаторах sh(1) и bash для задания актуального значения переменных окружения используется export. В этом примере для tcsh(1) значение по умолчанию для переменной EDITOR устанавливается равным /usr/local/bin/emacs:

```
% setenv EDITOR /usr/local/bin/emacs
```

Аналогичная команда для bash была бы следующей:

```
% export EDITOR="/usr/local/bin/emacs"
```

Чтобы раскрыть значение переменной окружения для того, чтобы посмотреть её текущее значение, в командной строке наберите символ $ перед именем переменной. Например, echo $TERM выведет актуальное значение для $TERM.

Командные процессоры обрабатывают специальные символы, называемые метасимволами, как особое обозначение данных. Самым общеупотребительным метасимволом является *, который обозначает любое количество символов в имени файла. Метасимволы могут использоваться для выполнения массовых операций с именами файлов. Например, команда echo * равнозначна команде ls, поскольку оболочка выбирает все файлы, соответствующие *, а echo выдаёт их список в командной строке.

Чтобы предотвратить обработку специального символа командным процессором, экранируйте его, предварив наклонной чертой влево \ (обратным слэшем). Например, echo $TERM выведет значение настройки терминала, тогда как echo \$TERM выведет в буквальном смысле строку $TERM.

3.9.1. Смена командного процессора

Самым простым способом замены командного процессора, используемого по умолчанию, на постоянной основе является использование команды `chsh`. При запуске этой команды открывается редактор, настроенный в переменной окружения `EDITOR`, значение которой по умолчанию равно vi(1). Измените строку `Shell:`, указав полный путь для нового командного процессора.

Альтернативным способом является использование команды `chsh -s`, которая настроит указанную оболочку без открытия редактора. Например, для замены командного процессора на `bash`:

```
% chsh -s /usr/local/bin/bash
```

Введите ваш пароль в строке приглашения и нажмите `Return` для смены вашего командного процессора. Для того, чтобы начать использовать новую оболочку, выйдите из системы и войдите в неё снова.

Новая оболочка *обязательно* должна присутствовать в файле `/etc/shells`. Если командный процессор был установлен из Коллекции портов FreeBSD, как это описано в главе Установка приложений: порты и пакеты, то он должен быть добавлен в этот файл автоматически. Если его там нет, добавьте его при помощи следующей команды, заменив путь на маршрут соответствующей оболочки:

```
# echo /usr/local/bin/bash >> /etc/shells
```

После этого запустите chsh(1) повторно.

3.9.2. Расширенные функции оболочки

Оболочка UNIX® является не только лишь интерпретатором команд, она также выступает в роли мощного инструмента, позволяющего пользователям выполнять команды, перенаправлять их результирующий и входной потоки, а также выстраивать последовательность команд для улучшения выдачи финализирующей команды. Когда такая функциональность объединяется со встроенными командами, пользователь получает окружение, которое может дать максимальный эффект.

Перенаправление на уровне оболочки представляет собой действие по отправке результата работы или входного потока какой-либо команды в другую команду или в файл. Для записи результата работы команды ls(1), например, в файл, перенаправьте выходной поток:

```
% ls > directory_listing.txt
```

Список содержимого каталога теперь будет находиться в `directory_listing.txt`. Некоторые команды, подобные sort(1), могут использоваться для чтения входного потока. Для

сортировки этого списка перенаправьте входной поток:

```
% sort < directory_listing.txt
```

Входной поток будет отсортирован и размещён на экране. Для перенаправления этого ввода в другой файл можно перенаправить выходной поток sort(1), объединив направление:

```
% sort < directory_listing.txt > sorted.txt
```

Во всех примерах выше команды выполняют перенаправление при помощи файловых дескрипторов. В каждой системе UNIX® имеются файловые дескрипторы, среди которых имеются стандартный ввод (stdin), стандартный вывод (stdout) и стандартная диагностика (stderr). У каждого из них имеется своё назначение, и здесь вводом может быть клавиатура или мышь, что-то, формирующее входной поток. Выводом может быть экран или бумага в принтере. А диагностикой может быть что угодно, используемое для диагностических сообщений или сообщений об ошибках. Все три рассматриваются как файловые дескрипторы ввода-вывода и иногда рассматриваются как потоки.

Посредством использования этих дескрипторов командный процессор обеспечивает прохождение ввода и вывода через различные команды и их перенаправление в файл или из файла. Ещё одним методом перенаправления является оператор конвейера.

Оператор конвейера UNIX®, "|", позволяет прямую передачу или перенаправление вывода одной команды в другую программу. Проще говоря, конвейер позволяет передавать стандартный вывод какой-либо команды в качестве стандартного ввода другой команде, к примеру:

```
% cat directory_listing.txt | sort | less
```

В этом примере содержимое `directory_listing.txt` будет отсортировано, а вывод передан в less(1). Это позволяет пользователю просматривать результат в собственном темпе и не позволяет выходить за рамки экрана.

3.10. Текстовые редакторы

Большинство настроек FreeBSD осуществляется редактированием текстовых файлов. В силу этого обстоятельства хорошей идеей является освоение текстового редактора. Несколько редакторов поставляются с FreeBSD в составе базового комплекта системы, и гораздо больше доступно в Коллекции портов.

Простым в освоении редактором является ee(1), что означает "easy editor" ("лёгкий редактор"). Чтобы запустить его, наберите `ee filename`, где *filename* является именем редактируемого файла. Внутри редактора все команды для управления его функциями перечислены вверху экрана. Карет (^) обозначает клавишу `Ctrl`, таким образом, `^e` означает комбинацию клавиш `Ctrl` + `e`. Чтобы выйти из ee(1), нажмите клавишу `Esc`, затем выберите

пункт "leave editor" в главном меню. Редактор запросит сохранение изменений, если файл был изменён.

Во FreeBSD также имеются более мощные текстовые редакторы типа vi(1), поставляемого как часть базового системного комплекта. Другие редакторы, подобные editors/emacs и editors/vim, являются частью Коллекции портов FreeBSD. Эти редакторы обладают большей функциональностью, но также они более сложны в изучении. Изучение более мощных редакторов типа vim или Emacs может сэкономить вам больше времени в долгосрочной перспективе.

Многие приложения, модифицирующие файлы или требующие текстового ввода, автоматически открывают текстовый редактор. Чтобы сменить редактор, заданный по умолчанию, определите значение переменной окружения `EDITOR`, как это описано в разделе Командные процессоры.

3.11. Устройства и файлы устройств

Термин "устройство" используется в основном по отношению к аппаратному обеспечению системы, такому как диски, принтеры, графические адаптеры и клавиатуры. При загрузке FreeBSD основной объём выдаваемых сообщений относится к обнаруживаемым устройствам. Копии сообщений, выдаваемых при загрузке, сохраняются в `/var/run/dmesg.boot`.

Каждое устройство имеет имя и номер. Например, `ada0` соответствует первому приводу жёстких дисков с интерфейсом SATA, а `kbd0` представляет собой клавиатуру.

Во FreeBSD доступ к большинству устройств обязательно должен осуществляться через специальные файлы, называемые узлами устройств, которые размещаются в `/dev`.

+== Страницы Справочника

Пожалуй, самая полная документация по FreeBSD имеет форму страниц справочной системы. Практически каждое приложение или утилита имеют соответствующую страницу (часто не одну), описывающую основы работы и различные параметры. Эти справочники можно просматривать при помощи `man`:

```
% man command
```

Здесь *command* является названием команды, информацию о которой нужно получить. Например, чтобы узнать больше о команде ls(1), наберите:

```
% man ls
```

Страницы Справочника разделены на разделы, соответствующие различному типу содержимого. Во FreeBSD имеются следующие разделы:

1. Пользовательские команды.

2. Системные вызовы и коды ошибок.
3. Функции стандартных библиотек.
4. Драйверы устройств.
5. Форматы файлов.
6. Развлечения и игры.
7. Дополнительная информация.
8. Команды системного администрирования.
9. Интерфейсы ядра системы.

В некоторых случаях одна и та же тема может появиться в различных разделах справочника. Например, существуют пользовательская команда `chmod` и системный вызов `chmod()`. Для указания команде man(1) искомого раздела задайте его номер:

```
% man 1 chmod
```

При этом будет выведена справка о пользовательской команде chmod(1). По традиции документирования ссылки на конкретный раздел онлайнового справочника указываются в скобках, так что chmod(1) относится к пользовательской команде, chmod(2) указывает на соответствующий системный вызов.

Если название страницы Справочника неизвестно, воспользуйтесь `man -k` для поиска по ключевым словам, встречающимся в описаниях страниц Справочника:

```
% man -k mail
```

Эта команда выдаёт список команд, имеющих ключевое слово "mail" в своих описаниях. Это равнозначно использованию команды apropos(1).

Чтобы прочитать описания всех команд из `/usr/sbin`, наберите:

```
% cd /usr/sbin
% man -f * | more
```

или

```
% cd /usr/sbin
% whatis * |more
```

3.11.1. Файлы GNU Info

FreeBSD поставляется с некоторым количеством приложений и утилит, выпущенных Free Software Foundation (FSF). В дополнение к страницам справочника, с этими программами

могут поставляться гипертекстовые документы в виде так называемых файлов `info`. Они могут быть просмотрены с помощью команды info(1) или, если установлен пакет editors/emacs, в режиме info редактора emacs.

Чтобы использовать info(1), наберите:

```
% info
```

Вызвать на экран краткое введение можно набрав `h`. Краткий список команд можно получить, набрав `?`.

Глава 4. Установка приложений: порты и пакеты

4.1. Обзор

Вместе с FreeBSD в составе базового комплекта системы поставляется богатый набор системный утилит. Однако для выполнения какой-то реальной работы очень скоро возникает необходимость в установке дополнительных приложений сторонних разработчиков. FreeBSD дает две взаимодополняющих технологии для установки программного обеспечения сторонних разработчиков: Коллекция Портов FreeBSD (для установки из исходных кодов) и пакеты (для установки из откомпилированных двоичных файлов). Любая из этих систем может быть использована для установки приложений с локальных носителей или прямо из сети.

После чтения этой главы вы будете знать:

- Как устанавливать бинарные пакеты с программным обеспечением сторонних разработчиков.
- Как собирать из исходных кодов программное обеспечение сторонних разработчиков при помощи Коллекции Портов.
- Как удалять ранее установленные пакеты или порты.
- Как переопределить значения, используемые по умолчанию в Коллекции Портов.
- Как найти необходимое программное обеспечение.
- Как обновить установленные приложения.

4.2. Обзор установки программного обеспечения

Стандартная процедура установки программного обеспечения сторонних разработчиков на UNIX®-систему выглядит примерно так:

1. Загрузка программного обеспечения, которое может распространяться в форме исходных текстов или двоичных файлов.
2. Распаковка программного обеспечения из дистрибутивного формата (обычно tar-архива, сжатого при помощи compress(1), gzip(1) или bzip2(1)).
3. Поиск документации в файлах INSTALL, README или в каком-то файле из подкаталога doc/ и её чтение в поиске описания установки программного обеспечения.
4. Если программное обеспечение распространялось в форме исходных текстов, его компиляция. Сюда может быть включено редактирования файла Makefile, запуск скрипта `configure` и другие работы.
5. Тестирование и установка программного обеспечения.

Если вы устанавливаете программный пакет, который не был специально перенесён на FreeBSD, то вам может даже потребоваться редактировать код для того, чтобы он нормально заработал.

FreeBSD предоставляет две технологии, которые выполняют эту работу за вас. На момент написания таким образом доступно более 36000 сторонних приложений.

Каждый пакет содержит уже откомпилированные копии всех команд приложения, а также все конфигурационные файлы и документацию. С файлом пакета можно работать командами управления пакетами FreeBSD, такими как pkg_add(1), pkg_delete(1), pkg_info(1) и так далее.

Каждый порт FreeBSD является набором файлов, предназначенных для автоматизации процесса компиляции приложения из исходного кода. Файлы, из которых состоит порт, содержат всю необходимую информацию для выполнения автоматической загрузки, извлечения, применения патчей, компиляции и установки приложения.

Также система портов может использоваться для генерации пакетов, которые в последствии становятся объектом работы для команд управления пакетами FreeBSD.

Как пакеты, так и порты принимают во внимание *зависимости*. Если при инсталляции приложения при помощи pkg_add(1) или Коллекции Портов будет обнаружено, что необходимая библиотека не была установлена, то первым делом будет выполнена установка библиотеки.

Несмотря на то, что обе технологии весьма похожи, и пакеты, и порты имеют свои преимущества. Выберите технологию, которая соответствует вашим требованиям к установке конкретного приложения.

Преимущества пакетов

- Сжатый tar-архив пакета обычно меньше, чем сжатый tar-архив, содержащий исходный код приложения.
- Пакеты не требуют времени на компиляцию. Для больших приложений, таких как Mozilla, KDE или GNOME, это может быть важно, особенно при работе на медленной системе.
- Пакеты не требуют понимания процесса компиляции программного обеспечения во FreeBSD.

Преимущества портов

- Пакеты обычно компилируются с консервативными параметрами, потому что они должны работать на максимальном количестве систем. При установке из порта становится возможным изменение опций компиляции.
- Некоторые приложения имеют опции времени компиляции, позволяющие определять необходимые функциональные возможности. К примеру, Apache может быть настроен с широким набором различных опций.

  В некоторых случаях для одного и того же приложения будут иметься несколько пакетов с разными предварительными настройками. Например, Ghostscript доступен как пакет

ghostscript и как пакет ghostscript-nox11 - в зависимости от того, установлен ли сервер X11. Создание нескольких пакетов одного приложения быстро становится бессмысленным, если приложение имеет более одного-двух параметров компиляции.

- Условия лицензирования некоторого программного обеспечения запрещают распространение в двоичном виде. Оно должно распространяться в виде исходного кода и компилироваться конечным пользователем.
- Некоторые пользователи не доверяют дистрибутивам в двоичном виде или предпочитают прочесть исходный код и попытаться найти потенциальные проблемы.
- Если у вас есть собственные патчи, вам нужен исходный код для того, чтобы их применять.

Чтобы отслеживать обновления портов, подпишитесь на Список рассылки, посвящённый Портам FreeBSD и Список рассылки, посвящённый ошибкам в портах FreeBSD.

Перед установкой любого приложения необходимо зайти на http://vuxml.freebsd.org/, где находится информация по вопросам безопасности приложений, или установить ports-mgmt/portaudit. После установки наберите `portaudit -F -a` для проверки всех установленных приложений на наличие известных уязвимостей.

В оставшейся части главы будет рассказано, как использовать пакеты и порты для установки и управления программным обеспечением сторонних разработчиков во FreeBSD.

4.3. Поиск программного обеспечения

Список имеющихся для FreeBSD приложений постоянно растет. Существует несколько способов найти то, что нужно:

- На сайте FreeBSD по адресу http://www.FreeBSD.org/ports/ поддерживается обновляемый список всех имеющихся приложений для FreeBSD, в котором можно выполнять поиск. Поиск порта можно выполнить либо по имени приложения, либо по названию категории.
- Dan Langille поддерживает сайт FreshPorts, на котором есть удобный поиск, а также на нём отслеживаются изменения в приложениях из Коллекции Портов. Зарегистрированным пользователям доступна возможность создавать собственные списки наблюдаемых портов и автоматически получать оповещения об их обновлениях по электронной почте.
- Если вы не знаете названия нужного вам приложения, попытайтесь воспользоваться сайтом типа Freecode (http://www.freecode.com/) для поиска приложения, а затем возвратитесь на сайт FreeBSD, чтобы проверить, есть ли порт для этого приложения.
- Если вам необходимо определить, в какой категории находится порт, наберите `whereis file`, где *file* - программа, которую вы хотите установить:

```
# whereis lsof
```

```
lsof: /usr/ports/sysutils/lsof
```

Как вариант, можно воспользоваться echo(1):

```
# echo /usr/ports/*/*lsof*
/usr/ports/sysutils/lsof
```

Учтите, что в выводе также будут присутствовать совпадающие с шаблоном имена файлов, сохраненные в /usr/ports/distfiles.

- Ещё одним способом поиска программного обеспечения является использование встроенной возможности поиска в Коллекции Портов. Чтобы ею воспользоваться, зайдите в /usr/ports и выполните команду `make search name=program-name`, где *program-name* - это название программы, которую вы хотите найти. Например, если вы ищете `lsof`:

```
# cd /usr/ports
# make search name=lsof
Port:   lsof-4.56.4
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
Maint:  obrien@FreeBSD.org
Index:  sysutils
B-deps:
R-deps:
```

Команда `make search` выполняет поиск в файле с индексной информацией. Если получено сообщение, что требуется файл INDEX, запустите `make fetchindex` для загрузки актуального индексного файла. После загрузки файла INDEX команда `make search` сможет выполнить запрошенный поиск.

Строка "Path:" указывает, где находится порт.

Чтобы получить лаконичный вывод, задайте цель `quicksearch`:

```
# cd /usr/ports
# make quicksearch name=lsof
Port:   lsof-4.87.a,7
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
```

Для выполнения более глубокого поиска используйте `make search key=string` или `make quicksearch key=string`, где *string* представляет собой некоторый текст, относящийся к искомому порту. Текст ищется в комментариях, описаниях или зависимостях. Этот способ можно использовать для поиска портов, связанных с некоторой темой, когда название программы неизвестно.

В обоих случаях (search и quicksearch) строка поиска нечувствительна к регистру. Поиск "LSOF" приводит к тому же самому результату, что и поиск "lsof".

4.4. Использование бинарных пакетов

Во FreeBSD есть несколько утилит для управления пакетами:

- Для установки, удаления и получения перечня установленных пакетов на работающей системе может быть запущена утилита sysinstall. Обратитесь к Установка пакетов (Install Packages) за более детальной информацией.
- Утилиты командной строки для управления пакетами, которые являются темой данного раздела.

4.4.1. Установка пакета

Для установки бинарного пакета FreeBSD из локального файла или с сервера в сети используйте pkg_add(1).

Пример 17. Загрузка пакета вручную и его локальная установка

```
# ftp -a ftp2.FreeBSD.org
Connected to ftp2.FreeBSD.org.
220 ftp2.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-     This machine is in Vienna, VA, USA, hosted by Verio.
230-         Questions? E-mail freebsd@vienna.verio.net.
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/sysutils/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz' (92375 bytes).
100% |***************************************************| 92375       00:00 ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
# pkg_add lsof-4.56.4.tgz
```

Если у вас нет источника пакетов, например, такого как набор CD-ROM дисков с FreeBSD, то добавьте опцию -r для pkg_add(1). Утилита автоматически определит правильный формат объектных файлов и релиз, а затем загрузит и установит пакет с сервера FTP без какого-

либо дополнительного взаимодействия с пользователем.

```
# pkg_add -r lsof
```

Чтобы задействовать альтернативное зеркало пакетов FreeBSD, укажите его адрес в переменной окружения `PACKAGESITE`. Для загрузки файлов утилита pkg_add(1) использует fetch(3). Последняя учитывает значения различных переменных окружения, включая `FTP_PASSIVE_MODE`, `FTP_PROXY` и `FTP_PASSWORD`. Если вы находитесь за сетевым экраном, или для работы с FTP/HTTP вам необходимо использовать прокси, то определите соответствующие переменные. Обратитесь к справочной странице по fetch(3) для получения полного списка переменных. Заметьте, что в примере выше вместо `lsof-4.56.4` используется `lsof`. В случае загрузки из сети номер версии в имени пакета должен быть опущен.

> Если вы используете FreeBSD-CURRENT или FreeBSD-STABLE, то утилита pkg_add(1) загрузит последнюю версию устанавливаемой программы. Если же вы используете версию -RELEASE, то pkg_add(1) установит версию пакета, который был собран для конкретного релиза. Это поведение возможно изменить переопределив значение `PACKAGESITE`. Например, в системе FreeBSD 8.1-RELEASE pkg_add(1) по умолчанию попытается скачать пакеты с `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8.1-release/Latest/`. Чтобы обязать pkg_add(1) загружать пакеты для FreeBSD 8-STABLE, присвойте `PACKAGESITE` значение `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8-stable/Latest/`.

Файлы пакетов распространяются в форматах .tgz и .tbz. Пакеты находятся по адресу ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages или в каталоге /packages дистрибутива FreeBSD на DVD. Структура каталогов с пакетами подобна тому, как организовано дерево /usr/ports. Каждая категория имеет собственный каталог, и каждый пакет помещается в каталог All.

4.4.2. Управление пакетами

Для вывода перечня установленных пакетов и их описаний может быть задействована pkg_info(1).

```
# pkg_info
colordiff-1.0.13    A tool to colorize diff output
docbook-1.2         Meta-port for the different versions of the DocBook DTD
...
```

Утилита pkg_version(1) выводит отчёт о версиях всех установленных пакетов и сравнивает их версии с текущими версиями соответствующих приложений, доступных из локального дерева портов.

```
# pkg_version
colordiff                   =
```

```
docbook                    =
...
```

Символы во второй колонке указывают сравнительную разницу в возрасте установленной версии и версии, находящейся в локальном дереве портов.

Символ	Значение
=	Версия установленного пакета соответствует версии, находящейся в локальном дереве портов.
<	Версия установленного пакета старее, чем та, что имеется в локальном дереве портов.
>	Версия установленного пакета новее чем та, что есть в дереве портов. Скорее всего, локальное дерево портов устарело.
?	В индексном файле портов установленный пакет не найден. Это может случиться если установленный порт был переименован или удалён из Коллекции Портов.
*	Имеется несколько версий пакета.
!	Установленный пакет есть в индексном файле, но по какой-то причине утилите `pkg_version` не удалось сравнить номер версии установленного пакета и соответствующей записи в файле.

4.4.3. Удаление пакета

Для удаления ранее установленных пакетов с программным обеспечением используйте утилиту pkg_delete(1).

```
# pkg_delete xchat-1.7.1
```

Следует отметить, что для pkg_delete(1) требуется полное имя пакета и номер версии; вышеприведенная команда не сработала бы, если б ей было указано *xchat* вместо *xchat-1.7.1*. Для нахождения версии установленного пакета задействуйте утилиту pkg_version(1). Или же, напечатайте групповой символ (wildcard) вместо номера версии:

```
# pkg_delete xchat\*
```

в этом случае будут удалены все пакеты, имена которых начинаются на `xchat`.

4.4.4. Разное

Вся информация о пакетах, включая перечни файлов и описания каждого установленного пакета, хранится в каталоге /var/db/pkg.

4.5. Использование pkgng для управления бинарными пакетами

pkgng - это усовершенствованный инструмент, пришедший на смену традиционным утилитам управления пакетами FreeBSD. Он обладает множеством функциональных возможностей, ускоряющих и облегчающих работу с бинарными пакетами. Первый релиз pkgng состоялся в августе 2012 года.

pkgng не является заменой для утилит управления портами, таких как ports-mgmt/portmaster или ports-mgmt/portupgrade. В то время, как ports-mgmt/portmaster и ports-mgmt/portupgrade позволяют устанавливать приложения сторонних разработчиков как из бинарных пакетов, так и из Коллекции Портов, утилита pkgng дает возможность устанавливать приложения исключительно из бинарных пакетов.

4.5.1. Начало работы с pkgng

Во FreeBSD 9.1 и более поздние включена программа "самонастройки" ("bootstrap") pkgng. Она скачивает и устанавливает основную утилиту pkgng.

Для запуска самонастройки, выполните:

```
# /usr/sbin/pkg
```

Для более ранних версий FreeBSD утилиту pkgng необходимо установить из Коллекции Портов или из бинарных пакетов.

Для установки pkgng из порта, запустите следующее:

```
# cd /usr/ports/ports-mgmt/pkg
# make
# make install clean
```

Для установки из бинарного пакета, выполните:

```
# pkg_add -r pkg
```

Действующие инсталляции FreeBSD требуют преобразования базы данных установленных пакетов утилиты pkg_install к новому формату. Для выполнения конвертирования, запустите:

```
# pkg2ng
```

Вышеприведенный шаг не требуется для новых инсталляций, в которых не было установлено программ сторонних разработчиков.

Этот шаг необратим. После перевода базы данных установленных пакетов к формату pkgng утилитами pkg_install более пользоваться не следует.

В процессе конвертирования базы данных установленных пакетов могут возникать сообщения об ошибках. На данном этапе их можно игнорировать, так как перечень программ сторонних разработчиков, информация о которых не была преобразована, будет перечислен по завершении работы `pkg2ng`. Над этим перечнем придется поработать вручную.

Чтобы убедиться, что Коллекция Портов FreeBSD регистрирует новые программы при помощи pkgng, а не pkg_install, для версий FreeBSD, предшествующих 10.*X*, в файл /etc/make.conf необходимо внести следующую запись:

```
WITH_PKGNG= yes
```

4.5.2. Настройка окружения pkgng

Система управления пакетами pkgng при выполнении большинства операций обращается к пакетному репозиторию. Адрес используемого по умолчанию репозитория указан в /usr/local/etc/pkg.conf или в переменной окружения `PACKAGESITE`. Последняя переопределяет адрес, указанный в конфигурационном файле.

Дополнительные опции конфигурации pkgng описаны в pkg.conf(5).

4.5.3. Основные операции pkgng

Информацию по работе с pkgng можно найти на странице справочника pkg(8), или в выводе утилиты `pkg`, запущенной без аргументов.

Каждый аргумент команды pkgng описан на соответствующей странице справочника. Например, чтобы ознакомиться со страницей справочника для `pkg install`, запустите любую из двух нижеследующих команд:

```
# pkg help install
```

```
# man pkg-install
```

4.5.3.1. Получение информации об установленных пакетах при помощи pkgng

Информация об установленных в системе пакетах может быть отображена при помощи команды `pkg info`. Подобно до pkg_info(1), в выводе перечисляются версии и описания всех установленных пакетов.

Если необходима информация о конкретном пакете, выполните:

```
# pkg info packagename
```

Например, для получения версии pkgng, который установлен в системе, запустите:

```
# pkg info pkg
pkg-1.0.2           New generation package manager
```

4.5.3.2. Установка и удаление пакетов при помощи pkgng

В общем, бинарные пакеты устанавливаются при помощи:

```
# pkg install packagename
```

Команда `pkg install` обращается к пакетному репозиторию, это упоминалось в Настройка окружения pkgng. Команда pkg-add(8), напротив, не выполняет обращений к пакетному репозиторию, а также игнорирует переменную `PACKAGESITE`. Как следствие - зависимости не отслеживаются, и необходимые зависимые компоненты не скачиваются с удаленного источника. В этом разделе описана работа с `pkg install`. За подробностями работы с `pkg add` обратитесь к справочной странице по pkg-add(8).

Утилита `pkg install` может устанавливать дополнительные бинарные пакеты. Например, для установки curl, выполните:

```
# pkg install curl
Updating repository catalogue
Repository catalogue is up-to-date, no need to fetch fresh copy
The following packages will be installed:

    Installing ca_root_nss: 3.13.5
    Installing curl: 7.24.0

The installation will require 4 MB more space

1 MB to be downloaded

Proceed with installing packages [y/N]: y
ca_root_nss-3.13.5.txz      100%    255KB   255.1KB/s 255.1KB/s 00:00
curl-7.24.0.txz         100%    1108KB  1.1MB/s 1.1MB/s     00:00
Checking integrity... done
```

```
Installing ca_root_nss-3.13.5... done
Installing curl-7.24.0... done
```

Новый пакет, как и любые дополнительные пакеты, которые были установлены как зависимости, перечисляются в списке установленных пакетов:

```
# pkg info
ca_root_nss-3.13.5  The root certificate bundle from the Mozilla Project
curl-7.24.0 Non-interactive tool to get files from FTP, GOPHER, HTTP(S) servers
pkg-1.0.2   New generation package manager
```

Пакеты, в которых более нет необходимости, могут быть удалены при помощи `pkg delete`. Например, если выяснится, что curl не нужен:

```
# pkg delete curl
The following packages will be deleted:

    curl-7.24.0_1

The deletion will free 3 MB

Proceed with deleting packages [y/N]: y
Deleting curl-7.24.0_1... done
```

4.5.3.3. Обновление установленных пакетов при помощи pkgng

Пакеты, версии которых устарели, можно найти при помощи команды `pkg version`. Версии установленных пакетов сравниваются с версиями приложений из локального дерева портов, а в случае отсутствия портов pkg-version(8) обращается к удаленному репозиторию пакетов.

При помощи pkgng можно обновлять пакеты до новых версий. Предположим, вышла новая версия curl. Установленный пакет можно обновить к новой версии, выполнив:

```
# pkg upgrade
Updating repository catalogue
repo.txz        100%    297KB 296.5KB/s 296.5KB/s   00:00
The following packages will be upgraded:

    Upgrading curl: 7.24.0 -> 7.24.0_1

1 MB to be downloaded

Proceed with upgrading packages [y/N]: y
curl-7.24.0_1.txz   100% 1108KB 1.1MB/s 1.1MB/s     00:00
Checking integrity... done
```

```
Upgrading curl from 7.24.0 to 7.24.0_1... done
```

4.5.3.4. Аудит безопасности пакетов при помощи pkgng

Изредка в приложениях из Коллекции Портов обнаруживаются уязвимости. В pkgng встроена возможность выполнять аудит безопасности, действующая подобно приложению из ports-mgmt/portaudit. Для выполнения аудита установленных в систему программ, выполните:

```
# pkg audit -F
```

4.5.4. Сложные вопросы работы с pkgng

4.5.4.1. Автоматическое удаление неиспользуемых зависимостей при помощи pkgng

После удаления пакета в системе могут остаться неиспользуемые зависимости, наподобие security/ca_root_nss из примера выше. Такие пакеты остаются установленными, несмотря на то, что они более не требуются другим пакетам. Определить и удалить неиспользуемые пакеты, которые были установлены как зависимости, можно при помощи:

```
# pkg autoremove
Packages to be autoremoved:
    ca_root_nss-3.13.5

The autoremoval will free 723 kB

Proceed with autoremoval of packages [y/N]: y
Deinstalling ca_root_nss-3.13.5... done
```

4.5.4.2. Резервное копирование базы данных установленных пакетов pkgng

В отличие от традиционной системы управления пакетами, pkgng располагает своим собственным механизмом резервного копирования базы данных. Для ручного создания резервной копии базы данных установленных пакетов, выполните:

```
# pkg backup -d pkgng.db
```

Замените имя файла *pkgng.db* на более подходящее.

В дополнение, pkgng содержит скрипт periodic(8), позволяющий выполнять ежедневное автоматическое копирование базы данных установленных пакетов. Для активации данной возможности задайте переменной `daily_backup_pkgng_enable` значение `YES` в файле periodic.conf(5).

Для предотвращения периодического запуска аналогичного скрипта

системы pkg_install, также выполняющего резервное копирование базы данных установленных пакетов, задайте переменной daily_backup_pkgdb_enable значение NO в файле periodic.conf(5).

Для восстановления содержимого базы данных установленных пакетов из резервной копии, выполните:

```
# pkg backup -r /path/to/pkgng.db
```

4.5.4.3. Удаление копий устаревших пакетов в системе pkgng

По умолчанию, pkgng сохраняет копии установленных бинарных пакетов в каталог, указанный переменной PKG_CACHEDIR в pkg.conf(5). При обновлении пакетов командой pkg upgrade старые версии обновленных пакетов автоматически не удаляются.

Для удаления устаревших версий бинарных пакетов из кеш-каталога, выполните:

```
# pkg clean
```

4.5.4.4. Изменение метаданных пакетов pkgng

Со временем программы из Коллекции Портов могут претерпевать изменения старшего (major) номера версии. В отличие от pkg_install, pkgng располагает встроенной командой для обновления информации о происхождении (origin) пакета. Например, изначально порт lang/php5 был версии 5.3. Позже этот порт был переименован в lang/php53, а под именем lang/php5 был создан порт версии 5.4. Утилитам системы pkg_install для обновления информации о происхождении (origin) пакета в собственной базе данных установленных пакетов потребовалась бы помощь дополнительного программного обеспечения, такого как ports-mgmt/portmaster.

В отличие от ports-mgmt/portmaster и ports-mgmt/portupgrade для портов, порядок перечисления новой и старой версий отличаются. Для pkgng необходим следующий порядок:

```
# pkg set -o category/oldport:category/newport
```

Например, в вышеприведенном случае для замены информации о происхождении пакета, выполните:

```
# pkg set -o lang/php5:lang/php53
```

Еще один пример: для изменения информации о происхождении пакета с lang/ruby18 на lang/ruby19, выполните:

```
# pkg set -o lang/ruby18:lang/ruby19
```

И последний пример: для замены информации о происхождении пакета разделяемой библиотеки libglut с graphics/libglut на graphics/freeglut, запустите:

```
# pkg set -o graphics/libglut:graphics/freeglut
```

Выполняя замену информации о происхождении пакетов, в большинстве случаев также требуется переустановить пакеты, которые зависят от изменившегося пакета. Для принудительной переустановки зависящих пакетов, выполните:

```
# pkg install -Rf graphics/freeglut
```

4.6. Использование Коллекции Портов

В этом разделе даны базовые сведения по использованию Коллекции Портов для установки или удаления программ. Детальное описание существующих целей `make` и переменных окружения находится в ports(7).

В июле 2012 года проект Портов FreeBSD сменил систему контроля версий: на смену CVS пришел Subversion. Рекомендуемым способом работы с деревом портов является Portsnap. Пользователи, которым требуется локальная подгонка портов (то есть, поддержание дополнительных локальных патчей), возможно предпочтут непосредственное использование Subversion. 28 февраля 2013 года сервис CVSup был выведен из эксплуатации, и дальнейшее использование последнего не рекомендуется.

4.6.1. Получение Коллекции Портов

Коллекция Портов - это набор файлов, состоящий из Makefile, патчей и файлов описаний, хранимых в /usr/ports. Этот набор файлов предназначен для построения и установки приложений во FreeBSD. В нижеследующих разделах описано несколько способов получения Коллекции Портов на тот случай, если Коллекция не была установлена во время инсталляции FreeBSD.

Procedure: Метод Portsnap

Portsnap это быстрый и удобный инструмент для получения Коллекции Портов, и в то же время - предпочитаемый выбор большинства пользователей.

1. Скачайте сжатый снэпшот Коллекции Портов в /var/db/portsnap.

```
# portsnap fetch
```

2. Если вы запускаете Portsnap впервые, извлеките снэпшот в /usr/ports:

```
# portsnap extract
```

3. По завершении первого запуска Portsnap, как было показано выше, /usr/ports может быть обновлен при помощи:

```
# portsnap fetch
# portsnap update
```

Procedure: Метод Subversion

Если необходим контроль за деревом портов (например, для поддержания локальных изменений), то для получения Коллекции Портов может быть задействован Subversion. Обратитесь к Subversion Primer за детальным описанием Subversion.

1. Для создания рабочей копии дерева портов необходимо иметь установленный Subversion. Если порты есть в наличии, то установите Subversion выполнив следующее:

```
# cd /usr/ports/devel/subversion
# make install clean
```

Если портов нет, то Subversion может быть установлен при помощи системы пакетов:

```
# pkg_add -r subversion
```

Если же для управления пакетами используется pkgng, то Subversion устанавливается при помощи следующей команды:

```
# pkg install subversion
```

2. Создайте рабочую копию дерева портов. Для ускорения процесса вместо *svn0.us-east.FreeBSD.org* укажите ближайшее к вам зеркало Subversion. Коммиттерам необходимо сначала прочитать Subversion Primer, чтобы удостовериться, что выбран корректный протокол.

```
# svn checkout https://svn0.us-east.FreeBSD.org/ports/head /usr/ports
```

3. При наличии рабочей копии /usr/ports все последующие обновления выполняются просто:

```
# svn update /usr/ports
```

Procedure: Метод с использованием Sysinstall

Этот метод подразумевает использование sysinstall для установки Коллекции Портов из установочного носителя. Учтите, что в итоге будет установлена старая копия Коллекции Портов, которая была актуальна на момент создания релиза. Если у вас есть подключение к Интернет, то вам необходимо пользоваться одним из вышеупомянутых методов.

1. Работая как пользователь `root`, запустите `sysinstall` так, как это показано ниже:

```
# sysinstall
```

2. Опуститесь вниз и выберите Configure, нажмите `Enter`
3. Опуститесь вниз и выберите Distributions, затем нажмите `Enter`
4. Опуститесь вниз к пункту ports, нажмите клавишу `Пробел`
5. Поднимитесь вверх к Exit, нажмите `Enter`
6. Выберите желаемый носитель для установки, например, CDROM, FTP и так далее.
7. Перейдите на пункт меню Exit и нажмите `Enter`.
8. Нажмите `X` для выхода из sysinstall.

4.6.2. Миграция с CVSup/csup на portsnap

Начиная с 28 февраля 2013 года дерево портов более не экспортируется в CVS, поэтому CVSup и csup не будут доставлять обновления для дерева портов.

Procedure: Миграция на Portsnap

Для миграции потребуется около 1 ГБ дискового пространства в /usr, в добавок, для Portsnap необходимо около 150 МБ дискового пространства в /var.

1. Если у вас настроено автоматическое обновление портов, например при помощи задания cron(8), запускающего CVSup или csup, то его необходимо будет отключить.

2. Переместите существующее дерево портов во временный каталог:

```
# mv /usr/ports /usr/ports.old
```

3. При помощи Portsnap скачайте новое дерево портов и извлеките его в /usr/ports:

```
# portsnap fetch extract
```

4. Переместите дистрибутивные файлы (distfiles) и сохраненные пакеты в новое дерево портов:

```
# mv /usr/ports.old/distfiles /usr/ports
# mv /usr/ports.old/packages /usr/ports
```

5. Удалите старое дерево портов:

```
# rm -rf /usr/ports.old
```

6. Если ранее использовался CVSup, то сейчас его можно удалить:

```
# pkg_delete -r -v cvsup-without-gui-\*
```

Пользователям pkgng необходимо запустить следующую команду:

```
# pkg delete cvsup-without-gui
```

4.6.3. Установка портов

Скелетом порта является набор файлов, который указывает вашей системе FreeBSD, как откомпилировать и установить программу. Скелет каждого порта включает:

- Makefile: этот файл содержит различные директивы, которые определяют, как приложение должно быть откомпилировано и куда в вашей системе оно должно быть установлено.
- distinfo: этот файл содержит информацию о файлах, которые должны быть загружены для сборки порта, а также их контрольные суммы (sha256(1)) для проверки того, что файлы не были повреждены в процессе загрузки.
- files: этот каталог содержит патчи, необходимые для компиляции и установки программы в вашей системе FreeBSD. Этот каталог также может содержать другие файлы, используемые для построения порта.
- pkg-descr: этот файл содержит более подробное описание программы.

- pkg-plist: это список всех файлов, которые будут установлены портом. В нем также содержатся указания системе портов на удаление определенных файлов во время удаления порта.

В некоторых портах присутствуют и другие файлы, такие, как pkg-message. Система портов использует эти файлы для обработки особых ситуаций. Если вы хотите узнать более подробно об этих файлах и о портах вообще, то обратитесь к Руководству по созданию портов для FreeBSD.

Порт не содержит собственно исходного кода, также известного как "дистрибутивный файл" (distfile). Способ распространения исходного кода определяется предпочтениями автора программы. Ниже описаны два способа установки порта FreeBSD.

Для установки портов вы должны войти в систему как пользователь `root`.

Перед установкой любого порта необходимо убедиться в наличии свежей Коллекции Портов и заглянуть на http://vuxml.freebsd.org/, где могут освещаться вопросы безопасности, связанные с конкретным портом. Если у вас установлен ports-mgmt/portaudit, то перед установкой нового порта запустите `portaudit -F` для загрузки свежей базы данных уязвимостей. Проверка безопасности и обновление базы данных будут выполняться при ежедневной проверке безопасности системы. За дальнейшей информацией обратитесь к страницам справочника portaudit(1) и periodic(8).

Использование Коллекции Портов предполагает наличие работающего подключения к Интернет. В противном случае вам придется раздобыть и поместить копию дистрибутивного файла в каталог /usr/ports/distfiles вручную.

Первым делом переместитесь в каталог устанавливаемого порта:

```
# cd /usr/ports/sysutils/lsof
```

Для компиляции (или построения - "build") порта наберите команду `make`. Вы должны увидеть вывод команды, подобный следующему:

```
# make
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===>  Extracting for lsof-4.57
...
[вывод команды при распаковке опущен]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
===>  Patching for lsof-4.57
===>  Applying FreeBSD patches for lsof-4.57
===>  Configuring for lsof-4.57
...
[вывод команды при конфигурации опущен]
```

```
...
===>  Building for lsof-4.57
...
[вывод команды при компиляции опущен]
...
#
```

По завершении компиляции вы снова вернетесь к приглашению командного процессора. Следующим шагом является установка порта при помощи `make install`:

```
# make install
===>  Installing for lsof-4.57
...
[вывод команды при установке опущен]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.57
===>   Registering installation for lsof-4.57
===>  SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
#
```

На этом этапе, получив приглашение оболочки, вы уже можете запустить установленное приложение. Так как `lsof` является программой, которая запускается с повышенными правами, выдаётся предупреждение о безопасности. Во время построения и установки портов следует обращать внимание на любые возникающие предупреждения.

Хорошей идеей является удаление рабочего подкаталога, содержащего временные файлы, использовавшиеся во время компиляции. Такое действие помогает беречь дисковое пространство и минимизирует вероятность возникновения проблем в дальнейшем, при обновлении до более новой версии порта.

```
# make clean
===>  Cleaning for lsof-4.57
#
```

Вы можете сэкономить два лишних шага, просто выдав команду `make install clean` вместо `make`, `make install` и последующей `make clean` в трех отдельных шагах.

Установка порта единственной командой `make install` чревата возможными частыми остановками процесса инсталляции из-за ожидания вмешательства пользователя: некоторым портам требуется ввод опций. Чтобы избежать траты времени, особенно для портов с многими зависимостями, запустите сначала `make config-recursive` чтобы выполнить

конфигурирование всех опций за один заход. Далее, запустите `make install [clean]`.

Когда используется `config-recursive`, перечень подлежащих настройке портов собирается целью `all-depends-list` команды make(1). Часто рекомендуется повторять запуск `make config-recursive` до тех пор, пока не будут определены все опции зависимых портов, а меню dialog(1) выбора опций портов появляться перестанут. При этом можно быть уверенным, что все опции были настроены как и намеревалось.

Некоторые командные процессоры для ускорения поиска исполняемых файлов и команд кэшируют имена программ, доступных для вызова из каталогов, перечисленных в переменной окружения `PATH`. Если вы используете `tcsh`, то вам может потребоваться набрать `rehash`, после чего свежеустановленную программу можно будет вызывать без указания полного пути. Для командного интерпретатора `sh` выполните `hash -r`. Дополнительную информацию можно найти в документации к вашему командному процессору.

В некоторых имеющихся в продаже комплектах DVD от третьих лиц, таких как the FreeBSD Toolkit от FreeBSD Mall, содержатся дистрибутивные файлы (distfiles). Их можно использовать с Коллекцией Портов. Смонтируйте DVD в /cdrom. Если вы используете иную точку монтирования, укажите её в переменной make(1) `CD_MOUNTPTS`. Если необходимые для построения порта дистрибутивные файлы находятся на диске, то они будут задействованы автоматически.

Лицензии некоторых портов не позволяют помещать их на DVD. Причиной тому может служить обязательность заполнения регистрационной формы перед загрузкой, или запрет на дальнейшее распространение. Если вы хотите установить порт, которого нет на DVD, вам нужно иметь подключение к Интернет.

Для загрузки файлов система портов использует утилиту fetch(1), которая проверяет значения некоторых переменных окружения, включая `FTP_PASSIVE_MODE`, `FTP_PROXY` и `FTP_PASSWORD`. Если вы находитесь за сетевым экраном или для работы с FTP/HTTP вам необходимо использовать прокси, то определите соответствующие переменные. Обратитесь к справочной странице по fetch(3) для получения полного списка переменных.

Пользователям, которые не могут быть постоянно подключены к сети, поможет команда `make fetch`. Запустите эту команду в каталоге /usr/ports, и требуемые файлы будут загружены. Эта команда также работает и с вложенными категориями, например: /usr/ports/net. Заметьте, что если порт имеет зависимости от библиотек или других портов, то команда *не будет* загружать дистрибутивные файлы для зависимых портов. Для загрузки всех зависимых дистрибутивных файлов задействуйте команду `make fetch-recursive`.

Вы можете построить все порты в категории за раз, запустив команду `make` в каталоге верхнего уровня. Однако это опасно, так как некоторые порты не

могут сосуществовать. В других случаях некоторые порты могут устанавливать два различных файла с одним и тем же именем.

В некоторых редких случаях пользователям необходимо получить tar-архивы с сайтов, отличающихся от указанных по умолчанию в MASTER_SITES. Вы можете переопределить значение MASTER_SITES посредством следующей команды:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

В этом примере значение переменной MASTER_SITES изменено на ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/.

Некоторые порты позволяют указывать опции, которые включают или выключают построение отдельных частей приложения, изменяют некоторые параметры безопасности, а также задают прочие настройки. Примерами таких портов могут служить: www/firefox, security/gpgme и mail/sylpheed-claws. Если для порта существуют опции компиляции, то перед началом построения будет отображено меню.

4.6.3.1. Переназначение рабочего и целевого каталогов

При помощи переменных WRKDIRPREFIX и PREFIX можно переопределить назначаемые по умолчанию рабочий и целевой каталоги. Например:

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

будет компилировать порт в каталоге /usr/home/example/ports, а установит всё в /usr/local.

```
# make PREFIX=/usr/home/example/local install
```

будет компилировать порт в каталоге /usr/ports, а установит в /usr/home/example/local.

И, конечно же,

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

использует обе возможности.

Альтернативным способом является задание этих каталогов в переменных окружения. Обратитесь к страницам справки по вашему командному процессору для получения инструкций о том, как установить переменные окружения.

4.6.3.2. Повторная настройка опций портов

Некоторые порты предоставляют ncurses-меню, содержащее опции построения. Будучи однажды успешно определены, опции сохраняются, и при повторном построении порта меню не вызывается. Для изменения сохраненных опций существует несколько способов повторного вызова меню. Один из них - зайти в каталог порта и набрать `make config`. Второй способ - запустить команду `make showconfig`. Еще один способ - выполнить команду `make rmconfig`, которая удалит все ранее отмеченные опции и позволит вам начать конфигурирование сначала. Эти и другие способы детально описаны на странице ports(7).

4.6.4. Удаление установленных портов

Установленные порты и пакеты удаляются при помощи команды pkg_delete(1):

```
# pkg_delete lsof-4.57
```

4.6.5. Обновление портов

Прежде всего, при помощи pkg_version(1) просмотрите, нет ли в Коллекции Портов более новых версий установленных приложений:

```
# pkg_version -v
```

4.6.5.1. Прочитайте /usr/ports/UPDATING

После обновления Коллекции Портов, и перед тем, как обновить приложение из порта, сверьтесь с файлом /usr/ports/UPDATING. В нем дана информация по различным вопросам и дополнительным шагам, которые могут быть необходимы для обновления порта, включая информацию о таких вещах как изменения форматов файлов, изменения в расположении конфигурационных файлов, или другие нестыковки с предыдущими версиями.

Если UPDATING противоречит чему-либо, написанному в этом документе, то доверьтесь информации из UPDATING.

4.6.5.2. Обновление Портов при помощи Portupgrade

Утилита portupgrade создана для простого обновления установленных портов. Она доступна из порта ports-mgmt/portupgrade. Установите её как и любой иной порт при помощи команды `make install clean`:

```
# cd /usr/ports/ports-mgmt/portupgrade
# make install clean
```

Проверьте перечень установленных портов командой `pkgdb -F` и устраните все несоответствия, о которых сообщит утилита. Хорошей идеей является выполнение этого действия регулярно, перед каждым обновлением.

Используйте `portupgrade -a` для обновления всех устаревших портов, установленных в вашей системе. Добавьте флаг `-i` если вы желаете получать запрос на каждый обновляемый порт.

```
# portupgrade -ai
```

Для обновления конкретного приложения, а не всех установленных портов, запустите `portupgrade pkgname`. Включите флаг `-R` при необходимости обновить все порты, требуемые данным приложением.

```
# portupgrade -R firefox
```

Для использования при установке пакетов, а не портов, укажите флаг `-P`. С этим параметром portupgrade будет искать пакеты в локальных каталогах, указанных в переменной окружения `PKG_PATH`, а если не найдет их, то загрузит с удаленного сайта. Если пакеты не могут быть найдены локально или загружены удаленно, portupgrade использует порты. Чтобы запретить использование портов, укажите `-PP`.

```
# portupgrade -PP gnome2
```

Для простой загрузки дистрибутивных файлов без построения или установки чего бы то ни было, задайте флаг `-F`. Дополнительную информацию можно получить на странице справки по утилите portupgrade(1).

4.6.5.3. Обновление портов при помощи portmaster

Утилита ports-mgmt/portmaster - это еще один инструмент для обновления установленных портов. Утилита portmaster опирается на возможности "базовой" системы и не зависит от других портов. Она использует информацию из /var/db/pkg/ для определения портов, подлежащих обновлению. Для установки утилиты выполните следующее:

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

Portmaster делит порты на четыре категории:

- Корневые порты: не зависят от других портов и нет портов, зависимых от корневых;
- Стволовые порты: не зависят от других портов, но есть порты, зависящие от стволовых;
- Веточные порты: зависят от других портов и есть порты, зависящие от веточных;
- Листьевые порты: зависят от других портов, но нет портов, зависящих от листьевых.

Чтобы проверить наличие обновлений для всех установленных портов, задайте утилите флаг `-L`:

```
# portmaster -L
===>>> Root ports (No dependencies, not depended on)
===>>> ispell-3.2.06_18
===>>> screen-4.0.3
        ===>>> New version available: screen-4.0.3_1
===>>> tcpflow-0.21_1
===>>> 7 root ports
...
===>>> Branch ports (Have dependencies, are depended on)
===>>> apache-2.2.3
        ===>>> New version available: apache-2.2.8
...
===>>> Leaf ports (Have dependencies, not depended on)
===>>> automake-1.9.6_2
===>>> bash-3.1.17
        ===>>> New version available: bash-3.2.33
...
===>>> 32 leaf ports

===>>> 137 total installed ports
        ===>>> 83 have new versions available
```

Все установленные порты могут быть обновлены при помощи одной команды:

```
# portmaster -a
```

По умолчанию, portmaster создаст дублирующий пакет перед удалением установленного порта. Если обновление порта прошло успешно, portmaster удалит дублирующий пакет. При помощи опции `-b` можно проинструктировать portmaster не выполнять автоматическое удаление дублирующего пакета. Указание опции `-i` запустит portmaster в интерактивном режиме, запрашивающим подтверждение пользователя перед обновлением каждого порта.

Если во время процесса обновления возникнут ошибки, задействуйте опцию `-f` для обновления/перестройки всех портов:

```
# portmaster -af
```

Также, portmaster может быть задействован для установки новых портов в систему, автоматически обновляя другие зависимые порты перед построением и установкой нового порта:

```
# portmaster shells/bash
```

За подробной информацией обратитесь к portmaster(8).

4.6.6. Порты и дисковое пространство

Работа с Коллекцией Портов со временем приводит к увеличению занимаемого дискового пространства. После построения и установки программы из порта удалите временный каталог work при помощи команды `make clean`. Для очистки всей Коллекции Портов наберите:

```
# portsclean -C
```

По прошествии некоторого времени у вас соберется множество дистрибутивных файлов в каталоге distfiles. Следующая команда удалит все дистрибутивные файлы, которые более не связаны ни с какими портами:

```
# portsclean -D
```

Для удаления всех дистрибутивных файлов, не связанных ни с одним установленным в вашу систему портом, наберите:

```
# portsclean -DD
```

Утилита `portsclean` является частью порта ports-mgmt/portupgrade.

Удобная утилита для автоматизации удаления портов, в которых вы более не нуждаетесь, доступна из порта ports-mgmt/pkg_cutleaves.

4.7. Действия после установки

После установки нового приложения вам обычно требуется прочесть любую имеющуюся документацию, отредактировать необходимые конфигурационные файлы и убедиться, что приложение запускается во время загрузки системы.

Очевидно, что шаги, в точности требуемые для конфигурации каждого приложения, отличаются. Однако, если если вы только что установили новое приложение и вам интересно, "Что же дальше?", то вам могут помочь следующие советы:

- Воспользуйтесь командой pkg_info(1) для определения того, куда и какие файлы были установлены. К примеру, если вы только что установили FooPackage версии 1.0.0, то по команде

  ```
  # pkg_info -L foopackage-1.0.0 | less
  ```

  будет выведен список всех файлов, установленных пакетом. Обратите особое внимание

на файлы, установленные в каталоги man/, которые являются справочными страницами, etc/, которые являются конфигурационными файлами, и doc/, которые являются более полной документацией.

Чтобы определить, какая версия приложения была установлена, выполните:

```
# pkg_info | grep -i foopackage
```

команда выведет список всех установленных пакетов, в названии которых присутствует *foopackage*. Замените *foopackage* на искомый фрагмент текста.

- Как только вы определите, куда были установлены справочные страницы приложения, просмотрите их при помощи команды man(1). Просмотрите примеры конфигурационных файлов, а также любую дополнительную документацию, если она была установлена.
- Если у приложения имеется веб-сайт, поищите там дополнительную документацию, ответы на часто задаваемые вопросы и так далее. Если вы не уверены, каков адрес веб-сайта, он может быть указан в выводе команды

```
# pkg_info foopackage-1.0.0
```

Строка `WWW:`, если она есть, должна содержать URL Web-сайта приложения.

- Как правило, порты приложений, которые должны запускаться при загрузке системы, устанавливают стартовые скрипты в каталог /usr/local/etc/rc.d. Просмотрите скрипт на предмет его корректности и, если необходимо, отредактируйте или переименуйте его. Обратитесь к разделу о Запуске сервисов за более подробной информацией.

4.8. Обработка нерабочих портов

Если вы встретили порт, который не компилируется:

1. Выясните, нет ли для порта решения проблемы в Problem Report database. Если оно есть, то вы можете воспользоваться предложенным решением.
2. Попросите помощи у мейнтейнера порта. Чтобы найти его адрес наберите команду `make maintainer` или просмотрите Makefile. Не забудьте указать имя и версию порта (скопировав строчку `$FreeBSD:` из файла Makefile) и включите в письмо весь вывод, предшествующий возникновению ошибки.

Некоторые порты поддерживаются группой людей из списка рассылки, а не отдельными людьми. В большинстве таких случаев адрес мейнтейнера выглядит подобно следующему: freebsd-listname@FreeBSD.org. Пожалуйста, учтите это при формулировании ваших вопросов.

В частности, если мейнтейнер порта - ports@FreeBSD.org, то такой порт

> вообще никем не поддерживается. Решение проблем и поддержка, если и имеют место, то приходят от общества, которое подписано на тот список рассылки. Волонтёры требуются всегда!

Если вы не получили ответ, то воспользуйтесь командой send-pr(1) для посылки сообщения о проблеме (изучите составление сообщений о проблеме во FreeBSD).

3. Исправьте его! В Руководстве по созданию портов содержится подробная информация об инфраструктуре портов, так что вы сможете исправить редкий неработающий порт или даже предложить свой собственный!
4. Воспользуйтесь pkg_add(1) и установите пакет вместо порта.

Глава 5. Το Σύστημα X Window

5.1. Обзор

FreeBSD использует X11 для того, чтобы дать пользователям мощный графический интерфейс. X11 является свободно доступной версией X Window System, реализованной в Xorg и XFree86™ (а также других программных пакетах, здесь не рассматриваемых). В версиях FreeBSD до и включая FreeBSD 5.2.1-RELEASE сервером X11 по умолчанию был XFree86™, выпускаемый The XFree86™ Project, Inc. Начиная с FreeBSD 5.3-RELEASE, официальной версией X11 по умолчанию стал Xorg, разработанный X.Org Foundation под лицензией, очень похожей на ту, которая используется FreeBSD. Под FreeBSD существуют также коммерческие X серверы.

Эта глава посвящена установке и настройке X11 в системе FreeBSD, с акцентом на релиз Xorg 7.7. За информацией о настройке XFree86™ (в более старых релизах FreeBSD XFree86™ был реализацией X11 по умолчанию), или более старых релизов Xorg, всегда можно обратиться к старым версиям Руководства FreeBSD по адресу http://docs.FreeBSD.org/doc/.

За дополнительной информацией по видео оборудованию, поддерживаемому X11, обратитесь к веб сайту Xorg.

После чтения этой главы вы будете знать:

- Как установить и настроить X11.
- О различных компонентах X Window System и их взаимодействии.
- Как установить и использовать различные оконные менеджеры.
- Как использовать шрифты TrueType® в X11.
- Как настроить вашу систему на графический интерфейс входа (XDM).

Перед чтением этой главы вам потребуется:

- Узнать, как устанавливать дополнительное программное обеспечение сторонних разработчиков (Установка приложений. порты и пакеты).

5.2. Основы X

Первое знакомство с X может оказаться чем-то вроде шока для тех, кто работал с другими графическими системами, такими, как Microsoft® Windows® или Mac OS®.

Хотя нет необходимости вникать во все детали различных компонентов X и их взаимодействия, некоторые базовые знания делают возможным использование сильных сторон X.

5.2.1. Почему именно X?

X не является первой оконной системой для UNIX®, но она самая популярная из них. До

работы над X команда ее разработчиков трудилась над другой оконной системой. Та система называлась "W" (от "Window"). X была просто следующей буквой в романском алфавите.

X можно называть "X", "X Window System", "X11" и множеством других терминов. Факт использования названия "X Windows" для X11 может задеть интересы некоторых людей; дополнительную информацию по этому поводу можно найти на странице справочной системы X(7).

5.2.2. Модель клиент/сервер в X

X изначально разрабатывалась, чтобы быть системой, ориентированной на работу в сети с использованием модели "клиент-сервер".

В модели работы X "X-сервер" работает на компьютере с клавиатурой, монитором и мышью. Область ответственности сервера включает управление дисплеем, обработку ввода с клавиатуры, мыши и других устройств ввода или вывода (например, "планшет" может быть использован в качестве устройства ввода, а видеопроектор в качестве альтернативного устройства вывода). Каждое X-приложение (например, XTerm или getenv(3)) является "клиентом". Клиент посылает сообщения серверу, такие, как "Пожалуйста, нарисуй окно со следующими координатами", а сервер посылает в ответ сообщения типа "Пользователь только что щёлкнул мышью на кнопке OK".

В случае использования дома или в офисе, сервер и клиенты X как правило будут работать на том же самом компьютере. Однако реально возможно запускать X-сервер на менее мощном настольном компьютере, а приложения X (клиенты) на, скажем, мощной и дорогой машине, обслуживающей целый офис. В этом сценарии X-клиент и сервер общаются через сеть.

Некоторых это вводит в заблуждение, потому что терминология X в точности обратна тому, что они ожидают. Они полагают, что "X-сервер" будет большой мощной машиной, стоящей на полу, а "X-клиентом" является машина, стоящая на их столах.

Важно помнить, что X-сервером является машина с монитором и клавиатурой, а X-клиенты являются программами, выводящими окна.

В протоколе нет ничего, что заставляет машины клиента и сервера работать под управлением одной и той же операционной системы, или даже быть одним и тем же типом компьютера. Определённо возможно запускать X-сервер в Microsoft® Windows® или Mac OS® от Apple, и есть множество свободно распространяемых и коммерческих приложений, которые это реализуют.

5.2.3. Оконный менеджер

Философия построения X очень похожа на философию построения UNIX®, "инструменты, не политика". Это значит, что X не пытаются диктовать то, как должна быть выполнена работа. Вместо этого пользователю предоставляются инструменты, а за пользователем остается принятие решения о том, как использовать эти инструменты.

Этот подход расширен в X тем, что не задается, как окна должны выглядеть на экране, как их двигать мышью, какие комбинации клавиш должны использоваться для переключения между окнами (то есть `Alt` + `Tab`, в случае использования Microsoft® Windows®), как должны выглядеть заголовки окон, должны ли в них быть кнопки для закрытия, и прочее.

Вместо этого X делегирует ответственность за это приложению, которое называется "Window Manager" (Менеджер Окон). Есть десятки оконных менеджеров для X: AfterStep, Blackbox, ctwm, Enlightenment, fvwm, Sawfish, twm, WindowMaker и другие. Каждый из этих оконных менеджеров предоставляет различные внешние виды и удобства; некоторые из них поддерживают "виртуальные рабочие столы"; некоторые из них позволяют изменять назначения комбинаций клавиш, используемых для управления рабочим столом; в некоторых есть кнопка "Start" или нечто подобное; некоторые поддерживают "темы", позволяя изменять внешний вид, поменяв тему. Эти оконные менеджеры, а также множество других, находятся в категории x11-wm коллекции портов.

Кроме того, оболочки KDE и GNOME имеют собственные оконные менеджеры, которые интегрированы в оболочку.

Каждый оконный менеджер также имеет собственный механизм настройки; некоторые предполагают наличие вручную созданного конфигурационного файла; некоторые предоставляют графические инструменты для выполнения большинства работ по настройке; по крайней мере один (Sawfish) имеет конфигурационный файл, написанный на диалекте языка Lisp.

Политика фокусирования

Другой особенностью, за которую отвечает оконный менеджер, является "политика фокусирования" мыши. Каждая оконная система должна иметь некоторый способ выбора окна для активации получения нажатий клавиш, а также визуальную индикацию того, какое окно активно.

Широко известная политика фокусировки называется "щелчок-для-фокуса" ("click-to-focus"). Эта модель используется в Microsoft® Windows®, когда окно становится активным после получения щелчка мыши.

X не поддерживает никакой конкретной политики фокусирования. Вместо этого менеджер окон управляет тем, какое окно владеет фокусом в каждый конкретный момент времени. Различные оконные менеджеры поддерживают разные методы фокусирования. Все они поддерживают метод щелчка для фокусирования, и большинство из них поддерживают некоторые другие методы.

Самыми популярными политики фокусирования являются:

фокус следует за мышью (focus-follows-mouse)

Фокусом владеет то окно, что находится под указателем мыши. Это не обязательно будет окно, которое находится поверх всех остальных. Фокус меняется при указании на другое окно, при этом также нет нужды щёлкать на нём.

нечеткий фокус (sloppy-focus)

С политикой focus-follows-mouse если мышь помещается поверх корневого окна (или заднего фона), то никакое окно фокус не получает, а нажатия клавиш просто пропадают. При использовании политики нечёткого фокуса он меняется только когда курсор попадает на новое окно, но не когда уходит с текущего окна.

щелчок для выбора фокуса (click-to-focus)

Активное окно выбирается щелчком мыши. Затем окно может быть "поднято" и появится поверх всех других окон. Все нажатия клавиш теперь будут направляться в это окно, даже если курсор переместится к другому.

Многие оконные менеджеры поддерживают и другие политики, а также вариации перечисленных. Обязательно обращайтесь к документации по оконному менеджеру.

5.2.4. Виджеты

Подход X, заключающийся в предоставлении инструментов, а не политики, распространяется и на виджеты, которые располагаются на экране в каждом приложении.

"Виджет" (widget) является термином для всего в пользовательском интерфейсе, на чём можно щёлкать или каким-то образом управлять; кнопки, зависимые (radio buttons) и независимые (check boxes) опции, иконки, списки и так далее. В Microsoft® Windows® это называется "элементами управления" ("controls").

Microsoft® Windows® и Mac OS® от Apple имеют очень жёсткую политику относительно виджетов. Предполагается, что разрабатываемые приложения обязательно должны иметь похожий внешний вид. Что касается X, то было решено, что не нужно требовать обязательного использования какого-то определённого графического стиля или набора виджетов.

В результате не стоит ожидать от X-приложений похожести во внешнем виде. Существует несколько популярных наборов виджетов и их разновидностей, включая оригинальный набор виджетов Athena от MIT, Motif® (по образу которого был разработан набор виджетов в Microsoft® Windows®, все эти скошенные углы и три разновидности серого цвета), OpenLook и другие.

В большинстве появляющихся в настоящее время приложений для X будет использоваться современно выглядящий набор виджетов, либо Qt, используемый в KDE, либо GTK+, используемый проектом GNOME. В этом отношении наблюдается унификация внешнего вида рабочего стола в UNIX®, что определённо облегчает жизнь начинающему пользователю.

5.3. Установка X11

Версией X11 по умолчанию для FreeBSD является Xorg. Xorg это сервер X дистрибутива

открытой реализации X Window System, выпущенной X.Org Foundation. Xorg основан на коде XFree86™ 4.4RC2 и X11R6.6. Версия Xorg, доступная на данный момент из коллекции портов FreeBSD: 7.7.

Для сборки и установки Xorg из Коллекции портов, выполните:

```
# cd /usr/ports/x11/xorg
# make install clean
```

Перед сборкой полной версии Xorg удостоверьтесь в наличии хотя бы 4 GB свободного места.

Кроме того, X11 может быть установлен непосредственно из пакетов. Бинарные пакеты, устанавливаемые pkg_add(1), доступны и для X11. Когда pkg_add(1) используется для удаленной загрузки пакетов, номер версии пакета необходимо удалить. pkg_add(1) автоматически установит последнюю версию приложения.

Таким образом, для загрузки и установки пакета Xorg, просто наберите:

```
# pkg_add -r xorg
```

В примерах выше будет установлен полный дистрибутив X11, включая серверы, клиенты, шрифты и так далее. Также доступны и отдельные пакеты и порты для различных частей X11.

В оставшейся части главы будет рассказано о том, как сконфигурировать X11 и настроить рабочее окружение.

5.4. Конфигурация X11

5.4.1. Перед тем, как начать

Перед настройкой X11 необходима следующая информация о конфигурируемой системе:

- Характеристики монитора
- Набор микросхем, используемый в видеоадаптере
- Объём видеопамяти

Характеристики монитора используются в X11 для определения рабочего разрешения и частоты. Эти характеристики обычно могут быть получены из документации, которая прилагается к монитору или с сайта производителя. Тут нужны два диапазона значений, для частоты горизонтальной развёртки и для частоты вертикальной синхронизации.

Набор микросхем графического адаптера определяет, модуль какого драйвера использует X11 для работы с графическим оборудованием. Для большинства типов микросхем это

может быть определено автоматически, но все же его полезно знать на тот случай, когда автоматическое определение не работает правильно.

Объём видеопамяти графического адаптера определяет разрешение и глубину цвета, с которым может работать система. Это важно, чтобы пользователь знал ограничения системы.

5.4.2. Конфигурирование X11

Начиная с версии 7.3, Xorg зачастую может работать без какого-либо файла настройки, для его запуска достаточно просто набрать:

```
% startx
```

Начиная с версии 7.4, Xorg может использовать HAL для автоматического поиска клавиатуры и мыши. Порты sysutils/hal и devel/dbus будут инсталлированы как зависимости x11/xorg, но для их включения необходимо иметь следующие записи в /etc/rc.conf file:

```
hald_enable="YES"
dbus_enable="YES"
```

Эти сервисы должны быть запущены (вручную или при загрузки системы) до последующей загрузки Xorg конфигурации.

Автоматическая конфигурация не всегда может сработать на некотором оборудовании, либо создать не совсем ту настройку, которая желаема. В этих случаях, необходима ручная настройка конфигурации.

Такие оконные менеджеры, как GNOME, KDE или Xfce имеют собственные утилиты, позволяющие пользователю легко устанавливать такие параметры, как разрешение экрана. Поэтому, если конфигурация по-умолчанию не подходящая и вы планируете инсталлировать эти оконные менеджеры, просто можете продолжить настройку рабочей среды, используя их собственные утилиты для установок параметров экрана.

Процесс настройки X11 является многошаговым. Первый шаг заключается в построении начального конфигурационного файла. Работая с правами суперпользователя, просто запустите:

```
# Xorg -configure
```

При этом в каталоге /root будет создан скелет конфигурационного файла X11 под именем xorg.conf.new (там, куда после su(1) или непосредственного входа будет указывать переменная `$HOME`). Программа X11 сделает попытку распознать графическое оборудование системы и запишет конфигурационный файл, загружающий правильные драйверы для обнаруженного оборудования в системе.

Следующим шагом является тестирование существующей конфигурации для проверки того, что Xorg может работать с графическим оборудованием в настраиваемой системе. Для этого выполните:

```
# Xorg -config xorg.conf.new
```

Начиная с Xorg 7.4 и выше, это тестирование покажет лишь черный экран, что делает диагностику не совсем полноценным. Старое поведение будет доступно при использовании опции `retro`

```
# Xorg -config xorg.conf.new -retro
```

Если появилась чёрно-белая сетка и курсор мыши в виде X, то настройка была выполнена успешно. Для завершения тестирования просто нажмите одновременно `Ctrl` + `Alt` + `Backspace`.

Данная комбинация включена по-умолчанию до Xorg версии 7.3. Для включения этого в версии 7.4 и выше, вы должны ввести следующую команду в любом эмуляторе X терминала:

```
% setxkbmap -option terminate:ctrl_alt_bksp
```

или создать конфигурационный файл клавиатуры для hald называемый x11-input.fdi и сохранить его в /usr/local/etc/hal/fdi/policy директории. Данный файл должен содержать следующие строчки:

```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbOptions"
type="string">terminate:ctrl_alt_bksp</merge>
    </match>
  </device>
</deviceinfo>
```

Вам может потребоваться перезагрузка системы для вступления параметров hald в силу.

Если мышь не работает, ее необходимо настроить. Обратитесь к Настройка мыши (Mouse Settings) в главе об установке FreeBSD. Дополнительно, начиная с версии 7.4, секция `InputDevice` в xorg.conf игнорируется в пользу автоматического поиска устройств. Для возвращения старого поведения, добавьте следующие строчки в секции `ServerLayout` или `ServerFlags`:

```
Option "AutoAddDevices" "false"
```

Устройства ввода могут конфигурированы затем как в предыдущих версиях, вместе с другими необходимыми опциями (такими, как переключение раскладок клавиатуры например).

Как ранее уже сообщалось, начиная с версии 7.4, по-умолчанию, hald демон будет пытаться распознать вашу клавиатуру автоматически. Есть возможность, что раскладка вашей клавиатуры или ее модель будут определены некорректно. Такие оконные менеджеры как GNOME, KDE или Xfce содержат свои инструменты для конфигурирования клавиатур. Тем не менее, можно установить параметры клавиатуры непосредственно с помощью утилиты setxkbmap(1) или через haldконфигурационные правила.

Например, если вы хотите использовать клавиши PC 102 клавиатуры, идущая с французской раскладкой, мы должны создать конфигурационный файл клавиатуры для hald называемый x11-input.fdi и сохранить в /usr/local/etc/hal/fdi/policy директории. Этот файл должен содержать следующие строчки:

```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbModel"
type="string">pc102</merge>
      <merge key="input.x11_options.XkbLayout" type="string">fr</merge>
    </match>
  </device>
</deviceinfo>
```

Если этот файл уже существует, просто скопируйте и добавьте эти строчки в данный файл.

Вы должны будете перезагрузить систему, чтобы заставить hald применить настройки.

Есть возможность проделать ту же конфигурацию из X терминала или скрипт следующей командой:

```
% setxkbmap -model pc102 -layout fr
```

Файл /usr/local/shared/X11/xkb/rules/base.lst содержит список различных клавиатур, доступные опции и раскладки.

Теперь выполните тонкую настройку в файле xorg.conf.new по своему вкусу. Откройте файл

в текстовом редакторе, таком, как emacs(1) или ee(1). Сначала задайте частоты для монитора. Они обычно обозначаются как частоты горизонтальной и вертикальной синхронизации. Эти значения добавляются в файл xorg.conf.new в раздел "Monitor":

```
Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        HorizSync    30-107
        VertRefresh  48-120
EndSection
```

Ключевых слов HorizSync и VertRefresh может и не оказаться в файле конфигурации. Если их нет, то они должны быть добавлены, с указанием корректных значений горизонтальной частоты синхронизации после ключевого слова HorizSync и вертикальной частоты синхронизации после ключевого слова VertRefresh. В примере выше были введены частоты монитора настраиваемой системы.

X позволяет использовать возможности технологии DPMS (Energy Star) с поддерживающими её мониторами. Программа xset(1) управляет временными задержками и может явно задавать режимы ожидания, останова и выключения. Если вы хотите включить использование возможностей DPMS вашего монитора, вы должны добавить следующую строку в раздел, описывающий монитор:

```
Option       "DPMS"
```

Пока файл конфигурации xorg.conf.new открыт в редакторе, выберите желаемые разрешение и глубину цвета, которые будут использоваться по умолчанию. Они задаются в разделе "Screen":

```
Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
        DefaultDepth 24
        SubSection "Display"
                Viewport  0 0
                Depth     24
                Modes     "1024x768"
        EndSubSection
EndSection
```

Ключевое слово DefaultDepth описывает глубину цвета, с которой будет работа по умолчанию. Это значение может быть переопределено при помощи параметра командной строки -depth для Xorg(1). Ключевое слово Modes описывает разрешение, с которым нужно работать при данной глубине цвета. Заметьте, что поддерживаются только те стандартные

режимы VESA, что определены графическим оборудованием настраиваемой системы. В примере выше глубина цвета по умолчанию равна двадцати четырём битам на пиксел. При такой глубине цвета принимается разрешение в 1024 на 768 точек.

Наконец, запишите конфигурационный файл и протестируйте его при помощи тестового режима, описанного выше.

При решении проблем могут помочь лог файлы X11, в которых находится информация по каждому устройству, к которому подключен сервер X11. Лог файлам Xorg названия даются в формате /var/log/Xorg.0.log. Имена лог файлам могут даваться от Xorg.0.log до Xorg.8.log и так далее.

Если все в порядке, то конфигурационный файл нужно установить в общедоступное место, где его сможет найти Xorg(1). Обычно это /etc/X11/xorg.conf или /usr/local/etc/X11/xorg.conf.

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

Теперь процесс настройки X11 завершен. Xorg теперь можно запустить с помощью startx(1). X11 можно также запустить через xdm(1).

5.4.3. Тонкие вопросы настройки

5.4.3.1. Конфигурирование при работе с графическими чипсетами Intel® i810

Конфигурирование при работе с интегрированными наборами микросхем Intel® i810 требует наличия agpgart, программного интерфейса AGP, посредством которого X11 будет управлять адаптером. Подробности смотрите на странице справочника agp(4).

Это позволит конфигурировать графическое оборудование точно так же, как и любой другой графический адаптер. Заметьте, что для систем, у которых драйвер agp(4) в ядро не вкомпилирован, попытка погрузить модуль с помощью kldload(8) окончится неудачно. Этот драйвер должен оказаться в ядре во время загрузки, либо вкомпилированным, либо подгруженным посредством /boot/loader.conf.

5.4.3.2. Настройка широкоэкранного режима

Для этого раздела необходимо несколько больше навыков настройки. Если после использования описанных выше инструментов настройки в результате рабочей конфигурации не получается, в лог файлах достаточно информации для доведения конфигурации до рабочего уровня. Для настройки используется текстовый редактор.

Существующие широкоэкранные стандарты (WSXGA, WSXGA+, WUXGA, WXGA, WXGA+, и т.д.) поддерживают форматы изображения 16:10 и 10:9, которые могут быть проблемными. Для формата 16:10, например, возможны следующие разрешения экрана:

- 2560x1600
- 1920x1200

- 1680x1050
- 1440x900
- 1280x800

Иногда достаточно добавить одно из этих разрешений в качестве параметра `Mode` в раздел `Section "Screen"` вот так:

```
Section "Screen"
Identifier "Screen0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
    Viewport  0 0
    Depth     24
    Modes     "1680x1050"
EndSubSection
EndSection
```

Xorg может извлечь информацию о разрешении из монитора посредством I2C/DDC, так что у него есть данные, какие частоты и разрешения может поддерживать монитор.

Если эти `ModeLines` не определены в драйверах, может потребоваться дополнительная настройка Xorg. Используя /var/log/Xorg.0.log, можно извлечь достаточно информации для создания рабочей строки `ModeLine` вручную. Просто обратитесь к следующей информации:

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border:
0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border:
0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max
170 MHz
```

Эта информация называется EDID. Создание `ModeLine` из сводится к расположению номеров в правильном порядке:

```
ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

Для нашего примера `ModeLine` в `Section "Monitor"` будет выглядеть так:

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
```

```
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

После завершения редактирования конфигурации, X должен запуститься в новом широкоэкранном разрешении.

5.5. Использование шрифтов в X11

5.5.1. Шрифты Type1

Шрифты, используемые по умолчанию и распространяемые вместе с X11, вряд ли можно назвать идеально подходящими для применения в обычных издательских приложениях. Большие презентационные шрифты выглядят рвано и непрофессионально, а мелкие шрифты в getenv(3) вообще невозможно разобрать. Однако есть некоторое количество свободно распространяемых высококачественных шрифтов Type1 (PostScript®), которые можно без изменений использовать с X11. К примеру, в наборе шрифтов URW (x11-fonts/urwfonts) имеются высококачественные версии стандартных шрифтов type1 (Times Roman™, Helvetica™, Palatino™ и другие). В набор Freefonts (x11-fonts/freefonts) включено ещё больше шрифтов, однако большинство из них предназначено для использования в программном обеспечении для работы с графикой, например, Gimp, и они не вполне пригодны для использования в качестве экранных шрифтов. Кроме того, X11 с минимальными усилиями может быть настроена на использование шрифтов TrueType®. Более детальная информация находится на странице справочной системы X(7) и в разделе о шрифтах TrueType® ниже.

Для установки вышеупомянутых коллекций шрифтов Type1 из коллекции портов выполните следующие команды:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

То же самое нужно будет сделать для коллекции freefont и других. Чтобы X-сервер обнаруживал этих шрифты, добавьте соответствующую строку в файл настройки X сервера (/etc/X11/xorg.conf), которая должна выглядеть так:

```
FontPath "/usr/local/lib/X11/fonts/URW/"
```

Либо из командной строки при работе с X выполните:

```
% xset fp+ /usr/local/lib/X11/fonts/URW
% xset fp rehash
```

Это сработает, но будет потеряно, когда сеанс работы с X будет закрыт, если эта команда не будет добавлена в начальный файл (~/.xinitrc в случае обычного сеанса через `startx` или

~/.xsession при входе через графический менеджер типа XDM). Третий способ заключается в использовании нового файла /usr/local/etc/fonts/local.conf: посмотрите раздел об антиалиасинге.

5.5.2. Шрифты TrueType®

В Xorg имеется встроенная поддержка шрифтов TrueType®. Имеются два модуля, которые могут обеспечить эту функциональность. В нашем примере используется модуль freetype, потому что он в большей степени похож на другие механизмы для работы с шрифтами. Для включения модуля freetype достаточно в раздел `"Module"` файла /etc/X11/xorg.conf добавить следующую строчку.

```
Load  "freetype"
```

Теперь создайте каталог для шрифтов TrueType® (к примеру, /usr/local/lib/X11/fonts/TrueType) и скопируйте все шрифты TrueType® в этот каталог. Имейте в виду, что напрямую использовать шрифты TrueType® с Macintosh® нельзя; для использования с X11 они должны быть в формате UNIX®/MS-DOS®/Windows®. После того, как файлы будут скопированы в этот каталог, воспользуйтесь утилитой ttmkfdir для создания файла fonts.dir, который укажет подсистеме вывода шрифтов X на местоположение этих новых файлов. `ttmkfdir` имеется в Коллекции Портов FreeBSD: x11-fonts/ttmkfdir.

```
# cd /usr/local/lib/X11/fonts/TrueType
# ttmkfdir -o fonts.dir
```

После этого добавьте каталог со шрифтами TrueType® к маршруту поиска шрифтов. Это делается точно также, как описано выше для шрифтов Type1, то есть выполните

```
% xset fp+ /usr/local/lib/X11/fonts/TrueType
% xset fp rehash
```

или добавьте строку `FontPath` в файл xorg.conf.

Это всё. Теперь getenv(3), Gimp, StarOffice™ и все остальные X-приложения должны увидеть установленные шрифты TrueType®. Очень маленькие (как текст веб-страницы на дисплее с высоким разрешением) и очень большие (в StarOffice™) шрифты будут теперь выглядеть гораздо лучше.

5.5.3. Антиалиасинг шрифтов

Антиалиасинг присутствует в X11 начиная с XFree86™, версии 4.0.2. Однако настройка шрифтов была довольно громоздка вплоть до появления XFree86™ 4.3.0. Начиная с версии XFree86™ 4.3.0, все шрифты, расположенные в каталогах /usr/local/lib/X11/fonts/ и ~/.fonts/, автоматически становятся доступными для применения антиалиасинга в приложениях, использующих Xft. Не все приложения могут использовать Xft, но во многих его поддержка присутствует. Примерами приложений, использующих Xft, является Qt версий 2.3 и более

поздних (это инструментальный пакет для оболочки KDE), GTK+ версий 2.0 и более поздних (это инструментальный пакет для оболочки GNOME), а также Mozilla версий 1.2 и более поздних.

Для применения к шрифтам антиалиасинга, а также для настройки параметров антиалиасинга, создайте (или отредактируйте, если он уже существует) файл /usr/local/etc/fonts/local.conf. Некоторые мощные возможности системы шрифтов Xft могут быть настроены при помощи этого файла; в этом разделе описаны лишь некоторые простые возможности. Для выяснения всех деталей, пожалуйста, обратитесь к fonts-conf(5).

Этот файл должен быть сформирован в формате XML. Обратите особое внимание на регистр символов, и удостоверьтесь, что все тэги корректно закрыты. Файл начинается обычным заголовком XML, за которым следуют DOCTYPE и тэг `<fontconfig>`:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

Как и говорилось ранее, все шрифты из каталога /usr/local/lib/X11/fonts/, а также ~/.fonts/ уже доступны для приложений, использующих Xft. Если вы хотите добавить каталог, отличный от этих двух, добавьте строчку, подобную следующей, в файл /usr/local/etc/fonts/local.conf:

```
<dir>/path/to/my/fonts</dir>
```

После добавления новых шрифтов, и особенно новых каталогов со шрифтами, вы должны выполнить следующую команду для перестроения кэшей шрифтов:

```
# fc-cache -f
```

Антиалиасинг делает границы несколько размытыми, что делает очень мелкий текст более читабельным и удаляет "лесенки" из текста большого размера, но может вызвать нечёткость при применении к тексту обычного размера. Для исключения размеров шрифтов, меньших 14, из антиалиасинга, добавьте такие строки:

```
<match target="font">
    <test name="size" compare="less">
        <double>14</double>
    </test>
    <edit name="antialias" mode="assign">
        <bool>false</bool>
    </edit>
</match>
<match target="font">
    <test name="pixelsize" compare="less" qual="any">
        <double>14</double>
    </test>
```

```
            <edit mode="assign" name="antialias">
                <bool>false</bool>
            </edit>
        </match>
```

Для некоторых моноширинных шрифтов антиалиасинг может также оказаться неприменимым при определении межсимвольного интервала. В частности, эта проблема возникает с KDE. Одним из возможных решений для этого является жесткое задание межсимвольного интервала в 100. Добавьте следующие строки:

```
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>fixed</string>
            </test>
            <edit name="family" mode="assign">
                <string>mono</string>
            </edit>
        </match>
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>console</string>
            </test>
            <edit name="family" mode="assign">
                <string>mono</string>
            </edit>
        </match>
```

(это создаст алиасы `"mono"` для других общеупотребительных имён шрифтов фиксированного размера), а затем добавьте:

```
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>mono</string>
            </test>
            <edit name="spacing" mode="assign">
                <int>100</int>
            </edit>
        </match>
```

С некоторыми шрифтами, такими, как Helvetica, при антиалиасинге могут возникнуть проблемы . Обычно это проявляется в виде шрифта, который наполовину вертикально обрезан. Хуже того, это может привести к сбоям таких приложений, как Mozilla. Во избежание этого следует добавить следующее в файл local.conf:

```
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>Helvetica</string>
```

```
        </test>
        <edit name="family" mode="assign">
            <string>sans-serif</string>
        </edit>
    </match>
```

После того, как вы закончите редактирование local.conf, удостоверьтесь, что файл завершен тэгом `</fontconfig>`. Если этого не сделать, ваши изменения будут проигнорированы.

Набор шрифтов по умолчанию, поставляемый с X11, не очень подходит, если включается антиалиасинг. Гораздо лучший набор шрифтов, используемых по умолчанию, можно найти в порте x11-fonts/bitstream-vera. Этот порт установит файл /usr/local/etc/fonts/local.conf, если такого ещё не существует. Если файл существует, то порт создаст файл /usr/local/etc/fonts/local.conf-vera. Перенесите содержимое этого файла в /usr/local/etc/fonts/local.conf, и шрифты Bitstream автоматически заменят используемые по умолчанию в X11 шрифты Serif, Sans Serif и Monospaced.

Наконец, пользователи могут добавлять собственные наборы посредством персональных файлов .fonts.conf. Для этого каждый пользователь должен просто создать файл ~/.fonts.conf. Этот файл также должен быть в формате XML.

И последнее замечание: при использовании дисплея LCD может понадобиться включение разбиения точек. При этом компоненты красного, зелёного и голубого цветов (разделяемые по горизонтали), рассматриваются как отдельные точки для улучшения разрешения экрана по горизонтали; результат может оказаться потрясающим. Для включения этого механизма добавьте такую строчку где-нибудь в файле local.conf:

```
    <match target="font">
        <test qual="all" name="rgba">
            <const>unknown</const>
        </test>
        <edit name="rgba" mode="assign">
            <const>rgb</const>
        </edit>
    </match>
```

В зависимости от типа дисплея, `rgb` может потребоваться заменить на `bgr`, `vrgb` или `vbgr`: пробуйте и смотрите, что работает лучше.

Антиалиасинг должен быть включен при следующем запуске X-сервера. Однако программы должны знать, как использовать его преимущества. В настоящее время инструментальный пакет Qt умеет ими пользоваться, так что вся оболочка KDE может использовать шрифты с антиалиасингом. GTK+ и GNOME также можно заставить использовать антиалиасинг посредством капплета "Font" (обратитесь к Шрифты с антиалиасингом и GNOME для выяснения всех подробностей). По умолчанию Mozilla версий 1.2 и выше будет автоматически использовать антиалиасинг. Для отмены использования антиалиасинга перестройте Mozilla с флагом `-DWITHOUT_XFT`.

5.6. Менеджеры экранов (Display Managers) X

5.6.1. Вступление

Менеджер Экранов X (XDM) это необязательный компонент X Window System, который используется для управления входом пользователей в систему. Это полезно в ряде ситуаций, например для минимальных "X Терминалов", десктопов, больших сетевых серверов экранов. Так как X Window System не зависит от сетей и протоколов, то существует множество различных конфигураций для X клиентов и серверов, запущенных на различных компьютерах, подключенных к сети. XDM предоставляет графический интерфейс для выбора сервера, к которому вы желаете подключится, и введения информации, авторизующей пользователя, например комбинации логина и пароля.

XDM можно рассматривать как аналог программы getty(8), предоставляющий такие же возможности для пользователей (смотрите Настройка для подробной информации). И это именно так, XDM производит вход в систему для подключенного пользователя и запускает управляющую сессию для пользователя (обычно это менеджер окон X). После этого XDM ожидает завершения приложения, означающее завершение пользователем работы и отключает управляющую сессию. Затем XDM может снова вывести приглашение к входу в систему и ожидать входа другого пользователя.

5.6.2. Использование XDM

Программой даемона XDM является /usr/local/bin/xdm. Эта программа может быть запущена от пользователя `root` в любой момент, и она начнёт управлять дисплеем X на локальной машине. Если XDM нужно запускать в фоновом режиме каждый раз при запуске компьютера, то наиболее правильный способ - это добавить новую запись в /etc/ttys. Для более подробной информации о формате и использовании этого файла смотрите Добавление строки в /etc/ttys. Вот строка, которую необходимо добавить в файл /etc/ttys для того, чтобы запустить даемон XDM на виртуальном терминале:

```
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off secure
```

По умолчанию эта запись отключена; для её включения нужно заменить пятое поле с `off` на `on` и перезапустить init(8), используя метод, описанный в Добавление строки в /etc/ttys. Первое поле это название терминала, которым будет управлять программа, `ttyv8`. Это означает, что XDM будет запущен на 9ом виртуальном терминале.

5.6.3. Конфигурирование XDM

Конфигурационные файлы XDM находятся в каталоге /usr/local/lib/X11/xdm. В нём размещаются несколько файлов, которые используются для изменения поведения и внешнего вида XDM. Обычно это следующие файлы:

Файл	Описание
Xaccess	Правила авторизации клиентов.

Файл	Описание
Xresources	Значения ресурсов X по умолчанию.
Xservers	Список локальных и удаленных экранов.
Xsession	Сценарий сессии по умолчанию.
Xsetup_*	Скрипт для запуска приложений до появления приглашения к входу в систему.
xdm-config	Глобальный конфигурационный файл для всех экранов запущенных на локальной машине
xdm-errors	Ошибки сгенерированные серверной программой.
xdm-pid	ID процесса запущенного XDM.

В этом каталоге также находятся несколько командных сценариев и программ, используемых для настройки рабочего стола (desktop) при запуске XDM. Назначение каждого из этих файлов будет вкратце описано. Точный синтаксис и информация по их использованию находятся в xdm(1).

В конфигурации по умолчанию выводится простое прямоугольное окно приглашения ко входу в систему с именем компьютера, написанным сверху большим шрифтом, и строками ввода "Login:" и "Password:" внизу. Это хорошая отправная точка для изменения внешнего вида экранов XDM.

5.6.3.1. Xaccess

Протокол, по которому происходит подключение дисплеев, управляемых XDM, называется X Display Manager Connection Protocol (XDMCP). Этот файл представляет собой набор правил для управления XDMCP соединениями с удалёнными машинами. Он игнорируется, пока стандартный файл xdm-config не содержит указаний по обслуживанию удалённых соединений.

5.6.3.2. Xresources

Это файл содержит установки по умолчанию для приложений, запущенных в экране выбора серверов и экране приглашения к входу в систему. В нем может быть изменён вид программы входа в систему. Формат этого файла идентичен файлу app-defaults, описанному в документации к X11.

5.6.3.3. Xservers

Это список удаленных экранов, которые XDM должен предоставить как варианты для входа в систему.

5.6.3.4. Xsession

Этот файл представляет из себя командный сценарий по умолчанию для пользователей,

вошедших в систему с использованием XDM. Обычно каждый пользователь имеет собственный сценарий входа в файле ~/.xsession, который используется вместо этого сценария.

5.6.3.5. Xsetup_*

Они запускаются автоматически перед тем, как показывается экран выбора сервера или экран входа в систему. Для каждого экрана (display) есть свой сценарий с именем Xsetup_, за которым следует локальный номер экрана (например, Xsetup_0). Обычно эти сценарии запускают одну или две программы в фоновом режиме, например `xconsole`.

5.6.3.6. xdm-config

Здесь содержатся настройки в формате app-defaults, которые применимы ко всем экранам данного компьютера.

5.6.3.7. xdm-errors

Здесь находится выдача X серверов, которые XDM пытается запустить. Если экран, который XDM пытается открыть, отключается по некоторым причинам, то это хорошее место для поиска сообщений об ошибках. Эти сообщения также записываются в пользовательский файл ~/.xsession-errors для каждого сеанса.

5.6.4. Использование сетевого сервера дисплеев

Для того, чтобы позволить другим клиентам подключаться к серверу дисплеев, необходимо отредактировать правила контроля доступа и включить обслуживание сетевых соединений. По умолчанию они выключены, что является хорошим решением с точки зрения обеспечения безопасности. Для того, чтобы позволить XDM принимать сетевые соединения, в первую очередь закомментируйте строку в файле xdm-config:

```
! БЕЗОПАСНОСТЬ: do not listen for XDMCP or Chooser requests
! Закомментируйте эти линии, если вы хотите управлять X терминалами с xdm
DisplayManager.requestPort:     0
```

и потом перезапустите XDM. Помните, что комментарии в файлах app-defaults начинаются с символа "!", а не как обычно, "\#". Может потребоваться более жёсткий контроль доступа - взгляните на примеры из Xaccess и обратитесь к странице справочника xdm(1) за дальнейшей информацией.

5.6.5. Замены для XDM

Существует несколько программ, заменяющих XDM. Одна из них, kdm (поставляемая вместе с KDE), описана далее в этой главе. В kdm имеется много визуальных и косметических улучшений, а также функциональность, позволяющая пользователям выбирать собственные оконные менеджеры во время входа в систему.

5.7. Графические оболочки

В этом разделе описываются различные графические оболочки, доступные в X для FreeBSD. Термин "графическая оболочка" может использоваться для чего угодно, от простого менеджера окон до полнофункционального набора приложений для рабочего стола, типа KDE или GNOME.

5.7.1. GNOME

5.7.1.1. О GNOME

GNOME является дружественной к пользователю графической оболочкой, позволяющей пользователям легко использовать и настраивать свои компьютеры. В GNOME имеется панель (для запуска приложений и отображения их состояния), рабочий стол (где могут быть размещены данные и приложения), набор стандартных инструментов и приложений для рабочего стола, а также набор соглашений, облегчающих совместную работу и согласованность приложений. Пользователи других операционных систем или оболочек при использовании такой мощной графической оболочки, какую обеспечивает GNOME, должны чувствовать себя в родной среде. Дополнительную информацию относительно GNOME во FreeBSD можно найти на сайте FreeBSD GNOME Project. Web сайт также содержит достаточно исчерпывающие FAQ'и, касающиеся установки, конфигурирования и управления GNOME.

5.7.1.2. Установка GNOME

Программу проще всего установить из пакета или коллекции портов:

Для установки пакета GNOME из сети, просто наберите:

```
# pkg_add -r gnome2
```

Для построения GNOME из исходных текстов используйте дерево портов:

```
# cd /usr/ports/x11/gnome2
# make install clean
```

После установки GNOME нужно указать X-серверу на запуск GNOME вместо стандартного оконного менеджера.

Самый простой путь запустить GNOME - это использовать GDM (GNOME Display Manager). GDM, который устанавливается, как часть GNOME (но отключен по умолчанию), может быть включён путём добавления `gdm_enable="YES"` в /etc/rc.conf. После перезагрузки, GNOME запустится автоматически после того, как вы зарегистрируйтесь в системе. Никакой дополнительной конфигурации не требуется.

GNOME может также быть запущен из командной строки с помощью конфигурирования файла .xinitrc. Если файл .xinitrc уже откорректирован, то просто замените строку, в которой

запускается используемый менеджер окон, на ту, что вызовет /usr/local/bin/gnome-session. Если в конфигурационном файле нет ничего особенного, то будет достаточно просто набрать:

```
% echo "/usr/local/bin/gnome-session" > ~/.xinitrc
```

Теперь наберите `startx`, и будет запущена графическая оболочка GNOME.

Если используется более старый менеджер дисплеев типа XDM, то это не сработает. Вместо этого создайте выполнимый файл .xsession с той же самой командой в нём. Для этого отредактируйте файл, заменив существующую команду запуска оконного менеджера на /usr/local/bin/gnome-session:

```
% echo "#!/bin/sh" > ~/.xsession
% echo "/usr/local/bin/gnome-session" >> ~/.xsession
% chmod +x ~/.xsession
```

Ещё одним вариантом является настройка менеджера дисплеев таким образом, чтобы он позволял выбирать оконный менеджер во время входа в систему; в разделе о KDE в подробностях описывается, как сделать это для kdm, менеджера дисплеев из KDE.

5.7.1.3. Шрифты с антиалиасингом и GNOME

X11 поддерживает антиалиасинг посредством своего расширения "RENDER". GTK+ 2.0 и более поздние версии (это инструментальный пакет, используемый GNOME) могут использовать такую функциональность. Настройка антиалиасинга описана в Антиалиасинг шрифтов. Таким образом, при наличии современного GNOME, возможно использование антиалиасинга. Просто перейдите в **Applications › Desktop Preferences › Font** и выберите либо **[Best shapes]**, **[Best contrast]**, либо **[Subpixel smoothing (LCDs)]**. Для приложений GTK+, которые не являются частью оболочки GNOME, задайте в качестве значения переменной окружения `GDK_USE_XFT``1` перед запуском программы.

5.7.2. KDE

5.7.2.1. О KDE

KDE является простой в использовании современной графической оболочкой. Вот лишь некоторые из преимуществ, которые даёт пользователю KDE:

- Прекрасный современный рабочий стол
- Рабочий стол, полностью прозрачный для работы в сети
- Интегрированная система помощи, обеспечивающая удобный и согласованный доступ к системе помощи по использованию рабочего стола KDE и его приложений
- Единообразный внешний вид и управление во всех приложениях KDE
- Стандартизированные меню и панели инструментов, комбинации клавиш, цветовые

схемы и так далее.

- Интернационализация: в KDE поддерживается более 40 языков
- Централизованное единообразное конфигурирование рабочего стола в диалоговом режиме
- Большое количество полезных приложений для KDE

Совместно с KDE поставляется веб-браузер под названием Konqueror, который является серьезным соперником другим браузерам для UNIX®-систем. Дополнительную информацию о KDE можно найти на веб-сайте KDE в FreeBSD. Для получения информации и информационных ресурсов, специфичных для KDE во FreeBSD, обратитесь к сайту команды FreeBSD-KDE team.

Имеется две версии KDE доступные на FreeBSD. Версия 3 была доступна очень долгое время и она является очень зрелой. Версия 4 - это следующее поколение, также доступное через Коллекцию Портов. Обе версии могут быть инсталлированы одновременно.

5.7.2.2. Установка KDE

Как и в случае с GNOME или любой другой графической оболочкой, программное обеспечение можно легко установить из пакета или из Коллекции Портов:

Для установки пакета KDE3 из сети, просто наберите:

```
# pkg_add -r kde
```

Для установки пакета KDE4 из сети, просто наберите:

```
# pkg_add -r kde4
```

pkg_add(1) автоматически загрузит самую последнюю версию приложения.

Для построения KDE3 из исходных текстов, воспользуйтесь деревом портов:

```
# cd /usr/ports/x11/kde3
# make install clean
```

Для построения KDE4 из исходных текстов, воспользуйтесь деревом портов:

```
# cd /usr/ports/x11/kde4
# make install clean
```

После установки KDE нужно указать X-серверу на запуск этого приложения вместо оконного менеджера, используемого по умолчанию. Это достигается редактированием файла .xinitrc:

Для KDE3:

```
% echo "exec startkde" > ~/.xinitrc
```

Для KDE4:

```
% echo "exec /usr/local/kde4/bin/startkde" > ~/.xinitrc
```

Теперь при вызове X Window System по команде `startx` в качестве оболочки будет использоваться KDE.

При использовании менеджера дисплеев типа XDM настройка несколько отличается. Вместо этого нужно отредактировать файл .xsession. Указания для kdm описаны далее в этой главе.

5.7.3. Более подробно о KDE

Теперь, когда KDE установлена в системе, можно узнать много нового из её справочных страниц или просто указанием и щелканьем по различным меню. Пользователи Windows® или Mac® будут чувствовать себя как дома.

Лучшим справочником по KDE является онлайновая документация. KDE поставляется с собственным веб-браузером, который называется Konqueror, десятками полезных приложений и подробной документацией. В оставшейся части этого раздела обсуждаются технические вопросы, трудные для понимания при случайном исследовании.

5.7.3.1. Менеджер дисплеев KDE

Администратору многопользовательской системы может потребоваться графический экран для входа пользователей в систему. Вы можете использовать XDM, как это описано ранее. Однако в KDE имеется альтернативный менеджер kdm, который был разработан более привлекательным и с большим количеством настраиваемых опций для входа в систему. В частности, пользователи могут легко выбирать (посредством меню), какую оболочку (KDE, GNOME или что-то ещё) запускать после входа в систему.

Для того, чтобы разрешить запуск kdm, измените в файле /etc/ttys строку, относящуюся к консоли `ttyv8`:

Для KDE3:

```
ttyv8 "/usr/local/bin/kdm -nodaemon" xterm on secure
```

Для KDE4:

```
ttyv8 "/usr/local/kde4/bin/kdm -nodaemon" xterm on secure
```

5.7.4. XFce

5.7.4.1. О XFce

XFce является графической оболочкой, построенной на основе инструментального пакета GTK+, используемого в GNOME, но она гораздо легче и предназначена для тех, кому нужен простой, эффективно работающий рабочий стол, который легко использовать и настраивать. Визуально он выглядит очень похоже на CDE, который есть в коммерческих UNIX®-системах. Вот некоторые из достоинств XFce:

- Простой, лёгкий в обращении рабочий стол
- Полностью настраиваемый при помощи мыши, с интерфейсом drag and drop и так далее
- Главная панель похожа на CDE, с меню, апплетами и возможностями по быстрому запуску приложений
- Интегрированный оконный менеджер, менеджер файлов, управление звуком, модуль совместимости с GNOME и прочее
- Возможность использования тем (так как использует GTK+)
- Быстрый, легкий и эффективный: идеален для устаревших/слабых машин или для машин с ограниченной памятью

Дополнительную информацию о XFce можно найти на сайте XFce.

5.7.4.2. Установка XFce

Для XFce имеется (на момент написания этого текста) бинарный пакет. Для его установки просто наберите:

```
# pkg_add -r xfce4
```

Либо, в случае построения из исходных текстов, используйте Коллекцию Портов:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

Теперь укажите X-серверу на запуск XFce при следующем запуске X. Просто наберите:

```
% echo "/usr/local/bin/startxfce4" > ~/.xinitrc
```

При следующем запуске X в качестве рабочего стола будет использоваться XFce. Как сказано выше, если используется менеджер дисплеев, такой, как XDM, создайте файл .xsession так, как это описано в разделе о GNOME, но с командой /usr/local/bin/startxfce4, либо настройте менеджер дисплеев так, чтобы он разрешил выбор рабочего стола во время входа в систему, как это описано в разделе о kdm.

Часть II: Общие задачи

Теперь, когда основы были пройдены, в данной части Руководства FreeBSD будут обсуждаться некоторые часто используемые возможности FreeBSD. В этих главах:

- Введение в популярные и полезные графические приложения: браузеры, бизнес приложения, программы просмотра документов и т.д.
- Представлены множество мультимедийных программ, доступных в FreeBSD.
- Описан процесс создания собственного ядра FreeBSD для включения дополнительных функций системы.
- Система печати разобрана в деталях, как для непосредственно подключенных принтеров, так и для принтеров, подключенных через сеть.
- Показано, как запускать приложения Linux в системе FreeBSD.

Перед прочтением некоторых из этих глав необходимо ознакомиться с предварительной информацией, что указано в кратком обзоре в начале каждой главы.

Глава 6. Приложения для настольного компьютера

6.1. Краткий обзор

FreeBSD может работать с широким кругом приложений для настольного компьютера (десктопа), таких как браузеры и текстовые процессоры. Большинство из них доступны в качестве пакетов или могут быть автоматически собраны из коллекции портов. Многим новым пользователям хотелось бы видеть эти приложения на своем компьютере. В этой главе показано как без усилий установить некоторые популярные приложения для настольного компьютера из пакетов или из коллекции портов.

Обратите внимание, что при установке программ из портов они компилируются из исходных текстов. Это может занять очень много времени, в зависимости от того, что вы собираете, и от скорости процессора вашего компьютера (компьютеров). Большинство программ, имеющихся в коллекции портов, могут быть установлены из прекомпилированных пакетов, если сборка из исходных текстов занимает недопустимо много времени.

Поскольку FreeBSD обеспечивает двоичную совместимость с Linux, многие приложения, первоначально разработанные для Linux, доступны и на вашем компьютере. Настоятельно рекомендуется прочитать Двоичная совместимость с Linux перед установкой любого из приложений Linux. Названия многих портов, использующих двоичную совместимость с Linux, начинаются с "linux-". Помните это при поиске отдельного порта, например с помощью whereis(1). Далее в статье подразумевается, что вы включили бинарную совместимость с Linux перед установкой какого-либо приложения Linux.

Вот несколько категорий, о которых пойдет речь в этой главе:

- Браузеры (такие как Mozilla, Opera, Firefox, Konqueror)
- Бизнес приложения (такие как KOffice, AbiWord, GIMP, OpenOffice.org)
- Программы просмотра документов (такие как Acrobat Reader®, gv, Xpdf, GQview)
- Финансовые программы (такие как GnuCash, Gnumeric, Abacus)

Перед прочтением этой главы вам потребуется:

- Узнать, как устанавливать дополнительные программы сторонних производителей (Установка приложений. порты и пакеты).
- Узнать, как устанавливать программы Linux (Двоичная совместимость с Linux).

Чтобы получить дополнительную информацию о настройке мультимедиа среды, прочтите Мультимедиа. Если вам нужна электронная почта, обратитесь к Электронная почта.

6.2. Браузеры

FreeBSD поставляется без предустановленного браузера. Вместо этого, в категории www коллекции портов содержится множество готовых к установке браузеров. Если у вас нет времени компилировать все (в некоторых случаях это может занять очень много времени), многие из них доступны в виде пакетов.

В KDE и GNOME уже есть HTML браузеры. Обратитесь к Графические оболочки за подробной информацией об установке этих полноценных десктопов.

Если вы ищете облегченный браузер, попробуйте www/dillo, www/links, или www/w3m из коллекции портов.

Этот раздел рассказывает о следующих приложениях:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
Mozilla	большая	тяжелая	Gtk+
Opera	малая	легкая	Доступны версии для FreeBSD и Linux. Для Linux версии необходимо наличие Linux Binary Compatibility и linux-openmotif
Firefox	средняя	тяжелая	Gtk+
Konqueror	средняя	тяжелая	Библиотеки KDE

6.2.1. Mozilla

Mozilla это наиболее современный и стабильный браузер; он полностью портирован на FreeBSD. Его достоинство в высокой совместимости со стандартами HTML. В нем есть почтовая и новостная программы. В нем даже найдется редактор HTML, если вам потребуется самостоятельно написать несколько веб-страничек. Пользователи getenv(3) найдут общие черты с Communicator, поскольку оба браузера имеют одну основу.

На медленных компьютерах с частотой CPU меньше 233MHz или с памятью меньше 64MB, Mozilla требует слишком много ресурсов, чтобы быть удобной в использовании. Вместо нее вы можете обратить внимание на браузер Opera, описанный ниже в этой главе.

Если вы не можете или не хотите компилировать Mozilla по какой-то причине, команда FreeBSD GNOME уже сделала это для вас. Просто установите пакет из сети с помощью:

```
# pkg_add -r mozilla
```

Если пакет недоступен, но у вас достаточно времени и места на диске, вы можете скачать

исходные тексты для Mozilla, скомпилировать их и установить в вашу систему. Это делается так:

```
# cd /usr/ports/www/mozilla
# make install clean
```

Порт Mozilla проверяет правильность установки путем запуска регистрации chrome с привилегиями пользователя `root`. Если вы хотите загрузить некоторые дополнения, например курсоры мыши, потребуется запустить Mozilla под `root` для их правильной установки.

После завершения установки Mozilla, больше не требуется работать под `root`. Вы можете запустить Mozilla в качестве браузера, набрав:

```
% mozilla
```

Вы можете также запустить непосредственно программу чтения почты и новостей, как показано ниже:

```
% mozilla -mail
```

6.2.2. Mozilla и Java™

Установка Mozilla проста, но к сожалению, установка Mozilla с поддержкой дополнений, таких как Java™ и Macromedia® Flash™ отнимает и время и место на диске.

Первое, что нужно сделать - загрузить файлы, которые будут использоваться с Mozilla. Зайдите с помощью имеющегося веб браузера на http://www.sun.com/software/java2/download.html и создайте учетную запись на этом веб-сайте. Сохраните имя пользователя и пароль, они могут понадобиться в будущем. Загрузите копию файлов jdk-1_5_0-bin-scsl.zip (JDK 5.0 SCSL Binaries) и jdk-1_5_0-src-scsl.zip (JDK 5.0 SCSL Source) и поместите их в каталог /usr/ports/distfiles, поскольку порт не может загрузить их автоматически в связи с лицензионными ограничениями. Загрузите с этого же сайта "java environment", http://javashoplm.sun.com/ECom/docs/Welcome.jsp?StoreId=22&PartDetailId=j2sdk-1.4.2_08-oth-JPR&SiteId=JSC&TransactionId=noreg, файл j2sdk-1_4_2_08-linux-i586.bin. Как и предыдущий, этот файл должен находиться в каталоге /usr/ports/distfiles/. Загрузите копию "java patchkit" с http://www.eyesbeyond.com/freebsddom/java/jdk15.html и поместите ее в /usr/ports/distfiles/. Наконец, установите порт java/jdk15 при помощи стандартной команды `make install clean`.

Запустите Mozilla и выберите пункт About Plug-ins в меню **Help**. В списке установленных плагинов должен присутствовать плагин Java™.

6.2.3. Mozilla и Macromedia® Flash™ plugin

Плагина Macromedia® Flash™ для FreeBSD не существует. Тем не менее, есть решение

(обертка, wrapper) для запуска плагина для Linux. Это решение также поддерживает плагины для Adobe® Acrobat®, RealPlayer и других.

Установите порт www/linuxpluginwrapper. Он требует для работы достаточно большого порта emulators/linux_base. Следуя инструкциям, исправьте файл /etc/libmap.conf! Примеры конфигураций вы можете найти в каталоге /usr/local/shared/examples/linuxpluginwrapper/.

Установите порт www/mozilla, если Mozilla еще не установлена.

Теперь просто запустите Mozilla:

```
% mozilla &
```

И войдите в пункт About Plug-ins меню **Help**. Должен появиться список со всеми доступными плагинами.

Плагин linuxpluginwrapper работает только на архитектуре i386™.

6.2.4. Opera

Opera это очень быстрый, полноценный и совместимый со стандартами браузер. Он также идет в комплекте с почтовой и новостной программами, клиентом IRC, модулем чтения RSS/Atom и другими. Несмотря на все это, браузер Opera относительно легок и быстр. Он поставляется в двух вариантах: "родная" для FreeBSD версия и версия, запускаемая в режиме эмуляции Linux.

Для работы в сети с помощью FreeBSD версии Opera установите пакет:

```
# pkg_add -r opera
```

На некоторых серверах FTP нет всех пакетов, но те же результаты можно получить с помощью коллекции портов, набрав:

```
# cd /usr/ports/www/opera
# make install clean
```

Для установки Linux версии Opera, замените `opera` на `linux-opera` в примере выше. Версия для Linux полезна в ситуации, когда требуются плагины, доступные только для Linux, такие как Adobe Acrobat Reader®. Во всех других отношениях версии для FreeBSD и Linux являются функционально идентичными.

6.2.5. Firefox

Firefox это браузер следующего поколения, основанный на коде Mozilla. Mozilla это полный набор приложений, таких как браузер, почтовый клиент, чат клиент и многое другое. Firefox это всего лишь браузер, что делает его меньше и быстрее.

Установите пакет, выполнив:

```
# pkg_add -r firefox
```

Вы можете также использовать коллекцию портов, если предпочитаете сборку из исходных текстов:

```
# cd /usr/ports/www/firefox
# make install clean
```

6.2.6. Konqueror

Konqueror это часть KDE, но может быть использован и отдельно от KDE, путем установки x11/kdebase3. Konqueror это гораздо больше чем просто браузер, это также менеджер файлов и программа просмотра мультимедиа.

Konqueror поставляется с набором плагинов, доступных из misc/konq-plugins.

Konqueror поддерживает также Flash™, документация How To для него доступна по адресу http://freebsd.kde.org/howto.php.

6.3. Бизнес приложения

В начале работы новые пользователи зачастую стремятся найти хороший офисный пакет или удобный текстовый процессор. Хотя некоторые десктопы, такие как KDE, поставляются с готовым офисным пакетом, приложения по умолчанию не существует. В FreeBSD есть все необходимое, кроме графической среды.

Этот раздел описывает следующие приложения:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
KOffice	малая	тяжелая	KDE
AbiWord	малая	легкая	Gtk+ или GNOME
The Gimp	малая	тяжелая	Gtk+
OpenOffice.org	большая	очень тяжелая	JDK™ 1.4, Mozilla

6.3.1. KOffice

Сообщество KDE предоставляет графическую среду с офисным пакетом, который может быть использован вне KDE. Он включает четыре стандартных компонента, встречающиеся и в других офисных пакетах. Текстовый процессор KWord, программа электронных таблиц KSpread, KPresenter для создания презентаций и программа векторной графики Kontour.

Перед установкой последней версии KOffice, убедитесь в наличии свежей версии KDE.

Для установки KOffice из пакета, выполните следующую команду:

```
# pkg_add -r koffice
```

Если пакет недоступен, используйте коллекцию портов. Например, для установки KOffice для KDE3, выполните:

```
# cd /usr/ports/editors/koffice-kde3
# make install clean
```

6.3.2. AbiWord

AbiWord это свободно распространяемый текстовый процессор, по внешнему виду и поведению очень похожий на Microsoft® Word. Он подходит для набора документов, писем, отчетов, напоминаний и так далее. Он очень быстр, содержит много новшеств и очень удобен в использовании.

AbiWord может импортировать и экспортировать множество файловых форматов, включая патентованный Microsoft .doc.

AbiWord доступен в виде пакета. Вы можете установить его так:

```
# pkg_add -r abiword
```

Если пакет недоступен, он может быть собран из коллекции портов, которая должна быть свежей. Это можно сделать командой:

```
# cd /usr/ports/editors/abiword
# make install clean
```

6.3.3. GIMP

Для создания и редактирования изображений есть продвинутая программа GIMP. Она может быть использована как простая программа рисования и как программа обработки фотографий. Поддерживается большое количество плагинов и предоставлен интерфейс для скриптов. GIMP может читать и записывать файлы многих форматов. Есть интерфейс со сканерами и планшетами.

Вы можете установить пакет, выполнив эту команду:

```
# pkg_add -r gimp
```

Если на вашем сервере FTP нет этого пакета, вы можете использовать коллекцию портов. Категория graphics коллекции портов содержит также раздел Руководство Gimp. Здесь

показано, как его установить:

```
# cd /usr/ports/graphics/gimp
# make install clean
# cd /usr/ports/graphics/gimp-manual-pdf
# make install clean
```

Категория graphics коллекции портов содержит версию GIMP для разработчиков в graphics/gimp-devel. HTML версия Руководства Gimp находятся в graphics/gimp-manual-html.

6.3.4. OpenOffice.org

OpenOffice.org включает все обязательные компоненты полноценного офисного пакета: текстовый процессор, программу электронных таблиц, программу управления презентациями и программу векторной графики. Интерфейс пользователя очень похож на другие офисные пакеты, возможен импорт и экспорт различных популярных файловых форматов. Приложение доступно в вариантах для множества разных языков, включая интерфейсы, проверку орфографии и словари.

Текстовый процессор OpenOffice.org использует чистый XML формат файлов для увеличения переносимости и гибкости. Программа для работы с текстовыми таблицами предоставляет макроязык и может работать с внешними базами данных. OpenOffice.org уже стабильна и существует в версиях для Windows®, Solaris™, Linux, FreeBSD, и Mac OS® X. Дополнительную информацию об OpenOffice.org можно найти на веб сайте OpenOffice.org. Получить специфичную для FreeBSD информацию и загрузить пакеты можно с веб сайта команды портирования OpenOffice на FreeBSD (FreeBSD OpenOffice.org Porting Team).

Для установки OpenOffice.org, выполните:

```
# pkg_add -r openoffice
```

Эта операция должна работать для любого релиза (-RELEASE) FreeBSD. Если вы используете иные версии (-STABLE, -CURRENT), нужный пакет может быть загружен с сайта группы поддержки OpenOffice.org и затем установлен при помощи pkg_add(1). На сайте вы найдете как последний стабильный релиз, так и текущую версию, находящуюся в разработке.

После установки пакета просто наберите следующую команду для запуска OpenOffice.org:

```
% openoffice.org
```

Во время первого запуска, вам будут заданы несколько вопросов и в вашей домашней директории будет создан каталог .openoffice.org2.

Если пакеты OpenOffice.org недоступны, можно выбрать компиляцию порта. Однако, вы должны помнить, что это потребует много места на диске и компиляция будет довольно долгой.

```
# cd /usr/ports/editors/openoffice.org-2.0
# make install clean
```

Если вы хотите собрать локализованную версию, то вместо предыдущей командной строки используйте следующее:

```
# make LOCALIZED_LANG=your_language install clean
```

Вам следует изменить *your_language* на корректный ISO код языка. Список поддерживаемых языковых кодов доступен в файле files/Makefile.localized, расположенный в директории порта.

После того, как это было сделано OpenOffice.org может быть запущен командой:

```
% openoffice.org
```

6.4. Программы просмотра документов

Некоторые новые форматы документов приобрели большую популярность. Стандартные программы для их просмотра могут отсутствовать в базовой системе. В этом разделе мы увидим, как их установить.

В разделе говорится о следующих приложениях:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
Acrobat Reader®	малая	легкая	Linux Binary Compatibility
gv	малая	легкая	Xaw3d
Xpdf	малая	легкая	FreeType
GQview	малая	легкая	Gtk+ или GNOME

6.4.1. Acrobat Reader®

Сейчас многие документы распространяются в формате PDF, аббревиатура для "Portable Document Format". Одна из рекомендованных программ для просмотра этого типа документов, это Acrobat Reader®, выпущенный Adobe для Linux. Поскольку FreeBSD может запускать исполняемые файлы Linux, он доступен также и для FreeBSD.

Для установки Acrobat Reader® 7 из Коллекции портов выполните:

```
# cd /usr/ports/print/acroread7
# make install clean
```

Пакет acroread7 недоступен из-за лицензионных ограничений.

6.4.2. gv

gv это программа просмотра PostScript® и PDF. Она разработана на основе ghostview, но выглядит лучше благодаря библиотеке Xaw3d. Она быстра, а ее интерфейс несложен. У gv есть множество функций, таких как выбор ориентации, размера бумаги, масштаба и сглаживание. Почти любая операция может быть выполнена как с клавиатуры, так и мышью.

Для установки gv из пакета, выполните:

```
# pkg_add -r gv
```

Если вы не можете получить пакет, используйте коллекцию портов:

```
# cd /usr/ports/print/gv
# make install clean
```

6.4.3. Xpdf

Если вам нужна небольшая программа просмотра PDF под FreeBSD, Xpdf это легкая и эффективная программа. Она требует очень небольшого количества ресурсов и очень стабильна. Используются стандартные шрифты X, Motif® или другие пакеты для X не нужны.

Для установки пакета Xpdf, выполните эту команду:

```
# pkg_add -r xpdf
```

Если пакет недоступен, или вы предпочитаете коллекцию портов, выполните:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

После завершения установки вы можете запустить Xpdf и использовать правую кнопку мыши для активации меню.

6.4.4. GQview

GQview это программа для работы с изображениями. Вы можете просмотреть файл одним кликом, запустить внешний редактор, получить миниатюры и многое другое. Еще в нем есть слайд-шоу и несколько основных файловых операций. Вы можете управлять коллекциями изображений и легко находить дубликаты. В GQview изображения можно просматривать во весь экран, его можно адаптировать к разным языкам.

Если вы хотите установить пакет GQview, выполните:

```
# pkg_add -r gqview
```

Если пакет недоступен, или вы предпочитаете использовать коллекцию портов, выполните:

```
# cd /usr/ports/graphics/gqview
# make install clean
```

6.5. Финансовые программы

Если по каким-то причинам вам нужно управлять своими финансами на десктопе FreeBSD, есть несколько мощных и простых в использовании приложений. Некоторые из них совместимы с широко распространенными форматами файлов, такими как документы Quicken или Excel.

В этом разделе говорится о следующих приложениях:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
GnuCash	малая	тяжелая	GNOME
Gnumeric	малая	тяжелая	GNOME
Abacus	малая	легкая	Tcl/Tk

6.5.1. GnuCash

GnuCash это часть проекта GNOME, который стремится предоставить дружественные к пользователю приложения с широким набором функций. С GnuCash вы можете отслеживать доходы и расходы, банковские счета или акции. Интуитивный интерфейс программы не мешает ей оставаться очень профессиональной.

GnuCash предоставляет интеллектуальный журнал записей, иерархическую систему учетных записей, множество клавиатурных сокращений и метод автозавершения. Он может разбивать одну транзакцию на несколько частей, детализируя ее. GnuCash может импортировать и присоединять файлы Quicken QIF. Он также работает с основными международными форматами дат и валютами.

Для установки GnuCash в вашу систему, выполните:

```
# pkg_add -r gnucash
```

Если пакет недоступен, вы можете использовать коллекцию портов:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

6.5.2. Gnumeric

Gnumeric это электронная таблица, часть графической среды GNOME. Она использует удобное автоматическое "угадывание" ввода пользователя в зависимости от формата ячейки и систему автозаполнения для множества последовательностей. Она может импортировать файлы нескольких популярных форматов, таких как Excel, Lotus 1-2-3, или Quattro Pro. Gnumeric работает с диаграммами через math/guppi. В ней множество встроенных функций, можно использовать обычные форматы ячеек: число, валюта, дата, время и многие другие.

Для установки Gnumeric из пакета, введите:

```
# pkg_add -r gnumeric
```

Если пакет недоступен, вы можете использовать коллекцию портов:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

6.5.3. Abacus

Abacus это небольшая и простая в использовании программа электронных таблиц. В ней много встроенных функций из нескольких областей, таких как статистика, финансы и математика. Она может импортировать и экспортировать файлы Excel. Abacus также может печатать PostScript®.

Для установки Abacus из пакета, выполните:

```
# pkg_add -r abacus
```

Если пакет недоступен, вы можете использовать коллекцию портов, выполнив:

```
# cd /usr/ports/deskutils/abacus
# make install clean
```

6.6. Итоги

Хотя FreeBSD популярна в основном среди провайдеров из-за стабильности и высокой производительности, на сегодняшний день она вполне готова к использованию в качестве десктопа. С несколькими тысячами приложений, доступных в виде пакетов или портов, вы можете создать прекрасный десктоп, отвечающий всем вашим потребностям.

После первой установки десктопа, вы можете попробовать сделать шаг вперед с misc/instant-workstation. Этот "мета-порт" позволяет вам собрать типичный набор портов для рабочей станции. Вы можете настроить его, редактируя /usr/ports/misc/instant-workstation/Makefile. Следуйте синтаксису существующего файла при добавлении и удалении портов, соберите порт как обычно. В конечном итоге, вы можете создать большой пакет, соответствующий вашему собственному десктопу, и установить его на другие рабочие станции!

Вот небольшой обзор всех графических приложений, о которых говорилось в этой главе:

Имя приложения	Имя пакета	Имя порта
Mozilla	`mozilla`	www/mozilla
Opera	`linux-opera`	www/linux-opera
Firefox	`firefox`	www/firefox
KOffice	`koffice-kde3`	editors/koffice-kde3
AbiWord	`abiword`	editors/abiword
The GIMP	`gimp`	graphics/gimp1
OpenOffice.org	`openoffice`	editors/openoffice
Acrobat Reader®	`acroread`	print/acroread7
gv	`gv`	print/gv
Xpdf	`xpdf`	graphics/xpdf
GQview	`gqview`	graphics/gqview
GnuCash	`gnucash`	finance/gnucash
Gnumeric	`gnumeric`	math/gnumeric
Abacus	`abacus`	deskutils/abacus

Глава 7. Мультимедиа

7.1. Краткий обзор

FreeBSD поддерживает большое количество различных звуковых карт, что позволяет вам насладится высококачественным звуком. Это также дает возможность записывать и воспроизводить звуковые файлы в формате MPEG Audio Layer 3 (MP3), WAV, Ogg Vorbis, а также во множестве других форматов. Коллекция Портов FreeBSD также содержит ряд приложений, позволяющих редактировать записанные звуковые файлы, добавлять звуковые эффекты, управлять подключенными MIDI устройствами.

FreeBSD может поддерживать воспроизведение видеофайлов и DVD. Количество приложений, позволяющих кодировать, преобразовывать и воспроизводить различные форматы видео, существенно меньше количества приложений для работы со звуком. Например, на время написания этого документа в Коллекции Портов FreeBSD не существовало хорошего приложения для преобразования видео, которое могло бы быть использовано для преобразований между разными форматами, как, например, audio/sox. Впрочем, ситуация в этой области меняется быстро.

Эта глава описывает необходимые шаги для настройки вашей звуковой карты. Настройка и установка X11 (X Window System) уже охватывает вопросы, связанные с аппаратными установками вашей видеокарты, хотя могут быть возможности дополнительной настройки для улучшения воспроизведения.

После прочтения этой главы вы будете знать:

- Как настроить систему так, чтобы звуковая карта была опознана.
- Методы проверки работы звуковой карты при помощи тестовых приложений.
- Как исправить проблемы, возникающие при работе со звуковыми картами.
- Как прослушать и создать MP3 и другие форматы.
- Как X сервер поддерживает видео.
- Некоторые проигрыватели и кодировщики видео, которые показывают хорошие результаты.
- Как воспроизвести DVD, .mpg и .avi файлы.
- Как скопировать информацию с CD и DVD в файлы.
- Как настроить ТВ тюнер.
- Как настроить сканер.

Перед чтением этой главы вам потребуется:

- Узнать, как конфигурировать и устанавливать новое ядро (Настройка ядра FreeBSD).

Попытка смонтировать аудио CD при помощи команды mount(8) как минимум, сообщит об ошибке и, как максимум, может привести к *панике*

> *ядра*. Эти носители имеют специальные форматы, которые отличны от обычной файловой системы ISO.

7.2. Настройка звуковой карты

7.2.1. Настройка системы

Перед тем как начать, определите модель вашей карты, процессор, который она использует, и интерфейс карты: PCI или ISA. FreeBSD поддерживает множество разных PCI и ISA карт. Сверьтесь со списком поддерживаемых аудио устройств в Информации об оборудовании, чтобы проверить, поддерживается ли ваша карта. Этот документ также содержит информацию о том, какой драйвер поддерживает вашу карту.

Для того, чтобы использовать звуковую карту, вы должный загрузить соответствующий драйвер устройства. Этого можно достигнуть двумя путями. Простейший способ - это просто загрузить соответствующий вашей карте модуль ядра используя kldload(8), что можно сделать или из командной строки:

```
# kldload snd_emu10k1
```

или добавлением соответствующей строки к файлу /boot/loader.conf:

```
snd_emu10k1_load="YES"
```

Эти примеры приведены для звуковой карты Creative SoundBlaster® Live!. Другие имеющиеся модули драйверов звуковых карты приведены в /boot/defaults/loader.conf Если вы не уверены, какой драйвер использовать, попробуйте загрузить snd_driver:

```
# kldload snd_driver
```

Это мета-драйвер, загружающий сразу все наиболее распространенные драйверы сразу. Это повышает скорость поиска правильного драйвера. Возможна также загрузка всех звуковых драйверов через /boot/loader.conf.

Для того чтобы узнать, какой именно драйвер требуется для вашей звуковой карты, вы можете проверить содержимое файла /dev/sndstat при помощи команды `cat /dev/sndstat`.

Другой способ заключается в добавлении статического драйвера в ядро. В разделе ниже дана более подробная информация о том, что вам нужно сделать для добавления поддержки оборудования. Более подробно о конфигурация ядра описана в Настройка ядра FreeBSD.

7.2.1.1. Настройка собственного ядра с поддержкой звука

Первое, что необходимо сделать, это добавить в ядро общий звуковой драйвер sound(4).

Добавьте в файл конфигурации ядра следующую строку:

```
device sound
```

Затем необходимо добавить поддержку имеющейся звуковой карты. Следовательно, нужно знать какой драйвер поддерживает карту. Для этого сверьтесь со списком поддерживаемых устройств из Информации об оборудовании. Например, звуковая карта Creative SoundBlaster® Live! поддерживается драйвером snd_emu10k1(4). Для добавления поддержки этой карты, используйте:

```
device snd_emu10k1
```

Прочтите страницу справочника драйвера, чтобы узнать, какой синтаксис использовать. Информация, относящаяся к синтаксису включения звуковых драйверов в файл конфигурации ядра, может быть также найдена в файле /usr/src/sys/conf/NOTES.

Не-PnP ISA карты могут потребовать включения в ядро информации о настройках звуковой карты (IRQ, I/O port, и т.д.). Эта информация добавляется редактированием файла /boot/device.hints. Во время загрузки системы loader(8) прочтет этот файл и передаст настройки ядру. Например, старая ISA не-PnP карта Creative SoundBlaster® 16 использует драйвер snd_sbc(4) совместно с snd_sb16(4). Для этой карты к файлу настройки ядра необходимо добавить следующие строки:

```
device snd_sbc
device snd_sb16
```

и со следующей информацией в /boot/device.hints:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

В данном случае, карта использует порт ввода-вывода `0x220` и IRQ `5`.

Синтаксис, используемый в файле /boot/device.hints, описан в справочной странице драйвера.

Установки, приведенные выше, используются по умолчанию. В некоторых случаях вам может потребоваться изменить IRQ или другие настройки в соответствии с настройками карты. За более подробной информацией обратитесь к странице справочника snd_sbc(4).

7.2.2. Тестирование звуковой карты

После перезагрузки модифицированного ядра, или после загрузки необходимого модуля, звуковая карта должна появиться в буфере системных сообщений (dmesg(8)) примерно так:

```
pcm0: <Intel ICH3 (82801CA)> port 0xdc80-0xdcbf,0xd800-0xd8ff irq 5 at device 31.5 on
pci0
pcm0: [GIANT-LOCKED]
pcm0: <Cirrus Logic CS4205 AC97 Codec>
```

Статус звуковой карты может быт проверен через файл /dev/sndstat:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm)
Installed devices:
pcm0: <Intel ICH3 (82801CA)> at io 0xd800, 0xdc80 irq 5 bufsz 16384
kld snd_ich (1p/2r/0v channels duplex default)
```

Вывод этой команды для вашей системы может отличаться. Если устройства pcm не появились, вернитесь назад и проверьте выполненные действия. Проверьте файл настройки ядра еще раз и убедитесь, что выбрано подходящее устройство. Часто встречающиеся проблемы приведены в Часто встречающиеся проблемы.

Если всё пройдет удачно, звуковая карта заработает. Если CD-ROM или DVD-ROM привод правильно подключён к звуковой карте, вы можете вставить CD в привод и воспроизвести его при помощи cdcontrol(1).

```
% cdcontrol -f /dev/acd0 play 1
```

Различные приложения, например audio/workman могут предоставить более дружественный пользователю интерфейс. Вы можете также установить приложения для прослушивания звуковых файлов MP3, как например audio/mpg123. Быстрым способом тестирования звуковой карты является отправка данных в файл /dev/dsp, как показано здесь:

```
% cat filename > /dev/dsp
```

где *filename* может быть любым файлом. Результатом выполнения этой команды станет шум, который означает, что звуковая карта на самом деле работает.

Уровни громкости звука могут быть изменены командой mixer(8). Более подробная информация находится на странице справочной системы mixer(8).

7.2.2.1. Часто встречающиеся проблемы

Ошибка	Решение
`unsupported subdevice XX`	Одно или более устройств не были правильно созданы. Повторите приведенные выше шаги.
`sb_dspwr(XX) timed out`	Порт ввода-вывода указан неправильно.
`bad irq XX`	IRQ установлен неправильно. Убедитесь, что настройки в системе и на карте одинаковы.
`xxx: gus pcm not attached, out of memory`	Для использования устройства недостаточно памяти.
`xxx: can't open /dev/dsp!`	Проверьте с помощью `fstat | grep dsp`, не занято ли устройство другим приложением. Создать проблемы могут esound и поддержка звука в KDE.

7.2.3. Использование нескольких источников звука

Достаточно часто встречается необходимость иметь несколько источников звука, которые должны воспроизводить одновременно, например когда esound или artsd не поддерживают совместное использование звукового устройства с некоторым приложением.

FreeBSD позволяет делать это при помощи *виртуальных звуковых каналов*, которые могут быть настроены с помощью sysctl(8). Виртуальные каналы позволяют вам мультиплексировать каналы воспроизведения звуковой карты, смешивая звук в ядре.

Для установки количества виртуальных каналов вы можете использовать две переменные sysctl, которые, если вы пользователь `root`, могут быть установлены таким образом:

```
# sysctl hw.snd.pcm0.vchans=4
# sysctl hw.snd.maxautovchans=4
```

В этом примере выделяются четыре виртуальных канала, чего вполне достаточно для повседневного использования. `hw.snd.pcm0.vchans` это количество виртуальных каналов устройства pcm0, оно может быть установлено сразу же, как только устройство было подключено. `hw.snd.maxautovchans` это количество виртуальных каналов, которые выделяются новому аудио устройству, когда оно подключается при помощи kldload(8). Так как модуль pcm может быть загружен независимо от аппаратных драйверов, `hw.snd.maxautovchans` может указывать количество виртуальных каналов для любых устройств, которые будут подключены позже.

Количество виртуальных каналов не может быть изменено, если аудио устройство занято. Вам потребуется предварительно закрыть все программы, работающие со звуком, такие как медиа-проигрыватели или звуковые даемоны.

Если вы не используете devfs(5), необходимо будет указать приложениям /dev/dsp0.*x*, где *x* это число от 0 до 3, если `hw.snd.pcm0.vchans` установлено в значение 4. Для системы, использующей devfs(5), вышеуказанные настройки будут сделаны автоматически прозрачно для пользователя.

7.2.4. Установка значений по умолчанию для каналов mixer

Значения по умолчанию для различных каналов mixer жестко прописаны в исходных текстах драйвера pcm(4). Существует множество различных приложений и даемонов, которые позволяют устанавливать значения для mixer, которые они запоминают и выставляют каждый раз при запуске, но это не совсем правильное решение, нам нужны значения по умолчанию на уровне драйвера. Они могут быть установлены путем указания в /boot/device.hints. Например:

```
hint.pcm.0.vol="100"
```

Установит значение для канала volume в значение по умолчанию 100, как только будет загружен модуль pcm(4).

7.3. Звук MP3

MP3 (MPEG Layer 3 Audio) достигает качества звука, близкого к CD, и нет причин не воспользоваться им на вашей рабочей станции.

7.3.1. Проигрыватели MP3

На данный момент наиболее популярным MP3-проигрывателем для X11 является XMMS (X Multimedia System). Скины приложения WinAMP могут быть использованы для XMMS так как графический интерфейс пользователя практически идентичен интерфейсу программы WinAMP от Nullsoft. XMMS поддерживает также собственные расширения.

XMMS может быть установлен из порта или пакета multimedia/xmms.

Интерфейс XMMS интуитивно понятен и включает в себя список песен, графический эквалайзер и многое другое. Те, кто знаком с WinAMP, найдут XMMS очень простым в использовании.

Порт audio/mpg123 является альтернативой, это MP3-проигрыватель для командной строки.

mpg123 может быть запущен с указанием звукового устройства и файла MP3 в командной строке как показано ниже:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3.
Version 0.59r (1999/Jun/15). Written and copyrights by Michael Hipp.
Uses code from various people. See 'README' for more!
THIS SOFTWARE COMES WITH ABSOLUTELY NO WARRANTY! USE AT YOUR OWN RISK!
```

```
Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

`/dev/dsp1.0` должно быть заменено соответствующим устройством dsp для вашей системы.

7.3.2. Копирование аудио дорожек с CD

Перед тем как преобразовывать CD или дорожку CD в MP3, аудио данные на CD должны быть скопированы на жёсткий диск. Это можно сделать путём копирования данных CDDA (CD Digital Audio) в файл WAV.

Утилита `cdda2wav`, которая является частью пакета sysutils/cdrtools, может быть использована для копирования аудио информации с CD, а также различной связанной информации.

Когда музыкальный CD находится в приводе, следующая команда может быть выполнена под `root` для того, чтобы скопировать весь CD в отдельные (один на каждую дорожку) WAV файлы:

```
# cdda2wav -D 0,1,0 -B
```

cdda2wav поддерживает ATAPI (IDE) приводы CDROM. Для копирования с IDE привода, укажите имя устройства вместо номеров SCSI. Например, для того, чтобы скопировать 7-ую аудио дорожку с IDE-привода:

```
# cdda2wav -D /dev/acd0 -t 7
```

Параметр `-D 0,1,0` указывает устройство SCSI 0,1,0, соответственно результату работы `cdrecord -scanbus`.

Для того, чтобы копировать отдельные дорожки, используйте параметр `-t` как показано ниже:

```
# cdda2wav -D 0,1,0 -t 7
```

Этот пример показывает как скопировать 7-ю дорожку музыкального CD. Для того чтобы скопировать набор дорожек, например, с первой по седьмую, укажите диапазон:

```
# cdda2wav -D 0,1,0 -t 1+7
```

Утилита dd(1) также может быть использована для копирования аудио дорожек на приводах ATAPI, для того, чтобы узнать больше об этом, прочитайте Копирование аудио CD.

7.3.3. Создание файлов MP3

На сегодняшний день наилучшим выбором программы для создания mp3 является lame.

Lame находится в дереве портов в подкаталоге audio/lame.

Используя скопированные файлы WAV, следующая команда преобразует audio01.wav в audio01.mp3:

```
# lame -h -b 128 \
--tt "Foo Song Title" \
--ta "FooBar Artist" \
--tl "FooBar Album" \
--ty "2001" \
--tc "Ripped and encoded by Foo" \
--tg "Genre" \
audio01.wav audio01.mp3
```

Частота 128 килобит является стандартом "де факто" для MP3. Многие, однако, используют более высокие частоты для получения лучшего качества, 160 или 192 килобита. Чем выше частота, тем больше дискового пространства будет занимать получаемый MP3, но качество будет выше. Параметр `-h` включает режим "лучшее качество, но меньше скорость". Параметры, начинающиеся с `--t` указывают теги ID3, которые обычно содержат информацию о песне, включаемую в файл MP3. О дополнительных настройках преобразования можно узнать, прочитав страницу руководства lame.

7.3.4. Декодирование MP3

Для того, чтобы записать музыкальный CD из файлов MP3, они должны быть преобразованы в несжатый формат WAV. Как XMMS, так и mpg123 поддерживают вывод MP3 в распакованный формат файлов.

Запись на диск в XMMS:

1. Запустите XMMS:
2. Нажмите правой кнопкой мыши в главном окне XMMS для того, чтобы показать меню.
3. Выберите `Preferences` (либо `Свойства`, если у вас локализованная версия XMMS) в `Options`.
4. Измените расширение вывода на "Disk Writer Plugin" (или "Расширение записи на диск", если у вас локализованная версия XMMS).
5. Нажмите `Configure` (или "Настройка", если у вас локализованная версия XMMS).
6. Введите (или выберите при помощи обзора) каталог, в который следует сохранять распакованные файлы.
7. Загрузите файл MP3 в XMMS как вы это делаете обычно. Установите громкость на 100% и отключите эквалайзер.
8. Нажмите `Воспроизвести` - XMMS будет выглядеть так же как и при обычном воспроизведении MP3, но самой музыки слышно не будет. На самом деле MP3 воспроизводится в файл.

9. Убедитесь, что вы установили расширение вывода таким, как оно было до этого, для того, чтобы снова слушать MP3.

Запись в stdout в mpg123:

1. Запустите `mpg123 -s audio01.mp3 > audio01.pcm`

XMMS записывает файл в формате WAV, в то время как mpg123 преобразовывает MP3 в простые аудио данные PCM. Оба формата могут быть использованы cdrecord для создания музыкальных CD. Для использования burncd(8) вам потребуются простые аудио данные PCM. Если же вы будете использовать файлы в формате WAV, то заметите небольшой щелчок в начале каждой аудио дорожки, этот щелчок - заголовок файла в формате WAV. Вы очень просто можете избавиться от него путём удаления заголовка WAV при помощи утилиты SoX (она может быть установлена из порта audio/sox или соответствующего пакета:

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Прочтите Запись и использование оптических носителей (CD) для того, чтобы узнать больше о записи CD в FreeBSD.

7.4. Воспроизведение видео

Воспроизведение видео является очень новой и быстро развивающейся областью применения. Будьте терпеливы. Не всё будет работать так беспроблемно, как это было со звуком.

Прежде, чем вы начнёте, определите модель видеокарты и чипсет, который она использует. Хотя Xorg и XFree86™ поддерживают множество различных видеокарт, только их малая часть показывает хорошую скорость воспроизведения видео. Для того, чтобы получить список расширений, поддерживаемых X-сервером, который используется вашей видеокартой, используйте команду xdpyinfo(1) во время работы X11.

Неплохо также иметь небольшой файл MPEG, который бы использовался как тестовый файл для проверки различных проигрывателей и настроек. Так как некоторые проигрыватели DVD будут искать носитель DVD как /dev/dvd по умолчанию или быть жёстко настроены на него, возможно будет полезно сделать символические ссылки на правильные устройства:

```
# ln -sf /dev/acd0 /dev/dvd
# ln -sf /dev/acd0 /dev/rdvd
```

Обратите внимание, природа devfs(5) такова, что такие созданные вручную ссылки не сохраняются при перезагрузке системы. Для автоматического создания символических ссылок при каждой загрузке системы добавьте в /etc/devfs.conf следующие строки:

```
link acd0 dvd
link acd0 rdvd
```

Кроме того, декодирование DVD требует доступа к специальным функциям DVD-ROM, поэтому должен быть доступ на запись для устройств DVD.

Для того, чтобы улучшить работу разделяемой памяти X11, рекомендуется увеличить значения некоторых переменных sysctl(8):

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

7.4.1. Определение возможностей видео

Существует несколько возможных путей отображения видео под X11. Что именно будет действительно работать, во многом зависит от аппаратного обеспечения. Каждый из описанных методов будет работать с различным качеством на разном аппаратном обеспечении. Во-вторых, воспроизведение видео в X11, это тема, которой уделяется достаточно много внимания последнее время, и с каждой новой версией Xorg или XFree86™ могут наблюдаться значительные улучшения.

Список наиболее часто используемых видеоинтерфейсов:

1. X11: обычный вывод X11 с использованием разделяемой памяти.
2. XVideo: расширение интерфейса X11, которое поддерживает видео в любом объекте X11.
3. SDL: the Simple Directmedia Layer.
4. DGA: the Direct Graphics Access - прямой доступ для графики.
5. SVGAlib: низкоуровневый доступ к графике на консоли.

7.4.1.1. XVideo

Xorg и XFree86™ 4.X включают в себя расширение, называющееся *XVideo* (также известное как Xvideo, Xv и xv), которое позволяет отображать видео прямо на объектах при помощи специального ускорения. Это расширение предоставляет очень хорошее качество воспроизведения даже на low-end машинах.

Для того чтобы проверить, работает ли это расширение, используйте команду `xvinfo`:

```
% xvinfo
```

XVideo поддерживается вашей видеокартой, если результат выглядит приблизительно так:

```
X-Video Extension version 2.2
screen #0
```

```
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
              client settable attribute
              client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
              client settable attribute
              client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
              client settable attribute
              client gettable attribute (current value is 0)
    maximum XvImage size: 1024 x 1024
    Number of image formats: 7
      id: 0x32595559 (YUY2)
        guid: 59555932-0000-0010-8000-00aa00389b71
        bits per pixel: 16
        number of planes: 1
        type: YUV (packed)
      id: 0x32315659 (YV12)
        guid: 59563132-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x30323449 (I420)
        guid: 49343230-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x36315652 (RV16)
        guid: 52563135-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x3e0, 0x7c00
      id: 0x35315652 (RV15)
        guid: 52563136-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
```

```
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x7e0, 0xf800
      id: 0x31313259 (Y211)
        guid: 59323131-0000-0010-8000-00aa00389b71
        bits per pixel: 6
        number of planes: 3
        type: YUV (packed)
      id: 0x0
        guid: 00000000-0000-0000-0000-000000000000
        bits per pixel: 0
        number of planes: 0
        type: RGB (packed)
        depth: 1
        red, green, blue masks: 0x0, 0x0, 0x0
```

Следует заметить, что перечисленные форматы (YUV2, YUV12 и т.п.) не присутствуют в каждой реализации XVideo и их отсутствие может быть помехой для некоторых проигрывателей.

Если результат выглядит так:

```
X-Video Extension version 2.2
screen #0
no adaptors present
```

то, возможно, XVideo не поддерживается для вашей видеокарты.

Если XVideo не поддерживается вашей видеокартой, то это всего лишь означает, что будет сложнее получить приемлемые для воспроизведения видео вычислительные мощности. В зависимости от вашей видеокарты и процессора, возможно, вы сможете получить удовлетворительный результат. Возможно, вы должны будете прочитать о путях улучшения производительности в Дальнейшее чтение.

7.4.1.2. Simple Directmedia Layer

SDL был задуман как уровень абстракции для разработки кросплатформенных приложений под Microsoft® Windows®, BeOS и UNIX®, позволяя им эффективно использовать звук и графику. SDL предоставляет низкоуровневые абстракции для аппаратного обеспечения, и может быть более эффективным чем интерфейс X11.

SDL есть в Коллекции портов FreeBSD: devel/sdl12.

7.4.1.3. Прямой доступ для графики (DGA)

DGA это расширение X11, которое позволяет программам напрямую изменять кадровый буфер (framebuffer) без участия X-сервера. Поскольку DGA основывается на низкоуровневом доступе к памяти, программы, которые используют его должны исполняться от пользователя `root`.

Расширение DGA может быть протестировано при помощи dga(1). Когда `dga` запущена, она изменяет цвета на экране при каждом нажатии клавиш. Для того, чтобы выйти из неё, используйте `q`.

7.4.2. Порты и пакеты для работы с видео

Этот раздел обсуждает программное обеспечение для работы с видео из Коллекции Портов FreeBSD. Воспроизведение видео является очень активной сферой разработок программного обеспечения и возможности различных приложений могут несколько отличаться от описанных здесь.

Во-первых, важно помнить, что многие приложения для работы с видео, которые работают на FreeBSD, были разработаны как приложения Linux. Многие из этих приложений все еще бета-качества. Вот некоторые проблемы, которые могут встретиться в работе видео пакетов на FreeBSD:

1. Приложение не может воспроизвести файл, который создало другое приложение.
2. Приложение не может воспроизвести файл, который создало само.
3. Одно и то же приложение на разных машинах, скомпилированное на каждой машине специально для неё, воспроизводит один и тот же файл различно.
4. Кажущийся тривиальным фильтр, например фильтр изменения размеров изображения, приводит к очень плохим "артефактам" из-за неправильной функции изменения размера.
5. Приложение часто не работает (оставляет core-файл).
6. Документация не устанавливается вместе с портом и может быть найдена лишь на сайте или в каталоге порта work.

Многие из этих приложений могут также проявлять "линуксизмы". Так, это могут быть некоторые проблемы, связанные со способом реализации некоторых стандартных библиотек в дистрибутивах Linux, или некоторыми дополнительными возможностями ядра Linux, которые авторы приложений посчитали существующими везде. Эти проблемы не всегда могут быть обнаружены людьми, поддерживающими порт (порты), вследствие чего могут возникнуть проблемы, сходные с нижеперечисленными:

1. Использование /proc/cpuinfo для того, чтобы определить характеристики процессора.
2. Неправильное использование нитей (threads), которое может привести к зависанию программы при завершении вместо нормального выхода.
3. Программного обеспечения, которое обычно используется совместно с данным приложением, ещё нет в Коллекции Портов FreeBSD.

Таким образом, разработчики этих приложений должны сотрудничать с людьми, поддерживающими порты, для того, чтобы минимизировать количество обходных путей, необходимых для портирования.

7.4.2.1. MPlayer

MPlayer это недавно разработанный и быстро развивающийся проигрыватель видео. Задачами команды разработчиков MPlayer являются скорость и гибкость при работе на Linux и других Unix-системах. Проект был начат, когда его основатель стал сыт по горло плохой производительностью и качеством проигрывателей того времени. Некоторые могут сказать, что графический интерфейс был принесён в жертву рационализированному дизайну. Однако, как только вы привыкнете к опциям командной строки MPlayer и его управлению с клавиатуры, всё будет хорошо.

7.4.2.1.1. Компиляция MPlayer

MPlayer находится в multimedia/mplayer. MPlayer производит различные тесты аппаратного обеспечения во время процесса компиляции, в результате чего полученные исполняемые модули не могут быть перенесены с одной системы на другую. Поэтому важно собирать его из портов, а не использовать бинарный пакет. Также, при сборке вы можете указать различные установки при помощи параметров командной строки `make`, как описывается в Makefile в начале сборки:

```
# cd /usr/ports/multimedia/mplayer
# make
N - O - T - E

Take a careful look into the Makefile in order
to learn how to tune mplayer towards you personal preferences!
For example,
make WITH_GTK1
builds MPlayer with GTK1-GUI support.
If you want to use the GUI, you can either install
/usr/ports/multimedia/mplayer-skins
or download official skin collections from
http://www.mplayerhq.hu/homepage/dload.html
```

Параметры порта по умолчанию должны подходить большинству пользователей. Однако, если вам необходим кодек XviD, необходимо указать в командной строке параметр `WITH_XVID`. Устройство DVD по умолчанию также может быть указано в командной строке параметром `WITH_DVD_DEVICE`, по умолчанию используется /dev/acd0.

На время написания данного документа порт MPlayer'а собирает и устанавливает свою документацию в формате HTML и два исполняемых файла, `mplayer` и `mencoder`, который является утилитой для перекодировки видео.

Документация к MPlayer очень информативна. Если читатель найдет информацию этой главы о аппаратном обеспечении для поддержки видео и интерфейсах недостаточной, то документация MPlayer будет очень хорошим дополнением. Обязательно уделите время чтению документации MPlayer, если вам нужна информация о поддержке видео под UNIX®.

7.4.2.1.2. Использование MPlayer

Каждый пользователь MPlayer должен создать подкаталог .mplayer в своем домашнем каталоге. Для того, чтобы его создать, выполните следующие действия:

```
% cd /usr/ports/multimedia/mplayer
% make install-user
```

Параметры для `mplayer` перечислены в страничке руководства mplayer. За более подробной информацией вы можете обратиться к документации в формате HTML. В этом разделе мы опишем несколько самых распространённых случаев использования mplayer.

Для того, чтобы воспроизвести файл, например testfile.avi через один из многих видеоинтерфейсов, используйте параметр `-vo`:

```
% mplayer -vo xv testfile.avi
```

```
% mplayer -vo sdl testfile.avi
```

```
% mplayer -vo x11 testfile.avi
```

```
# mplayer -vo dga testfile.avi
```

```
# mplayer -vo 'sdl:dga' testfile.avi
```

Стоит испробовать все варианты интерфейсов, так как их производительность зависит от множества факторов и будет заметно меняться в зависимости от аппаратного обеспечения.

Для того, чтобы воспроизвести DVD, замените testfile.avi на `dvd://N -dvd-device DEVICE`, где *N* является номером дорожки, с которой следует начать воспроизведение и DEVICE файл устройства привода DVD. Например, для того, чтобы воспроизвести дорожку 3 с /dev/dvd:

```
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```

Устройство DVD по умолчанию может быть определено во время сборки порта MPlayer параметром `WITH_DVD_DEVICE`. По умолчанию, это устройство /dev/acd0. Дополнительную информацию можно найти в Makefile порта.

Для того, чтобы остановить, приостановить или продолжить воспроизведение, воспользуйтесь привязкой клавиш, информация о которой может быть получена посредством запуска `mplayer -h`, либо на страничке документации.

Дополнительные, достаточно важные параметры воспроизведения: `-fs -zoom`, которые включают полноэкранный режим и `-framedrop`, который улучшает производительность на медленных системах.

Для того, чтобы командная строка запуска mplayer не становилась слишком большой, пользователь может создать файл .mplayer/config и установить параметры по умолчанию там:

```
vo=xv
fs=yes
zoom=yes
```

Также `mplayer` может быть использован для копирования дорожек DVD в .vob файлы. Для того, чтобы скопировать вторую дорожку DVD необходимо выполнить следующую команду:

```
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-device /dev/dvd
```

Полученный файл, out.vob, будет представлять собой MPEG, с которым можно производить различные действия при помощи программ, которые будут описаны далее в этом разделе.

7.4.2.1.3. mencoder

Перед использованием `mencoder`, было бы неплохо ближе ознакомится с его параметрами, используя документацию в формате HTML. Также существует страничка справочника mplayer, но она не очень полезна без HTML документации. Существует бесчисленное множество способов улучшения качества, снижения битрейта и изменения формата; и некоторые из этих приёмов могут влиять на производительность. Ниже приведено несколько примеров использования `mencoder`. Во-первых, простое копирование:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Неправильная комбинация параметров командной строки может привести к появлению файлов, которые невозможно будет воспроизвести даже `mplayer`. Поэтому, если вы хотите скопировать изображение в файл, лучше использовать только параметр `mplayer -dumpfile`.

Для того, чтобы преобразовать input.avi в MPEG4 со звуком в формате MPEG Audio Layer 3 (MP3) (требуется audio/lame):

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
     -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

Эта команда создаст файл, воспроизводимый `mplayer` и `xine`.

input.avi может быть заменён на `dvd://1 -dvd-device /dev/dvd` и `mplayer`, запущенный от пользователя `root`, будет преобразовывать дорожку DVD напрямую. Так как первый раз, скорее всего, вы будете недовольны полученными результатами, всё же рекомендуется

копировать дорожку в файл и работать затем с файлом.

7.4.2.2. Проигрыватель xine

xine - это большой проект, в задачи которого входит не только создание решения для видео все-в-одном, но и создание базовой библиотеки с возможностью расширения путем использования плагинов. Поставляется он как в виде порта, так и в виде пакета, multimedia/xine.

xine все еще несовершенен, но все-таки это хорошее начало. На практике xine требует либо быстрого процессора с быстрой видеокартой или поддержки расширения XVideo. Графический интерфейс можно использовать, но он все еще немного неуклюж.

На время написания этого документа в поставке xine не существовало модуля ввода, который бы мог воспроизводить DVD, закодированные по алгоритму CSS. Существуют сборки, в которых есть такой модуль, но ни одна из них не входит в Коллекцию Портов FreeBSD.

По сравнению с MPlayer, xine является более дружелюбным к пользователю, но, в то же время, скрывает более тонкие настройки и управление от пользователя. Также xine лучше работает на XVideo интерфейсах.

По умолчанию, xine запускается с графическим интерфейсом. Для открытия файлов используются меню.

```
% xine
```

В качестве альтернативы можно использовать его для запуска файла непосредственно, без GUI, следующей командой:

```
% xine -g -p mymovie.avi
```

7.4.2.3. Утилиты transcode

Приложение transcode не является проигрывателем. Это набор инструментов для преобразования видео и звуковых файлов. При помощи transcode можно объединять видеофайлы, исправлять поврежденные файлы, использовать инструменты командной строки для работы с потоками ввода/вывода stdin/stdout.

Большое количество опций может быть указано во время сборки порта multimedia/transcode. Для сборки transcode мы рекомендуем использовать следующую командную строку:

```
# make WITH_OPTIMIZED_CFLAGS=yes WITH_LIBA52=yes WITH_LAME=yes WITH_OGG=yes \
WITH_MJPEG=yes -DWITH_XVID=yes
```

Предложенных установок должно быть достаточно для большинства пользователей.

Для иллюстрации возможностей `transcode` приводится пример, показывающий как сконвертировать файл DivX формата в PAL MPEG-1 файл (PAL VCD):

```
% transcode -i input.avi -V --export_prof vcd-pal -o output_vcd
% mplex -f 1 -o output_vcd.mpg output_vcd.m1v output_vcd.mpa
```

Итоговый MPEG файл output_vcd.mpg может быть проигран с помощью MPlayer. Вы можете даже записать файл на CD-R носитель для создания Video CD. В этом случае, вам нужно будет установить и использовать программы multimedia/vcdimager и sysutils/cdrdao.

Существует страничка справочника для `transcode`, но вы также должны проконсультироваться с transcode wiki для получения более детальной информации и примеров.

7.4.3. Дальнейшее чтение

Различные пакеты видео программ для FreeBSD интенсивно разрабатываются. Очень возможно, что в ближайшем будущем многие обсуждаемые здесь проблемы разрешатся. Это займет время, и те, кто желает получить максимум от аудио/видео возможностей FreeBSD, должны будут собирать необходимые знания из нескольких списков часто задаваемых вопросов и обучающих статей, а также использовать различные приложения. Этот раздел существует для того, чтобы читатель мог получить указания на несколько источников дополнительной информации.

Документация MPlayer очень содержательна в техническом плане. Возможно, эти документы должны использоваться любым человеком, желающим получить высокий уровень знаний о видео на UNIX® системах. Список рассылки MPlayer враждебен для любого, кто не потрудился прочитать документацию, так что, если у вас есть желание сообщать о найденных ошибках, прочитайте вначале документацию.

xine HOWTO содержит главу об улучшении производительности, которая применима к любому проигрывателю.

Наконец, существует несколько многообещающих приложений, которые читатель может испробовать:

- Avifile, для которого также существует порт multimedia/avifile.
- Ogle, для которого также существует порт multimedia/ogle.
- Xtheater
- multimedia/dvdauthor, пакет с открытыми текстами для распространения DVD контента.

7.5. Настройка ТВ тюнеров

7.5.1. Введение

ТВ тюнеры предназначены для просмотра широковещательного или кабельного телевидения на компьютере. Большинство тюнеров поддерживают композитный видео

вход RCA или S-video, а некоторые из них поставляются с FM радио тюнером.

FreeBSD поддерживает PCI ТВ тюнеры, использующие Brooktree Bt848/849/878/879 или Conexant CN-878/Fusion 878a Video Capture Chip через драйвер bktr(4). Вы должны также убедиться, что тюнер поддерживается; обратитесь к странице справочника bktr(4) за списком поддерживаемых тюнеров.

7.5.2. Добавление драйвера

Для использования карты потребуется загрузить драйвер bktr(4), что можно сделать, добавив в /boot/loader.conf следующую строку:

```
bktr_load="YES"
```

В качестве альтернативы, вы можете статически скомпилировать ядро с поддержкой ТВ тюнера; добавьте следующие строки в файл конфигурации ядра:

```
device   bktr
device  iicbus
device  iicbb
device  smbus
```

Эти дополнительные драйвера устройств необходимы, поскольку компоненты карты соединены через шину I2C. Затем соберите и установите новое ядро.

Как только поддержка тюнера будет добавлена в систему, перегрузите компьютер. Во время загрузки TV карта должна отобразить примерно такие строки:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

Конечно, эти сообщения будут различаться на разном оборудовании. Тем не менее, проверьте, что тюнер определяется правильно; возможна перезапись параметров, определенных ядром, с помощью sysctl(8) MIB и параметров в файле настройки ядра. Например, если вы хотите указать, что это Philips SECAM тюнер, добавьте следующую строку к файлу настройки ядра:

```
options OVERRIDE_TUNER=6
```

или прямо задайте переменную sysctl(8):

```
# sysctl hw.bt848.tuner=6
```

Обратитесь к странице bktr(4) и файлу /usr/src/sys/conf/NOTES за более детальной информацией о доступных параметрах.

7.5.3. Полезные приложения

Для использования ТВ тюнера вам потребуется установить одно из следующих приложений:

- multimedia/fxtv предоставляет возможности ТВ-в-окне и захвата изображений/аудио/видео.
- multimedia/xawtv это также приложение для ТВ тюнера, с теми же, что и у fxtv возможностями.
- misc/alevt раскодирует и отображает видеотекст/телетекст.
- audio/xmradio, приложение для использования с FM радио тюнером, поставляемым с некоторыми ТВ тюнерами.
- audio/wmtune, это удобное приложение для радио тюнеров.

В коллекции портов FreeBSD можно найти и другие приложения.

7.5.4. Решение проблем

Если вы столкнулись с какой-либо проблемой, связанной с ТВ тюнером, проверьте в первую очередь поддержку микросхемы захвата видео и тюнера драйвером bktr(4), а также правильность установки параметров. За дальнейшей поддержкой и с вопросами о ТВ тюнере вы можете обращаться в Список рассылки, посвящённый поддержке средств мультимедиа под FreeBSD и использовать его архивы.

7.6. Сканеры

7.6.1. Введение

В FreeBSD доступ к сканерам обеспечивается программой SANE (Scanner Access Now Easy), обеспечивающей универсальный интерфейс (API) и доступной в коллекции портов FreeBSD. Для общения со сканерами SANE использует некоторые драйвера устройств FreeBSD.

FreeBSD поддерживает сканеры с интерфейсом как SCSI, так и USB. Убедитесь, что ваш сканер поддерживается SANE перед тем, как приступить к конфигурации. Для SANE существует список поддерживаемых устройств где находится информация о поддержке сканера и статусе этой поддержки. Кроме того, страница справочника uscanner(4) также перечисляет поддерживаемые устройства.

7.6.2. Конфигурация ядра

Как уже отмечалось, поддерживаются как SCSI, так и USB сканеры. В зависимости от интерфейса вашего сканера требуется поддержка разных драйверов устройств.

7.6.2.1. USB

Стандартное ядро GENERIC включает в себя драйвера, необходимые для поддержки USB сканеров. Если вы компилируете собственное ядро, убедитесь, что в его конфигурации присутствуют строки

```
device usb
device uhci
device ohci
device uscanner
```

В зависимости от чипсета USB, встроенного в вашу материнскую плату, потребуется лишь один из драйверов `device uhci` или `device ohci`, однако, наличие обеих строк в конфигурации ядра никому не повредит.

Если вы не хотите перестраивать ядро, и при этом ваше ядро не является стандартным (GENERIC), вы можете загрузить модуль драйвера поддержки сканеров uscanner(4) при помощи команды kldload(8):

```
# kldload uscanner
```

Для автоматической загрузки модуля при старте системы добавьте в файл /boot/loader.conf строку

```
uscanner_load="YES"
```

После перезагрузки с новым ядром или загрузки модуля подключите ваш USB сканер. В буфере системных сообщений (dmesg(8)) должна появиться строка об обнаружении сканера:

```
uscanner0: EPSON EPSON Scanner, rev 1.10/3.02, addr 2
```

В данном случае сканер будет использовать устройство /dev/uscanner0.

7.6.2.2. SCSI

Если ваш сканер имеет интерфейс SCSI, важно знать, к какому контроллеру он подключен. В зависимости от контроллера потребуются различные драйвера в файле конфигурации ядра. Стандартное ядро GENERIC поддерживает большинство распространенных SCSI-контроллеров. Внимательно прочитайте файл NOTES и добавьте необходимые строки в файл конфигурации вашего ядра. Помимо строки для драйвера адаптера, вам потребуются

следующие строки:

```
device scbus
device pass
```

После установки и загрузки нового ядра, в буфере системных сообщений должны появиться строки о вашем сканере, например:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

Если сканер не был включен в момент загрузки, его можно принудительно опознать, выполнив сканирование SCSI шины при помощи команды camcontrol(8):

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

После этого сканер должен появиться в списке устройств:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 (pass2,cd0)
```

Более подробная информация о устройствах SCSI доступна на страницах справочника scsi(4) и camcontrol(8).

7.6.3. Конфигурация SANE

Система SANE состоит из двух частей: аппаратной поддержки (backend, graphics/sane-backends) и программной поддержки (frontend, graphics/sane-frontends). Первая часть обеспечивает собственно доступ к сканеру. Список поддерживаемых устройствSANE содержит информацию о необходимом вам аппаратном модуле. Вторая часть обеспечивает графический интерфейс для сканирования (xscanimage).

В первую очередь следует установить порт или пакет graphics/sane-backends, после чего при помощи команды `sane-find-scanner` проверить поддержку сканера системой SANE:

```
# sane-find-scanner -q
```

```
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

В выводе должны присутствовать интерфейс сканера и имя используемого устройства. Производитель и модель сканера могут отсутствовать: это нормально.

Некоторым USB сканерам может потребоваться загрузка прошивки. Подробности смотрите в страницах справочника драйвера сканера, sane-find-scanner(1) и linprocfs(7).

Теперь необходимо убедиться, что сканер опознан программой графического интерфейса. В состав системы SANE входит утилита sane(1), позволяющая работать со сканером из командной строки. Опция -L используется для показа информации о сканере:

```
# scanimage -L
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
```

Отсутствие сообщений или сообщение об отсутствии устройств означает, что утилита sane(1) не смогла идентифицировать сканер. В этом случае вам потребуется отредактировать файл конфигурации аппаратного модуля и указать устройство, используемое сканером. Все файлы настройки находятся в каталоге /usr/local/etc/sane.d/. Такие проблемы присущи некоторым моделям USB сканеров.

Например, в случае USB сканера, описанного в USB, утилита sane-find-scanner выдаст следующую информацию:

```
# sane-find-scanner -q
found USB scanner (UNKNOWN vendor and product) at device /dev/uscanner0
```

Сканер обнаружен корректно, он использует интерфейс USB и доступен через устройство /dev/uscanner0. Теперь попробуем идентифицировать его:

```
# scanimage -L

No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Поскольку сканер не идентифицирован, нам потребуется изменить файл конфигурации /usr/local/etc/sane.d/epson.conf. В нашем примере использован сканер EPSON Perfection® 1650, так что мы знаем, что будет использоваться драйвер epson. Не забудьте прочитать комментарии в файле конфигурации. Требуемые изменения весьма просты: закомментируйте все строки, описывающие интерфейсы, не соответствующие интерфейсу вашего сканера (в нашем случае, все строки, начинающиеся со scsi: наш сканер использует интерфейс USB), и добавьте в конец файла строку, содержащую интерфейс и имя

использованного устройства. Мы добавим строку

```
usb /dev/uscanner0
```

Пожалуйста, прочтите комментарии в файле конфигурации, а также страницы справочника для более полной информации. Теперь мы можем проверить, что наш сканер опознан:

```
# scanimage -L
device `epson:/dev/uscanner0' is a Epson GT-8200 flatbed scanner
```

Наш USB сканер опознан. Не столь важно, что имя и номер модели не совпадают, главное, что используются правильные имя устройства и драйвер: `epson:/dev/uscanner0`.

После того как команда `scanimage -L` опознала сканер, конфигурация завершена. Все готово к сканированию.

Хотя утилита sane(1) позволяет производить сканирование из командной строки, как правило, для сканирования предпочтительнее использовать графический интерфейс. Для этого в состав SANE входит простая, но эффективная утилита xscanimage (graphics/sane-frontends).

Другой популярной программой графического интерфейса к сканеру является Xsane (graphics/xsane). Эта программа поддерживает такие расширенные возможности, как разные режимы сканирования (фотокопия, факс и т.п.), цветокоррекцию, потоковое сканирование и другие. Оба приложения пригодны для использования в качестве плагинов сканирования для GIMP.

7.6.4. Доступ к сканеру для других пользователей

Все описанные операции выполнялись нами с привилегиями суперпользователя (`root`). Вам может потребоваться дать доступ к сканеру другим пользователям. Для этого необходимо разрешить доступ на чтение и запись к файлу устройства, обслуживающему сканер. В нашем примере USB сканер использует устройство /dev/uscanner0, принадлежащее группе `operator`. Добавление пользователя `joe` в группу `operator` разрешит ему использовать сканер:

```
# pw groupmod operator -m joe
```

За подробностями обращайтесь к странице справочника pw(8). Вам также потребуется установить нужные права доступа (0660 или 0664) к устройству /dev/uscanner0, поскольку по умолчанию группа `operator` может лишь читать из него. Это достигается добавлением следующей строки в файл /etc/devfs.rules:

```
[system=5]
add path uscanner0 mode 660
```

Затем добавьте в файл конфигурации системы /etc/rc.conf такую строку (после чего перезагрузите систему):

```
devfs_system_ruleset="system"
```

Подробную информацию о правах на файлы устройств вы найдете на странице справочника devfs(8).

Разумеется, по соображениям безопасности, вы должны как следует подумать, прежде чем добавлять пользователя в другие группы, в особенности в группу `operator`.

Глава 8. Настройка ядра FreeBSD

8.1. Краткий обзор

Ядро - это основная часть операционной системы FreeBSD. Оно ответственно за управление памятью, параметры безопасности, работу с сетью, доступ к дискам и многое другое. Несмотря на то, что FreeBSD становится всё более динамически конфигурируемой, иногда приходится собирать собственное ядро.

После прочтения этой главы вы узнаете:

- Почему вам может понадобиться сборка собственного ядра.
- Как написать файл конфигурации ядра или изменить существующий.
- Как использовать файл конфигурации ядра для того, чтобы создать и собрать новое ядро.
- Как установить новое ядро.
- Что делать, если что-то не работает или работает не так, как должно.

Все команды, приводимые в этой главе в качестве примера, должны выполняться от пользователя `root`.

8.2. Зачем собирать собственное ядро?

Традиционно в FreeBSD использовалось так называемое "монолитное" ядро. Это означает, что ядро - это одна большая программа, которая поддерживает фиксированный набор устройств и в случае, если необходимо изменить его поведение, требуется сборка нового ядра и перезагрузка компьютера уже с новым ядром.

На сегодняшний день FreeBSD быстро продвигается к модели, в которой большая часть функциональности содержится в модулях, которые могут быть при необходимости динамически загружены и выгружены из ядра. Это позволяет ядру использовать устройства, которые "внезапно" появились в системе (например, устройства PCMCIA в лэптопе) или добавлять новую функциональность в ядро, которая не была необходима в момент первоначальной сборки ядра. Такой подход известен как модульность ядра.

Несмотря на это, всё ещё иногда бывает необходимо, чтобы некоторая функциональность была вкомпилирована в ядро статически. В некоторых случаях это продиктовано тем, что эта функциональность настолько сильно привязана к ядру, что не может быть динамически загружаемой. В других случаях это может быть просто потому, что никто не уделил время написанию динамически загружаемого модуля для этой функциональности.

Сборка собственного ядра - один из наиболее важных ритуалов, совершаемых опытными пользователями BSD. Несмотря на длительность этого процесса, ваша FreeBSD останется только в выигрыше. В отличие от ядра GENERIC, которое должно поддерживать широкий спектр аппаратного обеспечения, собственное ядро содержит поддержку аппаратного обеспечения только *вашего* компьютера. Это может давать следующие преимущества:

- Меньшее время загрузки. Поскольку ядро будет пытаться определить только то аппаратное обеспечение, которое установлено в вашем компьютере, время, которое потребуется системе для загрузки, может значительно уменьшиться.
- Уменьшение использования памяти. Собственное ядро часто использует меньше памяти, чем ядро GENERIC, так как из него исключены лишние драйвера и неиспользуемые функциональные возможности. Это важно тем, что часть оперативной памяти постоянно занята кодом ядра и поэтому не может быть выделена приложениям. Именно по этой причине собственное ядро особенно полезно при использовании систем с малым объемом оперативной памяти.
- Поддержка дополнительного аппаратного обеспечения. Собственное ядро позволяет вам добавить поддержку устройств, отсутствующих в ядре GENERIC.

8.3. Определение аппаратного обеспечения

Перед тем, как углубиться в конфигурирование ядра, было бы разумно составить перечень установленного в компьютер аппаратного обеспечения. Если FreeBSD не является основной операционной системой, то перечень оборудования может быть легко составлен на основании анализа конфигурации текущей операционной системы. Например, Диспетчер устройств (Device Manager) от Microsoft® обычно содержит необходимую информацию об установленных устройствах. Диспетчер устройств находится на панели управления (control panel).

> У некоторых версий Microsoft® Windows® есть значок Система (System), вызов которого отобразит экран, содержащий среди прочих и Диспетчер устройств.

Если других операционных систем на машине не установлено, системному администратору придется искать эту информацию самостоятельно. Один из методов подразумевает использование утилиты dmesg(8) и команды man(1). У большинства драйверов во FreeBSD есть страницы справочника, содержащие список поддерживаемого оборудования, а найденные во время начальной загрузки устройства будут перечислены в dmesg(8). К примеру, следующие строки информируют о том, что драйвер psm обнаружил мышь:

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

Этот драйвер необходимо будет включить в конфигурацию собственного ядра или загрузить посредством loader.conf(5).

В некоторых случаях `dmesg` отображает только системные сообщения вместо сообщений начальной загрузки. В таких случаях необходимо обращаться к файлу /var/run/dmesg.boot.

Еще один метод нахождения аппаратного обеспечения подразумевает использование достаточно информативной утилиты pciconf(8). Например:

```
ath0@pci0:3:0:0:        class=0x020000 card=0x058a1014 chip=0x1014168c rev=0x01
hdr=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

Эта часть вывода, полученная в результате запуска команды `pciconf -lv`, показывает, что драйвер ath обнаружил беспроводное Ethernet устройство. Набрав `man ath`, вы получите страницу справочника ath(4).

Также, для извлечения необходимой информации, можно воспользоваться ключом `-k` к команде man(1). В вышеприведенном случае можно набрать:

```
# man -k Atheros
```

чтобы получить страницы справочника, содержащие определенное слово:

```
ath(4)                   - Atheros IEEE 802.11 wireless network driver
ath_hal(4)               - Atheros Hardware Access Layer (HAL)
```

Теперь, имея в распоряжении перечень аппаратного оборудования, можно безбоязненно приступить к сборке специализированного ядра.

8.4. Драйвера, подсистемы и модули ядра

Перед построением специализированного ядра, обдумайте причины, побудившие вас к этому. Если требуется поддержка специального оборудования, то она наверняка уже реализована в виде модуля.

Модули ядра находятся в каталоге /boot/kernel, и они могут быть динамически включены в работающее ядро при помощи kldload(8). Если не все, то большинство драйверов существуют в виде модулей, и у них есть соответствующая страница справочника. К примеру, в предыдущем разделе упоминался драйвер ath беспроводного Ethernet устройства. Соответствующая ему страница справочника гласит:

```
Alternatively, to load the driver as a module at boot time, place the
following line in man:loader.conf[5]:

    if_ath_load="YES"
```

Как уже выше сказано, добавление строки `if_ath_load="YES"` в файл /boot/loader.conf позволит динамически загружать этот модуль во время загрузки системы.

В некоторых случаях, однако, интересующего вас модуля не существует. Чаще всего это

справедливо для определенных подсистем и очень важных драйверов. Например, поддержка файловой системы FreeBSD (FFS) является обязательной опцией в ядре. Как и поддержка сети (INET). К сожалению, единственный способ определить является ли драйвер обязательным - это проверить наличие соответствующего модуля.

Довольно легко удалить поддержку устройства или опцию, получив тем самым неработоспособное ядро. Например, если драйвер ata(4) изъят из конфигурации ядра, то система, использующая диски ATA, может не загрузиться без записи о модуле, добавленной в loader.conf. Если есть сомнения, проверьте наличие модуля, и только потом исключайте поддержку в ядре.

8.5. Сборка и установка собственного ядра

Для сборки ядра необходимо наличие всех исходных файлов FreeBSD.

Во-первых, давайте сделаем краткий обзор каталога, в котором будет происходить сборка ядра. Все каталоги, которые будут упоминаться, будут относительными по отношению к основному каталогу /usr/src/sys, который также доступен как каталог /sys. Этот каталог содержит множество подкаталогов, представляющих собой различные части ядра, но наиболее важным для нас будет каталог arch/conf, в котором вы будете редактировать конфигурационный файл ядра и в котором находится каталог compile, где будет собираться ваше ядро. *arch* может быть i386, amd64, ia64, powerpc, sparc64 или pc98 (альтернативная ветвь аппаратного обеспечения, популярная в Японии). Все, что находится внутри каталога определенной архитектуры, относится только к этой архитектуре; остальной код является машинно независимым и общим для всех платформ, на которые FreeBSD может быть потенциально портирована. Обратите внимание на логическую структуру каталогов, в которой каждое поддерживаемое устройство, каждая файловая система и каждая опция размещается в своём собственном каталоге.

В примерах этой главы подразумевается, что вы используете архитектуру i386. Если архитектура вашей системы отличается от используемой в примерах, то вам необходимо будет соответственно изменить имена каталогов.

Если каталог /usr/src/ отсутствует в вашей системе (или этот каталог пуст), то это значит, что исходные тексты не были установлены. Наиболее простой способ установить их - воспользоваться csup(1), как описано в Синхронизация исходных текстов. Далее, создайте символическую ссылку на /usr/src/sys/:

```
# ln -s /usr/src/sys /sys
```

Затем, перейдите в каталог arch/conf и скопируйте файл конфигурации GENERIC в файл с выбранным вами именем. Например:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYKERNEL
```

По традиции имя состоит из букв в верхнем регистре, и если вы поддерживаете несколько компьютеров FreeBSD на различном оборудовании, хорошая идея добавлять это имя к имени хоста. Мы назвали ядро MYKERNEL в этом примере.

Помещение файла конфигурации ядра в /usr/src может быть плохой идеей. Если вы испытываете проблемы, их можно решить удалив /usr/src и начав все с начала. После этого обычно требуется несколько секунд, чтобы понять, что вы удалили собственный файл настройки ядра. Не редактируйте непосредственно GENERIC, он может быть также перезаписан и при следующем обновлении дерева исходных текстов, и изменения ядра будут потеряны.

Вы можете сохранить файл конфигурации ядра в другом месте, а затем создать символическую ссылку на этот файл в каталоге i386.

Например:

```
# cd /usr/src/sys/i386/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL
```

Теперь отредактируйте файл MYKERNEL в своём любимом текстовом редакторе. Если вы только начинаете, единственным доступным редактором скорее всего будет vi, который слишком сложен для того, чтобы описать его здесь, но в библиографии перечислено множество книг, в которых его использование хорошо освещено. Однако FreeBSD предоставляет более простой редактор ее, который, если вы - новичок, подойдёт вам больше всего. Не стесняйтесь изменять строки комментариев в начале файла, с тем, чтобы отобразить вашу конфигурацию или изменения, которые вы сделали по сравнению с GENERIC.

Если вам приходилось собирать ядро для SunOS™ или какой-либо другой операционной системы типа BSD, многое из того, что содержится в этом файле будет очень знакомо вам. Если же вы, напротив, использовали другую операционную систему, такую как DOS, файл конфигурации GENERIC может показаться вам крайне сложным, поэтому следуйте инструкциям в разделе Конфигурационный файл медленно и внимательно.

Если вы синхронизируете дерево исходных текстов с деревом проекта FreeBSD, не забудьте свериться с файлом /usr/src/UPDATING перед обновлением. В этом файле описаны все важные вопросы и области исходного кода, требующие особого внимания. /usr/src/UPDATING всегда соответствует версии ваших исходных текстов FreeBSD, поэтому является более актуальным источником информации, чем это руководство.

Теперь вы должны скомпилировать ядро.

Procedure: Сборка ядра

Для сборки ядра необходимо наличие всех исходных файлов FreeBSD.

1. Перейдите в каталог /usr/src:

```
# cd /usr/src
```

2. Соберите ядро:

```
# make buildkernel KERNCONF=MYKERNEL
```

3. Установите новое ядро:

```
# make installkernel KERNCONF=MYKERNEL
```

По умолчанию, при построении ядра, *все* модули ядра так же будут пересобраны. Если вы хотите обновить ядро быстрее или построить только определённые модули, то вам нужно отредактировать файл /etc/make.conf перед началом процесса сборки ядра:

```
MODULES_OVERRIDE = linux acpi sound/sound sound/driver/ds1 ntfs
```

Эта переменная устанавливает список модулей, которые нужно построить вместо построения всех модулей.

```
WITHOUT_MODULES = linux acpi sound ntfs
```

В этой переменной перечисляются основные модули, которые необходимо исключить из процесса сборки. За другими переменными, которые вы можете посчитать полезными в процессе сборки ядра, обращайтесь к странице справочника make.conf(5).

Новое ядро будет скопировано в каталог /boot/kernel как /boot/kernel/kernel, а старое ядро будет перемещено в /boot/kernel.old/kernel. Теперь перезагрузите систему для того, чтобы использовать новое ядро. Если что-то пойдёт не так, вы можете обратиться к разделу Решение проблем в конце этой главы, который может оказаться полезен. Не забудьте прочитать раздел, который объясняет как исправить ситуацию, когда ядро не загружается.

Другие файлы, относящиеся к процессу загрузки, такие как загрузчик

(loader(8)) и его конфигурационные файлы, размещаются в /boot. Модули сторонних производителей могут быть помещены в /boot/kernel, хотя пользователи должны знать, что очень важно, чтобы модули были синхронизированы с собранным ядром. Модули, не рассчитанные на работу с собранным ядром, могут вызвать нестабильность и некорректность работы.

8.6. Конфигурационный файл

Формат конфигурационного файла достаточно прост. Каждая строка представляет собой ключевое слово и один или более аргументов. Для простоты большинство строк содержат только один аргумент. Всё, что следует за символом `#` является комментарием и игнорируется. Следующие разделы описывают каждый параметр, в порядке, в котором они появляются в GENERIC. За полным списком архитектурно-зависимых параметров и устройств обратитесь к файлу NOTES в том же каталоге, что и GENERIC. Архитектурно независимые параметры находятся в /usr/src/sys/conf/NOTES.

Директива `include` стала доступной для использования в конфигурационных файлах. Она позволяет включать в текущий конфигурационный файл содержимое другого файла, тем самым упрощая процесс внесения небольших изменений в существующий файл. Например, если вам необходимо добавить всего несколько дополнительных опций или драйверов в ядро GENERIC, то вам придется поддерживать только разницу к файлу GENERIC:

```
include GENERIC
ident MYKERNEL

options         IPFIREWALL
options         DUMMYNET
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPDIVERT
```

Большинство администраторов оценят значительные преимущества перед старым способом - написанием конфигурационного файла "с нуля": ваш конфигурационный файл будет отображать только изменения относительно GENERIC. А после обновлений исходного кода, новые функциональные возможности, появившиеся в GENERIC, будут добавлены и в вашу конфигурацию, если только не препятствовать этому директивами `nooptions` или `nodevice`. Далее в этом разделе описывается типовой конфигурационный файл, его опции и устройства, а также их роли.

Для сборки ядра со всеми возможными опциями (обычно используется для тестирования), выполните от имени суперпользователя (`root`) следующую команду:

```
# cd /usr/src/sys/i386/conf && make LINT
```

Это пример конфигурационного файла ядра GENERIC с различными дополнительными

комментариями, которые могут понадобиться для ясности. Этот пример должен совпадать с вашей копией в /usr/src/sys/i386/conf/GENERIC практически полностью.

```
machine        i386
```

Это архитектура машины. Она должна быть одной из следующих: amd64, i386, ia64, pc98, powerpc, или sparc64.

```
cpu          I486_CPU
cpu          I586_CPU
cpu          I686_CPU
```

Эта опция указывает тип процессора, который используется в вашей системе. В конфигурационном файле может быть несколько вхождений этой опции (например, если вы не уверены, какой из типов процессора необходимо использовать - I586_CPU или I686_CPU), но для собственного ядра лучше указывать только тот тип процессора, который установлен в вашей системе. Если вы не уверены, какой тип необходимо использовать вам, вы можете воспользоваться файлом /var/run/dmesg.boot, чтобы увидеть протокол загрузки системы.

```
ident          GENERIC
```

Этот параметр определяет "метку" ядра. Необходимо, чтобы она соответствовала названию файла конфигурации ядра, например MYKERNEL, если вы следовали инструкциям в предыдущих примерах. Значение, которое вы присвоите параметру ident будет выводиться в процессе загрузки, поэтому полезно давать новым ядрам другие имена для того, чтобы отличать их от обычного ядра (например, если вы хотите собрать экспериментальное ядро).

```
#To statically compile in device wiring instead of /boot/device.hints
#hints          "GENERIC.hints"         # Default places to look for devices.
```

device.hints(5) используются для настройки параметров драйверов устройств. Путь по умолчанию, который loader(8) будет проверять при загрузке - /boot/device.hints. Используя опцию hints вы можете вкомпилировать эти параметры статически в ваше ядро. В этом случае не требуется создавать файл device.hints в каталоге /boot.

```
makeoptions     DEBUG=-g          # Build kernel with gdb(1) debug symbols
```

При обычном построении ядра в сборку включается отладочная информация: опция -g передается компилятору gcc(1).

```
options          SCHED_ULE         # ULE scheduler
```

Планировщик по умолчанию во FreeBSD. Оставьте эту опцию.

```
options         PREEMPTION        # Enable kernel thread preemption
```

Позволяет высокоприоритетным нитям ядра вытеснять конкурентов, находящихся в режиме выполнения. Эта опция может помочь повысить реактивность системы по отношению к внешним воздействиям, например, за счет снижения латентности нитей, обрабатывающих прерывания.

```
options         INET              # InterNETworking
```

Поддержка сетевых возможностей. Оставьте эту опцию включенной, даже если вы не планируете подключаться к сети. Большинство программ требуют, чтобы работал хотя бы интерфейс обратной связи (loopback) (т.е. создание сетевых соединений внутри вашего ПК), так что эта опция в принципе является обязательной.

```
options         INET6             # IPv6 communications protocols
```

Включает поддержку коммуникационных протоколов IPv6.

```
options         FFS               # Berkeley Fast Filesystem
```

Включает поддержку основной файловой системы. Не удаляйте эту опцию, если вы планируете загружаться с жесткого диска.

```
options         SOFTUPDATES       # Enable FFS Soft Updates support
```

Этот параметр включает в ядре технологию Soft Updates, которая повышает скорость записи на диски. Несмотря на то, что эта технология включена в ядре, она должна быть включена для отдельных дисков. Просмотрите вывод команды mount(8) чтобы определить, включены ли Soft Updates для дисков вашей системы. Если вы не увидите параметр `soft-updates`, вам будет необходимо активировать его при помощи команды tunefs(8) (для существующих файловых систем) или команды newfs(8) (для новых файловых систем).

```
options         UFS_ACL           # Support for access control lists
```

Этот параметр включает в ядре поддержку списков управления доступом (ACL). Основывается на использовании расширенных атрибутов и UFS2, детальное описание вы сможете найти в Списки контроля доступа файловой системы (ACL). ACL включены по умолчанию и не должны выключаться в случае, если они ранее использовались на файловой системе, так как это удалит списки управления доступом и изменит то, как защищены файлы, непредсказуемым образом.

```
options         UFS_DIRHASH      # Improve performance on big directories
```

Эта опция включает функциональность, которая повышает скорость дисковых операций на больших каталогах в обмен на использование дополнительной памяти. Для большого сервера или рабочей станции рекомендуется оставить ее включенной, и выключить для системы, для которой более приоритетна память, чем скорость доступа к дискам, например для брандмауэра.

```
options         MD_ROOT          # MD is a potential root device
```

Этот параметр включает поддержку использования дисков в памяти для корневой файловой системы.

```
options         NFSCLIENT        # Network Filesystem Client
options         NFSSERVER        # Network Filesystem Server
options         NFS_ROOT         # NFS usable as /, requires NFSCLIENT
```

Сетевая файловая система. Если вы не планируете монтировать разделы с файлового сервера UNIX® через TCP/IP, вы можете исключить этот параметр из конфигурационного файла ядра.

```
options         MSDOSFS          # MSDOS Filesystem
```

Файловая система MS-DOS®. Если вы не собираетесь монтировать форматированный в DOS раздел жесткого диска в момент загрузки, вы можете безопасно закомментировать этот параметр. Необходимый модуль будет автоматически загружен, когда вы в первый раз смонтируете раздел DOS, так, как это описано ниже. Кроме того, замечательный пакет emulators/mtools позволяет получить доступ к DOS дискетам без необходимости монтировать и размонтировать их (и не требует наличия `MSDOSFS`).

```
options         CD9660           # ISO 9660 Filesystem
```

Файловая система ISO 9660 для компакт-дисков. Если у вас нет привода CDROM или вы будете лишь изредка монтировать компакт-диски с данными, закомментируйте эту строку, так как необходимый модуль будет загружен автоматически при первом монтировании компакт-диска с данными. Для использования звуковых компакт-дисков эта файловая система не потребуется.

```
options         PROCFS           # Process filesystem (requires PSEUDOFS)
```

Файловая система процессов. Это "виртуальная" файловая система монтируемая в /proc, которая позволяет таким приложениям, как ps(1) выдавать вам больше информации о

запущенных процессах. Использование PROCFS не требуется, так как большинство мониторинговых и отладочных инструментов было адаптировано для работы без PROCFS: система по умолчанию не монтирует файловую систему процессов.

```
options         PSEUDOFS         # Pseudo-filesystem framework
```

Ядра, которые используют PROCFS, должны также включать поддержку PSEUDOFS,

```
options         GEOM_PART_GPT    # GUID Partition Tables.
```

Добавляет поддержку Таблиц Разделов GUID. Этот параметр делает возможным наличие большого количества разделов на одном диске, до 128 в стандартной конфигурации.

```
options         COMPAT_43        # Compatible with BSD 4.3 [KEEP THIS!]
```

Совместимость с 4.3BSD. Не выключайте эту опцию; некоторые приложения будут вести себя странно, если этой опции не будет в ядре.

```
options         COMPAT_FREEBSD4  # Compatible with FreeBSD4
```

Эта опция требуется для поддержки приложений, собранных на более старых версиях FreeBSD, которые используют старые интерфейсы вызовов. Рекомендуется использовать данную опцию на всех системах на платформах i386™, на которых могут запускаться старые приложения; платформы, поддержка которых появилась только в FreeBSD 5.X, например ia64 и sparc64, не требуют этой опции.

```
options         COMPAT_FREEBSD5  # Compatible with FreeBSD5
```

Эта опция необходима для поддержки приложений, скомпилированных на FreeBSD 5.X и использующих интерфейс системных вызовов FreeBSD 5.X.

```
options         COMPAT_FREEBSD6  # Compatible with FreeBSD6
```

Эта опция требуется для поддержки приложений, собранных на FreeBSD версий 6.X, которые используют интерфейсы системных вызовов FreeBSD 6.X.

```
options         COMPAT_FREEBSD7  # Compatible with FreeBSD7
```

Эта опция требуется на системах FreeBSD версий 8 и более поздних для поддержки приложений, собранных для FreeBSD 7.X и использующих интерфейсы системных вызовов FreeBSD 7.X.

```
options         SCSI_DELAY=5000  # Delay (in ms) before probing SCSI
```

Этот параметр заставляет ядро приостановиться на 5 секунд перед тем, как идентифицировать каждое устройство SCSI в вашей системе. Если у вас установлены только жесткие диски IDE, вы можете игнорировать эту опцию, в противном случае, возможно, вы захотите уменьшить это число, для того чтобы ускорить загрузку. Естественно, если вы сделаете это, а у FreeBSD появятся проблемы с распознанием ваших устройств SCSI, необходимо будет увеличить этот параметр.

```
options         KTRACE           # ktrace(1) support
```

Включает поддержку трассировки процессов, что удобно при отладке.

```
options         SYSVSHM          # SYSV-style shared memory
```

Этот параметр предоставляет поддержку разделяемой памяти System V. Наиболее распространенное применение этого - расширение XSHM в X, которое многие приложения, интенсивно работающие с графикой, будут автоматически использовать для повышения скорости работы. Если вы используете X, эта опция будет необходима.

```
options         SYSVMSG          # SYSV-style message queues
```

Поддержка сообщений System V. Этот параметр добавляет в ядро всего лишь несколько сотен байт.

```
options         SYSVSEM          # SYSV-style semaphores
```

Поддержка семафоров System V. Не настолько часто используемая возможность, но в ядро добавляет всего несколько сотен байт.

Команда ipcs(1) с параметром `-p` покажет все процессы, которые используют любую из этих возможностей System V.

```
options         _KPOSIX_PRIORITY_SCHEDULING # POSIX P1003_1B real-time extensions
```

Расширения реального времени, добавленные 1993 POSIX®. Определенные приложения из коллекции используют их, например StarOffice™.

```
options         KBD_INSTALL_CDEV  # install a CDEV entry in /dev
```

Этот параметр разрешает формирование файлов устройств в /dev для клавиатур.

```
options         ADAPTIVE_GIANT    # Giant mutex is adaptive.
```

Giant - имя механизма защиты ("спящего" мьютекса) для крупных наборов ресурсов ядра. На нынешний момент Giant представляется фактически непригодным для использования в связи с серьезными потерями в производительности, и активно заменяется на механизмы, защищающие отдельные ресурсы ядра. Параметр `ADAPTIVE_GIANT` включает Giant в число адаптивных мьютексов: в случае, когда нить ядра нуждается в Giant, а он уже захвачен нитью, выполняющейся на другом процессоре, первая нить будет продолжать выполнение и ждать освобождения Giant. В норме нить должна была бы уснуть, пока не настанет очередной момент ее выполнения. Если вы не уверены, оставьте этот параметр в покое.

> Для FreeBSD 8.0-RELEASE и более поздних версий, все мьютексы являются адаптивными по умолчанию, если обратное не указано специально опцией `NO_ADAPTIVE_MUTEXES`. Следовательно, Giant также адаптивен по умолчанию, и поэтому опция `ADAPTIVE_GIANT` была удалена из файла конфигурации ядра.

```
device          apic              # I/O APIC
```

Устройство apic разрешает использование набора I/O APIC для распределения прерываний. Оно может быть использовано как с однопроцессорными, так и с многопроцессорными ядрами (для последних наличие apic является обязательным). Для поддержки многопроцессорности добавьте строку `options SMP`.

> Устройство apic существует только на архитектурах i386. На других архитектурах этот конфигурационный параметр использовать не следует.

```
device          eisa
```

Включите эту опцию если у вас материнская плата EISA. Это включает автоопределение и конфигурирование поддержки всех устройств на шине EISA.

```
device          pci
```

Включите этот параметр, если у вас материнская плата с поддержкой PCI. Это включит автоопределение карт PCI и проксирование из шины PCI в шину ISA.

```
# Floppy drives
device          fdc
```

Контроллер флоппи-диска.

```
# ATA and ATAPI devices
```

```
device          ata
```

Этот драйвер поддерживает все устройства ATA и ATAPI. Вам необходима только одна строка `device ata` в ядре для того, чтобы обнаружить все PCI устройства ATA/ATAPI в современных машинах.

```
device          atadisk                 # ATA disk drives
```

Эта строка необходима вместе с `device ata` для поддержки дисков ATA.

```
device          ataraid                 # ATA RAID drives
```

Эта строка необходима вместе с `device ata` для поддержки дисков ATA RAID.

```
device          atapicd                 # ATAPI CDROM drives
```

Поддержка приводов ATAPI CDROM. Используется вместе с `device ata`.

```
device          atapifd                 # ATAPI floppy drives
```

Поддержка флоппи-приводов ATAPI. Используется вместе с `device ata`.

```
device          atapist                 # ATAPI tape drives
```

Поддержка ленточных приводов ATAPI (стримеров). Используется вместе с `device ata`.

```
options         ATA_STATIC_ID           # Static device numbering
```

Заставляет драйвер нумеровать устройства статически; в противном случае происходит динамическая нумерация.

```
# SCSI Controllers
device          ahb        # EISA AHA1742 family
device          ahc        # AHA2940 and onboard AIC7xxx devices
options         AHC_REG_PRETTY_PRINT    # Print register bitfields in debug
                                        # output.  Adds ~128k to driver.
device          ahd        # AHA39320/29320 and onboard AIC79xx devices
options         AHD_REG_PRETTY_PRINT    # Print register bitfields in debug
                                        # output.  Adds ~215k to driver.
device          amd        # AMD 53C974 (Teckram DC-390(T))
device          isp        # Qlogic family
#device         ispfw      # Firmware for QLogic HBAs- normally a module
```

```
device          mpt         # LSI-Logic MPT-Fusion
#device         ncr         # NCR/Symbios Logic
device          sym         # NCR/Symbios Logic (newer chipsets + those of `ncr')
device          trm         # Tekram DC395U/UW/F DC315U adapters

device          adv         # Advansys SCSI adapters
device          adw         # Advansys wide SCSI adapters
device          aha         # Adaptec 154x SCSI adapters
device          aic         # Adaptec 15[012]x SCSI adapters, AIC-6[23]60.
device          bt          # Buslogic/Mylex MultiMaster SCSI adapters

device          ncv         # NCR 53C500
device          nsp         # Workbit Ninja SCSI-3
device          stg         # TMC 18C30/18C50
```

Контроллеры SCSI. Закомментируйте те, которых у вас в системе нет. Если у вас в системе исключительно IDE устройства, вы можете удалить все эти строки. Строки вида `*_REG_PRETTY_PRINT` включают режим отладки для соответствующих драйверов.

```
# SCSI peripherals
device          scbus       # SCSI bus (required for SCSI)
device          ch          # SCSI media changers
device          da          # Direct Access (disks)
device          sa          # Sequential Access (tape etc)
device          cd          # CD
device          pass        # Passthrough device (direct SCSI access)
device          ses         # SCSI Environmental Services (and SAF-TE)
```

Периферийные устройства SCSI. Опять-таки, закомментируйте те, которых у вас в системе нет, или, если у вас в наличии исключительно IDE, можете удалить все.

USB umass(4) драйвер (и некоторые другие драйверы) используют подсистему SCSI, хотя и не являются настоящими SCSI устройствами. Следовательно, вам необходимо сохранить поддержку SCSI, если какой-либо из этих драйверов включен в конфигурацию ядра.

```
# RAID controllers interfaced to the SCSI subsystem
device          amr         # AMI MegaRAID
device      arcmsr      # Areca SATA II RAID
device          asr         # DPT SmartRAID V, VI and Adaptec SCSI RAID
device          ciss        # Compaq Smart RAID 5*
device          dpt         # DPT Smartcache III, IV - See NOTES for options
device          hptmv       # Highpoint RocketRAID 182x
device          hptrr       # Highpoint RocketRAID 17xx, 22xx, 23xx, 25xx
device          iir         # Intel Integrated RAID
device          ips         # IBM (Adaptec) ServeRAID
device          mly         # Mylex AcceleRAID/eXtremeRAID
device          twa         # 3ware 9000 series PATA/SATA RAID
```

```
# RAID controllers
device          aac         # Adaptec FSA RAID
device          aacp        # SCSI passthrough for aac (requires CAM)
device          ida         # Compaq Smart RAID
device          mfi         # LSI MegaRAID SAS
device          mlx         # Mylex DAC960 family
device          pst         # Promise Supertrak SX6000
device          twe         # 3ware ATA RAID
```

Поддерживаемые RAID-контроллеры. Если у вас нет таковых, можете их закомментировать или удалить эти строки.

```
# atkbdc0 controls both the keyboard and the PS/2 mouse
device          atkbdc      # AT keyboard controller
```

Контроллер клавиатуры (`atkbdc`) предоставляет средства ввода/вывода для клавиатуры AT и PS/2 устройств. Этот контроллер необходим драйверу клавиатуры (`atkbd`) и PS/2 устройств (`psm`).

```
device          atkbd       # AT keyboard
```

Драйвер `atkbd` вместе с контроллером `atkbdc` предоставляет доступ к клавиатуре AT 84 или улучшенной клавиатуре AT, которая подключена к контроллеру AT клавиатуры.

```
device          psm         # PS/2 mouse
```

Используйте это устройство, если ваша мышь включается в порт PS/2.

```
device          kbdmux         # keyboard multiplexer
```

Поддержка мультиплексора клавиатур. Если использование двух и более клавиатур не планируется, можете смело исключать этот параметр.

```
device          vga         # VGA video card driver
```

Драйвер видеокарты.

```
device          splash      # Splash screen and screen saver support
```

Заставка при загрузке. Хранители экрана также требуют этого устройства.

```
# syscons is the default console driver, resembling an SCO console
device          sc
```

sc - это драйвер консоли по умолчанию, который имитирует консоль SCO. Так как большая часть консольных полноэкранных приложений обращаются к консоли через терминальную библиотеку termcap, вас не должно волновать, будете ли вы использовать этот драйвер, либо драйвер vt, который является VT220-совместимым драйвером консоли. Если у вас возникнут какие-либо проблемы с приложениями, работающими с этим драйвером консоли, установите переменную окружения TERM в значение scoansi.

```
# Enable this for the pcvt (VT220 compatible) console driver
#device          vt
#options         XSERVER          # support for X server on a vt console
#options         FAT_CURSOR       # start with block cursor
```

VT220-совместимый драйвер консоли, обратно совместимый с VT100/102. Он работает лучше на некоторых лэптопах, у которых возникают проблемы несовместимости с sc. Также, установите переменную окружения TERM в значение vt100 или vt220. Этот драйвер также может быть полезен в случаях подключения к большому количеству различных машин через сеть, на которых параметры для устройства sc для termcap или terminfo могут отсутствовать - vt100 присутствует практически на любой платформе.

```
device          agp
```

Включите эту опцию, если у вас есть AGP карта в системе. Это включит поддержку AGP и AGP GART для тех карт, которые поддерживают эту возможность.

```
# Power management support (see NOTES for more options)
#device          apm
```

Поддержка Advanced Power Management. Чаще всего используется в лэптопах, хотя и отключена по умолчанию.

```
# Add suspend/resume support for the i8254.
device           pmtimer
```

Устройство таймера для управления энергопотреблением, APM и ACPI.

```
# PCCARD (PCMCIA) support
# PCMCIA and cardbus bridge support
device          cbb              # cardbus (yenta) bridge
device          pccard           # PC Card (16-bit) bus
device          cardbus          # CardBus (32-bit) bus
```

Поддержка PCMCIA. Включите ее, если вы используете лэптоп.

```
# Serial (COM) ports
device          sio             # 8250, 16[45]50 based serial ports
```

Четыре последовательных порта, которые известны как COM порты в мире MS-DOS®/Windows®

Если у вас есть внутренний модем на COM4 и последовательный порт COM2, вам понадобится поменять IRQ модема на 2 (по непонятным техническим причинам IRQ2 = IRQ9) для того, чтобы получить к нему доступ из FreeBSD. Если у вас есть многопортовая карта с последовательными портами, ознакомьтесь с sio(4) чтобы узнать корректные значения для добавления в /boot/device.hints. Некоторые видеокарты (в частности те, что используют чипы S3) используют адреса ввода/вывода в форме `0x*2e8` и, так как многие дешевые последовательные карты не полностью раскодируют шестнадцатибитное пространство адресов ввода/вывода, они конфликтуют с этими картами, в итоге COM4 оказывается практически недоступным.

Каждый последовательный порт требует уникального IRQ (кроме тех случаев, когда вы используете мультипортовую карту, которая поддерживает совместное использование прерываний), поэтому значения IRQ по умолчанию для COM3 и COM4 не могут быть использованы.

```
# Parallel port
device          ppc
```

Интерфейс параллельного порта на шине ISA.

```
device          ppbus      # Parallel port bus (required)
```

Поддержка шины параллельного порта.

```
device          lpt        # Printer
```

Поддержка принтеров на параллельном порту.

Все три последних устройства необходимы для поддержка принтеров на параллельном порту.

```
device          plip       # TCP/IP over parallel
```

Драйвер TCP/IP через параллельный порт.

```
device          ppi         # Parallel port interface device
```

Поддержка ввода/вывода общего назначения ("geek port") + IEEE1284 ввода/вывода.

```
#device         vpo         # Requires scbus and da
```

Драйвер привода Iomega Zip. Требует наличия `scbus` и `da`. Наилучшая производительность достигается с портами в режиме EPP 1.9.

```
#device         puc
```

Раскомментируйте это устройство, если у вас есть "простая" последовательная или параллельная PCI карта, поддерживаемая драйвером puc(4).

```
# PCI Ethernet NICs.
device          de          # DEC/Intel DC21x4x (Tulip)
device          em          # Intel PRO/1000 adapter Gigabit Ethernet Card
device          ixgb        # Intel PRO/10GbE Ethernet Card
device          txp         # 3Com 3cR990 (Typhoon)
device          vx          # 3Com 3c590, 3c595 (Vortex)
```

Драйвера сетевых карт PCI. Закомментируйте или удалите драйвера тех карт, которые отсутствуют в вашей системе.

```
# PCI Ethernet NICs that use the common MII bus controller code.
# NOTE: Be sure to keep the 'device miibus' line in order to use these NICs!
device          miibus      # MII bus support
```

Поддержка шины MII требуется для некоторых PCI 10/100 Ethernet карт, которые используют MII-совместимые передатчики или реализуют интерфейс управления передатчиком, который имитирует MII. Добавление `device miibus` в конфигурационный файл ядра включает поддержку стандартного API miibus и всех драйверов PHY, включая стандартный для тех PHY, которые не обрабатываются специфическим образом конкретным драйвером.

```
device          bfe         # Broadcom BCM440x 10/100 Ethernet
device          bge         # Broadcom BCM570xx Gigabit Ethernet
device          dc          # DEC/Intel 21143 and various workalikes
device          fxp         # Intel EtherExpress PRO/100B (82557, 82558)
device      lge     # Level 1 LXT1001 gigabit ethernet
device          msk         # Marvell/SysKonnect Yukon II Gigabit Ethernet
device      nge     # NatSemi DP83820 gigabit ethernet
device          pcn         # AMD Am79C97x PCI 10/100 (precedence over 'lnc')
device          re          # RealTek 8139C+/8169/8169S/8110S
device          rl          # RealTek 8129/8139
```

```
device          sf          # Adaptec AIC-6915 (Starfire)
device          sis         # Silicon Integrated Systems SiS 900/SiS 7016
device          sk          # SysKonnect SK-984x > SK-982x gigabit Ethernet
device          ste         # Sundance ST201 (D-Link DFE-550TX)
device          stge        # Sundance/Tamarack TC9021 gigabit Ethernet
device          ti          # Alteon Networks Tigon I/II gigabit Ethernet
device          tl          # Texas Instruments ThunderLAN
device          tx          # SMC EtherPower II (83c170 EPIC)
device      vge    # VIA VT612x gigabit ethernet
device          vr          # VIA Rhine, Rhine II
device          wb          # Winbond W89C840F
device          xl          # 3Com 3c90x (Boomerang, Cyclone)
```

Драйвера, которые используют контроллер шины MII.

```
# ISA Ethernet NICs.  pccard NICs included.
device          cs          # Crystal Semiconductor CS89x0 NIC
# 'device ed' requires 'device miibus'
device          ed          # NE[12]000, SMC Ultra, 3c503, DS8390 cards
device          ex          # Intel EtherExpress Pro/10 and Pro/10+
device          ep          # Etherlink III based cards
device          fe          # Fujitsu MB8696x based cards
device          ie          # EtherExpress 8/16, 3C507, StarLAN 10 etc.
device          lnc         # NE2100, NE32-VL Lance Ethernet cards
device          sn          # SMC's 9000 series of Ethernet chips
device          xe          # Xircom pccard Ethernet

# ISA devices that use the old ISA shims
#device         le
```

Драйвера сетевых карт ISA. Ознакомьтесь с файлом /usr/src/sys/i386/conf/NOTES, чтобы узнать, какие сетевые карты каким драйвером поддерживаются.

```
# Wireless NIC cards
device          wlan            # 802.11 support
```

Generic 802.11 support. This line is required for wireless networking.

```
device          wlan_wep        # 802.11 WEP support
device          wlan_ccmp       # 802.11 CCMP support
device          wlan_tkip       # 802.11 TKIP support
```

Crypto support for 802.11 devices. These lines are needed if you intend to use encryption and 802.11i security protocols.

```
device          an          # Aironet 4500/4800 802.11 wireless NICs.
```

```
device          ath             # Atheros pci/cardbus NIC's
device          ath_hal         # Atheros HAL (Hardware Access Layer)
device          ath_rate_sample # SampleRate tx rate control for ath
device          awi        # BayStack 660 and others
device          wi         # WaveLAN/Intersil/Symbol 802.11 wireless NICs.
#device         wl         # Older non 802.11 Wavelan wireless NIC.
```

Поддержка различных беспроводных карт.

```
# Pseudo devices
device   loop          # Network loopback
```

Стандартное устройство обратной связи для TCP/IP. Если вы запускаете telnet или FTP по отношению `localhost` (он же `127.0.0.1`), то соединение пройдёт через это устройство. Этот параметр *обязателен*.

```
device   random        # Entropy device
```

Генератор случайных чисел для криптографической защиты.

```
device   ether         # Ethernet support
```

`ether` необходим лишь в случае, если у вас есть сетевая карта. Он включает поддержку стандартного кода протокола Ethernet.

```
device   sl            # Kernel SLIP
```

`sl` - это поддержка SLIP. SLIP был практически вытеснен PPP, который легче настраивается, лучше подходит для соединений модем-модем и имеет больше возможностей.

```
device   ppp           # Kernel PPP
```

Поддержка PPP в ядре для соединений dial-up. Также существует версия PPP, реализованного как приложение, использующее `tun`, и предлагающее большую гибкость и большее количество возможностей, как, например, соединение при необходимости (наличии обращения к сети).

```
device   tun           # Packet tunnel.
```

Используется пользовательским программным обеспечением PPP. Обратитесь к разделу PPP этой книги за дальнейшей информацией.

```
device   pty           # Pseudo-ttys (telnet etc)
```

"псевдо-терминал" или имитированный порт для входа. Используется входящими `telnet` и `rlogin`-сессиями, приложением xterm и некоторыми другими приложениями, такими как Emacs.

```
device   md            # Memory disks
```

Псевдо-устройства дисков в памяти.

```
device   gif           # IPv6 and IPv4 tunneling
```

Поддержка туннелирования IPv6 через IPv4, IPv4 через IPv6, IPv4 через IPv4 и IPv6 через IPv6. Устройство `gif` является "автоклонируемым", и будет срздавать файлы устройств по мере необходимости.

```
device   faith         # IPv6-to-IPv4 relaying (translation)
```

Это псевдо-устройство захватывает пакеты, которые были посланы ему и перенаправляет их даемону трансляции IPv4/IPv6.

```
# The `bpf' device enables the Berkeley Packet Filter.
# Be aware of the administrative consequences of enabling this!
# Note that 'bpf' is required for DHCP.
device   bpf           # Berkeley packet filter
```

Фильтр пакетов Berkeley. Это псевдо-устройство позволяет переводить сетевые интерфейсы в "неразборчивый" (promiscuous) режим, в котором перехватывается любой пакет в широковещательной сети (например ethernet). Эти пакеты могут быть сохранены на диск и/или исследованы при помощи tcpdump(1).

Устройство bpf(4) также используется программой dhclient(8) для того, чтобы получить адрес шлюза по умолчанию и т.п. Если вы используете DCHP, не удаляйте эту опцию.

```
# USB support
device          uhci          # UHCI PCI->USB interface
device          ohci          # OHCI PCI->USB interface
device      ehci          # EHCI PCI->USB interface (USB 2.0)
device          usb           # USB Bus (required)
#device         udbp          # USB Double Bulk Pipe devices
device          ugen          # Generic
device          uhid          # Human Interface Devices
```

```
device          ukbd          # Keyboard
device          ulpt          # Printer
device          umass         # Disks/Mass storage - Requires scbus and da
device          ums           # Mouse
device          ural          # Ralink Technology RT2500USB wireless NICs
device          urio          # Diamond Rio 500 MP3 player
device          uscanner      # Scanners
# USB Ethernet, requires mii
device          aue           # ADMtek USB Ethernet
device          axe           # ASIX Electronics USB Ethernet
device      cdce          # Generic USB over Ethernet
device          cue           # CATC USB Ethernet
device          kue           # Kawasaki LSI USB Ethernet
device          rue           # RealTek RTL8150 USB Ethernet
```

Поддержка различных USB устройств.

```
# FireWire support
device          firewire      # FireWire bus code
device          sbp           # SCSI over FireWire (Requires scbus and da)
device          fwe           # Ethernet over FireWire (non-standard!)
```

Поддержка различных устройств Firewire.

За дальнейшей информацией о дополнительных устройствах, поддерживаемых FreeBSD, обратитесь к файлу /usr/src/sys/i386/conf/NOTES.

8.6.1. Конфигурации с большим количеством оперативной памяти (PAE)

Машины с большим количеством оперативной памяти, в которых требуется более 4 гигабайт в пользовательском адресном пространстве и адресном пространстве ядра (User+Kernel Virtual Address, KVA) в обычном случае не смогут использовать более 4 гигабайт. Для решения этой проблемы Intel добавили поддержку 36-битной адресации в Pentium® Pro и более поздних моделях процессоров.

Расширение физического адресного пространства (PAE) в процессорах Intel® Pentium® Pro и более поздних позволяет использовать до 64 гигабайт оперативной памяти. FreeBSD имеет поддержку этой возможности посредством опции ядра `PAE`, доступной во всех текущих версиях FreeBSD. В связи с ограничениями архитектуры Intel, не делается никакого различия между памятью ниже или выше 4 гигабайт. Память, размещенная выше 4 гигабайт, просто добавляется к доступной памяти.

Для того, чтобы включить PAE в ядре, просто добавьте приведенную строку в конфигурационный файл ядра:

```
options           PAE
```

Поддержка PAE в FreeBSD существует только для процессоров Intel® IA-32. Также следует заметить, что PAE в FreeBSD не было полностью протестировано и должно считаться находящимся в состоянии бета-тестирования по сравнению с другими, стабильными возможностями FreeBSD.

Поддержка PAE в FreeBSD имеет следующие ограничения:

- Процесс не может получить доступ к более, чем 4 гигабайтам пространства VM.
- Драйверы устройств, которые не используют интерфейс bus_dma(9), приведут к повреждению информации в ядре с включенным PAE. Не рекомендуется использовать такие драйверы. По этой причине в FreeBSD включен конфигурационный файл ядра PAE, из которого удалены все драйверы, о которых известно, что они не работают при включенной поддержке PAE.
- Некоторые системные переменные определяют использование ресурсов памяти по количеству доступной физической памяти. Такие переменные могут привести к ненужному чрезмерному выделению памяти из-за особенностей работы системы PAE. Один из таких примеров - переменная `kern.maxvnodes`, которая управляет максимальным количеством vnode, разрешенных в ядре. Рекомендуется установить эту и подобные ей переменные вручную в адекватные значения.
- Возможно, понадобится увеличить пространство виртуальных адресов ядра (KVA) или уменьшить какую-либо переменную (см. выше), значение которой было неоправданно велико и могло привести к исчерпанию KVA. Для этого может быть использована опция ядра `KVA_PAGES`.

В случае сомнений относительно производительности и стабильности рекомендуется обратиться к странице руководства tuning(7). Страница руководства pae(4) содержит свежую информацию о поддержке PAE в FreeBSD.

8.7. Решение проблем

Существует четыре категории проблем, которые могут возникнуть при сборке собственного ядра. Вот они:

Не удаётся отработать команде `config`

Если команда config(8) не может отработать, то, скорее всего, вы допустили где-нибудь маленькую ошибку. К счастью, config(8) выведет номер проблемной строки, поэтому вы можете быстро найти строку, содержащую ошибку. Например, если вы видите:

```
config: line 17: syntax error
```

Убедитесь, что опция введена верно путём сравнения с файлом GENERIC или другим источником.

Не удаётся отработать команде `make`

Если не удаётся отработать команде `make`, обычно это означает ошибку в описании

конфигурации ядра, которая не достаточно тривиальна для того, чтобы config(8) мог обнаружить её. Опять-таки, просмотрите файл конфигурации, и, если вы все еще не можете решить проблему, напишите письмо в Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD, включив в письмо файл конфигурации ядра. Скорее всего проблема будет решена быстро.

Ядро не загружается:

Если ваше новое ядро не загружается или ему не удаётся обнаружить ваши устройства - не паникуйте! К счастью, в FreeBSD существует отличный механизм для восстановления после установки несовместимого ядра. Просто выберите ядро, которое хотите загрузить, в загрузчике FreeBSD. Доступ к нему вы можете получить, когда система находится в стартовом меню. Выберите шестой пункт ("Escape to a loader prompt"), введите команду `boot kernel.old`, или используйте любое другое ядро, которое загрузится без проблем. Во время переконфигурирования ядра всегда полезно оставлять копию ядра, о котором известно, что оно рабочее.

После загрузки с рабочим ядром вы можете проверить ваш файл конфигурации и попробовать собрать ядро опять. Очень полезным в данном случае окажется файл /var/log/messages, в котором, среди других записей, имеются сообщения ядра от каждой успешной загрузки. Также, команда dmesg(8) выведет сообщения ядра от текущей загрузки.

Если у вас возникли проблемы со сборкой ядра, убедитесь, что вы сохранили ядро GENERIC или другое рабочее ядро под другим именем, чтобы оно не было удалено при следующей сборке. Вы не можете использовать kernel.old, потому что при установке нового ядра kernel.old перезаписывается последним установленным ядром, которое может оказаться нерабочим. Также, как можно скорее переместите рабочее ядро в /boot/kernel, так как некоторые команды, такие как ps(1) будут работать некорректно. Для этого просто переместите каталог, содержащий работоспособное ядро:

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

Ядро работает, но ps(1) больше не работает

Если вы установили версию ядра отличную от той, с которой были собраны ваши системные утилиты, например, ядро от -CURRENT на системе -RELEASE, большая часть системных команд, таких как ps(1) и vmstat(8) не будут больше работать. Вам потребуется перекомпилировать и установить систему той же версии исходных текстов, что и ядро. Это одна из причин, по которой не следует использовать версию ядра, отличную от версии всей остальной системы.

Глава 9. Печать

9.1. Краткий обзор

FreeBSD можно использовать для печати на широком спектре принтеров, от старых матричных до новейших лазерных, без исключений, что позволяет создавать высококачественные распечатки из используемых приложений.

FreeBSD можно также сконфигурировать для работы в качестве сервера печати в сети; в этом качестве FreeBSD может получать задания печати от множества других компьютеров, включая другие компьютеры под управлением ОС FreeBSD, хосты Windows® и Mac OS®. FreeBSD будет гарантировать печать заданий по одному и может сохранять информацию о том, какие пользователи и машины выполняют основную часть печати, выдавать страницы-"баннеры", показывающие, кому принадлежит распечатка, и многое другое.

При прочтении этой главы вы узнаете:

- Как конфигурировать спулер печати FreeBSD.
- Как устанавливать фильтры печати для специфической обработки определенных заданий печати, включая преобразование поступающих на печать документов в форматы, которые понимает принтер.
- Как включить при печати колонтитулы или выдачу страниц-баннеров.
- Как печатать на принтеры, подключенные к другим компьютерам.
- Как печатать на принтеры, подключенные непосредственно к сети.
- Как задавать ограничения для принтера, включая ограничение размера заданий печати и запрет печати для отдельных пользователей.
- Как сохранять статистическую информацию о печати и учитывать использование принтера.
- Как решать проблемы печати.

Прежде чем читать эту главу, вы должны:

- Знать, как сконфигурировать и установить новое ядро (Настройка ядра FreeBSD).

9.2. Введение

Для использования принтеров в ОС FreeBSD вы можете настроить их для работы с системой спулинга печати Беркли (Berkeley line printer spooling system), также известной как система спулинга LPD. Это - стандартная система управления принтером во FreeBSD. В этой главе представлена система спулинга LPD и описано ее конфигурирование.

Если вы уже знакомы с LPD или другой системой спулинга печати, вы можете сразу перейти к разделу Базовая настройка.

LPD управляет всеми аспектами работы принтеров хоста. Она отвечает за несколько вещей:

- Она управляет доступом к непосредственно подключенным принтерам и принтерам, подключенным к другим хостам в сети.
- Она позволяет пользователям посылать файлы на печать; эти данные называют *заданиями*.
- Она предотвращает одновременный доступ к принтеру нескольких пользователей путем поддержки *очереди* для каждого принтера.
- Она позволяет печатать *страницы заголовка* (их также называют *баннерными* или *начальными* страницами), чтобы пользователи могли легко находить распечатанные задания в пачке распечаток.
- Она обеспечивает установку параметров взаимодействия для принтеров, подключенных к последовательным портам.
- Она может отправлять задания по сети спулеру LPD на другом хосте.
- Она может применять специальные фильтры для форматирования заданий для печати на разных языках описания страниц или задействования специфических возможностей принтера.
- Она учитывает использование принтера.

С помощью файла конфигурации (/etc/printcap) и за счет предоставления специальных программ фильтрования, можно потребовать от системы LPD выполнять все или некоторые из перечисленных выше функций на широком спектре принтерного оборудования.

9.2.1. Зачем использовать спулер

Если вы - единственный пользователь системы, вы можете спросить, зачем возиться со спулером, если управление доступом, страницы заголовка или учет использования принтера вам не нужны. Хотя можно обеспечить непосредственный доступ к принтеру, в любом случае следует использовать спулер, поскольку:

- LPD печатает задания в фоновом режиме; вам не придется ждать, пока данные будут скопированы на принтер.
- LPD позволяет легко пропустить задание печати через фильтры для добавления заголовков с датой/временем или преобразования специального формата файлов (такого как TeX DVI) в формат, который понимает принтер. Вам не придется выполнять эти шаги вручную.
- Многие свободно распространяемые и коммерческие программы, обеспечивающие возможность печати, обычно предполагают взаимодействие со спулером системы. Путем настройки системы спулинга вы упростите поддержку другого программного обеспечения, которое может быть добавлено в дальнейшем или уже установлено.

9.3. Основная настройка

Для использования принтеров с системой спулинга LPD, необходимо настроить как сам принтер, так и программное обеспечение LPD. Этот документ описывает два уровня настройки:

- См. раздел Простая настройка принтера, чтобы узнать, как подключить принтер, объяснить LPD, как с ним взаимодействовать, и отправлять на принтер простые текстовые файлы.
- См. раздел Расширенная настройка принтера, чтобы узнать, как печатать файлы множества специальных форматов, как печатать страницы заголовка, печатать по сети, управлять доступом к принтерам и учитывать использование принтера.

9.3.1. Простая настройка принтера

В этом разделе описано, как сконфигурировать принтер и программное обеспечение LPD для использования принтера. Здесь рассматриваются следующие вопросы:

- В разделе Настройка оборудования представлены советы по подключению принтера к порту компьютера.
- В разделе Настройка программного обеспечения показано, как настроить файл конфигурации спулера LPD (/etc/printcap).

Если вы настраиваете принтер, использующий для принятия заданий печати сетевой протокол, вместо локальных интерфейсов компьютера, см. раздел Принтеры с сетевыми интерфейсами.

Хотя этот раздел и назван "Простая настройка принтера", это, на самом деле, достаточно сложно. Заставить принтер работать с компьютером и спулером LPD - самая сложная часть. Расширенные опции, вроде выдачи страниц заголовков и учета использования, установить несложно, как только принтер заработает.

9.3.1.1. Настройка оборудования

В этом разделе описаны различные способы подключения принтера к ПК. Рассматриваются различные порты и кабели, а также параметры конфигурации ядра, которые может потребоваться установить, чтобы ОС FreeBSD могла взаимодействовать с принтером.

Если вы уже подключили ваш принтер и успешно печатали на него в другой операционной системе, можете перейти к разделу Настройка программного обеспечения.

9.3.1.1.1. Порты и кабели

Принтеры, которые продаются сегодня для использования на ПК, обычно поддерживают один или несколько из следующих интерфейсов:

- *Последовательные* интерфейсы, также известные как RS-232, или COM-порты, используют для посылки данных на принтер последовательный порт компьютера. Последовательные интерфейсы широко распространены в компьютерной индустрии, кабели для них легко найти и просто сделать. Для последовательных интерфейсов иногда нужны специальные кабели, и для их использования может потребоваться настраивать достаточно сложные опции взаимодействия. Большинство последовательных портов ПК имеют максимальную скорость передачи 115200 бит/сек, поэтому печатать через них большие графические задания неудобно.
- *Параллельные* интерфейсы используют параллельный порт компьютера для посылки

данных на принтер. Параллельные интерфейсы широко распространены на рынке ПК и работают быстрее, чем последовательные RS-232. Кабели легко найти, но сделать самостоятельно сложнее. При использовании параллельных интерфейсов опции взаимодействия обычно задавать не надо, что делает их конфигурирование существенно проще.

Параллельные интерфейсы иногда называют интерфейсами "Centronics", по названию типа разъема на принтере.

- Интерфейсы USB (сокращение от Universal Serial Bus - универсальная последовательная шина), могут работать на еще больших скоростях, чем параллельные или последовательные интерфейсы RS-232. Кабели для них - простые и дешевые. USB превосходит последовательный RS-232 и параллельный интерфейсы для печати, но не слишком хорошо поддерживается в UNIX®-системах. Обойти эту проблему можно, купив принтер с двумя интерфейсами, USB и параллельным, как у многих принтеров.

В общем случае, параллельные интерфейсы обычно обеспечивают только одностороннюю передачу (с компьютера на принтер), тогда как последовательные и USB поддерживают двустороннюю. Более новые параллельные порты (EPP и ECP) и принтеры могут взаимодействовать в обоих направлениях под FreeBSD, если используется кабель, соответствующий стандарту IEEE-1284.

Двустороннее взаимодействие с принтером через параллельный порт обычно выполняется одним из двух способов. Первый метод опирается на использование специально созданного драйвера принтера для FreeBSD, который поддерживает специфический язык данного принтера. Этот метод типичен для струйных принтеров и может использоваться для получения информации об уровне чернил и другой информации о состоянии. Второй метод используется, когда принтер поддерживает PostScript®.

Фактически, задания PostScript® являются программами, посылаемыми для выполнения принтеру; они вообще могут не выдавать результат на бумагу и возвращать его непосредственно компьютеру. PostScript® также использует двустороннее взаимодействие для сообщения компьютеру о проблемах, таких как ошибки в PostScript®-программе или замятие бумаги. Такая информация может пригодиться пользователям. Более того, лучший способ эффективного учета использования PostScript®-принтера требует двустороннего взаимодействия: вы запрашиваете у принтера значение счетчика страниц (сколько страниц напечатал принтер за все время существования), затем посылаете задание пользователя, затем снова запрашиваете значение его счетчика страниц. Вычитаем одно значение из другого, и узнаем, сколько бумаги потратил пользователь.

9.3.1.1.2. Параллельные порты

Для подключения принтера через параллельный интерфейс, соедините принтер и компьютер кабелем Centronics. Инструкции для принтера, для компьютера или обе должны полностью описывать эту процедуру.

Помните, какой параллельный порт компьютера вы использовали. Первый параллельный порт в ОС FreeBSD - ppc0; второй - ppc1, и так далее. Имена устройств для принтеров используют ту же схему: /dev/lpt0 для принтера на первом параллельном порту и т.д.

9.3.1.1.3. Последовательные порты

Для подключения принтера через последовательный интерфейс, соедините принтер с компьютером подходящим последовательным кабелем. Инструкции для принтера, для компьютера или обе должны полностью описывать эту процедуру.

Если вы не знаете, что такое "подходящий последовательный кабель", можете попробовать использовать один из следующих:

- *Модемный* кабель соединяет каждый штырёк на одном конце кабеля напрямую с соответствующим штырьком на другом конце. Кабель такого типа также называют кабелем "DTE-to-DCE". *

  Нуль-модемный кабель соединяет часть штырьков напрямую, другие - меняет (пересылку данных на приём данных, например), а некоторые - закорачивает на каждом разъеме. Кабель такого типа также называют кабелем "DTE-to-DTE" cable.

- Кабель *последовательного принтера*, необходимый для некоторых редко используемых принтеров, похож на нуль-модемный кабель, но посылает часть сигналов на соответствующие штырьки, а не закорачивает их.

Вам надо также настроить эти параметры взаимодействия с принтером, обычно - через элементы управления на лицевой панели или переключатели (DIP switches) на принтере. Выберите максимальную скорость передачи `bps` (бит в секунду, иногда - *baud rate*), которую могут поддерживать как компьютер, так и принтер. Выберите 7 или 8 битов данных; четность none, even или odd; и 1 или 2 стоп-бита. Также надо выбрать протокол управления передачей: none или XON/XOFF (также известный как "внутриполосный" или "программный"). Запомните выбранные установки для последующего конфигурирования программного обеспечения.

9.3.1.2. Настройка программного обеспечения

В этом разделе описана настройка программного обеспечения, необходимая для печати с помощью системы спулинга LPD в ОС FreeBSD.

Вот план действий, которые необходимо выполнить:

1. При необходимости, сконфигурировать в ядре поддержку порта, к которому подключен принтер; в разделе Конфигурирование ядра описано, что надо сделать.
2. Установить режим взаимодействия для параллельного порта, если используется параллельный порт; детали представлены в разделе Настройка режима взаимодействия для параллельного порта.
3. Проверить, может ли операционная система посылать данные на принтер. В разделе Проверка взаимодействия с принтером даны советы, как это сделать.
4. Настроить LPD для принтера, изменяя файл /etc/printcap. Как это сделать описано далее в этой главе.

9.3.1.2.1. Конфигурирование ядра

Ядро операционной системы компилируется для работы с конкретным набором устройств. Последовательный или параллельный интерфейс для принтера входит в этот набор. Поэтому может понадобиться добавить поддержку для дополнительного последовательного или параллельного порта, если он еще не сконфигурирован в ядре.

Чтобы узнать, поддерживает ли используемое в настоящий момент ядро последовательный интерфейс, наберите:

```
# grep sioN /var/run/dmesg.boot
```

Где *N* - номер последовательного порта, начиная с нуля. Если вы получаете результат, подобный следующему:

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
```

значит, ядро поддерживает порт.

Чтобы узнать, поддерживает ли ядро параллельный интерфейс, наберите:

```
# grep ppcN /var/run/dmesg.boot
```

Где *N* номер параллельного порта, начиная с нуля. Если вы получаете результат, подобный следующему:

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

значит, ядро поддерживает порт.

Может потребоваться переконфигурировать ядро, чтобы операционная система распознала и использовала параллельный или последовательный порт, используемый для подключения принтера.

Чтобы добавить поддержку последовательного порта, обратитесь к разделу, посвященному конфигурированию ядра. Чтобы добавить поддержку параллельного порта, почитайте этот же раздел *и* следующий раздел.

9.3.1.3. Настройка режима взаимодействия для параллельного порта

При использовании параллельного интерфейса можно выбрать, должна ли ОС FreeBSD взаимодействовать с принтером на основе прерываний или путем опроса. Универсальный драйвер принтера (lpt(4)) во FreeBSD использует систему ppbus(4), которая управляет

чипсетом порта с помощью драйвера ppc(4).

- Метод взаимодействия *на основе прерываний* является стандартным для ядра GENERIC. По этому методу, операционная система использует линию запроса прерывания (IRQ line) для определения готовности принтера к приему данных.
- Метод взаимодействия *путем опроса* требует от операционной системы постоянно запрашивать принтер, готов ли он к приему данных. Когда он отвечает, что готов, ядро посылает дополнительные данные.

Метод взаимодействия на основе прерываний обычно работает несколько быстрее, но использует ценную линию запроса прерывания. Про некоторые новые принтеры HP утверждают, что они работают некорректно в режиме взаимодействия на основе прерываний, вероятно, из-за некоторой (еще не вполне понятной) проблемы синхронизации. Для этих принтеров необходимо устанавливать режим опроса. Используйте тот режим, который работает. Некоторые принтеры будут работать в обоих режимах, но оказываются крайне медленными в режиме на основе прерываний.

Режим взаимодействия можно установить двумя способами: конфигурируя ядро или с помощью программы lptcontrol(8).

Для установки режима взаимодействия путем конфигурирования ядра:

1. Отредактируйте файл конфигурации ядра. Найдите запись `ppc0`. Если вы настраиваете второй параллельный порт, ищите запись `ppc1`. Используйте запись `ppc2` для третьего порта, и так далее.
   - Если необходимо установить режим на основе прерываний, отредактируйте следующую строку:

     ```
     hint.ppc.0.irq="N"
     ```

     в файле /boot/device.hints, заменив *N* соответствующим номером IRQ. Файл конфигурации ядра также должен содержать драйвер ppc(4):

     ```
     device ppc
     ```

   - Если необходимо установить режим опроса, удалите из файла /boot/device.hints следующую строку:

     ```
     hint.ppc.0.irq="N"
     ```

     В некоторых случаях, этого недостаточно для перевода порта в режим опроса под FreeBSD. Чаще всего, проблема связана с драйвером acpi(4), который может опрашивать и подключать устройства и, тем самым, управлять режимом доступа к порту принтера. Чтобы решить эту проблему, проверьте конфигурацию acpi(4).

2. Сохраните файл. Затем сконфигурируйте, соберите и установите ядро и перезагрузите систему. Подробнее см. в разделе конфигурирование ядра.

Для настройки режима взаимодействия с помощью утилиты lptcontrol(8):

1. Введите команду:

   ```
   # lptcontrol -i -d /dev/lptN
   ```

   для установки режима взаимодействия на основе прерываний для `lptN`.

2. Введите команду:

   ```
   # lptcontrol -p -d /dev/lptN
   ```

   для установки режима взаимодействия по опросу для `lptN`.

Вы можете поместить эти команды в файл /etc/rc.local для установки требуемого режима при каждой загрузке системы. Дополнительную информацию об этом ищите на странице справочного руководства lptcontrol(8).

9.3.1.4. Проверка взаимодействия с принтером

Прежде чем переходить к конфигурированию системы спулинга, надо убедиться, что операционная система может успешно посылать данные на принтер. Намного проще отлаживать взаимодействие с принтером и систему спулинга отдельно.

Для тестирования принтера мы пошлем на него текст. Для принтеров, которые могут непосредственно печатать посланные на них символы, идеально подходит программа lptest(1): она генерирует все 96 печатных символов ASCII в 96 строках.

Для PostScript®- (или основанного на другом языке) принтера, необходим более сложный тест. Подойдет небольшая PostScript®-программа, вроде следующей:

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto /Helvetica findfont 12 scalefont setfont
(Is this thing working?) show
showpage
```

Представленный выше PostScript®-код можно поместить в в файл и использовать, как показано в примерах в следующих разделах.

Когда в этом документе речь идет о языке принтера, подразумевается язык типа PostScript®, а не PCL компании Hewlett Packard. Хотя PCL имеет

> прекрасные функциональные возможности, в нем можно смешивать обычный текст с его управляющими последовательностями. PostScript® не позволяет непосредственно печатать обычный текст, и это язык принтера именно того рода, для которого надо выполнять специальные настройки.

9.3.1.4.1. Проверка параллельного принтера

В этом разделе описано, как проверить, может ли ОС FreeBSD взаимодействовать с принтером, подключенным к параллельному порту.

Для тестирования принтера на параллельном порту:

1. Станьте пользователем `root` с помощью команды su(1).
2. Пошлите данные на принтер.
   - Если принтер может печатать обычный текст, используйте утилиту lptest(1). Введите команду:

     ```
     # lptest > /dev/lptN
     ```

     Где *N* - номер параллельного порта, начиная с нуля.

   - Если принтер понимает PostScript® или другой язык принтера, пошлите на принтер небольшую программу. Введите команду:

     ```
     # cat > /dev/lptN
     ```

     Затем, построчно, *внимательно* введите программу, поскольку вы не сможете отредактировать строку после нажатия клавиши `RETURN` или `ENTER`. По окончании ввода программы, нажмите `CONTROL+D` или другую комбинацию клавиш, используемую для ввода символа конца файла.

     Можно также поместить программу в файл и выполнить команду:

     ```
     # cat file > /dev/lptN
     ```

     Где *file* - имя файла, содержащего программу, которую вы хотите послать принтеру.

Вы должны увидеть распечатку. Не переживайте, если текст выглядит не так, как предполагалось; этими проблемами мы займемся позже.

9.3.1.4.2. Проверка последовательного принтера

В этом разделе описано, как проверить, может ли ОС FreeBSD взаимодействовать с принтером, подключенным к последовательному порту.

Для тестирования принтера на последовательном порту:

1. Станьте пользователем `root` с помощью команды su(1).
2. Отредактируйте файл /etc/remote. Добавьте следующую запись:

   ```
   printer:dv=/dev/port:br#bps-rate:pa=parity
   ```

   Где *port* - специальный файл устройства для последовательного порта (`ttyd0`, `ttyd1` и т.д.), *bps-rate* - скорость обработки данных принтером, в битах в секунду, а *parity* - требуемая принтером четность (значение `even`, `odd`, `none` или `zero`).

   Вот пример записи для принтера, подключенного к третьему последовательному порту на скорости 19200 bps без четности:

   ```
   printer:dv=/dev/ttyd2:br#19200:pa=none
   ```

3. Подключитесь к принтеру с помощью tip(1). Введите команду:

   ```
   # tip printer
   ```

   Если этот шаг не срабатывает, снова отредактируйте файл /etc/remote и попробуйте использовать устройство /dev/cuaaN вместо /dev/ttydN.

4. Пошлите данные на принтер.
   - Если принтер может печатать обычный текст, используйте утилиту lptest(1). Введите команду:

     ```
     % $lptest
     ```

   - Если принтер понимает PostScript® или другой язык принтера, пошлите на принтер небольшую программу. Вводите программу, построчно, *очень внимательно*, поскольку нажатие клавиши Backspace или других клавиш редактирования может иметь значение для принтера. Может также понадобиться нажать специальную комбинацию клавиш, обозначающую конец файла, чтобы принтер понял, что получена вся программа. Для PostScript®-принтеров нажмите `CONTROL+D`.

     Можно также поместить программу в файл и ввести команду:

     ```
     % >file
     ```

     Где *file* - имя файла, содержащего программу. После того, как утилита tip(1) пошлет файл, нажмите требуемую для ввода признака конца файла

комбинацию клавиш.

Вы должны увидеть распечатку. Не переживайте, если текст выглядит не так, как предполагалось; этими проблемами мы займемся позже.

9.3.1.5. Включение спулера: файл /etc/printcap

Сейчас ваш принтер уже должен быть подключен, ядро (при необходимости) - сконфигурировано для взаимодействия с ним, и вы смогли послать на принтер простые данные. Теперь мы готовы к конфигурированию системы LPD для управления доступом к принтеру.

Система LPD конфигурируется путем редактирования файла /etc/printcap. Система спулинга LPD читает этот файл при каждом использовании спулера, так что, изменения в файле сразу же учитываются.

Формат файла printcap(5) прост. Используйте свой любимый текстовый редактор для изменения файла /etc/printcap. Формат файла идентичен формату других файлов, описывающих характеристики, например, /usr/shared/misc/termcap и /etc/remote. Полная информация о формате представлена на странице справочного руководства cgetent(3).

Простое конфигурирование спулера включает следующие шаги:

1. Выберите имя (и несколько удобных псевдонимов) для принтера и поместите их в файл /etc/printcap; подробнее об именовании см. в разделе Именование принтера.
2. Отключите выдачу начальных страниц (которые по умолчанию выдаются), вставив характеристику `sh`; подробнее об этом см. в разделе Подавление выдачи начальных страниц.
3. Создайте каталог для спулинга и укажите его местонахождение с помощью характеристики `sd`; подробнее об этом см. в разделе Создание каталога спулинга.
4. Выберите специальный файл устройства /dev для использования с принтером и укажите его в файле /etc/printcap с помощью характеристики `lp`; подробнее об этом см. в разделе Выбор устройства для принтера. Кроме того, если принтер подключен к последовательному порту, настройте параметры взаимодействия с помощью характеристики `ms#`, которая обсуждается в разделе Конфигурирование параметров взаимодействия для спулера.
5. Установите фильтр для обычного текста; подробнее об этом см. в разделе Установка текстового фильтра.
6. Проверьте настройку, напечатав что-нибудь с помощью команды lpr(1). Подробнее об этом см. в разделах Проверка и Выявление проблем.

Принтеры, использующие специальные языки, например, PostScript®-принтеры, не могут непосредственно печатать обычный текст. Простая настройка, представленная выше и описанная в следующих разделах, предполагает, что, если вы устанавливаете такой принтер, то будете

печатать только файлы, которые он может обработать.

Пользователи часто предполагают, что они могут печатать обычный текст на любом из установленных в системе принтеров. Программы, взаимодействующие для обеспечения печати с системой LPD, обычно исходят из этого же предположения. Если вы устанавливаете такой принтер и хотите иметь возможность посылать на печать задания на языке принтера *и* в виде обычного текста, настоятельно рекомендуется добавить дополнительный шаг к представленной выше простой последовательности настройки: установите программу автоматического преобразования обычного текста в PostScript® (или другой язык принтера). В разделе Прием заданий с обычным текстом на PostScript®-принтеры рассказано, как это сделать.

9.3.1.5.1. Именование принтера

Первый (простой) шаг - выбрать имя для принтера. На самом деле, не важно, выберете ли вы функциональное имя или причудливое, поскольку для принтера можно также задать несколько псевдонимов.

По крайней мере, один из принтеров, указанных в файле /etc/printcap, должен иметь псевдоним `lp`. Это - стандартное имя принтера. Если пользователи не установят переменную среды `PRINTER` и не укажут имя принтера в командной сроке при вводе любой команды системы LPD, по умолчанию для ее выполнения будет использован принтер `lp`.

Также широко распространена практика в качестве последнего псевдонима для принтера задавать полное его описание, включая производителя и модель.

После выбора имени и нескольких популярных псевдонимов поместите их в файл /etc/printcap. Имя принтера должно начинаться с крайнего левого столбца. Каждый псевдоним отделяйте вертикальной чертой, а после последнего псевдонима поместите двоеточие.

В следующем примере мы начнем со скелетного файла /etc/printcap, определяющего два принтера (построчный принтер Diablo 630 и лазерный PostScript®-принтер Panasonic KX-P4455):

```
#
#  /etc/printcap для хоста rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

В этом примере первый принтер назван `rattan` и ему заданы псевдонимы `line`, `diablo`, `lp` и `Diablo 630 Line Printer`. Поскольку у него есть псевдоним `lp`, он является стандартным принтером. Второму принтеру дано имя `bamboo` и ему заданы псевдонимы `ps`, `PS`, `S`, `panasonic` и `Panasonic KX-P4455 PostScript v51.4`.

9.3.1.5.2. Подавление выдачи начальных страниц

Система спулинга LPD будет по умолчанию печатать *заголовочную страницу* для каждого задания. Заголовочная страница содержит имя пользователя, отправившего задание, хост, с которого поступило задание, и имя задания, красивыми большими буквами. К сожалению, все эти дополнительные тексты мешают отладке простой настройки принтера, поэтому мы будет отключать выдачу начальных страниц.

Для подавления выдачи начальных страниц добавьте характеристику `sh` к записи принтера в файле /etc/printcap. Вот пример файла /etc/printcap с добавлением `sh`:

```
#
#  /etc/printcap для хоста rose - никаких начальных страниц
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:
```

Обратите внимание, как мы использовали правильный формат: первая строка начинается с самого левого столбца, а последующие строки смещены. Каждая строка в записи, кроме последней, завершается символом обратной косой черты.

9.3.1.5.3. Создание каталога для спулинга

Следующий шаг в простой настройке спулера - создать *каталог для спулинга*, каталог, в котором находятся задания печати, пока не будут напечатаны, и где находятся еще несколько других файлов для поддержки спулера.

Из-за присущих каталогам спулинга постоянных изменений, принято помещать эти каталоги в каталог /var/spool. Кроме того, не нужно создавать резервные копии содержимого каталогов спулинга. Пересоздать их можно с помощью простой команды mkdir(1).

Принято также задавать для каталога имя, совпадающее с именем принтера, как показано ниже:

```
# mkdir /var/spool/имя-принтера
```

Однако при наличии большого количества принтеров в сети может иметь смысл поместить все каталоги спулинга в один каталог, который просто резервируется для печати с помощью LPD. Мы сделаем это для наших двух принтеров, `rattan` и `bamboo`:

```
# mkdir /var/spool/lpd
# mkdir /var/spool/lpd/rattan
# mkdir /var/spool/lpd/bamboo
```

Если вас интересует конфиденциальность заданий, отправляемых пользователями на печать, можно защитить каталог спулинга, чтобы он не был общедоступным. Каталоги спулинга должны принадлежать и быть доступны на чтение, запись и просмотр содержимого пользователю daemon и группе daemon, и никому больше. Мы установим это для каталогов спулинга принтеров из нашего примера:

```
# chown daemon:daemon /var/spool/lpd/rattan
# chown daemon:daemon /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan
# chmod 770 /var/spool/lpd/bamboo
```

Наконец, надо сообщить системе LPD об этих каталогах с помощью файла /etc/printcap. Полное имя каталога спулинга задается с помощью характеристики `sd`:

```
#
#  /etc/printcap для хоста rose - добавлены каталоги спулинга
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:
```

Обратите внимание, что имя принтера начинается с самого первого столбца, а все последующие строки смещены, и каждая строка в записи, кроме последней, завершается символом обратной косой черты.

Если вы не зададите каталог спулинга с помощью характеристики `sd`, система спулинга будет использовать по умолчанию каталог /var/spool/lpd.

9.3.1.5.4. Выбор устройства для принтера

Мы выяснили, какой специальный файл устройства в каталоге /dev FreeBSD будет использовать для взаимодействия с принтером. Теперь мы сообщаем эту информацию системе LPD. Когда у системы спулинга есть задание для печати, она будет открывать указанное устройство от имени программы-фильтра (которая отвечает за передачу данных на принтер).

Задайте полное имя устройства /dev в файле /etc/printcap с помощью характеристики `lp`.

В нашем текущем примере давайте предположим, что принтер `rattan` подключен к первому параллельному порту, а принтер `bamboo` - к шестому последовательному порту; вот что нужно добавить в файл /etc/printcap:

```
#
#  /etc/printcap для хоста rose - указано, какие устройства использовать
```

```
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:
```

Если вы не укажете характеристику `lp` для принтера в файле /etc/printcap, система LPD использует по умолчанию устройство /dev/lp. Устройство /dev/lp сейчас в ОС FreeBSD не существует.

Если устанавливаемый принтер подключен к параллельному порту, перейдите к разделу Установка текстового фильтра. Иначе выполните сначала инструкции, представленные в следующем разделе.

9.3.1.5.5. Конфигурирование параметров взаимодействия спулера

Для принтеров на последовательных портах система LPD может устанавливать скорость передачи, четность и другие параметры взаимодействия через последовательных порт от имени программы-фильтра, которая посылает данные на принтер. Это полезно потому, что:

- Позволяет опробовать различные параметры взаимодействия, просто редактируя файл /etc/printcap; программу-фильтр перекомпилировать не нужно.
- Позволяет системе спулинга использовать одну и ту же программу-фильтр для нескольких принтеров, которые могут иметь различные установки для взаимодействия через последовательный порт.

Следующие характеристики в файле /etc/printcap задают параметры взаимодействия через последовательный порт для устройства, указанного в качестве значения характеристики `lp`:

`br#bps-rate`

Устанавливает скорость взаимодействия для устройства в *bps-rate*, где *bps-rate* может иметь значение 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600 или 115200 бит в секунду.

`ms#stty-mode`

Устанавливает опции для терминального устройства после открытия устройства. Поддерживаемые опции описаны на странице справочного руководства stty(1).

Когда система LPD открывает устройство, заданное характеристикой `lp`, она устанавливает опции устройства в соответствии со значением характеристики `ms#`. Наибольший интерес представляют режимы `parenb`, `parodd`, `cs5`, `cs6`, `cs7`, `cs8`, `cstopb`, `crtscts` и `ixon`, которые описаны на странице справочного руководства stty(1).

Давайте зададим опции для нашего принтера на шестом последовательном порту. Мы установим скорость передачи 38400. В качестве режима установим режим без четности с помощью `-parenb`, 8-битовые символы с помощью `cs8`, отсутствие модемного управления с

помощью clocal и аппаратное управление потоком с помощью опции crtscts:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:
```

9.3.1.5.6. Установка текстового фильтра

Теперь мы готовы задать системе LPD, какой текстовый фильтр использовать для посылки заданий на принтер. *Текстовый фильтр*, известный также как *входной фильтр*, - это программа, которую система LPD запускает при получении задания на печать. Когда система LPD запускает текстовый фильтр для принтера, она направляет на стандартный входной поток фильтра задание печати, а его стандартный выходной поток - на устройство принтера, заданное характеристикой lp. Предполагается, что фильтр прочитает задание из стандартного входного потока, выполнит все необходимые для принтера преобразования и выдаст результат в стандартный выходной поток, который и будет напечатан. Подробнее о текстовом фильтре см. в разделе Фильтры.

Для простой настройки принтера в качестве текстового фильтра можно задать небольшой скрипт командного интерпретатора, который просто выполняет /bin/cat для посылки задания на принтер. В составе FreeBSD поставляется другой фильтр, lpf, обрабатывающий забой и подчеркивание для принтеров, которые не слишком хорошо справляются с потоком данных, содержащих такие символы. И, конечно же, вы можете использовать любую другую необходимую программу-фильтр. Фильтр lpf детально описан в разделе lpf: текстовый фильтр.

Сначала давайте создадим скрипт командного интерпретатора /usr/local/libexec/if-simple для простого тестового фильтра. Поместите в этот файл следующий текст с помощью любимого текстового редактора:

```
#!/bin/sh
#
# if-simple - Простой фильтр входного текста для lpd
# Установлен в /usr/local/libexec/if-simple
#
# Просто копирует stdin в stdout. Игнорирует все аргументы фильтра.

/bin/cat && exit 0
exit 2
```

Сделайте этот файл выполняемым:

```
# chmod 555 /usr/local/libexec/if-simple
```

А теперь потребуйте от системы LPD его использовать, указав его в качестве значения характеристики if в файле /etc/printcap. Мы добавим его для двух принтеров, имеющихся

пока в примере файла /etc/printcap:

```
#
#  /etc/printcap для хоста rose - добавлен текстовый фильтр
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\ :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:\
        :if=/usr/local/libexec/if-simple:
```

Копию скрипта if-simple можно найти в каталоге /usr/shared/examples/printing.

9.3.1.5.7. Запуск системы LPD

Даемон lpd(8) запускается из /etc/rc, а необходимость запуска задается переменной `lpd_enable`. Эта переменная по умолчанию имеет значение `NO`. Если вы еще этого не сделали, добавьте строку:

```
lpd_enable="YES"
```

в файл /etc/rc.conf, а затем либо перезапустите машину, либо просто выполните команду lpd(8).

```
# lpd
```

9.3.1.5.8. Проверка

Вы добрались до конца простой настройки системы LPD. К сожалению, поздравлять вас еще рано, поскольку надо еще проверить настройку и устранить все выявленные проблемы. Для проверки настройки, попытайтесь что-то распечатать. Для печати с помощью системы LPD используется команда lpr(1), которая посылает задание на печать.

Можно скомбинировать lpr(1) с программой lptest(1), представленной в разделе Проверка взаимодействия с принтером, генерирующей тестовый текст.

Для тестирования простой настройки LPD:

Введите команду:

```
# lptest 20 5 | lpr -Pprinter-name
```

Где *printer-name* - имя (или псевдоним) принтера, заданное в файле /etc/printcap. Для проверки стандартного принтера, введите команду lpr(1) без аргумента `-P`. Как уже отмечалось, если тестируется принтер, предполагающий использование PostScript®, пошлите ему PostScript®-программу вместо использования утилиты lptest(1). Это можно сделать, поместив программу в файл и выполнив команду `lpr file`.

Для PostScript®-принтера вы должны получить результаты выполнения программы. Если вы используете lptest(1), ваши результаты должны иметь такой вид:

```
!"#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

Для дальнейшего тестирования принтера, попытайтесь загрузить программы побольше (для принтеров, поддерживающих определенный язык) или выполните команду lptest(1) с другими аргументами. Например, команда `lptest 80 60` выдаст 60 строк по 80 символов в каждой.

Если принтер не работает, см. раздел Выявление проблем.

9.4. Расширенная настройка принтера

В этом разделе описаны фильтры для печати специально сформатированных файлов, начальных страниц, печати по сети, ограничения и учета использования принтера.

9.4.1. Фильтры

Хотя система LPD поддерживает сетевые протоколы, очереди, контроль доступа и другие аспекты печати, большая часть *реальной* работы происходит в *фильтрах*. Фильтры - это программы, взаимодействующие с принтером и обеспечивающие учет особенностей устройства и специальных требований. При простой настройке принтера мы установили фильтр для обычного текста - крайне простой, который должен работать с большинством принтеров (см. раздел Установка текстового фильтра).

Однако, чтобы обеспечить преобразования формата, учет использования принтера и индивидуальных особенностей отдельных принтеров и т.п., надо разобраться, как работают фильтры. В конечном итоге, всеми этими аспектами печати должен заниматься фильтр. А плохая новость состоит в том, что, в большинстве случаев, вы *сами* должны предоставить соответствующие фильтры. Хорошая новость состоит в том, что многие фильтры общедоступны; а если подходящих нет, их обычно легко написать.

Кроме того, в составе ОС FreeBSD поставляется один фильтр, /usr/libexec/lpr/lpf, работающий со многими принтерами, которые могут печатать обычный текст. (Он обрабатывает символы забоя и табуляции в файле, выполняет учет использования, но и не более того.) Есть также ряд фильтров и компонентов фильтров в наборе портов FreeBSD.

Вот что вы найдете в этом разделе:

- В разделе Как работают фильтры сделана попытка дать обзор роли фильтра в процессе печати. Прочтите этот раздел, чтобы понять, что происходит "за кадром", когда система LPD использует фильтры. Это понимание поможет предвидеть и решать проблемы, с которыми вы можете столкнуться при добавлении дополнительных фильтров для каждого из принтеров.
- Система LPD предполагает, что каждый принтер, по умолчанию, может печатать обычный текст. Это проблематично для PostScript®-принтеров (или принтеров на базе другого языка), поскольку они не могут печатать обычный текст непосредственно. В разделе Прием заданий с обычным текстом на PostScript®-принтеры описано, что нужно сделать, чтобы решить эту проблему. Прочтите этот раздел, если используете PostScript®-принтер.
- PostScript® - популярный формат выдачи для многих программ. Некоторые люди даже пишут PostScript®-код непосредственно. К сожалению, PostScript®-принтеры дороги. В разделе Имитация PostScript® на не-PostScript® принтерах описано, как можно дополнительно изменить текстовый фильтр принтера для приема и печати данных PostScript® не *не-PostScript®* принтере. Прочтите этот раздел, если ваш принтер не поддерживает PostScript®.
- В разделе Фильтры преобразования описан способ автоматизации преобразования определенных форматов файлов, например, графики или данных для печатного станка, в форматы, которые может обработать ваш принтер. После чтения этого раздела вы сможете настроить свои принтеры так, что пользователи смогут выполнять команду `lpr -t` для печати данных troff, или `lpr -d` для печати данных TeX DVI, или `lpr -v` - для печати растровых изображений, и так далее. Я рекомендую прочитать этот раздел.
- В разделе Выходные фильтры описана не часто используемая возможность задавать выходные фильтры в системе LPD. Если только вы не печатаете начальные страницы (см. Начальные страницы), можно, пожалуй, вообще пропустить этот раздел.
- В разделе lpf: текстовый фильтр описана команда `lpf`, - достаточно полный, хотя и простой текстовый фильтр для строчных принтеров (и лазерных принтеров, работающих как строчные), поставляемый в составе ОС FreeBSD. Если надо быстро настроить учет использования принтера для обычного текста или если используется принтер, из которого при получении символов забоя идет дым, несомненно, стоит подумать об использовании `lpf`.

Различные скрипты, описанные далее, можно найти в каталоге /usr/shared/examples/printing.

9.4.1.1. Как работают фильтры

Как уже упоминалось, фильтр - это выполняемая программа, запускаемая системой LPD для поддержки специфических особенностей устройства при взаимодействии с принтером.

Когда системе LPD надо напечатать входящий в задание файл, она запускает программу-фильтр. Стандартный входной поток фильтра связывается с файлом, который надо распечатать, его стандартный выходной поток - с принтером, а стандартный поток ошибок

перенаправляется в файл регистрации ошибок (задается характеристикой `lf` в файле /etc/printcap, или используется стандартное устройство /dev/console).

Запускаемый системой LPD фильтр и его аргументы зависят от того, что указано в файле /etc/printcap, и какие аргументы указал пользователь для задания в команде lpr(1). Например, если пользователь ввел команду `lpr -t`, система LPD должна запустить фильтр troff, заданный характеристикой `tf` для соответствующего принтера. Если пользователь хочет печатать обычный текст, система должна запустить фильтр `if` (это верно в большинстве случаев: подробнее см. в разделе Выходные фильтры).

В файле /etc/printcap можно задавать три вида фильтров:

- *Текстовый фильтр*, который в документации LPD двусмысленно называют *входным фильтром*, обеспечивает печать обычного текста. Рассматривайте его как стандартный фильтр. Система LPD предполагает, что любой принтер может по умолчанию печатать обычный текст, а на текстовый фильтр возлагается задача обеспечить, чтобы символы забоя, табуляции или другие специальные символы не сбивали принтер с толку. Если вы работаете в среде, где надо учитывать использование принтера, текстовый фильтр должен также учитывать количество напечатанных страниц, обычно, подсчитывая количество напечатанных строк и сравнивая их с количеством строк на страницу, поддерживаемых принтером. Текстовый фильтр запускается со следующим списком аргументов:

  ```
  имя-фильтра [-c] -wширина -lдлина -iсдвиг -n имя-пользователя -h хост учетный-файл
  ```

  `-c`

  указывается, если задание послано командой `lpr -l`

  ширина

  значение из характеристики `pw` (page width - ширина страницы), указанной в файле /etc/printcap, по умолчанию - 132

  длина

  значение из характеристики `pl` (page length - длина страницы), по умолчанию - 66

  сдвиг

  сдвиг, заданный командой `lpr -i`, по умолчанию - 0

  имя-пользователя

  регистрационное имя пользователя, печатающего файл

  хост

  имя хоста, с которого было послано задание

  учетный-файл

  имя учетного файла, задаваемое характеристикой `af`.

- *Фильтр преобразования* преобразует специфичный формат файла в то, что принтер может воспроизвести на бумаге. Например, данные системы набора ditroff нельзя

печатать непосредственно, но можно установить фильтр преобразования для файлов ditroff, чтобы преобразовывать данные ditroff в тот вид, который принтер может воспринять и напечатать. В разделе Фильтры преобразования написано всё об этих фильтрах. Фильтры преобразования также необходимы для учета, если предполагается учет использования принтера. Фильтры преобразования запускаются со следующими аргументами:

`имя-фильтра` -x*ширина-пиксела* -y*высота-пиксела* -n *имя-пользователя* -h *хост учетный-файл* где *ширина-пиксела* - значение характеристики px (по умолчанию - 0), а *высота-пиксела* - значение характеристики py (по умолчанию - 0).

- *Выходной фильтр* используется только если нет текстового фильтра или если включена выдача начальных страниц. Судя по моему опыту, выходные фильтры используются редко. Они описаны в разделе Выходные фильтры. У выходного фильтра есть всего два аргумента:

  `имя-фильтра` -w*ширина* -l*длина* которые идентичны аргументам `-w` и `-l` текстового фильтра.

Фильтры также должны *завершать работу* со следующим статусом выхода:

exit 0

Если фильтр успешно напечатал файл.

exit 1

Если фильтр не смог напечатать файл, но хочет, чтобы система LPD попыталась распечатать файл ещё раз. Система LPD перезапустит фильтр, если его работа завершена с этим статусом.

exit 2

Если фильтр не смог напечатать файл и не хочет, чтобы система LPD пыталась его печатать еще раз. Система LPD удалит файл.

Поставляемый в составе FreeBSD текстовый фильтр /usr/libexec/lpr/lpf использует аргументы, задающие ширину и длину страницы для определения того, когда посылать символ прогона страницы (form feed) и как учитывать использование принтера. Он использует переданные в качестве аргументов имя пользователя, хост и учетный файл для внесения учетных записей.

При поиске фильтров убедитесь, что они совместимы с системой LPD. Если да, они должны поддерживать описанные выше списки аргументов. Если вы планируете создавать фильтры для общего использования, позаботьтесь о поддержке этих списков аргументов и кодов выхода.

9.4.1.2. Прием заданий с обычным текстом на PostScript®-принтеры

Если вы - единственный пользователь компьютера и PostScript®-принтера (или принтера на основе другого языка), и вы обещаете никогда не посылать на принтер обычный текст и никогда не использовать возможностей различных программ, требующих посылки на

принтер обычного текста, вам можно не заботиться о том, что описано в этом разделе.

Но, если вы хотите посылать на принтер как задания PostScript®, так и обычный текст, рекомендуется дополнить настройку принтера. Для этого надо, чтобы текстовый фильтр определял, является ли поступающее задание обычным текстом или программой на языке PostScript®. Все PostScript®-задания должны начинаться с `%!` (для других языков принтеров обратитесь к соответствующей документации). Если первые два символа в задании - именно эти, речь идет о PostScript®, и мы можем остальную часть задания передавать непосредственно. Если же первые два символа в файле - другие, фильтр будет преобразовывать текст в PostScript® и печатать результат.

Как нам это сделать?

Если вы используете последовательный принтер, хороший способ достичь поставленной цели состоит в установке `lprps`. `lprps` - это фильтр для PostScript®-принтера, выполняющий двустороннее взаимодействие с принтером. Он обновляет файл состояния принтера, помещая в него подробную информацию, выданную принтером, так что пользователи и администраторы могут узнать, в каком именно состоянии (например, `toner low` или `paper jam`) находится принтер. Но еще важнее, что он включает программу `psif`, которая определяет, является ли входящее задание обычным текстом, и вызывает `textps` (еще одну программу, поставляемую вместе с `lprps`) для преобразования его в PostScript®. Затем `lprps` посылает преобразованное задание на принтер.

`lprps` входит в набор портов FreeBSD (см. Набор портов). Вы, конечно, можете загрузить, собрать и установить его самостоятельно. После установки `lprps` просто укажите путь к программе `psif`, входящей в состав пакета `lprps`. Если вы установили `lprps` из Коллекции Портов, используйте следующий текст в записи для последовательного PostScript®-принтера в файле /etc/printcap:

```
:if=/usr/local/libexec/psif:
```

Надо также задать характеристику `rw`; она требует от системы LPD открывать принтер в режиме чтения и записи.

При использовании параллельного PostScript®-принтера (что не позволяет обеспечить двустороннее взаимодействие с принтером, необходимое для системы `lprps`), можно использовать в качестве текстового фильтра следующий скрипт командного интерпретатора:

```
#!/bin/sh
#
#  psif - Печать PostScript или обычного текста на PostScript-принтере
#  Скрипт, а НЕ версия, входящая в состав lprps
#  Установлен в /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`
```

```
if [ "$first_two_chars" = "%!" ]; then
    #
    #  Задание PostScript, печатать его.
    #
    echo "$first_line" && cat && printf "\004" && exit 0
    exit 2
else
    #
    #  Обычный текст, преобразовать его, а затем напечатать.
    #
    ( echo "$first_line"; cat ) | /usr/local/bin/textps && printf "\004" && exit 0
    exit 2
fi
```

В представленном выше скрипте, `textps` - отдельно установленная программа для преобразования обычного текста в PostScript®. Можно использовать любую программу преобразования текста в PostScript®. Коллекция Портов FreeBSD (см. материал о Коллекции Портов) включает полнофункциональную программу преобразования текста в PostScript® под названием `a2ps`, которую тоже можно попробовать использовать.

9.4.1.3. Имитация PostScript® на не-PostScript® принтерах

PostScript® является *фактическим* стандартом для высококачественного набора и печати. PostScript®, однако, - *дорогой* стандарт. К счастью, благодаря компании Aladdin Enterprises есть свободный аналог PostScript® под названием Ghostscript, который работает с FreeBSD. Ghostscript может читать большинство PostScript®-файлов и выдавать соответствующие страницы на множество устройств, включая многие моделей не-PostScript принтеров. Установив Ghostscript и используя специальный текстовый фильтр для принтера, можно заставить ваш не-PostScript® принтер работать фактически как PostScript®-принтер.

Ghostscript входит в набор портов FreeBSD, если вы хотите устанавливать его оттуда. Вы можете также легко загрузить, собрать и установить его самостоятельно.

Для имитации PostScript® надо, чтобы текстовый фильтр определял, печатается ли PostScript®-файл. Если нет, фильтр будет передавать файл на принтер непосредственно; в противном случае, он будет использовать Ghostscript, чтобы сначала преобразовать файл в формат, который поймет принтер.

Рассмотрим пример: следующий сценарий представляет собой текстовый фильтр для принтеров Hewlett Packard DeskJet 500. Для других принтеров замените аргумент `-sDEVICE` в команде `gs` (Ghostscript). (Введите команду `gs -h` для получения списка устройств, поддерживаемых установленной версией Ghostscript.)

```
#!/bin/sh
#
#  ifhp - Печать Ghostscript-эмулированного PostScript на DeskJet 500
#  Установлен в /usr/local/libexec/ifhp
```

```
#
#  Обрабатывать LF как CR+LF (чтобы избежать "эффекта ступенек"
#  на принтерах HP/PCL:
#
printf "\033&k2G" || exit 2

#
#  Прочитать первые два символа файла
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if [ "$first_two_chars" = "%!" ]; then
    #
    #  Это PostScript; используем Ghostscript для чтения, преобразования и печати.
    #
    /usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
        -sOutputFile=- - && exit 0
else
    #
    #  Обычный текст или HP/PCL, поэтому просто печатаем его напрямую; печатаем в
    #  конце символ прогона страницы, чтобы была выдана последняя страница.
    #
    echo "$first_line" && cat && printf "\033&l0H" &&
exit 0
fi

exit 2
```

Наконец, надо указать системе LPD, какой фильтр использовать, задав характеристику `if`:

```
:if=/usr/local/libexec/ifhp:
```

Вот и все. Теперь можно выполнять `lpr plain.text` и lpr whatever.ps, и обе команды должны успешно печатать.

9.4.1.4. Фильтры преобразования

После завершения простой настройки, описанной в разделе Простая настройка принтера, прежде всего, вам может потребоваться установить фильтры преобразования для любимых форматов файлов (кроме обычных текстов ASCII).

9.4.1.4.1. Зачем устанавливать фильтры преобразования?

Фильтры преобразования упрощают печать различного рода файлов. В качестве примера, предположим, что активно используется издательская система TeX и имеется PostScript®-принтер. При каждой генерации DVI-файла из TeX, мы не можем печатать его непосредственно, пока не преобразуем в PostScript®. Для этого используется такая последовательность команд:

```
% dvips seaweed-analysis.dvi
% lpr seaweed-analysis.ps
```

Установив фильтр преобразования для файлов DVI, мы можем не конвертировать файл каждый раз вручную, возложив эту задачу на систему LPD. Теперь при каждом получении DVI-файла нас от его распечатки отделяет только один шаг:

```
% lpr -d seaweed-analysis.dvi
```

Мы заставили систему LPD автоматически преобразовывать DVI-файл, указав опцию `-d`. Все опции преобразования представлены в разделе Опции форматирования и преобразования.

Для каждой из опций преобразования, которая должна поддерживаться принтером, установите *фильтр преобразования* и укажите его полное имя в файле /etc/printcap. Фильтр преобразования аналогичен текстовому фильтру для простой настройки принтера (см. раздел Установка текстового фильтра), но вместо печати обычного текста он преобразует файл в формат, который может понять принтер.

9.4.1.4.2. Какие фильтры преобразования следует устанавливать?

Устанавливать надо те фильтры преобразования, которые предполагается использовать. Если вы часто печатаете файлы DVI, значит, фильтр преобразования DVI необходим. Если вам часто приходится печатать результаты работы troff, может потребоваться фильтр troff.

В следующей таблице представлены фильтры, с которыми работает система LPD, их соответствующие характеристики для файла /etc/printcap, а также способ их вызова в команде `lpr`:

Тип файла	Характеристика /etc/printcap	Опция lpr
cifplot	`cf`	`-c`
DVI	`df`	`-d`
plot	`gf`	`-g`
ditroff	`nf`	`-n`
Текст на языке FORTRAN	`rf`	`-f`
troff	`tf`	`-f`
растровое изображение	`vf`	`-v`
обычный текст	`if`	никакой, `-p` или `-l`

В нашем примере использование `lpr -d` означает, что для принтера должна быть задана характеристика `df` в записи в файле /etc/printcap.

Вопреки мнению многих, форматы вроде текста на языке FORTRAN и plot, вероятно, устарели. У себя на машине вы можете дать новые значения этим или любым другим

опциям форматирования, установив соответствующие специализированные фильтры. Например, пусть необходимо напрямую печатать файлы Printerleaf (файлы настольной издательской системы Interleaf), но вообще вы не собираетесь печатать файлы типа plot. Можно установить фильтр преобразования Printerleaf в качестве значения характеристики `gf` и научить своих пользователей, что команда `lpr -g` означает "печатать файлы Printerleaf".

9.4.1.4.3. Установка фильтров преобразования

Поскольку фильтры преобразования представляют собой программы, не входящие в базовую поставку FreeBSD, их, видимо, надо помещать в каталоге /usr/local. Популярное местонахождение - каталог /usr/local/libexec, поскольку эти фильтры являются специализированными программами для выполнения системой LPD; обычным пользователям никогда не понадобится их выполнять.

Для включения фильтра преобразования, укажите его полное имя в качестве значения соответствующей характеристики для принтера в файле /etc/printcap.

В качестве примера, давайте добавим фильтр преобразования DVI в запись для принтера `bamboo`. Вот опять пример файла /etc/printcap, с новой характеристикой `df` для принтера `bamboo`.

```
#
#  /etc/printcap для хоста rose - добавлен фильтр df для bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Фильтр DVI - скрипт командного интерпретатора по имени /usr/local/libexec/psdf. Вот его текст:

```
#!/bin/sh
#
#  psdf - фильтр принтера, преобразующий DVI в PostScript
#  Установлен в /usr/local/libexec/psdf
#
# Вызывается системой lpd при выполнении пользователем команды lpr -d
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

Это скрипт выполняет команду dvips в режиме фильтрования (аргумент -f) входного потока, представляющего собой задание для печати. Затем запускается фильтр PostScript®-принтера lprps (см. раздел Прием заданий с обычным текстом на PostScript®-принтеры) с аргументами, переданными системой LPD этому скрипту. Команда lprps будет использовать эти аргументы для учета распечатанных страниц.

9.4.1.4.4. Дополнительные примеры фильтров преобразования

Поскольку нет фиксированного набора шагов для установки фильтров преобразования, я просто представлю дополнительные примеры. Используйте их в качестве руководства при создании собственных фильтров. Используйте их непосредственно, если нужно.

Следующий пример фильтра преобразует растровый файл (точнее, GIF-файл) для печати на принтере Hewlett Packard LaserJet III-Si:

```
#!/bin/sh
#
#  hpvf - Преобразовать GIF-файлы в HP/PCL и напечатать
#  Установлен в /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH
giftopnm | ppmtopgm | pgmtopbm | pbmtolj -resolution 300 \
    && exit 0 \
    || exit 2
```

Он работает путем преобразования GIF-файла в переносимый формат anymap, его - в переносимый формат graymap, затем - в переносимый bitmap, а уже его - в данные, подходящие для LaserJet/PCL.

Вот файл /etc/printcap с записью для принтера, в которой используется представленный выше фильтр:

```
#
#  /etc/printcap для хоста orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:
```

Следующий скрипт является фильтром преобразования для печати данных troff, получаемых из системы набора groff, на PostScript®-принтере bamboo:

```
#!/bin/sh
#
#  pstf - Преобразует выдаваемые groff данные troff в PS и печатает.
#  Установлен в /usr/local/libexec/pstf
#
```

```
exec grops | /usr/local/libexec/lprps "$@"
```

Представленный выше скрипт снова использует команду lprps для взаимодействия с принтером. Если принтер подключен к параллельному порту, придется использовать следующий скрипт:

```
#!/bin/sh
#
#  pstf - Преобразует выдаваемые groff данные troff в PS и печатает.
#  Установлен в /usr/local/libexec/pstf
#
exec grops
```

Вот и все. Вот какую запись надо добавить в файл /etc/printcap, чтобы включить этот фильтр:

```
:tf=/usr/local/libexec/pstf:
```

Вот пример, который пригодится старым специалистам по языку FORTRAN. Это фильтр для печати текста программы на языке FORTRAN на любом принтере, который может непосредственно печатать обычный текст. Мы установим его для принтера teak:

```
#!/bin/sh
#
# hprf - Фильтр текста на языке FORTRAN для LaserJet 3si:
# Установлен в /usr/local/libexec/hprf
#

printf "\033&k2G" && fpr && printf "\033&l0H" &&
 exit 0
exit 2
```

Надо добавить следующую строку к записи в файле /etc/printcap для принтера teak, чтобы включить этот фильтр:

```
:rf=/usr/local/libexec/hprf:
```

Перейдем к последнему, более сложному примеру. Мы добавим фильтр DVI для уже использовавшегося принтера LaserJet по имени teak. Сначала простая часть: изменить файл /etc/printcap, указав местонахождение фильтра DVI:

```
:df=/usr/local/libexec/hpdf:
```

А теперь - часть посложнее: создать фильтр. Для этого нам понадобится программа

преобразования DVI в LaserJet/PCL. Набор портов FreeBSD (см. Набор портов) содержит одну: соответствующий пакет называется dvi2xx. Установка этого пакета дает нам необходимую программу, dvilj2p, которая преобразует DVI в коды, подходящие для LaserJet IIp, LaserJet III и LaserJet 2000.

Команда dvilj2p требует создания достаточно сложного фильтра hpdf, поскольку она не может читать стандартный входной поток. Она хочет работать с именем файла. Что еще хуже, имя файла должно завершаться расширением .dvi, так что использование стандартного входного потока /dev/fd/0 тоже проблематично. Мы можем обойти эту проблему, создав (символическую) связь (с именем, завершающимся суффиксом .dvi) с устройством /dev/fd/0, тем самым, заставив команду dvilj2p читать из стандартного входного потока.

Единственная оставшаяся проблема состоит в том, что мы не можем создавать временную связь в каталоге /tmp. Символьные связи принадлежат пользователю и группе bin. Фильтр же работает от имени пользователя daemon. А у каталога /tmp установлен sticky bit. Фильтр сможет создать связь, но не сможет почистить за собой и удалить ее, поскольку связь будет принадлежать другому пользователю.

Вместо этого, фильтр будет создавать символическую связь в текущем рабочем каталоге, которым является каталог спулинга (задаваемый характеристикой sd в файле /etc/printcap). Это отличное место для выполнения фильтрами своих действий, особенно потому, что (иногда) в каталоге спулинга места больше, чем в /tmp.

Вот, наконец, и сам фильтр:

```
#!/bin/sh
#
#  hpdf - Печать данных DVI на принтере HP/PCL
#  Установлен в /usr/local/libexec/hpdf

PATH=/usr/local/bin:$PATH; export PATH

#
#  Определяем функцию для удаления временных файлов. Они существуют
#  в текущем каталоге - в каталоге спулинга для принтера.
#
cleanup() {
   rm -f hpdf$$.dvi
}

#
#  Определяем функцию для обработки критических ошибок: напечатать заданное
#  сообщение и выйти с кодом 2. Код выхода 2 сообщает системе LPD, что не
#  надо повторно пытаться печатать задание.
#
fatal() {
    echo "$@" 1>&2
    cleanup
    exit 2
```

```
}

#
#  Если пользователь удаляет задание, система LPD будет посылать сигнал SIGINT,
#  поэтому перехватываем SIGINT (и пару других сигналов), чтобы убрать за собой.
#
trap cleanup 1 2 15

#
#  Гарантируем, что не конфликтуем с существующими файлами.
#
cleanup

#
#  Связываем входной файл DVI со стандартным входным потоком (файлом для печати).
#
ln -s /dev/fd/0 hpdf$$.dvi || fatal "Cannot symlink /dev/fd/0"

#
#  Заменяем LF = CR+LF
#
printf "\033&k2G" || fatal "Cannot initialize printer"

#
#  Преобразуем и печатаем. Значение, возвращаемое программой dvilj2p, не надежно,
#  так что мы его игнорируем.
#
dvilj2p -M1 -q -e- dfhp$$.dvi

#
#  Убираем за собой и завершаем работу
#
cleanup
exit 0
```

9.4.1.4.5. Автоматизированное преобразование: альтернатива фильтрам преобразования

Все эти фильтры преобразования многое дают для среды печати, но требуют от пользователя указывать (в командной строке lpr(1)), какой именно фильтр использовать. Если пользователи не особенно разбираются в компьютерах, необходимость указывать опцию фильтра будет их раздражать. Что еще хуже, однако, при неправильном указании опции фильтрования может быть применен фильтр, не соответствующий типу файла, и принтер испортит несколько сотен страниц бумаги.

Вместо установки фильтров преобразования, можно попытаться заставить текстовый фильтр (поскольку он применяется по умолчанию) определять тип файла, который его попросили напечатать, и затем автоматически вызывать соответствующий фильтр преобразования. В этом могут помочь утилиты вроде `file`. Конечно, будет сложно различать *некоторые* типы файлов - и, конечно же, можно задавать фильтры преобразования только для них.

В наборе портов FreeBSD есть текстовый фильтр, выполняющий автоматическое преобразование; это `apsfilter`. Он может выявлять обычный текст, PostScript® и файлы DVI, выполнять соответствующие преобразования и печатать результат.

9.4.1.5. Выходные фильтры

Система спулинга LPD поддерживает еще один тип фильтров, который мы еще не рассматривали: выходные фильтры. Выходной фильтр предназначен только для печати обычного текста, как текстовый фильтр, но с множеством упрощений. Если вы используете выходной фильтр, а текстовый фильтр не задан, то:

- Система LPD запускает выходной фильтр один раз для всего задания, а не для каждого файла задания.
- Система LPD не пытается определить начало или конец файлов в задании для выходного фильтра.
- Система LPD не передает выходному фильтру имя пользователя или хоста, так что этот фильтр не предназначен для учета использования принтера. Фактически, он получает всего два аргумента:

  `имя-фильтра` -w*ширина* -l*длина*

  Где *ширина* берется из характеристики `pw`, а *длина* - из характеристики `pl` для соответствующего принтера.

Не соблазняйтесь простотой выходного фильтра. Если вы хотите, чтобы каждый файл в задании начинал печататься с новой страницы, выходной фильтр *не поможет*. Используйте текстовый фильтр (также известный как входной); см. раздел Установка текстового фильтра. Более того, выходной фильтр, фактически, - *более сложный*, поскольку он должен проверять посылаемый ему поток байтов в поисках специальных символов-флагов и посылать себе сигналы от имени системы LPD.

Однако выходной фильтр *необходим*, если надо выдавать начальные страницы и требуется посылать управляющие последовательности или другие строки инициализации, чтобы можно было напечатать начальную страницу. (Но он *не поможет*, если необходимо учитывать начальные страницы для пользователя, поскольку система LPD не передает выходному фильтру никакой информации о пользователе или хосте.)

На одном принтере система LPD позволяет совместно с выходным использовать текстовый или другие фильтры. В таких случаях, система LPD будет запускать выходной фильтр только для печати начальной страницы (см. раздел Начальные страницы). Система LPD затем предполагает, что выходной фильтр *остановится*, посылая ему два байта: ASCII 031 и ASCII 001. Когда выходной фильтр видит эти два байта (031, 001), он должен остановиться, посылая себе сигнал `SIGSTOP`. Когда система LPD закончит выполнение остальных фильтров, она перезапускает выходной фильтр, посылая ему сигнал `SIGCONT`.

Если есть выходной фильтр, но *нет* текстового, и система LPD обрабатывает задания с обычным текстом, LPD использует для выполнения задания выходной фильтр. Как уже было сказано, выходной фильтр будет печатать все файлы задания последовательно, без прогонов страниц или других настроек бумаги, а это *вряд ли* вас устроит. Почти во всех

случаях необходим текстовый фильтр.

Программа `lpf`, которую мы представили ранее как текстовый фильтр, может также работать как выходной фильтр. Если срочно необходим простой выходной фильтр, но вы не хотите писать код для выявления байтов и посылки сигнала, попробуйте использовать `lpf`. Можно также поместить `lpf` в скрипт командного интерпретатора для обработки любых кодов инициализации, которые может потребовать принтер.

9.4.1.6. `lpf`: текстовый фильтр

Программа /usr/libexec/lpr/lpf, поставляемая в составе двоичного дистрибутива FreeBSD, представляет собой текстовый (входной) фильтр, который может печатать с отступом (если задание послано командой `lpr -i`), пропускать все символы на печать (если задание послано командой `lpr -l`), настраивать позицию печати при получении в задании символов забоя и табуляции, а также учитывать количество напечатанных страниц. Она может также использоваться как выходной фильтр.

Программа `lpf` подходит для многих сред печати. И хотя она не позволяет посылать на принтер инициализационные последовательности, легко написать скрипт командного интерпретатора, который будет выполнять необходимую инициализацию, а затем вызывать `lpf`.

Чтобы программа `lpf` корректно выполняла учет страниц, ей необходимо указать корректные значения характеристик `pw` и `pl` в файле /etc/printcap. Она использует эти значения для определения того, сколько текста может поместиться на странице и сколько страниц было в задании пользователя. Подробнее об учете использования принтера см. в разделе Учет использования принтера.

9.4.2. Начальные страницы

При наличии *множества* пользователей, использующих различные принтеры, вероятно, можно считать *начальные страницы* неизбежным злом.

Начальные страницы, которые также называют *баннерными* или *разделительными страницами*, идентифицируют, кому принадлежат задания после их печати. Обычно информация на них выдается большими, жирными буквами, возможно, с декоративными рамочками, чтобы в пачке распечаток они отличались от реальных документов, образующих задания пользователей. Они позволяют пользователям быстро находить свои задания. Очевидный недостаток выдачи начальных страниц состоит в том, что для каждого задания надо печатать на одну страницу больше, причем, страница эта хоть сколько-нибудь нужна несколько минут, а затем она оказывается в мусорной корзине или сдается в макулатуру. (Учтите, что начальная страница выдается в начале задания, а не перед каждым файлом, так что бумаги может теряться не так уж и много.)

Система LPD может выдавать заголовочные страницы для ваших распечаток автоматически, *если* ваш принтер может непосредственно печатать обычный текст. Если используется PostScript®-принтер, потребуется внешняя программа для генерации начальной страницы; см. Начальные страницы на PostScript®-принтерах.

9.4.2.1. Включение выдачи начальных страниц

В разделе Простая настройка принтера мы отключили выдачу начальных страниц, задав характеристику sh (что означает "suppress header") в файле /etc/printcap. Для включения выдачи начальных страниц на принтер, просто удалите характеристику sh.

Кажется слишком просто, правда?

Вы правы. *Может* потребоваться задать выходной фильтр для посылки строк инициализации на принтер. Вот пример выходного фильтра для Hewlett Packard PCL-совместимых принтеров:

```
#!/bin/sh
#
#  hpof - Выходной фильтр для Hewlett Packard PCL-совместимых принтеров
#  Установлен в /usr/local/libexec/hpof

printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```

Задайте полное имя выходного фильтра в качестве значения характеристики of. Подробнее об этом см. в разделе Выходные фильтры.

Вот пример файла /etc/printcap для принтера teak, который мы представили ранее; мы включили выдачу начальных страниц и добавили показанный выше выходной фильтр:

```
#
#  /etc/printcap для хоста orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:\
        :of=/usr/local/libexec/hpof:
```

Теперь, когда пользователи выдают задания на принтер teak, они получают начальную страницу с каждым заданием. Если пользователи хотят тратить время на поиск своих распечаток, они могут подавить вывод начальных страниц, посылая задание с опцией lpr -h; другие опции lpr(1) см. в разделе Опции начальных страниц.

Система LPD выдает символ прогона страницы (form feed) после начальной страницы. Если ваш принтер использует другой символ или последовательность символов для выброса напечатанной страницы, укажите их в качестве значения характеристики ff в файле /etc/printcap.

9.4.2.2. Управление начальными страницами

Включая выдачу начальных страниц, система LPD будет выдавать длинный *длинный*

заголовок, целую страницу с большими буквами, идентифицирующими пользователя, хост и задание. Ниже представлен пример (kelly напечатала задание по имени outline с хоста `rose`):

```
k                   ll       ll
k                    l        l
k                    l        l
k   k     eeee       l        l     y    y
k  k     e    e      l        l     y    y
k k      eeeeee      l        l     y    y
kk k     e           l        l     y    y
k   k    e    e      l        l     y   yy
k    k    eeee      lll      lll     yyy y
                                          y
                                       y    y
                                        yyyy

                             ll
                    t         l        i
                    t         l
 oooo    u    u   ttttt       l       ii     n nnn     eeee
o    o   u    u     t         l        i     nn   n   e    e
o    o   u    u     t         l        i     n    n   eeeeee
o    o   u    u     t         l        i     n    n   e
o    o   u   uu     t  t      l        i     n    n   e    e
 oooo     uuu u      tt      lll      iii    n    n    eeee

r rrr     oooo     ssss     eeee
rr   r   o    o   s    s   e    e
r        o    o    ss      eeeeee
r        o    o      ss    e
r        o    o   s    s   e    e
r         oooo     ssss     eeee

                                        Job:  outline
                                        Date: Sun Sep 17 11:04:58 1995
```

Система LPD добавляет прогон страницы после этого текста, чтобы задание начиналось с новой страницы (если только вы не указали характеристику `sf` (suppress form feeds) в записи соответствующего принтера в файле /etc/printcap).

Если вы предпочитаете, чтобы система LPD создавала *короткий заголовок*, укажите характеристику `sb` (short banner) в файле /etc/printcap. Начальная страница будет иметь следующий вид:

```
rose:kelly  Job: outline  Date: Sun Sep 17 11:07:51 1995
```

Также по умолчанию система LPD печатает начальную страницу перед заданием. Для

изменения порядка на обратный, укажите характеристику `hl` (header last) в файле /etc/printcap.

9.4.2.3. Учет начальных страниц

Использование встроенных начальных страниц системы LPD порождает определенную парадигму учета использования принтера: начальные страницы пользователи *не должны оплачивать*.

Почему?

Поскольку выходной фильтр - единственная внешняя программа, управляющая выдачей начальных страниц, которая может выполнять учет, а ей не передают информацию о *пользователе или хосте* и учётный файл, так что, она не имеет никакого представления о том, на чей счет отнести использование принтера. Также недостаточно просто "добавлять одну страницу" в текстовом фильтре или в любом из фильтров преобразований (которые имеют информацию о пользователе и хосте), поскольку пользователи могут подавлять выдачу начальных страниц с помощью опции `lpr -h`. И их заставят оплачивать начальные страницы, которые они не печатали. Понятно, что опцию `lpr -h` будут использовать в большинстве случаев те, кто озабочен проблемами окружающей среды, но вы никак не можете стимулировать ее использование.

Также *недостаточно*, чтобы каждый из фильтров генерировал собственные начальные страницы (и, тем самым, мог их учитывать). Если пользователи захотят отказаться от выдачи начальных страниц и укажут опцию `lpr -h`, они все равно их получат, и будут вынуждены оплатить, поскольку система LPD не передает информации о наличии опции `-h` ни одному из этих фильтров.

Итак, что же вы можете сделать?

Вы можете:

- Принять парадигму системы LPD и сделать начальные страницы бесплатными.
- Установить альтернативную систему вместо LPD, такую как LPRng. В разделе Альтернативы стандартному спулеру представлена дополнительная информация о других системах спулинга, которые можно использовать вместо LPD.
- Написать *умный* выходной фильтр. Обычно выходной фильтр не предназначен для выполнения чего-то кроме инициализации принтера и простых преобразований символов. Он подходит для начальных страниц и заданий с обычным текстом (когда нет текстового (входного) фильтра). Но, если есть текстовый фильтр для заданий с обычным текстом, то система LPD будет запускать выходной фильтр только для начальных страниц. И выходной фильтр может анализировать текст начальной страницы, которую генерирует система LPD, чтобы определить, на счет какого пользователя и хоста отнести начальную страницу. Единственная проблема этого метода в том, что выходной фильтр все равно не знает, какой учетный файл использовать (ему не передают имя файла, заданное в качестве значения характеристики `af`), но при наличии хорошо известного учетного файла, его имя можно явно указать в выходном фильтре. Для упрощения этапа анализа задайте характеристику `sh` (short header) в файле /etc/printcap. Повторимся, что это может оказаться слишком сложным, и пользователи, несомненно, больше оценят

великодушного системного администратора, который сделает начальные страницы бесплатными.

9.4.2.4. Начальные страницы на PostScript®-принтерах

Как было описано выше, система LPD может генерировать начальную страницу в виде обычного текста, что подходит для многих принтеров. Конечно, PostScript®-принтеры не могут непосредственно печатать обычный текст, так что, для них возможность выдачи начальных страниц системы LPD бесполезна - или почти бесполезна.

Один очевидный способ получить начальные страницы - заставить каждый фильтр преобразования и текстовый фильтр генерировать начальную страницу. Эти фильтры должны использовать аргументы имя пользователя и хост для генерации соответствующей начальной страницы. Недостаток этого метода состоит в том, что пользователи будут всегда получать начальные страницы, даже если будут посылать задания с помощью команды `lpr -h`.

Давайте рассмотрим этот метод детально. Следующий сценарий принимает три аргумента (регистрационное имя пользователя, имя хоста и имя задания) и создает простую начальную страницу на языке PostScript®:

```
#!/bin/sh
#
#  make-ps-header - выдать начальную страницу на языке PostScript в stdout
#  Установлен в /usr/local/libexec/make-ps-header
#

#
#  Это единицы измерения PostScript (72 на дюйм). Измените значения для A4 или
#  другого используемого формата бумаги:
#
page_width=612
page_height=792
border=72

#
#  Проверяем аргументы
#
if [ $# -ne 3 ]; then
    echo "Usage: `basename $0` <user> <host> <job>" 1>&2
    exit 1
fi

#
#  Сохраняем значения в переменных, в основном, для упрощения понимания
#  последующего PostScript-кода.
#
user=$1
host=$2
job=$3
```

```
date=`date`

#
#  Посылаем PostScript-код в stdout.
#
exec cat <<EOF
%!PS

%
%  Гарантируем, что не будем влиять на следующее далее задание пользователя
%
save

%
%  Делаем тонкую некрасивую рамку по краям бумаги.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray

%
%  Выдаем регистрационное имя пользователя, красивыми, большими и рельефными буквами
%
/Helvetica-Bold findfont 64 scalefont setfont
$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub moveto
($user) show

%
%  Теперь выдаем всякие детали
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[ (Job:) (Host:) (Date:) ] {
200 y moveto show /y y 18 sub def }
forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[ ($job) ($host) ($date) ] {
        270 y moveto show /y y 18 sub def
} forall

%
% Вот и все
%
restore
showpage
```

```
EOF
```

Теперь, каждый из фильтров преобразования и текстовый фильтр может вызвать этот сценарий, чтобы сначала сгенерировать начальную страницу, а затем напечатать задание пользователя. Вот фильтр преобразования DVI, представленный ранее в этом документе, измененный для выдачи начальной страницы:

```
#!/bin/sh
#
#  psdf - фильтр преобразования DVI в PostScript
#  Установлен в /usr/local/libexec/psdf
#
#  Вызывается системой lpd при выполнении пользователем команды lpr -d
#

orig_args="$@"

fail() {
    echo "$@" 1>&2
    exit 2
}

while getopts "x:y:n:h:" option; do
    case $option in
        x|y)  ;; # Ignore
        n)    login=$OPTARG ;;
        h)    host=$OPTARG ;;
        *)    echo "LPD started `basename $0` wrong." 1>&2
              exit 2
              ;;
    esac
done

[ "$login" ] || fail "No login name"
[ "$host" ] || fail "No host name"

( /usr/local/libexec/make-ps-header $login $host "DVI File"
  /usr/local/bin/dvips -f ) | eval /usr/local/libexec/lprps $orig_args
```

Обратите внимание, как фильтр должен анализировать список аргументов, чтобы определить имя пользователя и имя хоста. Анализ аргументов в других фильтрах аргументов выполняется точно так же. Текстовый фильтр принимает, однако, немного другой набор аргументов (см. раздел Как работают фильтры).

Как уже упоминалось, представленная выше схема хотя и достаточно проста, но не позволяет учесть опцию "подавить вывод начальной страницы" (опция `-h`) команды `lpr`. Если пользователи хотят сберечь деревья (или несколько копеек, если вы берете деньги и за начальные страницы), они не смогут этого сделать, поскольку каждый фильтр будет выдавать начальную страницу для каждого задания.

Чтобы позволить пользователям отключать выдачу начальной страницы для отдельного задания, надо будет использовать прием, представленный в разделе Учет начальных страниц: написать выходной фильтр, который анализирует сгенерированную системой LPD начальную страницу и выдает ее PostScript®-версию. Если пользователь посылает задание командой `lpr -h`, система LPD не будет генерировать начальную страницу, как и ваш выходной фильтр. В противном случае, ваш выходной фильтр будет читать текст, полученный от системы LPD, и посылать на принтер соответствующий PostScript®-код для начальной страницы.

Если вы используете PostScript®-принтер с последовательным интерфейсом, можно использовать систему `lprps`, которая включает выходной фильтр, `psof`, делающий то, что описано выше. Помните, что программа `psof` не учитывает напечатанные пользователями начальные страницы.

9.4.3. Печать по сети

FreeBSD поддерживает печать по сети: посылку заданий на удаленные принтеры. Печатью по сети обычно называют две разные ситуации:

- Работа с принтером, подключенным к удаленному хосту. Вы устанавливаете принтер с обычным последовательным или параллельным интерфейсом на одном хосте. Затем, вы настраиваете систему LPD для обеспечения доступа к принтеру с других хостов в сети. В разделе "Принтеры, установленные на удаленных хостах" описано, как это сделать.
- Работа с принтером, подключенным непосредственно к сети. Принтер имеет сетевой интерфейс, кроме (или вместо) более традиционного последовательного или параллельного. Такой принтер может работать следующим образом:
  - Он может понимать протокол LPD и даже поддерживать очереди заданий с удаленных хостов. В этом случае, он работает просто как обычный хост с системой LPD. Для настройки такого принтера следуйте той же процедуре, которая описана в разделе "Принтеры, установленные на удаленных хостах".
  - Он может поддерживать получение потока данных по сети. В этом случае, вы "подключаете" принтер к одному из хостов в сети, делая этот хост ответственным за поддержку очередей заданий и их посылку на принтер. В разделе Принтеры с сетевыми интерфейсами представлен ряд советов по установке таких принтеров.

9.4.3.1. Принтеры, установленные на удаленных хостах

Система спулинга LPD имеет встроенную поддержку посылки заданий на другие хосты, на которых тоже работает система LPD (или совместимая с LPD). Это позволяет установить принтер на одном хосте и сделать его доступным с других хостов. Она также работает с принтерами, имеющими сетевые интерфейсы и понимающими протокол LPD.

Для обеспечения такого рода удаленной печати, сначала установите принтер на одном хосте, *хосте принтера*, с помощью процедуры, описанной в разделе Простая настройка принтера. Выполните любые необходимые дополнительные настройки, как описано в разделе Расширенная настройка принтера. Не забудьте протестировать принтер и убедиться, обеспечивает ли он заданные возможности системы LPD. Также проверьте, что *локальный хост* имеет право использовать службу LPD на *удаленном хосте* (см. раздел

Ограничение приема заданий с удаленных хостов).

Если вы используете принтер с сетевым интерфейсом, совместимый с системой LPD, упомянутым в обсуждении выше *хостом принтера* будет сам принтер, а в качестве *имени принтера* будет выступать имя, которое вы сконфигурировали для принтера. См. документацию, поставляемую с принтером и/или сетевым интерфейсом принтера.

Если вы используете Hewlett Packard Laserjet, то при задании принтеру имени `text` будет автоматически выполняться преобразование символа LF в последовательность CRLF, так что, сценарий hpif не понадобится.

Затем, на других хостах, для которых вы хотите обеспечить доступ к принтеру, создайте запись в их файлах /etc/printcap со следующими компонентами:

1. Дайте записи любое подходящее имя. Для простоты, однако, имеет смысл задавать такое же имя и псевдонимы, как и на хосте принтера.
2. Характеристику `lp` оставьте пустой, указав это явно (`:lp=:`).
3. Создайте каталог спулинга и укажите его местонахождение в характеристике `sd`. Система LPD будет сохранять задания в нем, прежде чем они будут посланы на хост принтера.
4. Укажите имя хоста принтера в качестве значения характеристики `rm`.
5. Укажите имя принтера на *хосте принтера* в качестве значения характеристики `rp`.

Вот и все. Не нужно перечислять фильтры преобразования, размеры страницы и вообще ничего больше в файле /etc/printcap.

Рассмотрим пример. На хосте `rose` есть два принтера, `bamboo` и `rattan`. Мы позволим пользователям хоста `orchid` печатать на эти принтеры. Вот файл /etc/printcap для хоста `orchid` (из раздела Включение выдачи начальных страниц). В нем уже есть запись для принтера `teak`; мы добавили две записи для принтеров на хосте `rose`:

```
#
#  /etc/printcap для хоста orchid - добавлены (удаленные) принтеры на rose
#

#
#  teak - локальный принтер; он подключен непосредственно к orchid:
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

#
#  rattan подключен к rose; посылать задания для rattan на хост rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
```

```
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

#
#  bamboo тоже подключен к rose:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:
```

Затем достаточно только создать каталоги спулинга на `orchid`:

```
# mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chown daemon:daemon /var/spool/lpd/rattan /var/spool/lpd/bamboo
```

Теперь пользователи хоста `orchid` могут печатать на принтеры `rattan` и `bamboo`. Если, например, пользователь на `orchid` выполнит команду

```
% lpr -P bamboo -d sushi-review.dvi
```

система LPD на `orchid` будет копировать задание в каталог спулинга /var/spool/lpd/bamboo и учтет, что печатается задание DVI. Как только на хосте `rose` появится место в каталоге спулинга принтера `bamboo`, две системы LPD передадут файл на хост `rose`. Файл будет ждать в очереди на `rose` пока, наконец, не будет напечатан. Он будет преобразован из формата DVI в PostScript® (поскольку `bamboo` является PostScript®-принтером) на хосте `rose`.

9.4.3.2. Принтеры с сетевыми интерфейсами

Часто при покупке сетевой карты для принтера можно приобрести две версии: эмулирующую спулер (более дорогая версия) или просто позволяющую принимать на принтер данные так, как если бы использовался последовательный или параллельный порт (более дешевая версия). В этом разделе описано, как использовать более дешёвую версию. Использование более дорогой версии описано в предыдущем разделе "Принтеры, установленные на удаленных хостах".

Формат файла /etc/printcap позволяет указывать, какой последовательный или параллельный интерфейс использовать, и (при использовании последовательного интерфейса), какую установить скорость, использовать ли управление потоком, размер отступов для табуляций, преобразование символов новой строки и другие параметры. Но нет способа указать подключение к принтеру, прослушивающему TCP/IP или другой сетевой порт.

Для посылки данных на подключенный к сети принтер, надо разработать программу взаимодействия, которую могут вызывать текстовый фильтр и фильтры преобразований. Вот один из примеров: скрипт `netprint` принимает все данные со стандартного входного потока и посылает их на принтер, подключенный к сети. Мы указываем имя хоста принтера в качестве первого аргумента, а номер порта, к которому надо подключаться - в качестве второго аргумента команды `netprint`. Учтите, что поддерживается только

одностороннее взаимодействие (с ОС FreeBSD на принтер); многие сетевые принтеры поддерживают двустороннее взаимодействие, и вы можете захотеть его использовать (для получения состояния принтера, учета и т.п.).

```
#!/usr/bin/perl
#
#  netprint - Текстовый фильтр для принтера, подключенного к сети
#  Установлен в /usr/local/libexec/netprint
#
$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
    = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);

socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
    || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

Затем можно использовать этот сценарий в различных фильтрах. Пусть у нас есть строчный принтер Diablo 750-N, подключенный к сети. Принтер принимает данные на печать через порт 5100. Имя хоста для принтера - scrivener. Вот текстовый фильтр для этого принтера:

```
#!/bin/sh
#
#  diablo-if-net - Текстовый фильтр для принтера Diablo `scrivener',
#  прослушивающего порт 5100. Установлен в /usr/local/libexec/diablo-if-net
#
exec /usr/libexec/lpr/lpf "$@" | /usr/local/libexec/netprint scrivener 5100
```

9.4.4. Ограничение использования принтера

В этом разделе представлена информация об ограничении доступа к принтеру. Система LPD позволяет управлять тем, кто может обращаться к принтеру, как локально, так и удаленно, смогут ли они печатать несколько копий, насколько большими могут быть их задания и насколько могут разрастаться очереди печати.

9.4.4.1. Ограничение количества копий

Система LPD позволяет пользователям легко печатать несколько копий файла. Пользователи могут печатать задания с помощью команды `lpr -#5` (например) и получать пять копий каждого файла в задании. Хорошо это или нет - решать вам.

Если вы считаете, что многочисленные копии только изнашивают ваши принтеры, можете отключить опцию `- команды lpr(1), добавив характеристику sc в файл /etc/printcap. Когда пользователи пошлют задания с опцией -#`, они увидят:

```
lpr: multiple copies are not allowed
```

Учтите, что если вы настроили удаленный доступ к принтеру (см. раздел "Принтеры, установленные на удаленных хостах"), необходимо задать характеристику `sc` также и в файлах /etc/printcap удаленных хостов, иначе пользователи все равно смогут посылать задания с несколькими копиями с других хостов.

Рассмотрим пример. Вот файл /etc/printcap для хоста `rose`. Принтер `rattan` вполне надежен, поэтому мы разрешим печатать на него несколько копий, но лазерный принтер `bamboo` несколько более изношен, поэтому мы отключим для него печать нескольких копий, добавив характеристику `sc`:

```
#
#  /etc/printcap для хоста rose - запрещает печать нескольких копий на bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Теперь нам также нужно добавить характеристику `sc` в файле /etc/printcap на хосте `orchid` (и раз уж мы его меняем, давайте отключим печать нескольких копий для принтера `teak`):

```
#
#  /etc/printcap для хоста orchid - отключена печать нескольких копий на
#  локальном принтере teak и на удаленном принтере bamboo
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:sc:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:
```

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:sc:
```

С помощью характеристики `sc` мы предотвращаем использование команды `lpr -#`, но это не мешает пользователям просто выполнить команду lpr(1) несколько раз или просто послать один и тот же файл несколько раз в одном задании следующим образом:

```
% lpr forsale.sign forsale.sign forsale.sign forsale.sign forsale.sign
```

Есть много способов предотвратить такое некорректное использование (включая его игнорирование), которые вы можете разработать самостоятельно.

9.4.4.2. Ограничение доступа к принтерам

Вы можете управлять тем, кто и на какие принтеры может печатать, с помощью механизма групп UNIX® и характеристики `rg` в файле /etc/printcap. Просто поместите пользователей, которым необходимо предоставить доступ к принтеру, в определенную группу, а затем укажите эту группу в качестве значения характеристики `rg`.

Пользователи, не входящие в эту группу (включая `root`) будут получать уведомление `lpr: Not a member of the restricted group` при попытке печатать на контролируемый принтер.

Как и в случае с характеристикой `sc` (подавить выдачу нескольких копий), при необходимости, надо указывать характеристику `rg` и на удаленных хостах, имеющих доступ к вашим принтерам (см. раздел "Принтеры, установленные на удаленных хостах").

Например, давайте разрешим всем обращаться к принтеру `rattan`, но только пользователи группы `artists` смогут использовать принтер `bamboo`. Вот знакомый уже файл /etc/printcap для хоста `rose`:

```
#
#  /etc/printcap для хоста rose - ограничение группы для bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Давайте не будем менять другой рассматриваемый файл /etc/printcap (для хоста `orchid`).

Конечно, в результате, любой пользователь orchid может печатать на bamboo. Возможно, на хосте orchid учетных записей и так немного, и вы хотите, чтобы все они имели доступ к принтеру. Или нет.

Для принтера может быть только одна ограниченная группа.

9.4.4.3. Контроль размеров посылаемых заданий

Если к принтеру обращается несколько пользователей, вам, возможно, понадобиться установить ограничение на максимальный размер файлов, которые пользователи могут посылать на печать. В конечном итоге, размер файловой системы, в которой находятся каталоги спулинга, ограничен, и надо гарантировать, что в нем останется место для заданий других пользователей.

Система LPD ограничить максимально допустимый размер файла в задании с помощью характеристики mx. Размер задается в блоках, размер которых, BUFSIZ, составляет 1024 байта. Если задать этой характеристике значение ноль, размер файла ограничиваться не будет; однако, если характеристика mx вообще не задана, то будет использоваться стандартное ограничение - 1000 блоков.

Ограничение применяется к *файлам в задании*, а *не* к общему размеру задания.

Система LPD не откажется печатать файл больше максимально допустимого для принтера размера. Вместо этого, она поставит в очередь часть файла до заданного предела, и она будет напечатана. Остальное не будет напечатано. Правильность такого поведения не бесспорна.

Давайте установим ограничения для принтеров из наших примеров, rattan и bamboo. Поскольку PostScript®-файлы этих художников обычно бывают весьма большими, мы ограничим их размер пятью мегабайтами. Мы не будем ограничивать использование обычного текстового строчного принтера:

```
#
#  /etc/printcap для хоста rose
#

#
#  Без ограничения на размер задания:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:mx#0:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

#
#  Размер файла - не более пяти мегабайт:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
```

```
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Опять таки, ограничения применяются только для локальных пользователей. Если вы настроили удаленный доступ к принтерам, для удаленных пользователей эти ограничения не действуют. Надо задать характеристику `mx` и в файлах /etc/printcap удаленных хостов. Более детальную информацию по удаленной печати см. в разделе "Принтеры, установленные на удаленных хостах".

Есть еще один специализированный способ ограничить размер заданий печати с удаленных принтеров; см. раздел Ограничение печати заданий с удаленных хостов.

9.4.4.4. Ограничение печати заданий с удаленных хостов

Система спулинга LPD обеспечивает несколько способов ограничить посылку заданий с удаленных хостов:

Ограничения хостов

Вы можете управлять тем, с каких удаленных хостов локальная система LPD принимает запросы, с помощью файлов /etc/hosts.equiv и /etc/hosts.lpd. Система LPD проверяет, поступает ли входящий запрос с хоста, указанного в одном из этих файлов. Если нет, система LPD отвергает запрос.

Формат этих файлов простой: по одному имени хоста в строке. Учтите, что файл /etc/hosts.equiv также используется протоколом ruserok(3) и влияет на программы rsh(1) и rcp(1), так что, будьте внимательны.

Например, вот файл /etc/hosts.lpd для хоста `rose`:

```
orchid
violet
madrigal.fishbaum.de
```

Это означает, что хост `rose` будет принимать запросы с хостов `orchid`, `violet` и `madrigal.fishbaum.de`. Если любой другой хост попытается обратиться к системе LPD хоста `rose`, его задание будет отвергнуто.

Ограничения размера

Вы можете управлять тем, сколько свободного места должно оставаться в файловой системе, в которой находится каталог спулинга. Создайте файл с именем minfree в каталоге спулинга для локального принтера. Вставьте в этот файл число, задающее, сколько блоков диска (по 512 байтов) должно быть свободными, чтобы удаленное задание было принято.

Это позволяет гарантировать, что удаленные пользователи не заполнят вашу файловую систему. Можно также использовать этот механизм для предоставления определенного

преимущества локальным пользователям: они смогут ставить задания в очередь еще долго после того, как свободного места на диске станет меньше, чем указано в файле minfree.

Например, давайте добавим файл minfree для принтера `bamboo`. Найдем в файле /etc/printcap каталог спулинга для этого принтера; вот запись для принтера `bamboo`:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:mx#5000:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Каталог спулинга задается характеристикой `sd`. Укажем, что в файловой системе должно быть три мегабайта (что составляет 6144 блоков диска) свободного места, чтобы система LPD принимала удаленные задания:

```
# echo 6144 > /var/spool/lpd/bamboo/minfree
```

Ограничения пользователей

Вы можете управлять тем, какие удаленные пользователи смогут печатать на локальные принтеры, задавая характеристику `rs` в файле /etc/printcap. Когда характеристика `rs` указана в записи для локально подключенного принтера, система LPD будет принимать задания с удаленных хостов, *если* пользователь, посылающий задание, также имеет учетную запись с тем же именем на локальном хосте. В противном случае, система LPD отвергает задание.

Эта возможность особенно полезна в среде, где есть, например, несколько отделов, совместно использующих сеть, и некоторые пользователи могут переходить из отдела в отдел. Если дать им учетные записи в системах, они смогут использовать принтеры из систем в своих отделах. Если вы хотели бы позволить им использовать *только* принтеры, но не остальные ресурсы вашего компьютера, можно дать им "формальные" учетные записи, без начального каталога и с бесполезным начальным командным интерпретатором вроде /usr/bin/false.

9.4.5. Учет использования принтера

Итак, вам надо брать деньги за распечатки. А почему нет? Бумага и чернила стоят денег. А есть еще и затраты на поддержку в работоспособном состоянии - принтеры имеют множество движущихся частей и склонны к поломкам. Вы проанализировали состояние принтеров, объемы использования и затраты на их эксплуатацию, и получили определенную стоимость страницы (или фута, метра или чего угодно). Теперь, как же начать реально учитывать распечатки?

Итак, плохая новость состоит в том, что система спулинга LPD в этом не сильно поможет. Учет сильно зависит от типа используемого принтера, форматов распечаток и *ваших* требований к оплате использования принтеров.

Для реализации учета надо изменить текстовый фильтр принтера (чтобы учитывать обычные текстовые задания) и фильтры преобразования (чтобы учитывать другие форматы файлов), для подсчета страниц или запроса количества напечатанных страниц у принтера. Не получится обойтись использованием простого выходного фильтра, поскольку он не может выполнять учет. См. раздел Фильтры.

Обычно есть два способа выполнения учета:

- *Периодический учет* - более распространенный способ, возможно, потому, что он проще. Когда кто-то печатает задание, фильтр регистрирует пользователя, хост и количество страниц в учетном файле. Каждый месяц, семестр, год или раз в любой желаемый период времени, вы собираете учетные файлы для различных принтеров, суммируете напечатанные каждым пользователем страницы и выставляете суммы за использование. Затем вы очищаете все регистрационные файлы, начиная с чистого листа новый отчетный период.
- *Постоянный учет* используется реже, вероятно, потому, что сложнее в реализации. Этот метод требует от фильтров выставлять пользователям суммы за распечатки сразу после использования принтеров. Как и проверка дисковых квот, этот учет выполняется немедленно. Вы можете не давать пользователям печатать, если баланс на их счету стал отрицательным, а также предоставить им способ проверить и изменить свои "квоты печати". Но этот метод требует поддержки базы данных для отслеживания пользователей и квот.

Система спулинга LPD легко поддерживает оба метода: поскольку вы (в большинстве случаев) должны предоставить фильтры, вам придется предоставить и код для учета. Но есть и положительный момент: методы учета могут быть сколько угодно гибкими. Например, можно выбрать периодический или постоянный учет. Можно выбрать, какую именно информацию регистрировать: имена пользователей, имена хостов, типы заданий, количество напечатанных страниц, квадратные метры использованной бумаги, продолжительность печати заданий, и т.д. Это делается путем изменения фильтров так, чтобы они сохраняли соответствующую информацию.

9.4.5.1. Простая система учета использования принтера

В составе FreeBSD поставляется две программы, которые можно сразу использовать для организации простой системы периодического учета. Речь идет о текстовом фильтре `lpf`, описанном в разделе lpf: текстовый фильтр, и о программе pac(8), обеспечивающей сбор и суммирование записей из учетных файлов принтеров.

Как уже упоминалось в разделе, посвященном фильтрам (Фильтры), система LPD при запуске текстового фильтра и фильтров преобразований передает им имя учетного файла в командной строке. Фильтры могут использовать соответствующий аргумент, чтобы определить, куда записывать учетную информацию. Имя этого файла берется из значения характеристики `af` в файле /etc/printcap и, если не заданно как абсолютное, интерпретируется относительно каталога спулинга.

Система LPD запускает `lpf` с аргументами ширины и длины страницы (которые берутся из характеристик `pw` и `pl`). Программа `lpf` использует эти аргументы для определения количества бумаги, которая будет использована. После посылки файла на принтер она

вносит запись в учетный файл. Эти записи имеют следующий вид:

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

Следует использовать отдельный учетный файл для каждого принтера, поскольку программа `lpf` не реализует механизм блокирования файлов, и два экземпляра `lpf` могут повредить записи друг друга, если записывают одновременно в один и тот же файл. Простой способ выделить отдельный учетный файл для каждого принтера - использовать характеристику `af=acct` в файле /etc/printcap. Тогда каждый учетный файл окажется в каталоге спулинга соответствующего принтера и будет назван acct.

Когда вы будете готовы выставить пользователям счет за распечатки, запустите программу pac(8). Просто перейдите в каталог спулинга принтера, учетную информацию которого вы хотите обработать, и введите команду `pac`. Вы получите итоговые суммы в долларах, как показано ниже:

```
  Login        pages/feet   runs    price
orchid:kelly                5.00    1   $  0.10
orchid:mary                31.00    3   $  0.62
orchid:zhang                9.00    1   $  0.18
rose:andy                   2.00    1   $  0.04
rose:kelly                177.00  104   $  3.54
rose:mary                  87.00   32   $  1.74
rose:root                  26.00   12   $  0.52

total                     337.00  154   $  6.74
```

Команда pac(8) принимает следующие аргументы:

`-P принтер`

По какому *принтеру* подсчитывать итоговые суммы. Эта опция работает, только если в качестве значения характеристики `af` в файле /etc/printcap указано абсолютное имя.

`-c`

Сортировать отчет по сумме, а не по имени пользователя в алфавитном порядке.

`-m`

Игнорировать имя хоста в учетных файлах. При указании этой опции, пользователь `smith` на хосте `alpha` считается тем же, что и пользователь `smith` на хосте `gamma`. Обычно эти пользователи считаются разными.

`-p стоимость`

Вычислять суммы из расчета *стоимость* долларов за страницу или за фут, вместо

использования значения характеристики pc в файле /etc/printcap, или двух центов (как принято по умолчанию). Можно задавать *стоимость* как число с плавающей запятой.

-r

Изменить порядок сортировки.

-s

Создать итоговый учетный файл и очистить учетный файл.

имя...

Выдать учетную информацию только для пользователей с заданными *именами*.

В стандартном отчете, который создает команда pac(8), выдается количество страниц, напечатанное каждым из пользователей с различных хостов. Если для вас хосты не имеют значения (поскольку пользователи могут работать на любом хосте), выполните команду pac -m для получения следующих итогов:

```
  Login        pages/feet   runs    price
andy                        2.00    1   $  0.04
kelly                     182.00  105   $  3.64
mary                      118.00   35   $  2.36
root                       26.00   12   $  0.52
zhang                       9.00    1   $  0.18

total                     337.00  154   $  6.74
```

Для получения сумм в долларах программа pac(8) использует значение характеристики pc в файле /etc/printcap (по умолчанию - 200, или 2 цента за страницу). Укажите в качестве значения этой характеристики, в сотых долях цента, стоимость страницы или фута, исходя из которой вы хотите брать деньги за распечатки. Это значение можно переопределить при вызове команды pac(8) с помощью опции -p. Но при использовании опции -p стоимость надо указывать в долларах, а не в сотых долях цента. Например, команда

```
# pac -p1.50
```

приводит к тому, что страница будет стоить один доллар пятьдесят центов. Используя эту опцию, можно фактически начинать грести деньги лопатой.

Наконец, при выполнении команды pac -s итоговая информация будет сохранена в итоговом учетном файле, имя которого строится как имя учетного файла принтера с суффиксом _sum. Затем учетный файл принтера очищается. Когда команда pac(8) выполняется повторно, она перечитывает итоговый файл для получения начальных сумм, а затем добавляет информацию из обычного учетного файла.

9.4.5.2. Как можно подсчитать количество напечатанных страниц?

Для выполнения хоть отдаленно точного учета надо уметь определять, сколько бумаги

использовано для печати задания. Это - основная проблема учета использования принтеров.

Для обычных текстовых заданий решить эту проблему не сложно: надо считать, сколько строк входит в задание, и сравнивать с количеством строк на страницу, которые поддерживает принтер. Не забывайте учитывать символы забоя в файлах, которые приводят к перепечатке строк поверх, а также длинные логические строки, которые переносятся на несколько физических.

Текстовый фильтр `lpf` (представленный в разделе lpf: текстовый фильтр) учитывает эти вещи при выполнении учета. Если вы пишете текстовый фильтр, который должен осуществлять учет, может иметь смысл просмотреть исходный код программы `lpf`.

Но как обрабатывать файлы других форматов?

Ну, для преобразования из DVI в формат LaserJet или из DVI в PostScript®, можно в фильтре анализировать диагностические результаты команды `dvilj` или `dvips`, чтобы определить, сколько страниц было преобразовано. Может оказаться возможным применить этот прием и для других форматов файлов и программ преобразования.

Но эти методы несовершенны из-за того, что принтер мог фактически и не напечатать все эти страницы. Например, он мог замять бумагу, в нем мог закончиться тонер или он мог вообще взорваться - и пользователю все равно пришлось бы платить.

Так что же делать?

Есть только один *надежный* способ *точного* учета. Купите принтер, который может сообщать, сколько бумаги он использовал, и подключите его через последовательный порт или по сети. Практически все PostScript®-принтеры поддерживают такую возможность. Другие модели - тоже (сетевые лазерные принтеры Imagen, например). Измените фильтры для этих принтеров так, чтобы получать количество использованных страниц после печати каждого задания, и пусть они записывают учетную информацию *только* на основе этого значения. Не надо ни считать строки, ни выполнять чреватую ошибками обработку файла.

Конечно, всегда можно поступить великодушно и не брать денег за распечатки.

9.5. Использование принтеров

В этом разделе описано, как использовать настроенные принтеры в ОС FreeBSD. Вот сводка команд пользовательского уровня:

lpr(1)

Печать заданий

lpq(1)

Проверка очередей принтеров

lprm(1)

Удаление заданий из очередей принтеров

Есть также административная команда, lpc(8), описанная в разделе Администрирование принтеров, используемая для управления принтерами и их очередями.

Все три команды, lpr(1), lprm(1) и lpq(1), поддерживают опцию `-P имя-принтера`, позволяющую указать, с каким принтером/очередью из указанных в файле /etc/printcap работать. Это позволяет посылать, удалять и проверять задания на разных принтерах. Если вы не используете опцию `-P`, эти команды используют принтер, указанный в качестве значения переменной среды `PRINTER`. Наконец, если переменная среды `PRINTER` не задана, эти команды по умолчанию направляются на принтер по имени `lp`.

Далее термин *стандартный принтер* означает принтер, указанный переменной среды `PRINTER` или принтер по имени `lp`, если переменная среды `PRINTER` не задана.

9.5.1. Задания печати

Для печати файлов, выполните команду:

```
% lpr имя-файла ...
```

Эта команда печатает каждый из перечисленных файлов на стандартный принтер. Если файлы не указаны, команда lpr(1) читает данные для печати со стандартного входного потока. Например, следующая команда печатает некоторые важные системные файлы:

```
% lpr /etc/host.conf /etc/hosts.equiv
```

Для выбора конкретного принтера, введите:

```
% lpr -P имя-принтера имя-файла ...
```

Следующая команда печатает подробный листинг текущего каталога на принтере `rattan`:

```
% ls -l | lpr -P rattan
```

Поскольку для команды lpr(1) файлы не указаны, команда `lpr` читает данные для печати из стандартного входного потока, который содержит результат выполнения команды `ls -l`.

Команда lpr(1) может также принимать множество опций для управления форматированием, применения преобразований, печати нескольких копий и т.д. Дополнительную информацию см. в разделе Опции печати.

9.5.2. Проверка заданий

При печати с помощью команды lpr(1), данные, которые надо напечатать, помещаются вместе в пакет, который называют "заданием печати", и посылаются системе спулинга LPD. Каждый принтер имеет очередь заданий, и ваше задание ждет в этой очереди вместе с

другими вашими заданиями и заданиями других пользователей. Принтер печатает эти задания по принципу первым пришло, первым выполнено.

Для получения очереди стандартного принтера, введите команду lpq(1). Чтобы указать конкретный принтер, используйте опцию -P. Например, команда

```
% lpq -P bamboo
```

показывает очередь для принтера по имени bamboo. Вот пример результатов выполнения команды lpq:

```
bamboo is ready and printing
Rank   Owner    Job  Files                              Total Size
active kelly    9    /etc/host.conf, /etc/hosts.equiv   88 bytes
2nd    kelly    10   (standard input)                   1635 bytes
3rd    mary     11   ...                                78519 bytes
```

Показано, что в очереди bamboo есть три задания. Первое задание, посланное пользователем kelly, получило "номер задания" 9. Каждое задание для принтера получает уникальный номер задания. В большинстве случаев номер задания можно игнорировать, но он потребуется, если надо будет отменить задание; подробнее об этом см. в разделе Удаление заданий.

Задание номер 9 состоит из двух файлов; несколько файлов, указанных в командной строке lpr(1), считаются частью одного задания. Это задание является текущим активным (обратите внимание на слово active в столбце "Rank"), т.е. принтер должен сейчас печатать это задание. Второе задание состоит из данных, передаваемых в качестве стандартного входного потока команде lpr(1). Третье задание послано пользователем mary; оно намного больше по объему. Полное имя файла, который печатается, слишком длинное и не помещается, поэтому команда lpq(1) просто выдает три точки.

Самая первая строка результатов команды lpq(1) тоже полезна: она говорит о том, что сейчас делает принтер (или, по крайней мере, что он делает по мнению системы LPD).

Команда lpq(1) также поддерживает опцию -l для генерации подробного длинного листинга. Вот пример результатов выполнения команды lpq -l:

```
waiting for bamboo to become ready (offline ?)
kelly: 1st                 [job 009rose]
       /etc/host.conf                    73 bytes
       /etc/hosts.equiv                  15 bytes

kelly: 2nd                 [job 010rose]
       (standard input)                  1635 bytes

mary: 3rd                                [job 011rose]
      /home/orchid/mary/research/venus/alpha-regio/mapping 78519 bytes
```

9.5.3. Удаление заданий

Если вы передумали печатать задание, можно удалить его из очереди заданий с помощью команды lprm(1). Часто можно использовать lprm(1) для удаления активного задания, но часть задания или даже все задание все равно может быть напечатано.

Для удаления задания со стандартного принтера сначала используйте команду lpq(1) для поиска номера задания. Затем введите команду:

```
% lprm номер-задания
```

Для удаления задания с указанного принтера, задайте опцию `-P` option. Следующая команда удаляет задание номер 10 из очереди заданий принтера `bamboo`:

```
% lprm -P bamboo 10
```

Для команды lprm(1) есть ряд сокращений:

lprm -

Удаляет все задания (со стандартного принтера), принадлежащие пользователю, который выполнил команду.

lprm *пользователь*

Удаляет все задания (для стандартного принтера), принадлежащие указанному *пользователю*. Суперпользователь может удалять задания других пользователей; обычный пользователь может удалять только собственные задания.

lprm

Если в командной строке не указаны номер задания, имя пользователя, или указана опция `-`, команда lprm(1) удаляет текущее активное задание на стандартном принтере, если оно принадлежит вам. Суперпользователь может удалять любое активное задание.

Добавьте опцию `-P` для любого из перечисленных выше сокращений, чтобы работать с любым необходимым принтером вместо стандартного. Например, следующая команда удаляет все задания текущего пользователя из очереди принтера по имени `rattan`:

```
% lprm -P rattan -
```

Если вы работаете в сетевой среде, команда lprm(1) позволит вам удалять задания только с хоста, с которого они были посланы, даже если тот же принтер доступен и с других хостов. Следующая последовательность команд демонстрирует это:

```
% lpr -P rattan myfile
% rlogin orchid
```

```
% lpq -P rattan
Rank   Owner      Job  Files                          Total Size
active seeyan      12    ...                            49123 bytes
2nd    kelly      13   myfile                         12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfA013rose dequeued
cfA013rose dequeued
```

9.5.4. Не только обычный текст: опции печати

Команда lpr(1) поддерживает несколько опций, управляющих форматированием текста, преобразованием графики и других форматов файлов, выдачей нескольких копий, обработкой задания и др. В этом разделе описаны эти опции.

9.5.4.1. Опции форматирования и преобразования

Следующие опции команды lpr(1) управляют форматированием файлов в задании. Используйте эти опции, если задание содержит не простой текст или если вы хотите сформатировать простой текст с помощью утилиты pr(1).

Например, следующая команда печатает файл DVI (из системы верстки TeX) по имени fish-report.dvi на принтере `bamboo`:

```
% lpr -P bamboo -d fish-report.dvi
```

Эти опции применяются для каждого файла в задании, так что нельзя смешивать (например) файлы DVI и ditroff в одном задании. Вместо этого посылайте однотипные файлы отдельными заданиями, используя для каждого задания соответствующие опции преобразования.

Все эти опции, кроме `-p` и `-T`, требуют наличия установленных для целевого принтера фильтров преобразования. Например, опция `-d` требует фильтра преобразования DVI. Подробнее см. в разделе Фильтры преобразования.

`-c`

Печать файлов cifplot.

`-d`

Печать файлов DVI.

`-f`

Печать текстовых файлов на языке FORTRAN.

`-g`

Печать графиков.

`-i число`

Сдвинуть результат вправо на *число* столбцов; если *число* не указано, сдвиг выполняется на 8 столбцов. Эта опция работает только с определенными фильтрами преобразования.

Не помещайте пробелы между `-i` и числом.

`-l`

Печать текстовых данных буквально, включая управляющие символы.

`-n`

Печать данных ditroff (device independent troff).

-p

Форматировать обычный текст перед печатью утилитой pr(1). Подробнее см. pr(1).

`-T заголовок`

Использовать указанный *заголовок* в колонтитуле pr(1) вместо имени файла. Эта опция учитывается только при использовании вместе с опцией `-p`.

`-t`

Печать данных troff.

`-v`

Печать растровых данных.

Вот пример: следующая команда печатает красиво сформатированную версию справочного руководства по команде ls(1) на стандартный принтер:

```
% zcat /usr/shared/man/man1/ls.1.gz | troff -t -man | lpr -t
```

Команда zcat(1) распаковывает исходный код страницы справочного руководства ls(1) и передает его команде troff(1), которая форматирует его и выдает результат в формате GNU troff, передаваемый команде lpr(1), посылающей задание спулеру LPD. Поскольку мы использовали опцию `-t` команды lpr(1), спулер при печати задания будет преобразовывать результат GNU troff в формат, понятный стандартному принтеру.

9.5.4.2. Опции обработки заданий

Следующие опции команды lpr(1) требуют от системы LPD специальной обработки задания:

-# копий

Выдавать указанное количество *копий* каждого файла в задании вместо одной. Администратор может отключить эту опцию для уменьшения износа принтера и поощрения использования ксерокса. См. раздел Ограничение количества копий.

В следующем примере на стандартный принтер печатается три копии файла parser.c, а затем - три копии parser.h:

```
% lpr -#3 parser.c parser.h
```

-m

Посылать почту после завершения задания печати. При указании этой опции, система LPD будет посылать почту на ваше имя после завершения обработки вашего задания. В сообщении будет сказано, выполнено ли задание успешно или по ходу была ошибка, и (часто) - в чем она состояла.

-s

Не копировать файлы в каталог спулинга, а сделать там на них символические связи.

Эту опцию имеет смысл использовать при печати больших заданий. Она экономит место в каталоге спулинга (ваше задание может занять все свободное место в файловой системе, в которой находится каталог спулинга). Она также экономит время, поскольку системе LPD не придется копировать каждый байт задания в каталог спулинга.

Есть, однако, и недостаток: поскольку система LPD будет ссылаться на исходные файлы непосредственно, вы не сможете изменять или удалять их, пока они не будут распечатаны.

Если вы печатаете на удаленный принтер, система LPD будет вынуждена, так или иначе, скопировать файлы с локального хоста на удаленный, поэтому опция `-s`сэкономит место только в локальном каталоге спулинга, но не в удаленном. Но, она все равно полезна.

-r

Удалять файлы в задании после копирования в каталог спулинга или после печати, если указана опция `-s`. Будьте внимательны при использовании этой опции!

9.5.4.3. Опции начальных страниц

Эти опции команды lpr(1) изменяют текст, который обычно выдается на начальной странице задания. Если выдача начальных страниц для целевого принтера отключена, эти опции не действуют. Информацию по настройке начальных страниц см. в разделе Начальные страницы.

-C *текст*

Заменить имя хоста на начальной странице *текстом*. Обычно на ней выдается имя хоста, с которого было послано задание.

-J *текст*

Заменить имя задания на начальной странице *текстом*. Имя задания обычно совпадает с именем первого файла в задании или имеет значение stdin, если печатается стандартный входной поток.

-h

Не выдавать начальной страницы.

В некоторых организациях эта опция может не действовать, что определяется способом генерации начальных страниц. Подробнее см. в разделе Начальные страницы.

9.5.5. Администрирование принтеров

Как администратор принтеров, вы должны их установить, настроить и протестировать. С помощью команды lpc(8) вы можете взаимодействовать с принтерами и другими способами. С помощью lpc(8) вы можете:

- Запускать и останавливать принтеры
- Включать и отключать их очереди
- Изменять порядок заданий в каждой очереди.

Начнем с замечания по терминологии: если принтер *остановлен*, он не будет печатать ничего из своей очереди. Пользователи могут продолжать посылать задания, которые будут ждать в очереди, пока принтер не будет *запущен* или пока очередь не будет очищена.

Если очередь *отключена*, ни один пользователь (кроме `root`) не может посылать задания на принтер. Во *включенную* очередь можно посылать задания. Принтер для отключенной очереди может быть *запущен*; при этом он будет продолжать печатать находящиеся в очереди задания, пока очередь не станет пустой.

В общем случае, для использования команды lpc(8) необходимо иметь привилегии `root`. Обычные пользователи могут использовать команду lpc(8) только для получения состояния принтера и перезапуска зависшего принтера.

Далее представлена сводка команд lpc(8). Большинство команд принимает аргумент *имя-принтера*, задающий, с каким принтером работать. Можно использовать значение `all` вместо *имени-принтера*, означающее все принтеры, перечисленные в файле /etc/printcap.

`abort имя-принтера`

Снять текущее задание и остановить принтер. Пользователи могут продолжать посылать задания, если очередь включена.

`clean имя-принтера`

Удалить старые файлы из каталога спулинга принтера. Иногда файлы, составляющие задание, не удаляются как положено системой LPD, особенно если в ходе печати были ошибки и выполнялось много административных действий. Эта команда находит файлы, не принадлежащие каталогу спулинга, и удаляет их.

`disable имя-принтера`

Отключить постановку новых заданий в очередь. Если принтер работает, он продолжит печатать задания, остающиеся в очереди. Суперпользователь (`root`) всегда может посылать задания, даже в отключенную очередь.

Эта команда полезна при тестировании вновь установленного принтера или фильтра: отключаем очередь и посылаем задания как `root`. Другие пользователи не смогут посылать задания, пока вы не закончите тестирование и не включите очередь повторно командой `enable`.

`down имя-принтера сообщение`

Отключить принтер. Аналогична последовательности команд `disable` и `stop`. Указанное *сообщение* выдается как состояние принтера при проверке пользователем очереди принтера с помощью lpq(1) или запросе его состояния командой `lpc status`.

`enable имя-принтера`

Включить очередь для принтера. Пользователи могут посылать задания, но принтер не будет их печатать, пока не будет запущен.

`help имя-команды`

Выдать справочную информацию по команде *имя-команды*. Если *имя-команды* не указано, выдает сводку по имеющимся командам.

`restart имя-принтера`

Перезапустить принтер. Обычные пользователи могут использовать эту команду, если в результате неких чрезвычайных обстоятельств система LPD зависла, но они не могут запустить принтер, остановленный командами `stop` или `down`. Команда `restart` эквивалентна последовательности команд `abort` и `start`.

`start имя-принтера`

Запустить принтер. Принтер будет печатать задания, находящиеся в его очереди.

`stop имя-принтера`

Остановить принтер. Принтер закончит печать текущего задания и больше ничего из очереди печатать не будет. Хотя принтер и остановлен, пользователи могут посылать задания во включенную очередь.

`topq имя-принтера задание-или-имя-пользователя`

Переупорядочить очередь для указанного принтера, помещая указанные по номеру *задания* или задания указанного по имени *пользователя* в начало очереди. Для этой команды нельзя использовать `all` в качестве *имени-принтера*.

`up имя-принтера`

Включить принтер; команда по действию противоположна команде `down`. Эквивалентна последовательности команд `start` и `enable`.

Утилита lpc(8) принимает перечисленные выше команды в командной строке. Если команда не указана, утилита lpc(8) входит в интерактивный режим, в котором можно вводить команды, пока не будет введена команда `exit`, `quit` или символ конца файла.

9.6. Альтернативы стандартному спулеру

Если вы прочитали все это руководство, к этому моменту вы знаете практически все, что

надо знать о системе спулинга LPD, входящей в состав ОС FreeBSD. Вы, возможно, уже осознали многие из ее недостатков, что, естественно, приводит к вопросу: "Какие еще системы спулинга существуют (и работают с ОС FreeBSD)"?

LPRng

Система LPRng, имя которой означает "LPR: the Next Generation" (LPR: следующее поколение) - это полностью переписанная система PLP. Патрик Пауэл (Patrick Powell) и Джастин Мейсон (Justin Mason) (основной специалист, занимающийся поддержкой PLP) объединили усилия для создания системы LPRng. Основной сайт по системе LPRng - http://www.lprng.org/.

CUPS

Система CUPS (сокращение от Common UNIX Printing System) предоставляет переносимый механизм печати для операционных систем, основанных на UNIX®. Она была разработана компанией Easy Software Products в качестве стандартного механизма печати для всех производителей и пользователей UNIX®.

Система CUPS использует протокол Internet Printing Protocol (IPP) для управления заданиями и очередями. Протоколы Line Printer Daemon (LPD), Server Message Block (SMB) и AppSocket (известный также как JetDirect) также поддерживаются, но с меньшими возможностями. Система CUPS добавляет поиск сетевых принтеров и опции печати на основе PostScript Printer Description (PPD), для поддержки практической печати в UNIX®.

Основной сайт по системе CUPS - http://www.cups.org/.

9.7. Выявление проблем

После выполнения простого тестирования с помощью команды lptest(1) вы можете получить один из следующих результатов вместо корректной распечатки:

Все работает, после определенной задержки; или не выдается распечатанная страница.

Принтер напечатал все, что нужно, но он на некоторое время задумывался и ничего не делал. Фактически, могло потребоваться нажать кнопку PRINT REMAINING или FORM FEED на принтере, чтобы результаты были выданы.

Если это произошло, вероятно, принтер ждал, нет ли в задании еще данных, прежде чем что бы то ни было печатать. Для решения этой проблемы можно посылать в текстовом фильтре на принтер символ FORM FEED (или любую необходимую последовательность символов). Этого обычно достаточно, чтобы принтер немедленно распечатал любой остающийся в его внутреннем буфере текст. Также полезно убедиться, что каждое задание печати заканчивается полной страницей, чтобы следующее задание не начиналось где-то с середины последней страницы предыдущего задания.

Следующий измененный скрипт командного интерпретатора /usr/local/libexec/if-simple выдает символ прогона страницы после посылки задания на принтер:

```
#!/bin/sh
#
```

```
# if-simple - Простой текстовый входной фильтр для lpd
# Установлен в /usr/local/libexec/if-simple
#
# Просто копирует stdin в stdout. Игнорирует все аргументы фильтра.
# Выдает символ прогона страницы (\f) после печати задания.

/bin/cat && printf "\f" && exit 0
exit 2
```

Принтер печатает "лесенкой".

Вы получаете на бумаге следующее:

```
!"#$%&'()*+,-./01234
               "#$%&'()*+,-./012345
                                #$%&'()*+,-./0123456
```

Вы стали очередной жертвой *эффекта лесенки*, вызванного различными интерпретациями того, какие символы должны обозначать новую строку. Операционные системы UNIX®-стиля используют один символ: ASCII-код 10, перевод строки (line feed - LF). MS-DOS®, OS/2® и другие используют пару символов, ASCII-код 10 *и* ASCII-код 13 (возврат каретки, carriage return или CR). Многие принтеры используют соглашение MS-DOS® для представления новых строк.

При печати из FreeBSD в тексте используется только символ перевода строки. Принтер, встретив символ перевода строки, переходит на следующую строку, но оставляет ту же горизонтальную позицию на строке для следующего печатаемого символа. Вот зачем нужен символ возврата каретки: чтобы перенести следующий печатаемый символ на левый край бумаги.

Вот что ОС FreeBSD хочет от принтера:

Принтер получает CR	Принтер печатает CR
Принтер получает LF	Принтер печатает CR + LF

Вот несколько способов этого добиться:

- Использовать переключатели конфигурации принтера или панель управления, чтобы изменить его интерпретацию этих символов. Поищите как это сделать в руководстве по своему принтеру.

  Если вы загружаете другие операционные системы, кроме FreeBSD, может иметь смысл *переконфигурировать* принтер для использования такой интерпретации символов CR и LF, которая принята в этих операционных системах. Затем можно использовать одно из представленных далее решений.

- Заставить драйвер последовательного порта FreeBSD автоматически

преобразовывать LF в CR+LF. Конечно, это подойдет *только* для принтеров, подключенных к последовательным портам. Для включения этой возможности используйте характеристику `ms#` и установите режим `onlcr` для принтера в файле /etc/printcap.

- Послать *управляющий код* на принтер, заставляющий его временно обрабатывать символы LF по-другому. Управляющие коды, которые может поддерживать ваш принтер, поищите в руководстве своего принтера. Когда найдете соответствующий управляющий код, измените текстовый фильтр для посылки сначала этого кода, а затем - задания печати.

  Вот пример текстового фильтра для принтеров, понимающих управляющие последовательности языка Hewlett-Packard PCL. Этот фильтр заставляет принтер обрабатывать символы LF как LF и CR; затем он посылает задание; наконец, он посылает символ прогона страницы для выдачи последней страницы задания. Он должен работать практически со всеми принтерами Hewlett Packard.

```
#!/bin/sh
#
# hpif - Простой текстовый входной фильтр для lpd для принтеров на базе HP-PCL
# Установлен в /usr/local/libexec/hpif
#
# Просто копирует stdin в stdout. Игнорирует все аргументы фильтра.
# Требует от принтера обрабатывать LF как CR+LF. Выдает страницу по окончании.

printf "\033&k2G" && cat && printf "\033&l0H" && exit 0
exit 2
```

  Вот пример файла /etc/printcap с хоста `orchid`. К нему через первый параллельный порт подключен один принтер, Hewlett Packard LaserJet 3Si, по имени `teak`. Для него в качестве текстового фильтра используется представленный выше скрипт:

```
#
#  /etc/printcap для хоста orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:
```

Строки напечатаны одна поверх другой.

Принтер так и не перешел на следующую строку. Все строки текста были напечатаны одна поверх другой, на одной строке.

Эта проблема "обратна" эффекту лесенки, описанному выше, и встречается намного реже. Каким-то образом, символы LF, которые ОС FreeBSD использует для завершения строк, обрабатывались как символы CR и вызывали перевод позиции печати на левый край бумаги, но не переход на следующую строку.

Используйте переключатели конфигурации принтера или панель управления для обеспечения следующей интерпретации символов LF и CR:

Принтер получает	Принтер печатает
CR	CR
LF	CR + LF

Принтер теряет символы.

По ходу печати принтер не печатает несколько символов в каждой строке. Проблема со временем может становиться все хуже, так что теряется все больше символов.

Проблема состоит в том, что принтер не справляется с той скоростью, с которой компьютер посылает данные по последовательной линии (эта проблема не должна возникать на принтерах, подключенных к параллельным портам). Есть два способа решить проблему:

- Если принтер поддерживает управление потоком XON/XOFF, заставить FreeBSD использовать его, указав режим `ixon` в характеристике `ms#`.
- Если принтер поддерживает управление несущим потоком (carrier flow control), укажите режим `crtscts` в характеристике `ms#`. Убедитесь, что кабель, соединяющий принтер с компьютером, правильно распаян для управления несущим потоком.

Напечатан мусор.

Принтер напечатал нечто похожее на случайный мусор, а не требуемый текст.

Это, обычно, - еще один симптом неправильных параметров взаимодействия с последовательным принтером. Перепроверьте скорость взаимодействия в характеристике `br` и установку четности в характеристике `ms#`; проверьте, что принтер использует те же установки, которые заданы в файле /etc/printcap.

Ничего не произошло.

Если ничего не произошло, проблема, вероятно, связана с FreeBSD, а не с оборудованием. Добавьте характеристику журнального файла (`lf`) в файл /etc/printcap для принтера, работу с которым отлаживаете. Например, вот запись для принтера `rattan` с характеристикой `lf`:

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:\
        :lf=/var/log/rattan.log
```

Затем попытайтесь напечатать снова. Поищите в журнальном файле (в нашем примере - /var/log/rattan.log) возможные сообщения об ошибках. На основе полученных сообщений попытайтесь решить проблему.

Если вы не зададите характеристику `lf`, система LPD использует по умолчанию

/dev/console.

Глава 10. Двоичная совместимость с Linux

10.1. Краткий обзор

FreeBSD предоставляет двоичную совместимость с несколькими другими UNIX®-подобными операционными системами, включая Linux. Возможно, вы спрашиваете себя, зачем FreeBSD нужно уметь работать с приложениями, написанными для Linux? Ответ на этот вопрос достаточно прост: многие компании и разработчики производят программное обеспечение только для Linux, так как эта операционная система очень быстро завоевала огромную популярность в компьютерном мире. Пользователям же FreeBSD приходится обращаться к этим компаниям и разработчикам с просьбами выпустить версии своих программ специально для FreeBSD. Проблема в том, что большинство производителей программного обеспечения не осознают, насколько бы увеличился их рынок сбыта, выпускай они также FreeBSD-версии, и продолжают разрабатывать только под Linux. Что же делать пользователям FreeBSD? В этой ситуации на помощь приходит двоичная совместимость FreeBSD с Linux.

Вкратце, эта совместимость позволяет пользователям FreeBSD работать с 90% приложений для Linux без каких-либо модификаций последних. Среди этих приложений: StarOffice™, Linux-версия getenv(3), Adobe® Acrobat®, RealPlayer, VMware, Oracle®, WordPerfect, Doom, Quake, и многие другие. Есть сведения, что в некоторых ситуациях эти Linux-программы показывали более высокую производительность при работе под FreeBSD, чем под Linux.

Конечно, существуют некоторые особенности Linux, которые не поддерживаются в полной мере под FreeBSD. Например, не будут работать приложения Linux, использующие специфичные вызовы i386™, такие как переключение в виртуальный режим 8086.

При чтении этой главы вы узнаете:

- Как включить двоичную совместимость с Linux в вашей системе.
- Как установить дополнительные совместно используемые (shared) библиотеки Linux.
- Как установить приложения Linux в систему FreeBSD.
- Детали реализации совместимости с Linux в ОС FreeBSD.

Перед прочтением этой главы вам потребуется:

- Узнать как устанавливать дополнительное программное обеспечение сторонних разработчиков (Установка приложений. порты и пакеты).

10.2. Установка

Двоичная совместимость с Linux не включена по умолчанию. Простейший способ включения этой функциональности заключается в загрузке KLD-объекта `linux` ("Kernel LoaDable object"). Вы можете загрузить этот модуль, набрав следующее, как пользователь `root`:

```
# kldload linux
```

Если вы хотите, чтобы совместимость с Linux была включена постоянно, необходимо добавить в /etc/rc.conf следующую строку:

```
linux_enable="YES"
```

Для проверки того, загружен ли модуль, может быть использована команда kldstat(8):

```
% kldstat
Id Refs Address    Size     Name
 1    2 0xc0100000 16bdb8   kernel
 7    1 0xc24db000 d000     linux.ko
```

Если по какой-либо причине вы не хотите или не можете загрузить KLD, вы можете статически включить поддержку Linux в ядро, добавив опцию `options COMPAT_LINUX` в файл конфигурации ядра. Затем соберите и установите новое ядро, следуя описанию в Настройка ядра FreeBSD.

10.2.1. Установка необходимых библиотек Linux

Установить все требуемые библиотеки можно двумя путями: либо используя порт linux_base, либо установив их вручную.

10.2.1.1. Установка с помощью порта linux_base

Этот метод является самым простым, и мы рекомендуем воспользоваться именно им. Процесс аналогичен установке любого другого порта из Коллекции Портов. Просто выполните следующие команды:

```
# cd /usr/ports/emulators/linux_base-fc4
# make install distclean
```

Теперь вы можете работать с приложениями для Linux. Некоторые программы, возможно, будут сообщать о несоответствии подверсий некоторых системных библиотек. Однако обычно это не вызывает каких-либо неудобств.

Возможно наличие нескольких версий порта emulators/linux_base, соответствующих различным версиям разных дистрибутивов Linux. Вы должны установить порт, наиболее близко соответствующий требованиям приложений Linux, которые будут установлены.

10.2.1.2. Установка библиотек вручную

Если у вас не установлена коллекция портов, можно установить требуемые библиотеки

вручную. Вам понадобятся совместно используемые библиотеки для Linux, которые нужны программам, и runtime-компоновщик. Вам также потребуется создать "теневой корневой каталог", /compat/linux, где будут расположены Linux-библиотеки. Если Linux-программе нужно загрузить какую-либо совместно используемую библиотеку, FreeBSD сперва будет пытаться найти ее в этом дереве. Так, если программа загружает, например, /lib/libc.so, FreeBSD попытается открыть /compat/linux/lib/libc.so, и если такого файла не существует, будет пытаться открыть /lib/libc.so. Разделяемые библиотеки должны находиться в теневом дереве, а не в каталогах, выдаваемых загрузчиком Linux `ld.so`.

Обычно вам придется добавлять совместно используемые библиотеки, от которых зависят Linux-программы, только при нескольких первых установках приложений Linux на вашу систему FreeBSD. По мере работы, у вас в системе накопится достаточный набор совместно используемых библиотек Linux для запуска новых Linux-программ без дополнительных действий.

10.2.1.3. Как установить дополнительные совместно используемые библиотеки

Что, если при установленном linux_base порте ваше приложение все равно сообщает об отсутствии необходимой библиотеки? Как узнать, какая именно нужна библиотека и где ее взять? В принципе, есть два способа. Вам необходимо иметь привилегии пользователя `root` для их осуществления.

Если у вас есть доступ к машине, на которой установлен Linux, узнайте, какие библиотеки использует Linux-приложение, и просто скопируйте из на свою машину. Рассмотрим следующий пример:

Допустим, вы скачали по FTP Linux-версию Doom и установили ее на Linux-машине. Вы можете узнать, какие совместно используемые библиотеки нужны Doom, с помощью команды `ldd linuxdoom`:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

Вам потребуются все файлы, перечисленные в последнем столбце. Скопируйте их в дерево /compat/linux на вашей системе, а также создайте символические ссылки на эти файлы с именами из первого столбца, соответственно. В итоге, у вас в системе FreeBSD должны быть следующие файлы:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Учтите, что если у вас уже есть совместно используемая библиотека Linux с соответствующим первому столбцу результатов `ldd` основным номером версии, вам не обязательно копировать файл, указанный в последнем столбце, в вашу систему. Уже существующий файл должен подойти. Рекомендуется, однако, все равно скопировать совместно используемую библиотеку, если ее версия новее. Предыдущую версию библиотеки можно удалить, если вы создали символическую ссылку на новую. Итак, если у вас в системе есть следующие библиотеки:

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

и какое-либо приложение требует библиотеку более поздней версии, судя по результатам команды `ldd`:

```
libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
```

Если версии немного отличаются в последней цифре, копировать /lib/libc.so.4.6.29 необязательно, так как программа, скорее всего, будет нормально работать и с немного устаревшей версией. Тем не менее, вы можете заменить libc.so:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Символические ссылки важны *только* для Linux-программ. Runtime-компоновщик FreeBSD самостоятельно подберет правильные номера версий библиотек, и вам не нужно об этом беспокоиться.

10.2.2. Установка двоичных файлов Linux ELF

Для ELF-файлов иногда требуется сделать так называемый "branding" (маркировать его). Если попытаться запустить не маркированный ELF-файл, вы получите следующее сообщение об ошибке:

```
% ./моя-linux-elf-программа
ELF binary type not known
Abort
```

Чтобы помочь ядру FreeBSD отличить ELF-файл FreeBSD от двоичного файла Linux, используется утилита brandelf(1):

```
% brandelf -t Linux моя-linux-elf-программа
```

В настоящее время набор инструментальных средств GNU (GNU toolchain) помещает необходимую маркировочную информацию в двоичные ELF-файлы автоматически, поэтому необходимость в этом действии возникает всё реже.

10.2.3. Конфигурирование распознавания имен хостов

Если DNS не работает или вы получаете это сообщение:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

то вам нужно создать (изменить) файл /compat/linux/etc/host.conf, содержащий:

```
order hosts, bind
multi on
```

Таким образом, вы указываете, то сначала производится поиск в файле /etc/hosts, а только затем запрашивается DNS. Когда файл /compat/linux/etc/host.conf отсутствует, Linux-приложения находят файл /etc/host.conf для FreeBSD и сообщают о несовместимом синтаксисе. Если вы не настраивали сервер имен с помощью файла /etc/resolv.conf, уберите `bind` из файла /compat/linux/etc/host.conf.

10.3. Установка Mathematica®

Ниже описано, как установить Linux-версию пакета Mathematica® 5.X на систему FreeBSD.

Linux версия Mathematica® или Mathematica® for Students можно заказать непосредственно в компании Wolfram по адресу http://www.wolfram.com/.

10.3.1. Использование установщика Mathematica®

Сначала вы должны указать FreeBSD, что Linux бинарники от Mathematica® используют Linux ABI. Самый простой путь сделать это-установить марку ELF Linux по умолчанию для всех немаркированных двоичных файлов с помощью команды:

```
# sysctl kern.fallback_elf_brand=3
```

FreeBSD будет считать, что все немаркированные двоичные ELF-файлы используют Linux ABI, и вы сможете запустить MathInstaller прямо с CDROM.

Теперь, скопируйте файл MathInstaller на ваш жёсткий диск:

```
# mount /cdrom
# cp /cdrom/Unix/Installers/Linux/MathInstaller /localdir/
```

и в этом файле замените `/bin/sh` в в первой строке на `/compat/linux/bin/sh`. Этим мы убедимся, что установщик будет выполняться Linux версией sh(1). Дальше, замените все вхождения `Linux)` на `FreeBSD)` с помощью текстового редактора или с помощью скрипта, представленного ниже, в следующей главе. Это укажет установщику Mathematica®, вызывающему `uname -s` для определения операционной системы, относиться к FreeBSD, как к Linux подобной операционной системе. Теперь, запуск `MathInstaller` установит Mathematica®.

10.3.2. Modifying the Mathematica® Executables

Скрипты командной оболочки, которые Mathematica® создала во время установки, должны быть изменены перед тем, как вы сможете использовать их. Если вы выбрали /usr/local/bin в качестве директории для помещения исполняемых файлов Mathematica®, то вы обнаружите в этом каталоге ссылки на файлы math, mathematica, Mathematica, и MathKernel. В каждом из них замените `Linux)` на `FreeBSD)` с помощью текстового редактора или с помощью следующего скрипта командной оболочки:

```
#!/bin/sh
cd /usr/local/bin
for i in math mathematica Mathematica MathKernel
  do sed 's/Linux)/FreeBSD)/g' $i > $i.tmp
  sed 's/\/bin\/sh/\/compat\/linux\/bin\/sh/g' $i.tmp > $i
  rm $i.tmp
  chmod a+x $i
done
```

10.3.3. Получение пароля к пакету Mathematica®

Когда вы запустите Mathematica® в первый раз, у вас будет запрошен пароль. Если вы еще не получили пароль от Wolfram, запустите программу `mathinfo` в директории установки для получения вашего "machine ID". Этот machine ID основан исключительно на MAC адресе вашей первичной Ethernet карты, так что, вы не сможете использовать вашу копию Mathematica® на разных машинах.

При регистрации по электронной почте, по телефону или по факсу вы сообщаете "machine ID", а в ответ получаете пароль, состоящий из нескольких групп чисел.

10.3.4. Использование интерфейса Mathematica® по сети

Mathematica® использует специальные шрифты для отображения некоторых символов, которые отсутствуют в стандартных шрифтах (символы интегралов, сумм, греческий алфавит и другие). Протокол X требует, чтобы эти шрифты были установлены *локально*. Это означает, что вы должны скопировать эти шрифты с компакт-диска или хоста, на котором установлена Mathematica®, на вашу машину. Обычно эти шрифты находятся в каталоге

/cdrom/Unix/Files/SystemFiles/Fonts компакт-диска или в каталоге /usr/local/mathematica/SystemFiles/Fonts на диске. Собственно файлы со шрифтами находятся в подкаталогах Type1 и X. О том, как их использовать, читайте ниже.

Можно просто скопировать их в один из существующих каталогов шрифтов в каталоге /usr/X11R6/lib/X11/fonts. В этом случае придётся отредактировать файл fonts.dir, добавив в него названия шрифтов и изменив число шрифтов в первой строке. Можно также запустить программу mkfontdir(1), находясь в том каталоге, куда вы скопировали шрифты.

Есть альтернативный способ: скопировать каталоги в /usr/X11R6/lib/X11/fonts:

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
# mkfontdir
```

Теперь добавьте каталоги с новыми шрифтами в путь к шрифтам:

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

Если вы используете сервер Xorg, то можно просто прописать эти каталоги в файле xorg.conf.

Для сервера XFree86™, файл конфигурации XF86Config.

Если на вашем компьютере *нет* каталога /usr/X11R6/lib/X11/fonts/Type1, замените MathType1 на Type1 в предыдущем примере.

10.4. Установка Maple™

Maple™ - коммерческая математическая программа, аналогичная Mathematica®. Это программное обеспечение надо купить у http://www.maplesoft.com/, а потом зарегистрироваться там для получения файла лицензии. Для установки этого программного обеспечения в ОС FreeBSD используется следующая последовательность простых шагов.

1. Выполните скрипт командного интерпретатора INSTALL из дистрибутива. Выберите опцию "RedHat", когда будет предложено программой установки. Обычно установка выполняется в каталог /usr/local/maple.

2. Если вы этого ещё не сделали, купите лицензию на Maple™ в компании Maple Waterloo Software (http://register.maplesoft.com/) и скопируйте ее в файл /usr/local/maple/license/license.dat.

3. Установите диспетчер лицензий FLEXlm, выполнив скрипт установки INSTALL_LIC, входящий в состав Maple™. Укажите основное имя хоста вашей машины для сервера лицензий.

4. Исправьте файл /usr/local/maple/bin/maple.system.type с помощью следующего патча:

```
   ----- snip ------------------
*** maple.system.type.orig      Sun Jul  8 16:35:33 2001
--- maple.system.type   Sun Jul  8 16:35:51 2001
***************
*** 72,77 ****
--- 72,78 ----
          # the IBM RS/6000 AIX case
          MAPLE_BIN="bin.IBM_RISC_UNIX"
          ;;
+     "FreeBSD"|\
      "Linux")
          # the Linux/x86 case
        # We have two Linux implementations, one for Red Hat and
   ----- snip end of patch -----
```

Учтите, что после `"FreeBSD"|\` не должно быть никаких пробелов.

Этот патч заставляет Maple™ распознавать "FreeBSD" как тип Linux-системы. Скрипт командного интерпретатора bin/maple вызывает скрипт bin/maple.system.type, который, в свою очередь, вызывает `uname -a` для получения имени операционной системы. В зависимости от имени ОС он определяет, какие двоичные модули использовать.

5. Запустите сервер лицензий.

Следующий скрипт, установленный в файл /usr/local/etc/rc.d/lmgrd.sh, обеспечивает удобный способ запуска `lmgrd`:

```
   ----- snip ------------

#! /bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bi
n
PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
export PATH

LICENSE_FILE=/usr/local/maple/license/license.dat
LOG=/var/log/lmgrd.log

case "$1" in
```

```
start)
        lmgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
        echo -n " lmgrd"
        ;;
stop)
        lmgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
        ;;
*)
        echo "Usage: `basename $0` {start|stop}" 1>&2
        exit 64
        ;;
esac

exit 0
   ----- snip ------------
```

6. Выполните тестовый запуск Maple™:

```
% cd /usr/local/maple/bin
% ./xmaple
```

Программа должна запуститься и работать. Не забудьте написать в Maplesoft и сообщить, что хотели бы видеть версию специально для FreeBSD!

10.4.1. Типичные проблемы

- Работать с диспетчером лицензий FLEXlm может быть непросто. Дополнительную документацию по нему можно найти на сайте http://www.globetrotter.com/.
- `lmgrd` очень требователен к файлу лицензии и выдает дамп памяти при выявлении любых проблем. Правильный файл лицензии должен иметь следующий вид:

```
# ========================================================
# License File for UNIX Installations ("Pointer File")
# ========================================================
SERVER chillig ANY
#USE_SERVER
VENDOR maplelmg

FEATURE Maple maplelmg 2000.0831 permanent 1 XXXXXXXXXXXX \
         PLATFORMS=i86_r ISSUER="Waterloo Maple Inc." \
         ISSUED=11-may-2000 NOTICE=" Technische Universitat Wien" \
         SN=XXXXXXXXX
```

Серийный номер и ключ забиты символами 'X'. `chillig` - имя хоста.

Редактирование файла лицензий возможно, если только не трогать строку "FEATURE"

(которая защищена лицензионным ключом).

10.5. Установка MATLAB®

Этот документ описывает процесс установки Linux-версии MATLAB® версии 6.5 на систему FreeBSD. Эта программа работает вполне нормально, за исключением Java Virtual Machine™ (см. Связь с Java™ Runtime Environment).

Linux-версию MATLAB® можно приобрести непосредственно в компании The MathWorks на сайте http://www.mathworks.com. Не забудьте получить файл лицензии или инструкции по его созданию. По ходу дела дайте знать производителю, что хотели бы увидеть версию их ПО специально для FreeBSD.

10.5.1. Процесс установки MATLAB®

Для установки MATLAB® выполните следующие шаги:

1. Вставьте инсталляционный CD и смонтируйте его. Станьте пользователем `root`, как рекомендует скрипт установки. Для запуска скрипта установки наберите:

   ```
   # /compat/linux/bin/sh /cdrom/install
   ```

   Запускается графический инсталлятор. Если вы получаете сообщения о невозможности открыть дисплей, наберите `setenv HOME ~USER`, где *USER* - пользователь, от имени которого выполнена команда su(1).

2. При запросе корневого каталога MATLAB®, наберите: `/compat/linux/usr/local/matlab`.

   Чтобы упростить набор остальных команд в ходе установки, выполните в командном интерпретаторе следующую команду: `set MATLAB=/compat/linux/usr/local/matlab`

3. Отредактируйте файл лицензии в соответствии с инструкциями в полученной лицензии MATLAB®.

   Этот файл можно подготовить заранее с помощью любого текстового редактора и скопировать его в $MATLAB/license.dat до того, как инсталлятор попросит его отредактировать.

4. Завершите процесс установки.

В этот момент ваша установка MATLAB® завершена. Следующие шаги позволяют "связать" эту программу с вашей системой FreeBSD.

10.5.2. Запуск диспетчера лицензий

1. Создайте символические ссылки для скриптов диспетчера лицензий:

```
# ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW
# ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW
```

2. Создайте файл запуска /usr/local/etc/rc.d/flexlm.sh. Представленный ниже пример - измененная версия входящего в дистрибутив файла $MATLAB/etc/rc.lm.glnx86. Изменены местонахождения файлов и диспетчер лицензий запускается под эмулятором Linux.

```
#!/bin/sh
case "$1" in
  start)
        if [ -f /usr/local/etc/lmboot_TMW ]; then
              /compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u username &&
echo 'MATLAB_lmgrd'
        fi
        ;;
  stop)
        if [ -f /usr/local/etc/lmdown_TMW ]; then
            /compat/linux/bin/sh /usr/local/etc/lmdown_TMW  > /dev/null 2>&1
        fi
        ;;
  *)
        echo "Usage: $0 {start|stop}"
        exit 1
        ;;
esac

exit 0
```

Этот файл надо сделать выполняемым:

```
# chmod +x /usr/local/etc/rc.d/flexlm.sh
```

Вы также должны заменить *username* именем пользователя в вашей системе (но не `root`).

3. Запустите диспетчер лицензий с помощью команды:

```
# /usr/local/etc/rc.d/flexlm.sh start
```

10.5.3. Связь с Java™ Runtime Environment

Измените ссылку Java™ Runtime Environment (JRE) так, чтобы он ссылалась на версию, работающую в FreeBSD:

```
# cd $MATLAB/sys/java/jre/glnx86/
# unlink jre; ln -s ./jre1.1.8 ./jre
```

10.5.4. Создание скрипта запуска MATLAB®

1. Поместите следующий скрипт запуска в файл /usr/local/bin/matlab:

   ```
   #!/bin/sh
   /compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"
   ```

2. Затем выполните команду `chmod +x /usr/local/bin/matlab`.

В зависимости от версии emulators/linux_base, при выполнении этого скрипта могут быть выданы сообщения об ошибках. Чтобы избежать этого, отредактируйте файл /compat/linux/usr/local/matlab/bin/matlab и измените строку вида:

```
if [ `expr "$lscmd" : '.*->.*'` -ne 0 ]; then
```

(в версии 13.0.1 это строка 410) на следующую строку:

```
if test -L $newbase; then
```

10.5.5. Создание скрипта остановки MATLAB®

Следующие действия необходимы для решения проблемы с некорректным завершением работы MATLAB®.

1. Создайте файл $MATLAB/toolbox/local/finish.m и поместите в него одну строку:

   ```
   ! $MATLAB/bin/finish.sh
   ```

`$MATLAB` - литерал.

В том же каталоге находятся файлы finishsav.m и finishdlg.m, которые позволяют сохранять рабочее пространство перед выходом. Если вы

используете любой из них, вставьте представленную выше строку сразу после команды `save`.

2. Создайте файл $MATLAB/bin/finish.sh, который будет содержать следующий скрипт:

```
#!/usr/compat/linux/bin/sh
(sleep 5; killall -1 matlab_helper) &
exit 0
```

3. Сделайте этот файл выполняемым:

```
# chmod +x $MATLAB/bin/finish.sh
```

10.5.6. Использование MATLAB®

В этот момент все готово для выполнения команды `matlab` и начала использования этой программы.

10.6. Установка Oracle®

10.6.1. Введение

Ниже описан процесс установки Oracle® 8.0.5 и Oracle® 8.0.5.1 Enterprise Edition для Linux на систему FreeBSD.

10.6.2. Установка Linux-среды

Удостоверьтесь, что порты emulators/linux_base и devel/linux_devtools установлены на вашей системе. Если у вас возникнут трудности с этими портами, воспользуйтесь пакетами или более ранними их версиями из Коллекции Портов.

Если вы хотите использовать интеллектуальный агент (intelligent agent), придется также установить пакет TCL от Red Hat: tcl-8.0.3-20.i386.rpm. Универсальная команда для установки пакетов с помощью официального порта RPM (archivers/rpm):

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm пакет
```

Установка этого *пакета* должна пройти без каких-либо ошибок.

10.6.3. Создание среды Oracle®

Прежде чем вы сможете установить Oracle®, необходимо настроить соответствующую среду. В этом документе описано, что *специально* нужно сделать, чтобы запустить Oracle® для Linux под FreeBSD - это не пересказ официального руководства по установке Oracle®.

10.6.3.1. Настройка ядра

Как описано в руководстве по установке Oracle®, необходимо установить максимальный размер совместно используемой (shared) памяти. Не используйте `SHMMAX` под FreeBSD. `SHMMAX` просто вычисляется, исходя из `SHMMAXPGS` и `PGSIZE`. Следовательно, нужно задавать `SHMMAXPGS`. За информацией о прочих опциях обратитесь к официальному руководству. Пример настроек:

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

Установите эти опции в зависимости от того, как и для чего вы будете использовать Oracle®.

Не забудьте добавить следующие строки в файл конфигурации ядра:

```
options SYSVSHM # совместно используемая память SysV
options SYSVSEM # семафоры SysV
options SYSVMSG # межпроцессное взаимодействие SysV
```

10.6.3.2. Учетная запись Oracle®

Создайте специальную учетную запись `oracle`, как и любую другую учетную запись. Единственное отличие в том, что для `oracle` необходимо указать командный интерпретатор Linux. Добавьте /compat/linux/bin/bash в /etc/shells и установите для `oracle` командный интерпретатор /compat/linux/bin/bash.

10.6.3.3. Переменные среды

Кроме стандартных переменных среды Oracle®, таких как `ORACLE_HOME` и `ORACLE_SID`, вам нужно будет установить следующие переменные среды:

Переменная	Значение
`LD_LIBRARY_PATH`	`$ORACLE_HOME/lib`
`CLASSPATH`	`$ORACLE_HOME/jdbc/lib/classes111.zip`
`PATH`	`/compat/linux/bin /compat/linux/sbin /compat/linux/usr/bin /compat/linux/usr/sbin /bin /sbin /usr/bin /usr/sbin /usr/local/bin $ORACLE_HOME/bin`

Желательно устанавливать все переменные среды в файле .profile. Вот реальный пример:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
```

```
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

10.6.4. Установка Oracle®

Из-за небольшой несовместимости с Linux-эмулятором, вам нужно будет создать подкаталог .oracle в каталоге /var/tmp прежде, чем можно будет начать установку. Сделайте ее владельцем пользователя `oracle`. Если вы все сделали правильно, то установка Oracle® должна пройти без проблем. Если какие-либо трудности все же возникли, проверьте еще раз все конфигурационные файлы и/или целостность дистрибутива Oracle®. После окончания установки Oracle® примените патчи, описанные в следующих двух подразделах.

Одна из часто возникающих проблем - неправильно установленный адаптер TCP-протокола. В результате, невозможно запустить процессы прослушивания TCP. Вот решение проблемы:

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

Не забудьте повторно запустить root.sh!

10.6.4.1. Изменение root.sh

При установке Oracle® необходимо выполнить некоторые действия от имени пользователя `root`. Они записаны в скрипте командного интерпретатора root.sh, который находится в каталоге orainst. Перед запуском, примените к нему следующий патч (исправляет местонахождение утилиты chown), либо запускайте его в командном интерпретаторе Linux.

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
***************
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
```

```
! CHOWN=/bin/chown
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

Если вы устанавливаете Oracle® не с компакт-диска, можно изменить исходный файл root.sh. Он называется rthd.sh и находится в каталоге orainst.

10.6.4.2. Изменение genclntsh

Скрипт `genclntsh` используется для того, чтобы создать единую совместно используемую клиентскую библиотеку, которая используется для создания демонстраций. Примените следующий патч, чтобы закомментировать определение переменной `PATH`:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
***************
*** 32,38 ****
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

10.6.5. Запуск Oracle®

Теперь, если вы все сделали правильно, то можете использовать Oracle® так же, как и в системе Linux.

10.7. Установка SAP® R/3®

Инсталляции систем SAP® на базе FreeBSD не будут поддерживаться службой поддержки SAP® - они предоставляют поддержку только на сертифицированных платформах.

10.7.1. Предисловие

Этот документ описывает возможный способ установки системы SAP® R/3® с СУБД Oracle® Database на машине с ОС FreeBSD, включая установку FreeBSD и Oracle®. Будут описаны две разные конфигурации:

- SAP® R/3® 4.6B (IDES) с Oracle® 8.0.5 на FreeBSD 4.3-STABLE
- SAP® R/3® 4.6C с Oracle® 8.1.7 на FreeBSD 4.5-STABLE

Хотя в этом документе мы пытаемся подробно описать все важные шаги, он не заменяет руководства по установке Oracle® и SAP® R/3®.

По специфическим вопросам SAP® и Oracle® обратитесь к документации, поставляемой в составе SAP® R/3® Linux edition, а также к другим источникам информации об Oracle® и SAP® OSS.

10.7.2. Программное обеспечение

Для установки SAP® были использованы следующие диски CD-ROM:

10.7.2.1. SAP® R/3® 4.6B, Oracle® 8.0.5

Имя	Номер	Описание
KERNEL	51009113	SAP Kernel Oracle / Installation / AIX, Linux, Solaris
RDBMS	51007558	Oracle / RDBMS 8.0.5.X / Linux
EXPORT1	51010208	IDES / DB-Export / Диск 1 из 6
EXPORT2	51010209	IDES / DB-Export / Диск 2 из 6
EXPORT3	51010210	IDES / DB-Export / Диск 3 из 6
EXPORT4	51010211	IDES / DB-Export / Диск 4 из 6
EXPORT5	51010212	IDES / DB-Export / Диск 5 из 6
EXPORT6	51010213	IDES / DB-Export / Диск 6 из 6

Кроме того, мы использовали CD Oracle® 8 Server (Опытная версия 8.0.5 для Linux, ядро версии 2.0.33), который не обязательно понадобится, и FreeBSD 4.3-STABLE (она вышла всего через несколько дней после 4.3 RELEASE).

10.7.2.2. SAP® R/3® 4.6C SR2, Oracle® 8.1.7

Имя	Номер	Описание
KERNEL	51014004	SAP Kernel Oracle / SAP Kernel Version 4.6D / DEC, Linux
RDBMS	51012930	Oracle 8.1.7/ RDBMS / Linux

Имя	Номер	Описание
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 1 из 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 2 из 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 3 bp 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 4 из 4
LANG1	51013954	Release 4.6C SR2 / Language / DE, EN, FR / Диск 1 из 3

В зависимости от языков, которые необходимо установить, могут потребоваться дополнительные CD. Здесь мы использовали только немецкий и английский языки, поэтому потребовался только первый языковой CD. Обратите внимание, что номера всех EXPORT CD идентичны. Все три языковых CD тоже имеют один номер (это отличается от нумерации CD в версии 4.6B IDES). На момент написания этого раздела (20.03.2002) установленное ПО работало на FreeBSD 4.5-STABLE.

10.7.3. Примечания по SAP®

Прочитайте следующие документы перед установкой SAP® R/3® - они пригодятся в ходе установки:

10.7.3.1. SAP® R/3® 4.6B, Oracle® 8.0.5

Номер	Название
0171356	SAP Software on Linux: Essential Comments
0201147	INST: 4.6C R/3 Inst. on UNIX - Oracle
0373203	Update / Migration Oracle 8.0.5 -→ 8.0.6/8.1.6 LINUX
0072984	Release of Digital UNIX 4.0B for Oracle
0130581	R3SETUP step DIPGNTAB terminates
0144978	Your system has not been installed correctly
0162266	Questions and tips for R3SETUP on Windows NT / W2K

10.7.3.2. SAP® R/3® 4.6C, Oracle® 8.1.7

Номер	Название
0015023	Initializing table TCPDB (RSXP0004) (EBCDIC)
0045619	R/3 with several languages or typefaces

Номер	Название
0171356	SAP Software on Linux: Essential Comments
0195603	RedHat 6.1 Enterprise version: Known problems
0212876	The new archiving tool SAPCAR
0300900	Linux: Released DELL Hardware
0377187	RedHat 6.2: important remarks
0387074	INST: R/3 4.6C SR2 Installation on UNIX
0387077	INST: R/3 4.6C SR2 Inst. on UNIX - Oracle
0387078	SAP Software on UNIX: OS Dependencies 4.6C SR2

10.7.4. Требования к аппаратному обеспечению

Следующего оборудования достаточно для установки SAP® R/3® System. Для производственного использования необходима более точная оценка параметров:

Компонент	4.6B	4.6C
Процессор	2 x 800MHz Pentium® III	2 x 800MHz Pentium® III
Память	1GB ECC	2GB ECC
Объем дисков	50-60GB (IDES)	50-60GB (IDES)

Для производственного использования рекомендуются процессоры Xeon™ с большим кешем, высокоскоростной доступ к дискам (SCSI, аппаратный RAID-контроллер), USV и ECC-RAM. Большой объем дискового пространства связан с заранее сконфигурированной системой IDES, которая создает 27 Гбайт файлов базы данных по ходу установки. Этого пространства также достаточно для исходных производственных систем и прикладных данных.

10.7.4.1. SAP® R/3® 4.6B, Oracle® 8.0.5

Было использовано следующее стандартное оборудование: двухпроцессорная материнская плата с двумя процессорами Pentium® III 800 MHz, SCSI-контроллером Adaptec® 29160 Ultra160 (для работы с 40/80 Гбайт стримером DLT и приводом CDROM), Mylex® AcceleRAID™ (2 канала, firmware 6.00-1-00 с 32 Мбайт RAM). К RAID-контроллеру Mylex® подключены два диска по 17 Гбайт (зеркалированы) и четыре диска по 36 Гбайт (RAID уровня 5).

10.7.4.2. SAP® R/3® 4.6C, Oracle® 8.1.7

Для этой установки был использован Dell™ PowerEdge™ 2500: двухпроцессорная плата с двумя процессорами Pentium® III 1000 MHz (256 Кбайт кэш), 2 Гбайта PC133 ECC SDRAM, PERC/3 DC PCI RAID-контроллер со 128 Мбайтами и приводом EIDE DVD-ROM. К RAID-контроллеру подключены два диска по 18 Гбайт (зеркалированы) и четыре диска по 36 Гбайт (RAID уровня 5).

10.7.5. Установка FreeBSD

Сначала надо установить FreeBSD. Есть несколько способов сделать это; подробнее см. Подготовка собственного источника установки.

10.7.5.1. Компоновка дисков

Для простоты диски при установке SAP® R/3® 46B и SAP® R/3® 46C SR2 использовались одинаково. Изменились только имена устройств, поскольку установка выполнялась на разное оборудование (/dev/da и /dev/amr, соответственно, так что при использовании AMI MegaRAID® будут задействованы устройства /dev/amr0s1a вместо /dev/da0s1a):

Файловая система	Размер (в блока по 1 Кбайту)	Размер (Гбайт)	Смонтирована в
/dev/da0s1a	1.016.303	1	/
/dev/da0s1b		6	swap
/dev/da0s1e	2.032.623	2	/var
/dev/da0s1f	8.205.339	8	/usr
/dev/da1s1e	45.734.361	45	/compat/linux/oracle
/dev/da1s1f	2.032.623	2	/compat/linux/sapmnt
/dev/da1s1g	2.032.623	2	/compat/linux/usr/sap

Конфигурируем и инициализируем заранее два логических диска с помощью ПО RAID Mylex® или PERC/3. Программы конфигурирования можно запустить в ходе загрузки BIOS.

Обратите внимание, что использованная компоновка дисков немного отличается от рекомендованной SAP®, поскольку SAP® рекомендует монтировать подкаталоги Oracle® (и некоторые другие) отдельно - мы решили просто создать подкаталоги.

10.7.5.2. `make world` и новое ядро

Загрузите последние исходные коды ветки -STABLE. Пересоздайте систему и ваше специализированное ядро после необходимых изменений в файле конфигурации ядра. В него надо включить параметры ядра, требуемые для SAP® R/3® и Oracle®.

10.7.6. Установка среды Linux

10.7.6.1. Установка базовой системы Linux

Сначала необходимо установить порт linux_base (от имени пользователя `root`):

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

10.7.6.2. Установка среды разработки Linux

Среда разработки Linux необходима, если вы хотите установить Oracle® в ОС FreeBSD, как описано в Установка Oracle®:

```
# cd /usr/ports/devel/linux_devtools
# make install distclean
```

Среда разработки Linux была установлена только для SAP® R/3® 46B IDES. Она не нужна, если сервер Oracle® не перекомпоновывается в системе FreeBSD. Именно так и происходит, если вы используете tar-архив (tarball) Oracle® с Linux-системы.

10.7.6.3. Установка необходимых пакетов RPM

Для запуска программы `R3SETUP` необходима поддержка модулей PAM. В ходе первой установки SAP® на ОС FreeBSD 4.3-STABLE мы попытались установить PAM со всеми необходимыми пакетами и, в конце концов, принудительно установили пакет PAM, что и сработало. Для SAP® R/3® 4.6C SR2 мы сразу принудительно установили PAM RPM, что тоже сработало, так что похоже, что пакеты, от которых декларирована зависимость, не нужны:

```
# rpm -i --ignoreos --nodeps --root /compat/linux --dbpath /var/lib/rpm \
pam-0.68-7.i386.rpm
```

Чтобы Oracle® 8.0.5 мог запустить интеллектуального агента, нам также пришлось установить пакет RedHat Tcl tcl-8.0.5-30.i386.rpm (иначе перекомпоновка в ходе установки Oracle® не выполнится). Есть и другие проблемы с перекомпоновкой Oracle®, но это проблема Oracle® на Linux, не связанная с особенностями FreeBSD.

10.7.6.4. Дополнительные советы

Может также иметь смысл добавить `linprocfs` в /etc/fstab; подробнее об этом см. на странице справочника linprocfs(5). Еще можно установить параметр `kern.fallback_elf_brand=3` в файле /etc/sysctl.conf.

10.7.7. Создание среды SAP® R/3®

10.7.7.1. Создание необходимых файловых систем и точек монтирования

Для простой установки достаточно создать следующие файловые системы:

точка монтирования	размер в Гбайтах
/compat/linux/oracle	45 GB
/compat/linux/sapmnt	2 GB
/compat/linux/usr/sap	2 GB

Также необходимо создать несколько ссылок. В противном случае, инсталлятор SAP® будет

выдавать сообщения об ошибках, поскольку он проверяет созданные ссылки:

```
# ln -s /compat/linux/oracle /oracle
# ln -s /compat/linux/sapmnt /sapmnt
# ln -s /compat/linux/usr/sap /usr/sap
```

Возможные сообщения об ошибках в ходе установки (в данном случае, для установки System *PRD* и SAP® R/3® 4.6C SR2):

```
INFO 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:200
    Checking existence of symbolic link /usr/sap/PRD/SYS/exe/dbg to
    /sapmnt/PRD/exe. Creating if it does not exist...

WARNING 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:400
    Link /usr/sap/PRD/SYS/exe/dbg exists but it points to file
    /compat/linux/sapmnt/PRD/exe instead of /sapmnt/PRD/exe. The
    program cannot go on as long as this link exists at this
    location. Move the link to another location.

ERROR 2002-03-19 16:45:36 R3LINKS_IND_IND Ins_SetupLinks:0
    can not setup link '/usr/sap/PRD/SYS/exe/dbg' with content
    '/sapmnt/PRD/exe'
```

10.7.7.2. Создание пользователей и каталогов

Для системы SAP® R/3® необходимы два пользователя и три группы. Имена пользователей зависят от идентификатора системы SAP® (SID), который состоит из трех букв. Некоторые из этих идентификаторов зарезервированы SAP® (например, `SAP` и `NIX`. Полный список см. в документации SAP®). Для установки IDES мы использовали `IDS`, а для установки 4.6C SR2 - `PRD`, поскольку эта система предназначалась для промышленного использования. Поэтому нам понадобились следующие группы (идентификаторы групп могут отличаться, мы просто указали наши значения, использованные при установке):

идентификатор группы	имя группы	описание
100	dba	Администратор базы данных
101	sapsys	Система SAP®
102	oper	Оператор базы данных

Для стандартной установки Oracle® используется только группа `dba`. В качестве группы `oper` используется та же группа `dba` (подробнее об этом см. в документации Oracle® и SAP®).

Нам также нужны следующие пользователи:

идентификатор пользователя	имя пользователя	общий вид имени	группа	дополнительные группы	описание
1000	idsadm/prdadm	_sid_adm	sapsys	oper	Администратор SAP®
1002	oraids/oraprd	ora_sid_	dba	oper	Администратор Oracle®

Добавление пользователей с помощью adduser(8) требует следующих параметров для "Администратора SAP®" (обратите внимание на командный интерпретатор и начальный каталог):

```
Name: sidadm
Password: ******
Fullname: SAP Administrator SID
Uid: 1000
Gid: 101 (sapsys)
Class:
Groups: sapsys dba
HOME: /home/sidadm
Shell: bash  (/compat/linux/bin/bash)
```

а для "Администратора Oracle®":

```
Name: orasid
Password: ******
Fullname: Oracle Administrator SID
Uid: 1002
Gid: 100 (dba)
Class:
Groups: dba
HOME: /oracle/sid
Shell: bash  (/compat/linux/bin/bash)
```

Для него также надо указать группу `oper`, если вы используете обе группы, `dba` и `oper`.

10.7.7.3. Создание каталогов

Эти каталоги обычно создаются в отдельных файловых системах. Впрочем, все зависит от ваших требований. Мы решили создавать их как обычные каталоги, поскольку в любом случае они находятся на одном массиве RAID 5:

Сначала мы установим владельцев и права для ряда каталогов (от имени пользователя `root`):

```
# chmod 775 /oracle
```

```
# chmod 777 /sapmnt
# chown root:dba /oracle
# chown sidadm:sapsys /compat/linux/usr/sap
# chmod 775 /compat/linux/usr/sap
```

Затем, мы создадим каталоги от имени пользователя `ora_sid_`. Все они будут размещены в каталоге /oracle/SID:

```
# su - orasid
# cd /oracle/SID
# mkdir mirrlogA mirrlogB origlogA origlogB
# mkdir sapdata1 sapdata2 sapdata3 sapdata4 sapdata5 sapdata6
# mkdir saparch sapreorg
# exit
```

Для установки Oracle® 8.1.7 потребуется несколько дополнительных каталогов:

```
# su - orasid
# cd /oracle
# mkdir 805_32
# mkdir client stage
# mkdir client/80x_32
# mkdir stage/817_32
# cd /oracle/SID
# mkdir 817_32
```

Каталог client/80x_32 должен иметь точно такое имя. Не заменяйте *x* числом или другим символом.

На третьем шаге мы создаем каталоги от имени пользователя `_sid_adm`:

```
# su - sidadm
# cd /usr/sap
# mkdir SID
# mkdir trans
# exit
```

10.7.7.4. Записи в файле /etc/services

Для системы SAP® R/3® необходим ряд записей в файле /etc/services, которые не будут правильно созданы при установке под FreeBSD. Пожалуйста, добавьте следующие записи (потребуются, по крайней мере, записи, соответствующие номеру экземпляра - в данном случае, `00`. Не повредит добавить все записи, от `00` до `99`, для `dp`, `gw`, `sp` и `ms`). Если вы собираетесь использовать SAProuter или вам необходим доступ к SAP® OSS, потребуется также запись для `99`, поскольку порт 3299 обычно используется для процесса SAProuter в целевой системе:

```
sapdp00  3200/tcp # SAP Dispatcher.      3200 + Instance-Number
sapgw00  3300/tcp # SAP Gateway.         3300 + Instance-Number
sapsp00  3400/tcp #                      3400 + Instance-Number
sapms00  3500/tcp #                      3500 + Instance-Number
sapmsSID 3600/tcp # SAP Message Server.  3600 + Instance-Number
sapgw00s   4800/tcp # SAP Secure Gateway   4800 + Instance-Number
```

10.7.7.5. Необходимые локали

Системе SAP® нужны, по крайней мере, две локали, не входящие в стандартную установку RedHat. SAP® предлагает необходимые пакеты RPM для загрузки со своего FTP-сервера (который доступен только для клиентов с доступом к OSS). См. список необходимых пакетов RPM в заметке 0171356.

Можно также просто создать соответствующие ссылки (например, с *de_DE* и *en_US*), но мы не рекомендуем это для производственной системы (хотя это и сработало для системы IDES безо всяких проблем). Необходимы следующие локали:

```
de_DE.ISO-8859-1
en_US.ISO-8859-1
```

Создайте ссылки следующим образом:

```
# cd /compat/linux/usr/shared/locale
# ln -s de_DE de_DE.ISO-8859-1
# ln -s en_US en_US.ISO-8859-1
```

Если их не будет, в ходе установки возникнет ряд проблем. Если их просто проигнорировать (установив `STATUS` для соответствующих шагов равным `OK` в файле CENTRDB.R3S), нельзя будет зарегистрироваться в системе SAP® без дополнительных усилий.

10.7.7.6. Настройка ядра

Системам SAP® R/3® надо много ресурсов. Поэтому мы добавили следующие параметры в файл конфигурации ядра:

```
# Для пожирателей памяти (SAP и Oracle):
options MAXDSIZ="(1024*1024*1024)"
options DFLDSIZ="(1024*1024*1024)"
# Необходимые опции System V.
options SYSVSHM #совместно используемая память в стиле SYSV
options SHMMAXPGS=262144 #макс. количество страниц совместно используемой
#options SHMMAXPGS=393216 #use this for the 46C inst.parameters
options SHMMNI=256 #макс. количество идентификаторов совместно используемой
options SHMSEG=100 #макс. количество сегментов разд. памяти на процесс
options SYSVMSG #очереди сообщений в стиле SYSV
```

```
options MSGSEG=32767 #макс. количество сегментов сообщений в системе
options MSGSSZ=32 #размер сегмента сообщений. ДОЛЖЕН быть степенью 2
options MSGMNB=65535 #макс. символов на очередь сообщений
options MSGTQL=2046 #макс. количество сообщений в системе
options SYSVSEM #семафоры в стиле SYSV
options SEMMNU=256 #количество структур UNDO семафоров
options SEMMNS=1024 #количество семафоров в системе
options SEMMNI=520 #количество идентификаторов семафоров
options SEMUME=100       #количество ключей UNDO
```

Минимальные значения указаны в документации, поставляемой вместе с SAP®. Поскольку описания для Linux нет, дополнительную информацию см. в разделе HP-UX (32-bit). Поскольку в системе, на которую устанавливалась версия 4.6C SR2, оперативной памяти больше, сегменты совместно используемой можно сделать больше как для SAP®, так и для Oracle®, поэтому задайте большее количество страниц совместно используемой памяти.

При стандартной установке FreeBSD на i386™, задайте значения `MAXDSIZ` и `DFLDSIZ` не более 1 Гбайта. В противном случае могут выдаваться странные ошибки вроде `ORA-27102: out of memory` и `Linux Error: 12: Cannot allocate memory`.

10.7.8. Установка SAP® R/3®

10.7.8.1. Подготовка дисков CDROM SAP®

В ходе установки придется монтировать и демонтировать много дисков CDROM. При наличии достаточного количества приводов CDROM, можно смонтировать их все. Мы же решили скопировать содержимое дисков CDROM в соответствующие каталоги:

```
/oracle/SID/sapreorg/имя_cd
```

где *имя_cd* - одно из следующих KERNEL, RDBMS, EXPORT1, EXPORT2, EXPORT3, EXPORT4, EXPORT5 и EXPORT6 для установки 4.6B/IDES, и KERNEL, RDBMS, DISK1, DISK2, DISK3, DISK4 и LANG для установки 4.6C SR2. Все имена файлов на смонтированных дисках должны быть в верхнем регистре, в противном случае, используйте при монтировании опцию `-g`. Поэтому используйте следующие команды:

```
# mount_cd9660 -g /dev/cd0a /mnt
# cp -R /mnt/* /oracle/SID/sapreorg/имя_cd
# umount /mnt
```

10.7.8.2. Запуск скрипта установки

Сначала надо подготовить каталог install:

```
# cd /oracle/SID/sapreorg
```

```
# mkdir install
# cd install
```

Затем запускается скрипт установки, который скопирует почти все необходимые файлы в каталог install:

```
# /oracle/SID/sapreorg/KERNEL/UNIX/INSTTOOL.SH
```

Дистрибутив IDES (4.6B) включает полностью настроенную демонстрационную систему SAP® R/3®, поэтому он включает шесть дисков EXPORT CD, а не три. В этот момент шаблон установки CENTRDB.R3S предназначен для установки стандартного центрального экземпляра (R/3® и базы данных), а не для установки центрального экземпляра IDES, поэтому необходимо скопировать соответствующий файл CENTRDB.R3S из каталога EXPORT1, иначе команда `R3SETUP` запросит только три диска EXPORT CDs.

Более новый релиз SAP® 4.6C SR2 включает четыре диска EXPORT CD. Шаги установки определяет файл параметров CENTRAL.R3S. В отличие от прежних релизов, больше нет отдельных шаблонов установки для центрального экземпляра с базой данных или без нее. SAP® использует отдельный шаблон для установки базы данных. Для перезапуска установки в дальнейшем, однако, достаточно перезапустить исходный файл.

В ходе установки и после нее SAP® требует, чтобы команда `hostname` возвращала только имя компьютера, не уточнённое именем домена. Поэтому либо задайте имя хоста в соответствии с этим требованием, либо настройте псевдоним с помощью команды `alias hostname='hostname -s'` для пользователей `ora_sid_` и `_sid_adm` (и для пользователя `root`, про крайней мере, в ходе шагов по установке, выполняемых от имени `root`). Можно также изменить файлы .profile и .login для обоих пользователей, которые создаются в ходе установки SAP®.

10.7.8.3. Запуск `R3SETUP` 4.6B

Проверьте, что переменная среды `LD_LIBRARY_PATH` установлена правильно:

```
# export LD_LIBRARY_PATH=/oracle/IDS/lib:/sapmnt/IDS/exe:/oracle/805_32/lib
```

Выполните команду `R3SETUP` от имени пользователя `root` из каталога установки:

```
# cd /oracle/IDS/sapreorg/install
# ./R3SETUP -f CENTRDB.R3S
```

Скрипт затем задает ряд вопросов (стандартные ответы даны в скобках, а затем представлены реальные ответы):

Вопрос	Стандартное значение	Ответ
Enter SAP System ID	[C11]	IDS`Enter`

Вопрос	Стандартное значение	Ответ
Enter SAP Instance Number	[00]	`Enter`
Enter SAPMOUNT Directory	[/sapmnt]	`Enter`
Enter name of SAP central host	[troubadix.domain.de]	`Enter`
Enter name of SAP db host	[troubadix]	`Enter`
Select character set	[1] (WE8DEC)	`Enter`
Enter Oracle server version (1) Oracle 8.0.5, (2) Oracle 8.0.6, (3) Oracle 8.1.5, (4) Oracle 8.1.6		1 `Enter`
Extract Oracle Client archive	[1] (Yes, extract)	`Enter`
Enter path to KERNEL CD	[/sapcd]	/oracle/IDS/sapreorg/KERNEL
Enter path to RDBMS CD	[/sapcd]	/oracle/IDS/sapreorg/RDBMS
Enter path to EXPORT1 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT1
Directory to copy EXPORT1 CD	[/oracle/IDS/sapreorg/CD4_DIR]	`Enter`
Enter path to EXPORT2 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT2
Directory to copy EXPORT2 CD	[/oracle/IDS/sapreorg/CD5_DIR]	`Enter`
Enter path to EXPORT3 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT3
Directory to copy EXPORT3 CD	[/oracle/IDS/sapreorg/CD6_DIR]	`Enter`
Enter path to EXPORT4 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT4
Directory to copy EXPORT4 CD	[/oracle/IDS/sapreorg/CD7_DIR]	`Enter`
Enter path to EXPORT5 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT5
Directory to copy EXPORT5 CD	[/oracle/IDS/sapreorg/CD8_DIR]	`Enter`
Enter path to EXPORT6 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT6
Directory to copy EXPORT6 CD	[/oracle/IDS/sapreorg/CD9_DIR]	`Enter`
Enter amount of RAM for SAP + DB		850 `Enter` (in Megabytes)
Service Entry Message Server	[3600]	`Enter`
Enter Group-ID of sapsys	[101]	`Enter`
Enter Group-ID of oper	[102]	`Enter`
Enter Group-ID of dba	[100]	`Enter`
Enter User-ID of _sid_adm	[1000]	`Enter`
Enter User-ID of ora_sid_	[1002]	`Enter`
Number of parallel procs	[2]	`Enter`

Если вы не скопировали диски в разные каталоги, инсталлятор SAP® не сможет найти необходимые CD (идентифицируемые файлом LABEL.ASC на диске) и попросит затем

вставить и смонтировать CD, и подтвердить или ввести точку его монтирования.

Файл CENTRDB.R3S может содержать ошибки. В нашем случае, он снова запросил EXPORT4 CD, но указал корректный ключ (6_LOCATION, затем 7_LOCATION и т.д.), так что, можно просто продолжать вводить корректные значения.

За исключением нескольких упомянутых ниже проблем, все должно идти нормально до момента, когда придется устанавливать программное обеспечение для работы с базой данных Oracle®.

10.7.8.4. Запуск `R3SETUP` 4.6C SR2

Проверьте, что переменная среды `LD_LIBRARY_PATH` установлена правильно. Это значение отличается от использованного при установке версии 4.6B с Oracle® 8.0.5:

```
# export LD_LIBRARY_PATH=/sapmnt/PRD/exe:/oracle/PRD/817_32/lib
```

Выполните команду `R3SETUP` от имени пользователя `root` из каталога установки:

```
# cd /oracle/PRD/sapreorg/install
# ./R3SETUP -f CENTRAL.R3S
```

Скрипт затем задаст ряд вопросов (стандартные значения даны в скобках, а затем идут реальные ответы):

Вопрос	Стандартное значение	Ответ
Enter SAP System ID	[C11]	PRD `Enter`
Enter SAP Instance Number	[00]	`Enter`
Enter SAPMOUNT Directory	[/sapmnt]	`Enter`
Enter name of SAP central host	[majestix]	`Enter`
Enter Database System ID	[PRD]	PRD `Enter`
Enter name of SAP db host	[majestix]	`Enter`
Select character set	[1] (WE8DEC)	`Enter`
Enter Oracle server version (2) Oracle 8.1.7		2 `Enter`
Extract Oracle Client archive	[1] (Yes, extract)	`Enter`
Enter path to KERNEL CD	[/sapcd]	/oracle/PRD/sapreorg/KERNEL
Enter amount of RAM for SAP + DB	2044	1800 `Enter` (in Megabytes)
Service Entry Message Server	[3600]	`Enter`
Enter Group-ID of sapsys	[100]	`Enter`

Вопрос	Стандартное значение	Ответ
Enter Group-ID of oper	[101]	`Enter`
Enter Group-ID of dba	[102]	`Enter`
Enter User-ID of `oraprd`	[1002]	`Enter`
Enter User-ID of `prdadm`	[1000]	`Enter`
LDAP support		3 `Enter` (no support)
Installation step completed	[1] (continue)	`Enter`
Choose installation service	[1] (DB inst,file)	`Enter`

Пока создание пользователей дает сообщение об ошибке при установке на стадиях OSUSERDBSID_IND_ORA (создание пользователя `ora_sid_`) и OSUSERSIDADM_IND_ORA (создание пользователя `_sid_adm`).

За исключением некоторых упомянутых далее проблем, все должно идти нормально до момента, когда придется устанавливать программное обеспечение для работы с базой данных Oracle®.

10.7.9. Установка Oracle® 8.0.5

Описания возможных проблем с Linux и Сервером Oracle® см. в соответствующих файлах SAP® Notes и Oracle® Readme. Большинство, если не все проблемы, связаны с несовместимыми библиотеками.

Подробнее об установке Oracle® см. в разделе Установка Oracle®.

10.7.9.1. Установка Oracle® 8.0.5 с помощью `orainst`

Если надо использовать Oracle® 8.0.5, для успешной перекомпоновки понадобится несколько дополнительных библиотек, поскольку Oracle® 8.0.5 был скомпонован со старой версией glibc (RedHat 6.0), но уже RedHat 6.1 использует новую библиотеку glibc. Так что, для успешной перекомпоновки нужно установить следующие дополнительные пакеты:

compat-libs-5.2-2.i386.rpm

compat-glibc-5.2-2.0.7.2.i386.rpm

compat-egcs-5.2-1.0.3a.1.i386.rpm

compat-egcs-c++-5.2-1.0.3a.1.i386.rpm

compat-binutils-5.2-2.9.1.0.23.1.i386.rpm

Дополнительную информацию см. в файлах SAP® Notes или Oracle® Readme. Если установить эти пакеты не представляется возможным (на момент установки у нас не было времени, чтобы это проверить), можно использовать исходные двоичные модули или перекомпонованные двоичные модули с исходной системы RedHat.

Для компиляции интеллектуального агента должен быть установлен пакет RedHat Tcl. Если вы не можете найти пакет tcl-8.0.3-20.i386.rpm, подойдет и более новый, вроде tcl-8.0.5-30.i386.rpm для RedHat 6.1.

За исключением перекомпоновки, установка выполняется просто:

```
# su - oraids
# export TERM=xterm
# export ORACLE_TERM=xterm
# export ORACLE_HOME=/oracle/IDS
# cd $ORACLE_HOME/orainst_sap
# ./orainst
```

Нажимайте на всех экранах клавишу `Enter`, пока программное обеспечение не будет установлено, убрав только пометку выбора с *Oracle® On-Line Text Viewer*, поскольку этого компонента для Linux сейчас нет. Oracle® затем захочет перекомпоновать модули с помощью `i386-glibc20-linux-gcc` вместо имеющихся `gcc`, `egcs` или `i386-redhat-linux-gcc`.

Из-за нехватки времени мы решили использовать двоичные модули из версии Oracle® 8.0.5 PreProduction после того, как первая попытка заставить работать версию с RDBMS CD провалилась, - попытки найти и загрузить требуемые пакеты RPM нам показались настоящим кошмаром.

10.7.9.2. Установка Oracle® 8.0.5 Pre-production Release для Linux (ядро 2.0.33)

Эту установку выполнить очень легко. Монтируем CD, запускаем инсталлятор. Затем он запрашивает местонахождение начального каталога Oracle® и копирует туда двоичные модули. Мы, однако, не удаляли остатки прежних попыток установить RDBMS.

В конечном итоге, базу данных Oracle® удалось запустить без проблем.

10.7.10. Установка tar-архива Oracle® 8.1.7 для Linux

Создайте tar-архив oracle81732.tgz каталога установки на Linux-системе и разархивируйте его в каталог /oracle/SID/817_32/.

10.7.11. Продолжение установки SAP® R/3®

Сначала проверьте настройку среды для пользователей `idsamd` (*sid_adm) и* `oraids` *(ora_sid*). У них обоих должны теперь быть файлы .profile, .login и .cshrc, использующие `hostname`. Если имя хоста в системе полностью уточнено, надо заменить `hostname` командой `hostname -s` во всех трех файлах.

10.7.11.1. Загрузка базы данных

Потом команду `R3SETUP` можно либо перезапустить, либо продолжить (в зависимости от того, была ли завершена ее работа). `R3SETUP` затем создает табличные пространства и загружает данные (для 46B IDES - с дисков от EXPORT1 до EXPORT6, для 46C - с дисков от DISK1 до DISK4) в базу данных с помощью утилиты `R3load`.

После завершения загрузки базы данных (это может занять несколько часов), будет запрошено несколько паролей. Для тестовых установок можно использовать хорошо известные стандартные пароли (но если защита важна - используйте другие!):

Вопрос	Ответ
Enter Password for sapr3	sap`Enter`
Confirum Password for sapr3	sap`Enter`
Enter Password for sys	change_on_install`Enter`
Confirm Password for sys	change_on_install`Enter`
Enter Password for system	manager`Enter`
Confirm Password for system	manager`Enter`

Мы столкнулись с несколькими проблемами с `dipgntab` при установке 4.6B.

10.7.11.2. Процесс прослушивания

Запустите процесс прослушивания (Oracle® Listener) от имени пользователя `ora_sid_` следующим образом:

```
% umask 0; lsnrctl start
```

В противном случае, вы можете получить сообщение об ошибке ORA-12546, поскольку у сокетов будут неправильные права доступа. См. SAP® Note 072984.

10.7.11.3. Обновление таблиц MNLS

Если вы планируете использовать в системе SAP® языки, для которых не подходит кодировка Latin-1, придется изменить таблицы Multi National Language Support. Эта процедура описана в SAP® OSS Notes 15023 и 45619. Если же нет, можете пропустить этот вопрос в ходе установки SAP®.

> Если вам не нужна поддержка MNLS, все равно необходимо проверить таблицу TCPDB и инициализировать ее, если это еще не было сделано. Дополнительную информацию см. в SAP® Note 0015023 и 0045619.

10.7.12. Шаги после установки

10.7.12.1. Запрос лицензионного ключа SAP® R/3®

Вы должны запросить ваш лицензионный ключ SAP® R/3®. Это необходимо, поскольку временная лицензия, использованная в ходе установки, действительна только четыре недели. Сначала получите ключ оборудования. Зарегистрируйтесь как пользователь `idsadm` и вызовите команду `saplicense`:

```
# /sapmnt/IDS/exe/saplicense -get
```

При вызове команды `saplicense` без параметров будет выдан список опций. После получения лицензионного ключа, его можно установить с помощью команды:

```
# /sapmnt/IDS/exe/saplicense -install
```

Затем вас попросят ввести следующие значения:

```
SAP SYSTEM ID   = SID, 3 символа
CUSTOMER KEY    = ключ оборудования, 11 символов
INSTALLATION NO = установка, 10 цифр
EXPIRATION DATE = yyyymmdd, обычно - "99991231"
LICENSE KEY     = лицензионный ключ, 24 символа
```

10.7.12.2. Создание пользователей

Создайте пользователя в клиенте 000 (некоторые задачи обязательно надо выполнять из клиента 000, от имени пользователя, отличающегося от `sap*` и `ddic`). В качестве имени пользователя мы обычно выбираем `wartung` (или `service`, по английски). Требуются профили `sap_new` и `sap_all`. Для дополнительной защиты надо изменить пароли стандартных пользователей на всех клиентах (в том числе, пользователей `sap*` и `ddic`).

10.7.12.3. Конфигурирование системы передачи, профиля, режимов работы и т.п.

В клиенте 000, от имени пользователя, отличающегося от `ddic` и `sap*`, выполните, как минимум, следующее:

Задача	Транзакция
Сконфигурируйте систему передачи, например, как *Stand-Alone Transport Domain Entity*	STMS
Создайте/Отредактируйте профиль для системы	RZ10
Сконфигурируйте режимы работы и экземпляры	RZ04

Эти и другие шаги, которые надо выполнить после установки, подробно описаны в руководствах по установке SAP®.

10.7.12.4. Редактирование initsid.sap (initIDS.sap)

Файл /oracle/IDS/dbs/initIDS.sap содержит профиль резервного копирования SAP®. Здесь надо задать размер используемой ленты, тип сжатия и т.д. Чтобы можно было использовать `sapdba` / `brbackup`, мы изменили следующие значения:

```
compress = hardware
archive_function = copy_delete_save
cpio_flags = "-ov --format=newc --block-size=128 --quiet"
cpio_in_flags = "-iuv --block-size=128 --quiet"
tape_size = 38000M
tape_address = /dev/nsa0
tape_address_rew = /dev/sa0
```

Объяснения:

`compress`: мы использовали ленту HP DLT1, которая поддерживает аппаратное сжатие.

`archive_function`: этот параметр задает стандартное поведение для сохранения архивных журналов Oracle®: новые журнальные файлы сохраняются на ленту, уже сохраненные файлы журнала сохраняются еще раз, а затем удаляются. Это предотвращает многочисленные проблемы, если потребуется восстановить базу данных, а одна из архивных лент окажется сбойной.

`cpio_flags`: по умолчанию используется `-B`, что устанавливает размер блока 5120 байт. Для лент DLT компания HP рекомендует размер блока не меньше 32 Кбайт, поэтому мы использовали значение `--block-size=128` для задания размера блока 64 Кбайта. Опция `--format=newc` необходима, поскольку у нас есть индексные дескрипторы (inodes) с номерами больше 65535. Последняя опция, `--quiet` необходима потому, что иначе команда `brbackup` выдает сообщение об ошибке, как только команда `cpio` выдаст количество сохраненных блоков.

`cpio_in_flags`: флаги, необходимые для загрузки данных с ленты. Формат распознается автоматически.

`tape_size`: обычно этот параметр задает реальную ёмкость ленты. Из соображений надежности (мы используем аппаратное сжатие), задано значение несколько меньше фактического.

`tape_address`: устройство без перемотки для использования в команде `cpio`.

`tape_address_rew`: устройство с перемоткой для использования в команде `cpio`.

10.7.12.5. Проблемы конфигурирования после установки

Следующие параметры SAP® надо настроить после установки (примеры для IDES 46B, 1 Гбайт памяти):

Имя	Значение
ztta/roll_extension	250000000
abap/heap_area_dia	300000000
abap/heap_area_nondia	400000000
em/initial_size_MB	256

Имя	Значение
em/blocksize_kB	1024
ipc/shm_psize_40	70000000

SAP® Note 0013026:

Имя	Значение
ztta/dynpro_area	2500000

SAP® Note 0157246:

Имя	Значение
rdisp/ROLL_MAXFS	16000
rdisp/PG_MAXFS	30000

При указанных выше параметрах в системе с 1 Гбайт памяти можно обнаружить примерно следующее использование памяти:

```
Mem: 547M Active, 305M Inact, 109M Wired, 40M Cache, 112M Buf, 3492K
Free
```

10.7.13. Проблемы в ходе установки

10.7.13.1. Перезапуск R3SETUP после устранения проблемы

R3SETUP останавливается при выявлении ошибки. Если вы просмотрели соответствующие журнальные файлы и исправили ошибку, придется запускать R3SETUP снова, обычно выбирая REPEAT как опцию для последнего шага, на котором команда R3SETUP выдала сообщение об ошибке.

Для перезапуска команды R3SETUP просто запустите её с соответствующим файлом R3S:

```
# ./R3SETUP -f CENTRDB.R3S
```

для 4.6B или с файлом

```
# ./R3SETUP -f CENTRAL.R3S
```

для 4.6C, независимо от того, произошла ли ошибка при работе с файлом CENTRAL.R3S или DATABASE.R3S.

На некоторых стадиях команда R3SETUP предполагает, что запущены и работают процессы как сервера базы данных, так и SAP® (поскольку эти

шаги уже выполнены). Если возникнут ошибки и, например, запустить сервер базы данных не получится, придется вручную запускать сервер базы данных и SAP® после исправления ошибок и до повторного запуска R3SETUP.

Не забудьте также снова запустить процесс прослушивания Oracle® (как пользователь ora_sid_ с помощью команды umask 0; lsnrctl start), если он тоже был остановлен (например, из-за необходимой перезагрузки системы).

10.7.13.2. OSUSERSIDADM_IND_ORA в ходе R3SETUP

Если R3SETUP выдает сообщения об ошибках на этом этапе, отредактируйте используемый при этом файл шаблона R3SETUP (CENTRDB.R3S (4.6B), либо CENTRAL.R3S или DATABASE.R3S (4.6C)). Найдите раздел [OSUSERSIDADM_IND_ORA] или поищите единственную запись STATUS=ERROR и отредактируйте следующие значения:

```
HOME=/home/sidadm (было пусто)
STATUS=OK (был статус ERROR)
```

Затем надо снова перезапустить R3SETUP.

10.7.13.3. OSUSERDBSID_IND_ORA в ходе R3SETUP

Возможно, команда R3SETUP также выдаст сообщения об ошибке на этой стадии. Ошибка здесь аналогична возникающей на стадии OSUSERSIDADM_IND_ORA. Просто отредактируйте используемый файл шаблона R3SETUP (CENTRDB.R3S (4.6B), либо CENTRAL.R3S или DATABASE.R3S (4.6C)). Найдите раздел [OSUSERDBSID_IND_ORA] или поищите единственную запись STATUS=ERROR и отредактируйте следующее значение в этом разделе:

```
STATUS=OK
```

Затем перезапустите R3SETUP.

10.7.13.4. oraview.vrf FILE NOT FOUND в ходе установки Oracle®

Вы не сняли выбор с *Oracle® On-Line Text Viewer* перед началом установки. Он помечен для установки, хотя этот продукт и не доступен сейчас для Linux. Снимите пометку с этого продукта в меню установки Oracle® и перезапустите установку.

10.7.13.5. TEXTENV_INVALID в ходе R3SETUP, RFC или запуска SAPgui

Если возникает эта ошибка, не найдена нужная локаль. SAP® Note 0171356 перечисляет необходимые пакеты RPM, которые надо установить (например, saplocales-1.0-3, saposcheck-1.0-1 для RedHat 6.1). Если игнорировать все ошибки и менять STATUS соответствующих шагов с ERROR на OK (в файле CENTRDB.R3S) каждый раз, когда R3SETUP сообщает об ошибке и просто перезапуск ать R3SETUP, система SAP® не будет правильно сконфигурирована, и вы затем не сможете подключиться к системе с помощью SAPgui, хотя запустить систему и получится. Попытка подключения с помощью старой Linux-версии SAPgui приведет к

выдаче следующих сообщений:

```
Sat May 5 14:23:14 2001
*** ERROR => no valid userarea given [trgmsgo. 0401]
Sat May 5 14:23:22 2001
*** ERROR => ERROR NR 24 occured [trgmsgi. 0410]
*** ERROR => Error when generating text environment. [trgmsgi. 0435]
*** ERROR => function failed [trgmsgi. 0447]
*** ERROR => no socket operation allowed [trxio.c 3363]
Speicherzugriffsfehler
```

Это связано с тем, что система SAP® R/3® не может корректно назначить локаль и сама не была надлежащим образом сконфигурирована (не хватает записей в некоторых таблицах базы данных). Чтобы можно было подключиться к SAP®, добавьте следующие записи в файл DEFAULT.PFL (см. Note 0043288):

```
abap/set_etct_env_at_new_mode = 0
install/collate/active = 0
rscp/TCP0B = TCP0B
```

Перезапустите систему SAP®. Теперь вы можете подключиться к системе, хотя специфические для страны языковые установки могут работать не так, как предполагалось. После исправления настроек страны (и добавления соответствующих локалей) эти записи можно удалить из файла DEFAULT.PFL и перезапустить систему SAP®.

10.7.13.6. ORA-00001

Эта ошибка возникает только с Oracle® 8.1.7 на FreeBSD. Причина в том, что сервер Oracle® не может правильно проинициализироваться и аварийно завершает работу, оставляя не освобожденными в системе семафоры и совместно используемую память. При следующей попытке запустить сервер базы данных выдается ошибка ORA-00001.

Найдите оставшиеся семафоры и сегменты памяти с помощью команды `ipcs -a` и удалите с помощью `ipcrm`.

10.7.13.7. ORA-00445 (фоновый процесс PMON не запущен)

Эта ошибка произошла с Oracle® 8.1.7. Она выдается, если сервер был запущен с помощью обычного скрипта `startsap` (например, `startsap_majestix_00`) от имени пользователя `prdadm`.

Возможный способ обхода - запускать сервер базы данных от имени пользователя `oraprd` с помощью `svrmgrl`:

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> startup;
SVRMGR> exit
```

10.7.13.8. ORA-12546 (запускайте процесс прослушивания с правильными правами)

Запускайте процесс прослушивания Oracle® от имени пользователя `oraids` следующими командами:

```
# umask 0; lsnrctl start
```

В противном случае, вы можете получить сообщение об ошибке ORA-12546, поскольку сокеты не будут иметь нужных прав доступа. См. SAP® Note 0072984.

10.7.13.9. ORA-27102 (не хватает памяти)

Эта ошибка произошла при попытке использовать значения `MAXDSIZ` и `DFLDSIZ` больше 1 Гбайта (1024x1024x1024). Кроме того, мы получили `Linux Error 12: Cannot allocate memory`.

10.7.13.10. [DIPGNTAB_IND_IND] в ходе `R3SETUP`

В общем случае, см. SAP® Note 0130581 (прекращается работа `R3SETUP` на шаге `DIPGNTAB`). В ходе установки IDES-версии по каким-то причинам процесс установки использовал вместо правильного имени системы SAP®, "IDS", пустую строку, `""`. Это приводит к небольшим проблемам при доступе к каталогам, поскольку пути генерируются динамически на базе *SID* (в данном случае, IDS). Поэтому вместо обращения к:

```
/usr/sap/IDS/SYS/...
/usr/sap/IDS/DVMGS00
```

используются следующие пути:

```
/usr/sap//SYS/...
/usr/sap/D00
```

Чтобы продолжить установку мы создали ссылку и дополнительный каталог:

```
# pwd
/compat/linux/usr/sap
# ls -l
total 4
drwxr-xr-x 3  idsadm sapsys 512 May 5 11:20 D00
drwxr-x--x 5  idsadm sapsys 512 May 5 11:35 IDS
lrwxr-xr-x 1  root   sapsys 7 May 5 11:35 SYS -> IDS/SYS
drwxrwxr-x 2  idsadm sapsys 512 May 5 13:00 tmp
drwxrwxr-x 11 idsadm sapsys 512 May 4 14:20 trans
```

Мы также нашли документы SAP® Notes (0029227 и 0008401), описывающие это поведение. Мы не столкнулись с подобными проблемами при установке SAP® 4.6C.

10.7.13.11. [RFCRSWBOINI_IND_IND] в ходе `R3SETUP`

В ходе установки SAP® 4.6C, эта ошибка возникла в результате другой ошибки, произошедшей ранее по ходу установки. В данном случае придется просмотреть соответствующие журнальные файлы и устранить исходную проблему.

Если после просмотра журнальных файлов выявлена только эта ошибка (проверьте SAP® Notes), можно поменять `STATUS` соответствующего шага с `ERROR` на `OK` (в файле CENTRDB.R3S) и перезапустить `R3SETUP`. После установки надо выполнить отчет `RSWBOINS` из транзакции SE38. Дополнительную информацию о стадиях `RFCRSWBOINI` и `RFCRADDBDIF` см. в SAP® Note 0162266.

10.7.13.12. [RFCRADDBDIF_IND_IND] в ходе `R3SETUP`

Здесь применяются те же ограничения: проверьте путем просмотра журнальных файлов, что эта ошибка не вызвана какими-то предыдущими проблемами.

Если подтверждается, что применим документ SAP® Note 0162266, просто поменяйте `STATUS` соответствующего шага с `ERROR` на `OK` (в файле CENTRDB.R3S) и перезапустите `R3SETUP`. После установки надо выполнить отчет `RADDBDIF` из транзакции SE38.

10.7.13.13. sigaction sig31: File size limit exceeded

Это сообщение об ошибке выдается в ходе запуска процессов SAP®*disp+work*. Если SAP® запускается скриптом `startsap`, запускаются отдельные подпроцессы, выполняющие грязную работу по запуску всех остальных процессов SAP®. В результате, сам скрипт не получит уведомления, если что-то пойдет не так.

Чтобы проверить, нормально ли запустились процессы SAP®, посмотрите на состояние процессов с помощью команды `ps ax | grep SID`, которая выдаст список всех процессов Oracle® и SAP®. Если похоже, что некоторых процессов не хватает или вы не можете подключиться к системе SAP®, просмотрите соответствующие журнальные файлы, которые можно найти в каталоге /usr/sap/SID/DVEBMGSnr/work/. Надо просматривать файлы dev_ms и dev_disp.

Сигнал 31 выдается, если объем памяти, совместно используемой Oracle® и SAP®, превосходит заданный в файле конфигурации ядра, и от него можно избавиться, указав большее значение:

```
# большее значение для производственных систем 46C:
options SHMMAXPGS=393216
# меньшее значение, достаточное для 46B:
#options SHMMAXPGS=262144
```

10.7.13.14. Сбой при запуске `saposcol`

Есть ряд проблем с программой `saposcol` (версии 4.6D). Система SAP® использует `saposcol` для сбора данных о производительности системы. Эта программа не нужна для использования системы SAP®, так что проблему можно отнести к несерьезным. Более старые версии (4.6B) работают, но собирают не все данные (многие вызовы просто возвращают 0, например, для

использования процессора).

10.8. Дополнительные сведения

Если вы интересуетесь, как обеспечивается двоичная совместимость с Linux, этот раздел для вас. Большинство материала взято из электронного письма, адресованного Terry Lambert tlambert@primenet.com в Список рассылки, посвящённый неформальным беседам о FreeBSD (ID письма: <199906020108.SAA07001@usr09.primenet.com>).

10.8.1. Как все это устроено?

FreeBSD поддерживает абстракцию, называемую "загрузчик выполняемых классов". Фактически, он является первой стадией системного вызова execve(2).

На самом деле, FreeBSD имеет список загрузчиков вместо одного, завершающийся загрузчиком `#!` для запуска любых командных интерпретаторов и скриптов.

Исторически сложилось, что единственный загрузчик в UNIX® системах проверял "магическое число" (чаще всего первые 4 или 8 байт файла), чтобы определить, известен ли формат двоичного файла системе, и если да, то вызвал соответствующий загрузчик.

Если файл не опознавался системой как двоичный, системный вызов execve(2) возвращал ошибку, и текущий командный интерпретатор начинал выполнять файл как скрипт.

По умолчанию скрипт выполнялся "текущим командным интерпретатором".

Позднее, sh(1) был модифицирован, так, чтобы проверять первые два символа в файле, и если они оказывались `:\n`, то файл выполнялся как сценарий для csh(1) (утверждается, что SCO были первыми, кто сделал эту модификацию).

FreeBSD сейчас ведет себя по-другому: пробегает по списку загрузчиков,включающему специальный загрузчик `#!`, который вызывает нужный интерпретатор, указанный после этих символов до следующего пробела, или /bin/sh, если не нашел подходящего.

Для поддержки Linux ABI FreeBSD ищет магическое число, соответствующее двоичному файлу ELF (на этой стадии не различаются FreeBSD, Solaris™, Linux или любая другая ОС поддерживающая формат ELF).

Далее, ELF-загрузчик определяет "марку" (brand) двоичного файла ELF (специальный комментарий в ELF-файле, отсутствующий в двоичных файлах ELF SVR4/Solaris™).

Соответственно, Linux программы должны быть "маркированы" для `Linux` (например, с помощью утилиты brandelf(1)):

```
# brandelf -t Linux file
```

Когда это сделано, загрузчик ELF выявит марку `Linux` в файле.

Когда ELF-загрузчик находит "марку" `Linux`, он заменяет соответствующий указатель в

структуре proc. Все системные вызовы индексируются через этот указатель (в традиционной UNIX® системе это массив структур sysent[], содержащий системные вызовы). Кроме того, процесс помечается для специальной обработки вектора обработчиков сигналов, а также ряда других (небольших) исправлений, которые осуществляются специальным модулем ядра для поддержки Linux.

Вектор системных вызовов Linux содержит, среди прочего, список записей sysent[], адреса которых находятся в модуле ядра.

При выполнении системного вызова из двоичного файла Linux, код обработчика разыменовывает указатель на функцию системного вызова из структуры proc, и получает точки входа системных вызовов Linux, а не FreeBSD.

Плюс ко всему, в Linux-режиме динамически "изменяется корень" файловой системы при поиске файлов; фактически так же, как и параметр union при монтировании файловых систем (не путать с unionfs!). Сперва, файл ищется в каталоге /compat/linux/исходное_полное_имя и только затем, в случае неудачи, в /исходное_полное_имя. Это гарантирует, что программы, которым требуются другие программы, смогут работать (например, весь набор инструментальных средств Linux сможет работать в среде поддержки Linux ABI). Это также дает возможность Linux программам выполнять FreeBSD команды, если не найдется соответствующих Linux команд. Например, можно скопировать FreeBSD uname(1) в дерево каталогов /compat/linux, и Linux-программы не смогут разобраться, что они работают не в Linux.

Фактически, имеется ядро Linux в ядре FreeBSD; различные базовые функции, реализующие все услуги ядра, идентичны как в записях таблицы системных вызовов FreeBSD, так и в записях таблицы системных вызовов Linux: операции с файловой системой, виртуальная память, средства доставки сигналов, System V IPC ... Единственное отличие в том, что FreeBSD-программы получают *интерфейсные* функции FreeBSD, а Linux-программы получают *интерфейсные* функции Linux (в большинстве более старых ОС есть только их собственные интерфейсные функции: функции берутся из статического глобального массива структур sysent[], а не из массива, полученного разыменованием динамически проинициализированного указателя в структуре proc процесса, выполняющего вызов).

Какая же реализация ABI для FreeBSD "родная"? Это не имеет значения. Единственное различие (на данный момент, в будущем все может и, вероятно, изменится), пожалуй, в том, что функции системных вызовов FreeBSD зашиты в ядро, а для Linux они могут быть либо статически скомпонованы в ядро, либо получаться через модуль ядра.

Да, но можно ли назвать это эмуляцией? Нет. Это реализация ABI, а не эмуляция. Как таковой, эмулятор (или симулятор) отсутствует.

В таком случае, почему же иногда говорят об "эмуляции Linux"? Чтобы "насолить" FreeBSD! Фактически, причина в том, что на момент первой реализации не существовало слова, которое бы точнее описывало этот процесс. Нельзя было сказать, что FreeBSD запускает приложения Linux (без перекомпиляции или загрузки соответствующего модуля ядра это невозможно). Но надо было как-то описать, что загружается - отсюда и "эмулятор Linux".

Часть III: Системное администрирование

Оставшиеся главы Руководства охватывают все аспекты администрирования FreeBSD системы. Каждая глава начинается с описания того, что вы сможете изучить в результате прочтения этой главы.

Эти главы спланированы так, что вы можете прочитать их когда вам нужно узнать какую-либо информацию. Вам не нужно читать их в определенном порядке, и не нужно прочитать их все перед тем, как начать пользоваться FreeBSD.

Глава 11. Настройка и оптимизация

11.1. Введение

Один из важных аспектов FreeBSD это настройка системы. Правильная настройка системы поможет избежать головной боли при последующих обновлениях. Эта глава описывает большую часть процесса настройки FreeBSD, включая некоторые параметры, которые можно установить для оптимизации системы FreeBSD.

После прочтения этой главы вы узнаете:

- Как эффективно работать с файловыми системами и разделами подкачки.
- Основы настройки rc.conf и системы запуска приложений /usr/local/etc/rc.d.
- Как настроить и протестировать сетевую карту.
- Как настроить виртуальные хосты на сетевых устройствах.
- Как использовать различные файлы конфигурации в /etc.
- Как оптимизировать FreeBSD, используя переменные `sysctl`.
- Как увеличить скорость работы дисков и изменить ограничения, накладываемые ядром.

Перед прочтением этой главы вам следует:

- Понять основы UNIX® и FreeBSD (Основы UNIX).
- Ознакомиться с основами конфигурации/компиляции ядра (Настройка ядра FreeBSD).

11.2. Начальное конфигурирование

11.2.1. Разделы диска

11.2.1.1. Основы построения разделов

Во время разметки жёсткого диска с помощью bsdlabel(8) или sysinstall(8), важно помнить, что скорость чтения и записи данных уменьшается от внешних к внутренним трекам диска. Самые маленькие и самые часто используемые файловые системы (корневая и раздел подкачки) должны быть расположены в начале диска, в то время как самые большие, такие, как /usr, в конце. Самым оптимальным считается следующий порядок расположения файловых систем: root, swap, /var, /usr.

Размер файловой системы /var определяется предназначением машины. /var используется для хранения почтовых ящиков, очередей печати и лог файлов. Размер почтовых ящиков и лог файлов может расти неограниченно в зависимости от количества пользователей системы и от того, как долго хранятся лог-файлы. Большинству пользователей никогда не потребуется гигабайт, но помните, что /var/tmp должен быть достаточно большим для пакетов.

В разделе /usr содержит большинство файлов, необходимых для поддержки системы, порты

(ports(7), рекомендуется) и исходные тексты (опционально). Оба эти каталога опциональны при установке. Для этого раздела рекомендуется как минимум 2 гигабайта.

При установке размера разделов, не забудьте принять во внимание рост размера требуемого системе дискового пространства. Переполнение одного раздела даже при наличии свободного места на другом может вызвать затруднения.

Многие пользователи обнаружили, что размер разделов, предлагаемый sysinstall(8)'ом по умолчанию, иногда меньше подходящего для разделов /var и /. Тщательно планируйте размер разделов и не жалейте места.

11.2.1.2. Раздел подкачки

Как правило, размер раздела подкачки должен быть равен удвоенному размеру оперативной памяти. Например, если на машине установлено 128 мегабайт памяти, раздел подкачки должен быть 256 мегабайт. Системы с меньшим количеством памяти могут работать лучше с большим объёмом раздела подкачки. Не рекомендуется устанавливать размер раздела подкачки меньше 256 мегабайт, необходимо также принять во внимание возможное наращивание объема установленной на машине памяти. Алгоритмы кэширования VM настроены на максимальное быстродействие, когда размер раздела подкачки равен как минимум удвоенному размеру памяти. Заниженный размер раздела подкачки может привести к неэффективной работе постраничного сканирования VM и вызвать проблемы при увеличении объёма памяти.

На больших системах с несколькими SCSI дисками (или несколькими IDE дисками, находящимися на разных контроллерах), рекомендуется создавать раздел подкачки на каждом диске (до четырёх дисков). Разделы подкачки должны быть примерно одного размера. Ядро не накладывает ограничений на размер раздела подкачки, но внутренние структуры позволяют иметь общий размер разделов подкачки, равный наибольшему, умноженному на четыре. Выделение под разделы подкачки примерно одинакового места позволить ядру оптимально расположить разделы подкачки. Установка размера подкачки больше требуемого нормальна, даже если этот объем не используется. В этих условиях может быть проще восстановиться после зависания программы перед тем, как возникнет необходимость перезагрузки.

11.2.1.3. Зачем нужны разделы?

Некоторые пользователи считают, что лучше использовать один большой раздел, но есть несколько причин, по которым этого лучше не делать. Во-первых, у каждого раздела свои характеристики, и отделяя их, можно выполнить соответствующие настройки. Например, корневая и файловая система и /usr в основном предназначены для чтения, без большого объема записи. В то же время множество операций чтения и записи выполняется в /var и /var/tmp.

При правильном размещении и выборе размера разделов системы, фрагментация в более маленьких разделах, куда часто записываются данные, не перенесётся на остальные разделы. Размещение самых часто используемых разделов ближе к началу диска увеличит скорость ввода/вывода там, где она нужна больше всего. Хотя производительность важна и для больших дисков, передвижение их ближе к концу диска не повлечёт значительного

уменьшения быстродействия по сравнению с перемещением ближе к концу диска /var. И, наконец, разделы существуют и из соображений безопасности. Наличие маленького аккуратного корневого раздела, доступного только для чтения даёт значительные шансы на "выживание" после краха системы.

11.3. Основные настройки

Основные настройки системы располагаются в /etc/rc.conf. Этот файл вмещает широкий спектр конфигурационной информации, используемой при загрузке системы. Имя этого файла прямо отражает его назначение, это файл настройки для файлов rc*.

Администратор должен сделать записи в rc.conf, чтобы переопределить строки по умолчанию из /etc/defaults/rc.conf. Файлы по умолчанию нельзя копировать в /etc - они вмещают значения по умолчанию, а не примеры значений. Все специфичные для данной системы изменения должны быть сделаны в файле rc.conf.

Существует несколько методов для отделения общей конфигурации для группы систем от конкретной для данной системы в целях уменьшения объема работы администратора. Рекомендуемый метод - прописать общую конфигурацию в отдельный файл, например, в /etc/rc.conf.site, и включить его название в /etc/rc.conf, который вмещает только специфичную для данной системы информацию.

Поскольку rc.conf читается sh(1), есть тривиальный способ сделать это. Например:

- rc.conf:

```
. /etc/rc.conf.site
hostname="node15.example.com"
network_interfaces="fxp0 lo0"
ifconfig_fxp0="inet 10.1.1.1"
```

- rc.conf.site:

```
defaultrouter="10.1.1.254"
saver="daemon"
blanktime="100"
```

Файл rc.conf.site может быть распространён на все системы, используя `rsync` или подобную ей программу, в то время, как rc.conf должен остаться только на одной машине.

Обновление системы с помощью sysinstall(8) или `make world` не повлекут за собой перезапись rc.conf. Вся информация в этом файле сохранится.

11.4. Настройка приложений

Обычно, установленные приложения имеют свои конфигурационные файлы, со своим собственным синтаксисом. Важно хранить эти файлы отдельно от файлов основной

системы, чтобы их можно было легко администрировать с помощью средств управления пакетами.

Обычно эти файлы устанавливаются в /usr/local/etc. В случае, если приложению нужно большое количество конфигурационных файлов, для их хранения будет создан подкаталог.

Обычно, вместе с установкой портов и пакетов, устанавливаются и примеры конфигурационных файлов. Обычно они имеют расширение .default. Если не существует конфигурационных файлов для этого приложения, они будут созданы путём копирования .default файлов.

Например, /usr/local/etc/apache:

```
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf.default
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf.default
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic.default
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types.default
-rw-r--r--  1 root  wheel   7980 May 20  1998 srm.conf
-rw-r--r--  1 root  wheel   7933 May 20  1998 srm.conf.default
```

Размеры файлов показывают, что только файл srm.conf был изменён. При следующем обновлении Apache этот файл уже не будет перезаписан.

11.5. Запуск сервисов

Многие пользователи предпочитают устанавливать программы сторонних производителей в FreeBSD из набора портов. В подобных случаях может потребоваться сконфигурировать программы так, чтобы они запускались при инициализации системы. Сервисы, такие как mail/postfix или www/apache13, - это лишь два примера множества программных пакетов, которые можно запускать при инициализации системы. В этом разделе описывается процедура, предназначенная для запуска программ сторонних разработчиков.

Большинство входящих в FreeBSD сервисов, таких как cron(8), запускается с помощью стартовых скриптов системы. Эти скрипты могут различаться в зависимости от версии FreeBSD или ее производителя; однако важнее всего учитывать, что их начальную конфигурацию можно задать с помощью простых стартовых скриптов.

До появления rc.d приложения должны были помещать простой стартовый скрипт в каталог /usr/local/etc/rc.d, который затем читался скриптами инициализации системы. Эти скрипты затем выполнялись в ходе последующих стадий запуска системы.

Хотя много разработчиков потратили часы на попытки внедрить старый стиль конфигурирования в новую систему, остаётся фактом, что для некоторых утилит сторонних производителей по-прежнему необходим скрипт, помещённый в указанный выше каталог.

Незначительные различия в скриптах зависят от того, используется ли rc.d. До версии FreeBSD 5.1 использовались скрипты в старом стиле, и почти во всех случаях скрипты в новом стиле должны подойти так же хорошо.

Хотя каждый скрипт должен соответствовать некоторым минимальным требованиям, в большинстве случаев эти требования не зависят от версии FreeBSD. Каждый скрипт должен иметь в конце расширение .sh и каждый скрипт должен быть выполняемым. Последнее требование может быть выполнено путем установки командой `chmod` уникальных прав доступа `755`. Также, как минимум, должна быть опция `start` для запуска приложения и опция `stop` для его остановки.

Простейший стартовый скрипт, пожалуй, будет похож на следующий:

```
#!/bin/sh
echo -n ' utility'

case "$1" in
start)
        /usr/local/bin/utility
        ;;
stop)
        kill -9 `cat /var/run/utility.pid`
        ;;
*)
        echo "Usage: `basename $0` {start|stop}" 2
        exit 64
        ;;
esac

exit 0
```

Этот скрипт поддерживает опции `stop` и `start` для приложения, которое мы здесь называем просто - `utility`.

А можно запускать его и вручную, с помощью команды:

```
# /usr/local/etc/rc.d/utility.sh start
```

Хотя и не все программы сторонних производителей требуют добавления строки в файл rc.conf, практически каждый день очередной новый порт меняется так, чтобы поддерживать подобную конфигурацию. Поищите в результатах, выдаваемых после установки более детальную информацию по конкретному приложению. Некоторые программы сторонних производителей будут включать стартовые скрипты, позволяющие использовать приложение с rc.d; но это мы еще обсудим в следующем разделе.

11.5.1. Расширенное конфигурирование приложения

Теперь, когда FreeBSD включает rc.d, конфигурирование запуска приложений стало более

оптимальным; фактически, оно стало более тщательным. С помощью ключевых слов, рассмотренных в разделе rc.d, приложения теперь можно настроить для запуска после других заданных сервисов, например, DNS; можно разрешить передачу дополнительных флагов через rc.conf вместо жесткого задания флагов в стартовых скриптах, и т.д. Простой скрипт может иметь следующий вид:

```
#!/bin/sh
#
# PROVIDE: utility
# REQUIRE: DAEMON
# KEYWORD: shutdown

. /etc/rc.subr

name=utility
rcvar=utility_pidfile

command="/usr/local/sbin/utility"

load_rc_config $name

#
# НЕ МЕНЯЙТЕ ЗДЕСЬ ЭТИ СТАНДАРТНЫЕ ЗНАЧЕНИЯ
# ЗАДАВАЙТЕ ИХ В ФАЙЛЕ /etc/rc.conf
#
utility_enable=${utility_enable-"NO"}
pidfile=${utility_pidfile-"/var/run/utility.pid"}

run_rc_command "$1"
```

Этот скрипт будет гарантировать, что указанное приложение utility будет запущено после сервиса `daemon`. Он также предоставляет метод для создания и отслеживания файла идентификатора процесса, PID.

Для этого приложения затем можно поместить следующую строку в файл /etc/rc.conf:

```
utility_enable="YES"
```

Этот новый метод также позволяет легко работать с аргументами командной строки, включать стандартные функции из файла /etc/rc.subr, обеспечивает совместимость с утилитой rcorder(8) и упрощает конфигурирование с помощью файла rc.conf.

11.5.2. Использование сервисов для запуска сервисов

Другие сервисы, такие как даемоны сервера POP3, IMAP, и т.п. могут быть запущены с помощью inetd(8). Для этого необходимо установить сервисную утилиту из набора портов и добавить соответствующую строчку конфигурации в файл /etc/inetd.conf или раскомментировать подходящую строку конфигурации из уже имеющихся. Работа с

даемоном inetd и его конфигурирование подробно описаны в разделе inetd.

В некоторых случаях использование для запуска системных служб даемона cron(8) может оказаться более приемлемым. Этот подход имеет несколько преимуществ, поскольку даемон `cron` запускает эти процессы от имени владельца файла crontab. Это позволяет обычным пользователям запускать и поддерживать некоторые приложения.

Утилита `cron` поддерживает уникальную возможность, `@reboot`, - это значение можно использовать вместо спецификации времени. В результате, задание будет выполнено при запуске cron(8), обычно - в ходе инициализации системы.

11.6. Настройка утилиты `cron`

Одна из наиболее полезных утилит FreeBSD это cron(8). Утилита `cron` работает в фоновом режиме и постоянно проверяет файл /etc/crontab. Утилита `cron` проверяет также каталог /var/cron/tabs в поиске новых файлов crontab. Файлы crontab содержат информацию об определенных функциях, которые `cron` выполняет в указанное время.

Утилита `cron` использует два разных типа конфигурационных файлов, системный и пользовательский. Все различие между этими двумя форматами заключается в шестом поле. В системном файле шестое поля это имя пользователя, с правами которого будет запущена команда. Это позволяет запускать команды из системного crontab от любого пользователя. В пользовательском файле шестое поле указывает запускаемую команду, и все команды запускаются от пользователя, который создал crontab; это важно для безопасности.

Пользовательские crontab позволяют индивидуальным пользователям планировать задачи без привилегий суперпользователя (`root`). Команды из crontab пользователя запускаются с привилегиями этого пользователя.

Пользователь `root` может использовать собственный crontab, как и любой другой пользователь. Он будет отличаться от системного crontab /etc/crontab. Поскольку существует системный crontab, обычно не требуется создавать пользовательский crontab для `root`.

Давайте заглянем в файл /etc/crontab (системный crontab):

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD: src/etc/crontab,v 1.32 2002/11/22 16:13:39 tom Exp $
#①
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ②
HOME=/var/log
#
#
#minute hour    mday    month   wday    who command ③
#
```

```
#
*/5 *   *   *   *   root    /usr/libexec/atrun ④
```

① Как и в большинстве файлов настройки FreeBSD, символы "#" означают комментарии. Комментарии нужны для напоминания о том, что означает строка и зачем она добавлена. Комментарии не могут находиться на той же строке, что и команда, или они будут восприняты как часть команды; располагайте их на новой строке. Пустые строки игнорируются.

② Сначала должны быть заданы переменные окружения. Знак равно (`=`) используется для задания переменных окружения, в этом примере `SHELL`, `PATH`, и `HOME`. Если переменная для оболочки не задана, `cron` использует оболочку по умолчанию, `sh`. Если не задана переменная `PATH`, значение по умолчанию не устанавливается и пути к файлам должны быть полными. Если не задана переменная `HOME`, `cron` будет использовать домашний каталог соответствующего пользователя.

③ В строке всего семь полей. Их значения `minute`, `hour`, `mday`, `month`, `wday`, `who` (кто), и `command`. Значение полей почти очевидно. `minute` это время в минутах, когда будет запущена команда. `hour` означает то же самое для часов. `mday` означает день месяца. `month`, это то же самое, что час и минута, но для месяцев. Параметр `wday` это день недели. Все эти поля должны быть в числовом формате, время в двадцатичетырехчасовом исчислении. Поле `who` имеет специальное значение, и присутствует только в файле /etc/crontab. Это поле определяет пользователя, с правами которого должна быть запущена команда. Когда пользователь устанавливает собственный файл crontab, он не указывает этот параметр. Последний параметр `command`. Он указывает команду, которая должна быть запущена.

④ Последняя строка определяет параметры, описанные выше. Здесь задано значение `*/5`, и несколько символов `\*`. Эти символы `*` означают "первый-последний", и могут быть интерпретированы как *каждый*. Таким образом, для этой строки соответствующая команда `atrun` вызывается под пользователем `root` каждые пять минут независимо от дня или месяца. За дополнительной информацией по команде `atrun` обращайтесь к странице справочника atrun(8).Команды могут принимать любое количество параметров; однако команды, состоящие из нескольких строк, должны быть объединены символом "\".

Этот формат одинаков для каждого файла crontab, за исключением одной детали. Шестое поле, где указано имя пользователя, присутствует только в файле /etc/crontab. Это поле должно быть исключено из crontab файлов пользователей.

11.6.1. Установка crontab

Вы не должны использовать процедуру, описанную здесь, для установки системного crontab. Просто используйте свой любимый текстовый редактор: утилита `cron` узнает о том, что файл изменился и сразу начнет использовать обновленную версию. Обратитесь к этой части FAQ за дальнейшей информацией.

Для установки готового crontab пользователя, сначала создайте в вашем любимом редакторе файл соответствующего формата, а затем воспользуйтесь утилитой `crontab`. Обычно она запускается так:

```
% crontab crontab-file
```

В этом примере, crontab-file это имя файла crontab, который только что был создан.

Существует также параметр для просмотра установленных файлов crontab: задайте `crontab` параметр `-l`.

Для пользователей, составляющих crontab вручную, без временного файла, существует параметр `crontab -e`. Она вызовет редактор с пустым файлом. Когда файл будет сохранен, `crontab` автоматически установит его.

Если позднее вы захотите полностью удалить свой crontab, используйте `crontab` с параметром `-r`.

11.7. Использование rc во FreeBSD 5.X и последующих версиях

Во FreeBSD недавно была интегрирована из NetBSD система rc.d, используемая для старта системы. Многие из файлов в каталоге /etc/rc.d предназначены для основных сервисов, они могут управляться параметрами `start`, `stop`, и `restart`. Например, sshd(8) может быть перезапущен следующей командой:

```
# /etc/rc.d/sshd restart
```

Эта процедура похожа для других сервисов. Конечно, сервисы обычно запускаются автоматически при загрузке системы, как указано в rc.conf(5). Например, включение даемона Network Address Translation при запуске выполняется простым добавлением следующей строки в /etc/rc.conf:

```
natd_enable="YES"
```

Если `natd_enable="NO"` уже присутствует, просто измените `NO` на `YES`. Скрипты rc автоматически загрузят все другие зависимые сервисы, как описано ниже.

Поскольку система rc.d в основном предназначена для запуска/отключения сервисов во время запуска/отключения системы, стандартные параметры `start`, `stop` и `restart` будут работать только если установлена соответствующая переменная в /etc/rc.conf. Например, команда выше `sshd restart` будет работать только если переменная `sshd_enable` в файле /etc/rc.conf установлена в `YES`. Для выполнения скриптов независимо от установок в /etc/rc.conf, параметры `start`, `stop` или `restart` необходимо задавать с префиксом "force". Например, для перезапуска `sshd` независимо от установок в /etc/rc.conf, выполните следующую команду:

```
# /etc/rc.d/sshd forcerestart
```

Проверить состояние переменной в файле /etc/rc.conf легко: запустите соответствующий скрипт из rc.d с параметром `rcvar`. Проверка переменной для `sshd` выполняется следующей командой:

```
# /etc/rc.d/sshd rcvar
# sshd
$sshd_enable=YES
```

Вторая строка (`# sshd`) это вывод команды `sshd`, а не консоль `root`.

Чтобы определить, запущен ли сервис, существует параметр `status`. Например для проверки того, запущен ли `sshd`, выполните:

```
# /etc/rc.d/sshd status
sshd is running as pid 433.
```

В некоторых случаях возможна также перегрузка (`reload`) сервиса. Скрипт, запущенный с этим параметром, попытается отправить сервису сигнал, вызывающий перезагрузку файлов настройки. В большинстве случаев это означает отправку сервису сигнала `SIGHUP`. Следует помнить, что эту функцию поддерживают не все сервисы.

Система rc.d используется не только для сетевых серверов, она отвечает также за большую часть инициализации системы. Рассмотрим, к примеру, файл bgfsck. Во время выполнения этот скрипт выводит следующее сообщение:

```
Starting background file system checks in 60 seconds.
```

Следовательно, этот файл используется для фоновой проверки файловых систем, которая выполняется только в процессе инициализации системы.

Функционирование многих сервисов системы зависит от корректной работы других сервисов. Например, NIS и другие основанные на RPC сервисы могут не запуститься, пока не загрузится `rpcbind` (portmapper). Для разрешения этой проблемы, в начале каждого скрипта в комментарии включаются информация о зависимостях и другие метаданные. Программа rcorder(8) используется для разбора этих комментариев во время старта системы для определения порядка, в котором должны вызываться системные сервисы в соответствии с зависимостями. В начало каждого стартового файла должны быть включены следующие строки:

- `PROVIDE`: Задает имя сервиса, предоставляемого этим файлом.
- `REQUIRE`: Список сервисов, необходимых этому сервису. Этот файл будет запущен *после* указанных сервисов.
- `BEFORE`: Список сервисов, зависящих от этого сервиса. Этот файл будет запущен *до* указанных сервисов.

Используя этот метод, администратор может легко контролировать системные сервисы без использования "уровней запуска", как в некоторых других операционных системах UNIX®.

Дополнительную информацию о системе rc.d можно найти на страницах справочника rc(8) и rc.subr(8).

11.8. Настройка карт сетевых интерфейсов

В наши дни мы не представляем себе компьютера без сетевого подключения. Добавление и настройка сетевой карты это обычная задача любого администратора FreeBSD.

11.8.1. Поиск подходящего драйвера

В первую очередь определите тип используемой карты (PCI или ISA), модель карты и используемый в ней чип. FreeBSD поддерживает многие PCI и ISA карты. Обратитесь к Списку поддерживаемого оборудования вашего релиза чтобы узнать, поддерживается ли карта.

Как только вы убедились, что карта поддерживается, потребуется определить подходящий драйвер. В файлах /usr/src/sys/conf/NOTES и /usr/src/sys/arch/conf/NOTES находится список драйверов сетевых интерфейсов с информацией о поддерживаемых чипсетах/картах. Если вы сомневаетесь в том, какой драйвер подойдет, прочтите страницу справочника к драйверу. Страница справочника содержит больше информации о поддерживаемом оборудовании и даже о проблемах, которые могут возникнуть.

Если ваша карта широко распространена, вам скорее всего не потребуется долго искать драйвер. Драйверы для широко распространенных карт представлены в ядре GENERIC, так что ваша карта должна определиться при загрузке, примерно так:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38
000ff irq 15 at device 11.0 on pci0
dc0: Ethernet address: 00:a0:cc:da:da:da
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30
000ff irq 11 at device 12.0 on pci0
dc1: Ethernet address: 00:a0:cc:da:da:db
miibus1: <MII bus> on dc1
ukphy1: <Generic IEEE 802.3u media interface> on miibus1
ukphy1:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
```

В этом примере две карты используют имеющийся в системе драйвер dc(4).

Если драйвер вашей сетевой карты отсутствует в GENERIC, для ее использования потребуется загрузить подходящий драйвер. Это может быть сделано одним из двух способов:

- Простейший способ - просто загрузить модуль ядра сетевой карты с помощью kldload(8). Не все драйверы доступны в виде модулей; например, модули отсутствуют для ISA карт.
- Вместо этого, вы можете статически включить поддержку карты, скомпилировав собственное ядро. Информацию о том, какие параметры нужно включать в ядро, можно получить из /usr/src/sys/conf/NOTES, /usr/src/sys/arch/conf/NOTES и страницы справочника драйвера сетевой карты. За более подробной информацией о сборке собственного ядра обращайтесь к Настройка ядра FreeBSD. Если карта была обнаружена вашим ядром (GENERIC) во время загрузки, собирать ядро не потребуется.

11.8.2. Настройка сетевой карты

Как только для сетевой карты загружен подходящий драйвер, ее потребуется настроить. Как и многое другое, сетевая карта может быть настроена во время установки с помощью sysinstall.

Для вывода информации о настройке сетевых интерфейсов системы, введите следующую команду:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        ether 00:a0:cc:da:da:da
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
dc1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
        ether 00:a0:cc:da:da:db
        media: Ethernet 10baseT/UTP
        status: no carrier
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet 127.0.0.1 netmask 0xff000000
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
```

Старые версии FreeBSD могут потребовать запуска ifconfig(8) с параметром `-a`, за более подробным описанием синтаксиса ifconfig(8) обращайтесь к странице справочника. Учтите также, что строки, относящиеся к IPv6 (`inet6` и т.п.) убраны из этого примера.

В этом примере были показаны следующие устройства:

- dc0: первый Ethernet интерфейс
- dc1: второй Ethernet интерфейс
- lp0: интерфейс параллельного порта
- lo0: устройство loopback
- tun0: туннельное устройство, используемое ppp

Для присвоения имени сетевой карте FreeBSD использует имя драйвера и порядковый номер, в котором карта обнаруживается при инициализации устройств. Например, sis2 это третья сетевая карта, использующая драйвер sis(4).

В этом примере, устройство dc0 включено и работает. Ключевые признаки таковы:

1. `UP` означает, что карта настроена и готова.
2. У карты есть интернет (`inet`) адрес (в данном случае `192.168.1.3`).
3. Установлена маска подсети (`netmask`; `0xffffff00`, то же, что и `255.255.255.0`).
4. Широковещательный адрес (в данном случае, `192.168.1.255`).
5. Значение MAC адреса карты (`ether`) `00:a0:cc:da:da:da`
6. Выбор физической среды передачи данных в режиме автовыбора (`media: Ethernet autoselect (100baseTX full-duplex)`). Мы видим, что dc1 была настроена для работы с `10baseT/UTP`. За более подробной информацией о доступных драйверу типах среды обращайтесь к странице справочника.
7. Статус соединения (`status`) `active`, т.е. несущая обнаружена. Для dc1, мы видим `status: no carrier`. Это нормально, когда Ethernet кабель не подключен к карте.

Если ifconfig(8) показывает примерно следующее:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
            ether 00:a0:cc:da:da:da
```

это означает, что карта не была настроена.

Для настройки карты вам потребуются привилегии пользователя `root`. Настройка сетевой карты может быть выполнена из командной строки с помощью ifconfig(8), но вам потребуется делать это после каждой перезагрузки системы. Подходящее место для настройки сетевых карт это файл /etc/rc.conf.

Откройте /etc/rc.conf в текстовом редакторе. Вам потребуется добавить строку для каждой сетевой карты, имеющейся в системе, например, в нашем случае, было добавлено две строки:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Замените dc0, dc1, и так далее на соответствующие имена ваших карт, подставьте соответствующие адреса. Обратитесь к страницам справочника сетевой карты и ifconfig(8), за подробной информацией о доступных опциях и к странице справочника rc.conf(5) за дополнительной информацией о синтаксисе /etc/rc.conf.

Если вы настроили сетевую карту в процессе установки системы, некоторые строки, касающиеся сетевой карты, могут уже присутствовать. Внимательно проверьте /etc/rc.conf перед добавлением каких-либо строк.

Отредактируйте также файл /etc/hosts для добавления имен и IP адресов различных компьютеров сети, если их еще там нет. За дополнительной информацией обращайтесь к man.hosts.5; и к /usr/shared/examples/etc/hosts.

11.8.3. Тестирование и решение проблем

Как только вы внесете необходимые изменения в /etc/rc.conf, перегрузите компьютер. Изменения настроек интерфейсов будут применены, кроме того будет проверена правильность настроек.

Как только система перезагрузится, проверьте сетевые интерфейсы.

11.8.3.1. Проверка Ethernet карты

Для проверки правильности настройки сетевой карты, попробуйте выполнить ping для самого интерфейса, а затем для другой машины в локальной сети.

Сначала проверьте локальный интерфейс:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

Затем проверьте другую машину в локальной сети:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Вы можете также использовать имя машины вместо `192.168.1.2`, если настроен файл /etc/hosts.

11.8.3.2. Решение проблем

Решение проблем с аппаратным и программным обеспечением всегда вызывает сложности, которые можно уменьшить, проверив сначала самые простые варианты. Подключен ли сетевой кабель? Правильно ли настроены сетевые сервисы? Правильно ли настроен брандмауэр? Поддерживается ли используемая карта в FreeBSD? Всегда проверяйте информацию об оборудовании перед отправкой сообщения об ошибке. Обновите FreeBSD до последней версии STABLE. Просмотрите архивы списков рассылки, или поищите информацию в интернет.

Если карта работает, но производительность низка, может помочь чтение страницы справочника tuning(7). Проверьте также настройки сети, поскольку неправильные настройки могут стать причиной низкой скорости соединения.

Некоторые пользователи встречаются с несколькими `device timeouts`, что нормально для некоторых сетевых карт. Если это продолжается и надоедает, убедитесь, что устройство не конфликтует с другим устройством. Внимательно проверьте подключение кабеля. Возможно также, что вам просто надо установить другую карту.

Время от времени, пользователи видят несколько ошибок `watchdog timeout`. Первое, что требуется сделать, это проверить сетевой кабель. Многие карты требуют поддержки Bus Mastering слотом PCI. На некоторых старых материнских платах, только один PCI слот имеет такую поддержку (обычно слот 0). Сверьтесь с документацией на сетевую карту и материнскую плату, чтобы определить, может ли это быть проблемой.

Сообщение `No route to host` появляются, если система не в состоянии доставить пакеты к хосту назначения. Это может случиться, если не определен маршрут по умолчанию, или кабель не подключен. Проверьте вывод команды `netstat -rn` и убедитесь, что к соответствующему хосту есть работающий маршрут. Если это не так, прочтите Сложные вопросы работы в сети.

Сообщения `ping: sendto: Permission denied` зачастую появляются при неправильно настроенном брандмауэре. Если `ipfw` включен в ядре, но правила не определены, правило по умолчанию блокирует весь трафик, даже запросы ping! Прочтите Межсетевые экраны с более подробной информацией.

Иногда скорость карты недостаточна, или ниже среднего. В этих случаях лучше всего изменить режим выбора типа подключения с `autoselect` на правильный тип. Обычно это работает для большинства оборудования, но не может решить проблему во всех случаях. Проверьте еще раз настройки сети и прочтите страницу справочника tuning(7).

11.9. Настройка виртуальных серверов

Очень часто FreeBSD используется для размещения сайтов, когда один сервер работает в сети как несколько серверов. Это достигается присвоением нескольких сетевых адресов одному интерфейсу.

У сетевого интерфейса всегда есть один "настоящий" адрес, хотя он может иметь любое количество "синонимов" (alias). Эти синонимы обычно добавляются путём помещения

соответствующих записей в /etc/rc.conf.

Синоним для интерфейса fxp0 выглядит следующим образом:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

Заметьте, что записи синонимов должны начинаться с `alias0` и идти далее в определенном порядке (например, `_alias1`, `_alias2`, и т.д.). Конфигурационный процесс остановится на первом по порядку отсутствующем числе.

Определение маски подсети для синонима очень важно, но к счастью, так же просто. Для каждого интерфейса должен быть один адрес с истинной маской подсети. Любой другой адрес в сети должен иметь маску подсети, состоящую из всех единичек (что выражается как `255.255.255.255` или как `0xffffffff`).

Например, рассмотрим случай, когда интерфейс fxp0 подключён к двум сетям, к сети `10.1.1.0` с маской подсети `255.255.255.0` и к сети `202.0.75.16` с маской `255.255.255.240`. Мы хотим, чтобы система была видна по IP, начиная с `10.1.1.1` по `10.1.1.5` и с `202.0.75.17` по `202.0.75.20`. Как было сказано выше, только первый адрес в заданном диапазоне (в данном случае, `10.0.1.1` и `202.0.75.17`) должен иметь реальную маску сети; все остальные (с `10.1.1.2` по `10.1.1.5` и с `202.0.75.18` по `202.0.75.20`) должны быть сконфигурированы с маской сети `255.255.255.255`.

Для этого в файл /etc/rc.conf должны быть внесены следующие записи:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

11.10. Файлы настройки

11.10.1. Каталог /etc

Во FreeBSD определён ряд каталогов, предназначенных для хранения конфигурационных файлов. Это:

/etc	Основные файлы конфигурации системы. Тут размещены системно-зависимые данные.

/etc/defaults	Версии системных конфигурационных файлов по умолчанию.
/etc/mail	Дополнительные конфигурационные файлы sendmail(8), другие конфигурационные файлы MTA.
/etc/ppp	Настройка для user- и kernel-ppp программ.
/etc/namedb	Основное место расположения данных named(8). Обычно named.conf и файлы зон расположены здесь.
/usr/local/etc	Конфигурационные файлы установленных приложений. Могут содержать подкаталоги приложений.
/usr/local/etc/rc.d	Скрипты запуска/остановки установленных приложений.
/var/db	Автоматически генерируемые системно-специфичные файлы баз данных, такие как база данных пакетов, и так далее

11.10.2. Имена хостов

11.10.2.1. /etc/resolv.conf

/etc/resolv.conf определяет, как резолвер (resolver) FreeBSD получает доступ к Системе Доменных Имён (DNS).

Основные записи resolv.conf:

`nameserver`	IP адрес сервера имён. Сервера опрашиваются в порядке описания. Максимальное количество адресов - три.
`search`	Список доменов для поиска с помощью hostname lookup. Обычно определяется доменом, в котором находится компьютер.
`domain`	Домен, в котором находится компьютер.

Типичный вид resolv.conf:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```

Опции `search` и `domain` нельзя использовать совместно.

Если вы используете DHCP, dhclient(8) обычно перезаписывает resolv.conf информацией,

полученной от серверов DHCP.

11.10.2.2. /etc/hosts

/etc/hosts - простая текстовая база данных, напоминающая старый Интернет. Она работает совместно с DNS и NIS, сопоставляя доменные имена IP адресу. Отдельные компьютеры, соединённые с помощью локальной сети, могут быть записаны тут вместо named(8) сервера с целью упрощения. Кроме того, /etc/hosts используется для записи IP адресов и соответствующих им доменов, избавляя от внешнего трафика, используемого для запросов к DNS серверам.

```
# $FreeBSD$
#
# Host Database
# This file should contain the addresses and aliases
# for local hosts that share this file.
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1                     localhost localhost.my.domain myname.my.domain
127.0.0.1               localhost localhost.my.domain myname.my.domain

#
# Imaginary network.
#10.0.0.2               myname.my.domain myname
#10.0.0.3               myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#       10.0.0.0        -   10.255.255.255
#       172.16.0.0      -   172.31.255.255
#       192.168.0.0     -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers.  PLEASE PLEASE PLEASE do not try
# to invent your own network numbers but instead get one from your
# network provider (if any) or from the Internet Registry (ftp to
# rs.internic.net, directory `/templates').
#
```

Формат /etc/hosts:

```
[IP адрес в Интернете] [имя компьютера] [alias1] [alias2] ...
```

Например:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

За дополнительной информацией обращайтесь к hosts(5).

11.10.3. Настройка лог файлов

11.10.3.1. syslog.conf

syslog.conf is является файлом конфигурации для syslogd(8). В нём указываются, типы сообщений генерируемые `syslog`, и лог файлы, в которые они записываются.

```
# $FreeBSD$
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manual page.
*.err;kern.debug;auth.notice;mail.crit          /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.*                                      /var/log/security
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/log/cron
*.err                                           root
*.notice;news.err                               root
*.alert                                         root
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote log host named loghost
#*.*                                            @loghost
# uncomment these if you're running inn
# news.crit                                     /var/log/news/news.crit
# news.err                                      /var/log/news/news.err
# news.notice                                   /var/log/news/news.notice
!startslip
*.*                                             /var/log/slip.log
!ppp
*.*                                             /var/log/ppp.log
```

За более полной информацией обратитесь к syslog.conf(5).

11.10.3.2. newsyslog.conf

newsyslog.conf - конфигурационный файл newsyslog(8), программы, обычно контролируемой

cron(8). newsyslog(8) определяет, когда лог-файлы нуждаются в архивировании и перегруппировке. logfile перемещается в logfile.0, logfile.0 перемещается в logfile.1, и так далее. Другое именование получится при архивировании с помощью gzip(1): logfile.0.gz, logfile.1.gz, и т.д.

newsyslog.conf показывает, какие лог файлы должны быть проинспектированы, сколько их должно быть сохранено, и когда они должны быть пересмотрены. Лог файлы могут быть перегруппированы и/или заархивированы, когда они либо достигнут определённого размера, либо при достижении определённых даты/времени.

```
# configuration file for newsyslog
# $FreeBSD$
#
# filename          [owner:group]    mode count size when [ZB] [/pid_file] [sig_num]
/var/log/cron                           600  3     100  *     Z
/var/log/amd.log                        644  7     100  *     Z
/var/log/kerberos.log                   644  7     100  *     Z
/var/log/lpd-errs                       644  7     100  *     Z
/var/log/maillog                        644  7     *    @T00  Z
/var/log/sendmail.st                    644  10    *    168   B
/var/log/messages                       644  5     100  *     Z
/var/log/all.log                        600  7     *    @T00  Z
/var/log/slip.log                       600  3     100  *     Z
/var/log/ppp.log                        600  3     100  *     Z
/var/log/security                       600  10    100  *     Z
/var/log/wtmp                           644  3     *    @01T05 B
/var/log/daily.log                      640  7     *    @T00  Z
/var/log/weekly.log                     640  5     1    $W6D0 Z
/var/log/monthly.log                    640  12    *    $M1D0 Z
/var/log/console.log                    640  5     100  *     Z
```

За дополнительной информацией обращайтесь к newsyslog(8).

11.10.4. sysctl.conf

sysctl.conf очень похож на rc.conf. Значения устанавливаются в виде `variable=value`. Указанные значения устанавливаются после перевода системы в многопользовательский режим. Однако не все переменные могут быть установлены в этом режиме.

Пример sysctl.conf, настроенного для выключения протоколирования фатальных ошибок программ и разрешения Linux-программам определять, что они запускаются под FreeBSD:

```
kern.logsigexit=0       # Do not log fatal signal exits (e.g. sig 11)
compat.linux.osname=FreeBSD
compat.linux.osrelease=4.3-STABLE
```

11.11. Настройка с помощью sysctl

sysctl(8) - это интерфейс, позволяющий вам вносить изменения в работающую систему FreeBSD. Эти изменения касаются многих опций стека TCP/IP и виртуальной памяти; опытный системный администратор может использовать их для существенного увеличения производительности. Более пяти тысяч системных переменных могут быть прочитаны и записаны с помощью sysctl(8).

По своей сути, sysctl(8) выполняет две функции: чтение и изменение настроек системы.

Для просмотра всех доступных для чтения переменных:

```
% sysctl -a
```

Чтобы прочитать определённую переменную, например, `kern.maxproc`, введите:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Для присвоения значения переменной, используйте выражение вида *переменная=значение*:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Изменяемые с помощью sysctl переменные обычно принимают значения либо строкового, либо целого, либо булевого типа. Переменные булевого типа могут принимать два значения (`1` (истина) и `0` (ложь)).

Если вы хотите устанавливать некоторые переменные автоматически при каждой загрузке компьютера, добавьте их в файл /etc/sysctl.conf. За дополнительной информацией обращайтесь к странице справочника sysctl.conf(5) и к sysctl.conf.

11.11.1. Переменные sysctl(8) только для чтения

В некоторых случаях желательно изменить переменные sysctl(8) только для чтения. Иногда другого способа решить проблему нет; при этом, результат может быть достигнут только на этапе начальной загрузки.

Например, на некоторых моделях лэптопов диапазон памяти устройства cardbus(4) не определяется и выдается приблизительно такая ошибка:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

Ситуации, похожие на эту, требуют изменения некоторых значений sysctl(8), модификация

которых запрещена. Для разрешения этой ситуации пользователь может поместить sysctl(8) "OID" в файл /boot/loader.conf. Значения по умолчанию хранятся в файле /boot/defaults/loader.conf.

Решение проблемы, приведенной выше, потребует помещения строки `hw.pci.allow_unsupported_io_range=1` в вышеупомянутый файл. Теперь cardbus(4) будет работать нормально.

11.12. Оптимизация дисков

11.12.1. Переменные Sysctl

11.12.1.1. `vfs.vmiodirenable`

Значением переменной `vfs.vmiodirenable` может быть установлено в 0 (выключено) или 1 (включено); по умолчанию 1. Эта переменная отвечает за метод кэширования каталогов. Размер большинства каталогов невелик. Они могут поместиться в одном фрагменте (обычно 1K), и могут занимать ещё меньше места (обычно 512 байт) в кэше буфера. При отключении этой переменной (при установке значения 0) буфер прокэширует только заданное число каталогов даже если у вас много памяти. При включении (при установке значения 1) эта переменная sysctl позволит использовать страничное кэширование VM, делая доступным для кэширования каталогов весь объём памяти. Однако, минимальный объём памяти, используемой для кэширования каталогов стал равен объёму страницы (обычно 4 K) вместо 512 байт. Мы рекомендуем оставлять эту опцию включенной, если ваш компьютер исполняет программы, манипулирующие значительным количеством файлов. Примером таких программ могут быть кэширующие прокси-серверы, большие почтовые серверы и серверы новостей. Обычно включение этой опции не понижает производительности, однако лучше поэкспериментировать, чтобы узнать оптимальное значение для вашей машины.

11.12.1.2. `vfs.write_behind`

Переменная sysctl `vfs.write_behind` по умолчанию установлена в `1` (включено). Она указывает системе выполнять запись на носитель по кластерам, что обычно делается для больших файлов. Идея в том, чтобы избежать заполнения кэша неполными буферами, когда это не увеличивает производительность. Однако, это может заблокировать процессы и в некоторых случаях вам может понадобиться отключить этот параметр.

11.12.1.3. `vfs.hirunningspace`

Переменная sysctl `vfs.hirunningspace` определяет число запросов записи на диск, которые могут быть поставлены в очередь. Значение по умолчанию обычно подходит, но на компьютерах с большим количеством дисков вы можете увеличить его до четырех или пяти *мегабайт*. Учтите, что установка слишком большого значения (превышающего размер буфера записи) может привести к очень значительному падению общей производительности. Не делайте это значение произвольно большим! Большие значения могут привести к задержкам чтения, выполняемого в то же время

Есть много других переменных sysctl, относящихся к кэшированию в буфер и страничному кэшированию VM. Мы не рекомендуем изменять эти значения, поскольку система VM делает отличную работу по автоматической самонастройке.

11.12.1.4. vm.swap_idle_enabled

Переменная sysctl vm.swap_idle_enabled полезна в больших многопользовательских системах, где есть много пользователей, входящих и выходящих из системы, и множество ожидающих процессов. Такие системы обычно генерируют большое количество запросов на выделение памяти. Включение этой переменной и настройка задержки выгрузки (swapout hysteresis, в секундах) установкой переменных vm.swap_idle_threshold1 и vm.swap_idle_threshold2 позволит освобождать страницы памяти, занятые ожидающими процессами, более быстро, чем при нормальном алгоритме выгрузки. Это помогает даемону выгрузки страниц. Не включайте этот параметр, пока он на самом деле вам не понадобится, поскольку его действие в сущности заключается в более ранней выгрузке страниц из памяти; это повышает нагрузку на подкачку и диск. В малых системах эффект от включения этого параметра предсказуем, но в больших системах нагруженной на подкачкой этот параметр позволяет системе VM проще загружать и выгружать процессы из памяти.

11.12.1.5. hw.ata.wc

Во FreeBSD 4.3 кэширование записи на IDE диски было отключено. Это понижало производительность IDE дисков в тестах, но было необходимо для лучшей сохранности данных. Проблема состоит в том, что IDE диски неправильно указывают время завершения записи на диск. При включенном кэшировании IDE диски могут не только записать данные в неправильном порядке - при большой нагрузке на диск некоторые блоки могут задержаться до бесконечности. Сбой, или отключение питания могут могут стать причиной серьёзных повреждений в файловой системе. Поэтому для безопасности системы значение по умолчанию этого параметра было изменено. К сожалению, результатом этого стало столь значительная потеря производительности, что после выхода релиза значение этого параметра было возвращено в первоначальное состояние. Вам следует проверить значение переменной sysctl hw.ata.wc на вашей машине. Если кэширование выключено - вы можете включить его, установив значение переменной ядра, равное 1. Это должно быть сделано при помощи загрузчика при загрузке. Если вы сделаете это позже - изменения не будут иметь силы.

За более подробной информацией обращайтесь к ata(4).

11.12.1.6. SCSI_DELAY (kern.cam.scsi_delay)

Параметр настройки ядра SCSI_DELAY может использоваться для уменьшения времени загрузки системы. Значение по умолчанию велико и может составлять более 15 секунд в процессе загрузки. Уменьшение его до 5 секунд обычно работает (особенно с современными дисками). В новых версиях FreeBSD (5.0 и выше) должен использоваться параметр kern.cam.scsi_delay, настраиваемый во время загрузки. Этот параметр и параметр настройки ядра принимают значения в *миллисекундах*, а *не* в *секундах*.

11.12.2. Soft Updates

Программа tunefs(8) используется для настройки файловой системы. Эта программа может принимать большое количество параметров, но мы рассмотрим лишь один из них - включение и выключение Soft Updates, что может быть достигнуто следующим образом:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

Нельзя изменять файловую систему с помощью tunefs(8) когда она смонтирована. Самое подходящее время для включения "Soft Updates" - перед монтированием разделов, в однопользовательском режиме.

Soft Updates существенно увеличивают скорость создания и удаления файлов путём использования кэширования. Мы рекомендуем использовать Soft Updates на всех ваших файловых системах. Однако у Soft Updates есть и обратные стороны: во-первых, Soft Updates гарантирует целостность файловой системы в случае сбоя, но может наблюдаться задержка в несколько секунд (или даже минуту!) перед записью на жесткий диск. Если система зависнет - вы можете потерять больше, чем, если бы вы не включили Soft Updates. Во-вторых, Soft Updates задерживает освобождение блоков файловой системы. Если ваша файловая система заполнена, выполнение значительного обновления, например, `make installworld`, может вызвать переполнение.

11.12.2.1. Дополнительная информация о Soft Updates

Есть два традиционных способа записи метаданных файловых систем на диск (пример метаданных: индексные дескрипторы и каталоги).

Исторически, поведение по умолчанию заключается в синхронном обновлении метаданных. Если каталог был изменен, система ждет, пока изменение не будет физически записано на диск. Содержимое файлов проходит через кэш и записывается на диск асинхронно. Преимущество этого способа в его надежности. При сбое во время обновления метаданные остаются в нормальном состоянии. Файл либо создается целиком, либо вообще не создается. Если блоки данных не были записаны в файл из буфера во время сбоя, fsck(8) сможет определить это и восстановить файловую систему, установив длину файла в 0. Кроме того, реализация этого способа проста и понятна. Недостаток в том, что обновление метаданных занимает много времени. Команда `rm -r`, например, последовательно удаляет все файлы в каталоге, и каждое изменение в каталоге (удаление файла) будет синхронно записано на диск. Сюда включаются обновления самого каталога, таблицы индексных дескрипторов, и возможно блоков, занятых файлом. Те же соглашения работают при распаковке больших иерархий (`tar -x`).

Другой вариант это асинхронное обновление метаданных. Это поведение по умолчанию для Linux/ext2fs и *BSD ufs с параметром `mount -o async`. Все обновления метаданных просто пропускаются через кэш буфера, как и содержимое файлов. Преимущество этой реализации в том, что нет необходимости ждать каждый раз, пока метаданные будут записаны на диск, поэтому все операции с большим объемом обновления метаданных будут происходить гораздо быстрее, чем при синхронном обновлении. Кроме того, реализация все еще проста

и понятна, поэтому риск появления ошибок в коде невелик. Недостаток в том, что нет никаких гарантий исправности файловой системы. Если во время обновления большого объема метаданных произойдет сбой (например, отключение питания, или нажатие кнопки reset), файловая система останется в непредсказуемом состоянии. Нет возможности определить состояние файловой системы после такого сбоя; блоки данных файла могут быть уже записаны на диск, а обновления таблицы индексных дескрипторов нет. Невозможно реализовать `fsck`, которая могла бы исправить получившийся хаос (поскольку необходимой информации нет на диске). Если файловая система была уничтожена во время восстановления, единственный способ восстановления - запустить newfs(8) и воспользоваться резервной копией.

Обычное решение этой проблемы состояло в реализации *протоколировании проблемной области (dirty region logging)*, известном как *журналирование*, хотя этот термин использовался неправильно и порой также применялся к другим формам протоколирования транзакций. Обновление метаданных как и прежде происходит синхронно, но в отдельную область диска. Позже они перемещаются туда, где должны быть. Поскольку область протоколирования это небольшая, последовательная область диска, головкам жесткого диска не приходится перемещаться на большие расстояния даже во время значительных обновлений, поэтому такой способ быстрее, чем синхронные обновления. Кроме того, сложность реализации довольно ограничена, поэтому риск внесения ошибок невелик. Недостаток в том, что все обновления метаданных записываются дважды (один раз в область протоколирования и один раз окончательно), поэтому при обычной работе производительность может понизиться. С другой стороны, в случае сбоя все незаконченные действия с метаданными могут быть быстро отменены, или завершены после загрузки системы, поэтому система после сбоя загружается быстрее.

Kirk McKusick, разработчик Berkeley FFS, решил эту проблему с помощью Soft Updates: все незавершенные обновления метаданных находятся в памяти и записываются на диск в упорядоченном виде ("упорядоченное обновления метаданных"). При значительных обновлениях метаданных более поздние обновления "присоединяются" к предыдущим, если они все еще находятся в памяти и еще не записаны на диск. Поэтому все операции, скажем, над каталогом, обычно выполняются в памяти перед записью обновления на диск (блоки данных сортируются в соответствии с их положением, так что они не будут записаны на диск до метаданных. При крахе операционной системы выполняется "откат": считается, что все операции, не записанные на диск, никогда не происходили. Файловая система находится в том состоянии, в котором она была за 30-60 секунд до сбоя. Используемый алгоритм гарантирует, что все используемые ресурсы маркированы соответствующим образом в своих областях: блоки и индексные дескрипторы. После сбоя могут остаться только ошибки выделения ресурсов, они помечаются как "используемые", хотя на самом деле "свободны". fsck(8) разбирается в ситуации и освобождает более не используемые ресурсы. После сбоя система может быть безопасно смонтирована с опцией `mount -f`. Для освобождения ресурсов, которые могут не использоваться, в дальнейшем потребуется запустить fsck(8). Эта идея лежит в основе *background (фоновая) fsck*: во время запуска системы записывается только *снимок* файловой системы. Все системы могут быть смонтированы в "грязном" состоянии, и система загружается в многопользовательский режим. Затем, фоновые `fsck` ставятся в очередь для всех систем, где это требуется, чтобы освободить неиспользуемые ресурсы. (Файловые системы, где не используются Soft Updates, все еще требуют запуска `fsck` в обычном режиме).

Преимущество этого способа в том, что обновления метаданных происходят почти так же быстро, как при асинхронных обновлениях (т.е. быстрее, чем при *журналировании*, когда метаданные записываются дважды). Недостаток в сложности кода (подразумевающим больший риск появления ошибок в области, где вероятность потери данных пользователя особенно высока) и в более высоких требованиях к объему памяти. К тому же могут возникнуть некоторые странные на первый взгляд ситуации. После сбоя состояние файловой системы несколько более "старое". В ситуации, когда стандартный способ синхронизации оставит несколько файлов нулевой длины после выполнения `fsck`, в файловой системе с Soft Updates их не останется вовсе, поскольку ни метаданные, ни содержимое файлов не были записаны на диск. Дисковое пространство не будет освобождено пока обновления не будут записаны на диск, что может занять некоторое время после выполнения `rm`. Это может повлечь проблемы при установке большого количества файлов на файловую систему, где не хватает места для помещения всех файлов дважды.

11.13. Изменение ограничений, накладываемых ядром

11.13.1. Ограничения на Файлы/Процессы

11.13.1.1. `kern.maxfiles`

Значение `kern.maxfiles` может быть увеличено или уменьшено в зависимости от потребностей вашей системы. Эта переменная определяет максимальное число дескрипторов файлов. Когда таблица дескрипторов файлов полна, в очереди системных сообщений появится сообщение `file: table is full`. Это сообщение может быть прочитано с помощью команды `dmesg`.

Каждый открытый файл, сокет или буфер использует дескриптор файла. Широкомасштабному серверу может понадобиться много тысяч дескрипторов файлов, в зависимости от количества программ, одновременно выполняемых на сервере.

Стандартное значение `kern.maxfile` определяется переменной `maxusers` в вашем файле конфигурации ядра. Значение `kern.maxfiles` увеличивается пропорционально значению `maxusers`. При компилировании ядра, нужно установить эту переменную согласно потребностям вашей системы. Исходя из значения этой переменной, ядро устанавливает значения большинства предопределённых переменных. Даже если предполагается, что к компьютеру не будут одновременно подсоединяться 256 пользователей, требуемые ресурсы могут быть такими же, как у крупномасштабного сервера.

Система автоматически настроит `maxusers`, если вы явно установите его в `0`. Если вы желаете выставить значение самостоятельно, то задайте `maxusers` по меньшей мере равным 4, особенно если вы используйте X Window System или компилируйте программное обеспечение. Причина в том, что самая значимая таблица, устанавливаемая `maxusers` - это максимальное количество процессов, которая устанавливается равным `20 + 16 * maxusers`, и поэтому, если вы установите `maxusers` в 1, то вы сможете иметь только 36 одновременных процессов, включая 18 или около того, что система запустит во время загрузки и 15 или

около того, что вы создадите при запуске X Window System. Даже простая задача, как чтение страницы справочника породит 9 процессов для фильтрации, декомпрессии и её просмотра. Установка `maxusers` в 64 позволит иметь вам до 1044 одновременных процессов, чего должно быть достаточно примерно для всех использований. Если, тем не менее, вы увидите пугающую ошибку при попытке запуска другой программы, или вы используйте сервер с большим количеством одновременных пользователей (как `ftp.FreeBSD.org`), то вы всегда можете увеличить значение и пересобрать систему.

`maxusers` *не* ограничивает количество пользователей, которые могут заходить на вашу машину. Оно просто устанавливает различные размеры таблиц в разумные значения, учитывая максимальное количество пользователей, вы вероятно будете иметь на вашей системе и как много процессов каждый из них сможет запускать. Ключевое слово, которое ограничивает количество одновременных удаленных входов и терминальных X окон - это `pseudo-device pty 16`. С FreeBSD 5.X вам не надо беспокоиться об этом значении, так как pty(4) драйвер является "автоматически клонирующим"; вы просто используйте `device pty` в вашем конфигурационном файле.

11.13.1.2. `kern.ipc.somaxconn`

Переменная sysctl `kern.ipc.somaxconn` ограничивает размер очереди для приема новых TCP соединений. Значение по умолчанию `128` слишком мало для надежной обработки новых соединений для нагруженного web сервера. Для такого сервера рекомендуется увеличить это значение до `1024` или выше. Даемон сервиса может сам ограничивать очередь приема новых соединений (например, sendmail(8), или Apache), но обычно в файле настройки даемона есть директива для настройки длины очереди. Более длинная очередь также помогает избежать атак Denial of Service ().

11.13.2. Сетевые Ограничения

Опция ядра `NMBCLUSTERS` обуславливает количество Mbuf, доступных на машине. На сервере с большим трафиком и маленьким Mbuf производительность будет пониженной. Каждый кластер представлен двумя килобайтами памяти, поэтому значение 1024 означает 2 мегабайта памяти ядра, зарезервированной для сетевых буферов. Для определения оптимального значения необходимо провести простые вычисления. Если у вас веб сервер, который может обслуживать 1000 одновременных соединений, и каждое соединение "съедает" 16 K буфера приема и 16 K буфера отправки, вам потребуется 32 MB памяти под буферы. Хорошее правило - умножение этого значения на 2, 2x32 MB / 2 KB = 64 MB / 2 kB = 32768. Мы рекомендуем значения между 4096 и 32768 для машин с большим объемом памяти. Не указывайте произвольно большое значение параметра, это может привести к падению системы при загрузке. Используйте netstat(1) с опцией `-m` для определения количества используемых сетевых кластеров.

Для настройки в процессе загрузки используйте в loader переменную `kern.ipc.nmbclusters`. Только в старых версиях FreeBSD потребуется пересобрать ядро (config(8)) с измененным параметром `NMBCLUSTERS`.

Для нагруженных серверов, интенсивно использующих системный вызов sendfile(2), может потребоваться увеличения буферов sendfile(2) с помощью параметра конфигурации ядра `NSFBUFS`, или изменения значения путем установки переменной в /boot/loader.conf (обратитесь к loader(8) за подробностями). Общий признак того, что параметр требуется изменить - состояние процессов `sfbufa`. Переменная sysctl `kern.ipc.nsfbufs` установлена только для чтения. Этот параметр увеличивается вместе с `kern.maxusers`, хотя может потребоваться увеличить его отдельно.

> Даже если сокет помечен как неблокирующий, вызов sendfile(2) на неблокирующем сокете может вызвать блокирование sendfile(2), пока не станет доступным достаточное количество `struct sf_buf`.

11.13.2.1. `net.inet.ip.portrange.*`

Переменные sysctl `net.inet.ip.portrange.*` контролируют диапазоны номеров портов, автоматически привязываемых к TCP и UDP сокетам. Есть три диапазона: нижний диапазон, диапазон по умолчанию и верхний диапазон. Большинство сетевых программ используют диапазон по умолчанию, контролируемый `net.inet.ip.portrange.first` и `net.inet.ip.portrange.last`, установленными соответственно в 1024 и 5000. Диапазоны портов привязки используются для исходящих соединений и при некоторых условиях портов может не хватить. Это чаще всего происходит на сильно загруженном прокси сервере. Диапазон портов не становится проблемой при работе серверов, которые обрабатывают в основном входящие соединения, или с небольшим количеством исходящих соединений, например mail relay. Для ситуаций, когда возможен недостаток портов, рекомендуется немного увеличить `net.inet.ip.portrange.last`. Может подойти значение `10000`, `20000`, или `30000`. Учтите также возможное влияние брандмауэра при изменении диапазона портов. Некоторые могут блокировать большие диапазоны портов (обычно с небольшими номерами) и вынуждают использовать более высокие диапазоны для исходящих соединений. По этой причине не рекомендуется уменьшать значение `net.inet.ip.portrange.first`.

11.13.2.2. TCP Bandwidth Delay Product

TCP Bandwidth Delay Product Limiting похоже на TCP/Vegas в NetBSD. Оно может быть включено установкой переменной sysctl `net.inet.tcp.inflight.enable` в `1`. Система попытается вычислить задержку пакетов для каждого соединения и ограничить объем данных в очереди сети до значения, требуемого для поддержания оптимальной пропускной способности.

Эта возможность полезна при передаче данных через модемы, Gigabit Ethernet, или даже через высокоскоростные WAN соединения (или любые другие соединения с большой задержкой передачи), особенно если вы также используете изменение размера окна или настроили большое окно передачи. Если вы включили этот параметр, убедитесь также, что переменная `net.inet.tcp.inflight.debug` установлена в `0` (отладка выключена), а для использования в реальных задачах может понадобиться установка переменной `net.inet.tcp.inflight.min` к значению как минимум `6144`. Но учтите, что установка большого значения этой переменной может фактически отключить ограничение в зависимости от вида соединения. Ограничение уменьшает количество данных на определенном маршруте

и управляет очередью пакетов, как и уменьшает общее количество данных в очереди локального интерфейса хоста. С меньшим количеством пакетов в очереди двусторонние интерактивные соединения, особенно на медленных линиях, могут проходить быстрее. Но имейте ввиду, что эта функция работает только при передаче данных (передача данных / сторона сервера). Она не работает при получении данных (загрузке).

Изменение значения переменной `net.inet.tcp.inflight.stab` *не* рекомендуется. Этот параметр по умолчанию равен 20, что означает добавление 2 пакетов к вычислению задержки передачи. Дополнительное окно требуется для стабилизации алгоритма и улучшения ответной реакции на изменение условий, но также приводит к большему времени ping на медленных соединениях (задержка все же гораздо меньше, чем без алгоритма inflight). Вы можете попробовать уменьшить этот параметр до 15, 10 или 5; а также уменьшить `net.inet.tcp.inflight.min` (например, до 3500) для получения желаемого эффекта. Уменьшение значений этих параметров может использоваться только как крайняя мера.

11.13.3. Виртуальная память

11.13.3.1. `kern.maxvnodes`

Файлы и каталоги в ядре представлены при помощи vnode (виртуальных узлов). Увеличение их числа может помочь уменьшить нагрузку на дисковую подсистему. Как правило, специальной настройки это значение не требует, однако, в некоторых случаях дисковая активность является узким местом, и система исчерпывает таблицу vnode, значение этой переменной следует увеличить. При этом необходимо оценить объем неактивной и свободной памяти.

Текущее количество использованных vnode можно посмотреть при помощи команды:

```
#  sysctl vfs.numvnodes
vfs.numvnodes: 91349
```

Максимальное количество vnode, доступных системе:

```
#  sysctl kern.maxvnodes
kern.maxvnodes: 100000
```

Если количество использованных vnode близко к максимуму, значение переменной `kern.maxvnodes` следует увеличить на 1000. Следите за динамикой изменения `vfs.numvnodes`. Если оно увеличивается, приближаясь к вновь установленному максимуму, процесс следует повторить. Изменение в распределении памяти должно быть видно в выводе утилиты top(1): больше памяти перейдет в разряд активной.

11.14. Увеличение объема подкачки

Вне зависимости от того, что вы планировали, иногда система ведет себя неожиданно. Если

вам потребовался дополнительный объем подкачки, его довольно просто добавить. Есть три способа увеличения объема подкачки: добавить новый жесткий диск, включить подкачку по NFS, или создать файл подкачки на существующем разделе.

За информацией о криптовании раздела подкачки обращайтесь к Шифрование области подкачки данного Руководства.

11.14.1. Подкачка на новом жестком диске

Лучший способ добавить подкачку, конечно, использовать еще один жесткий диск. Вы можете сделать это в любой момент. Если такой способ подходит, прочтите еще раз информацию по пространству подкачки в Начальное конфигурирование Руководства, где рассказывается о наилучшем способе организации раздела подкачки.

11.14.2. Подкачка через NFS

Подкачка через NFS рекомендуется только в том случае, если в системе отсутствует жесткий диск; подкачка через NFS ограничена скоростью сетевого подключения и к тому же дополнительно нагружает NFS сервер.

11.14.3. Файлы подкачки

Вы можете создать файл определенного размера и использовать его как файл подкачки. В нашем примере будет использован файл /usr/swap0 размером 64MB. Конечно, вы можете использовать любое имя.

Пример 18. Создание файла подкачки в FreeBSD

1. Убедитесь, что в файле настройки ядра присутствует драйвер виртуального диска (md(4)). Он есть в ядре GENERIC.

```
device   md   # Memory "disks"
```

2. Создайте файл подкачки (/usr/swap0):

```
# dd if=/dev/zero of=/usr/swap0 bs=1024k count=64
```

3. Установите подходящие права на (/usr/swap0):

```
# chmod 0600 /usr/swap0
```

4. Включите файл подкачки в /etc/rc.conf:

```
swapfile="/usr/swap0"   # Set to name of swapfile if aux swapfile desired.
```

5. Перегрузите компьютер или для включения подкачки прямо сейчас введите:

```
# mdconfig -a -t vnode -f /usr/swap0 -u 0  swapon /dev/md0
```

11.15. Управление питанием и ресурсами

Очень важно использовать аппаратные ресурсы эффективно. До того, как появился ACPI, управление потреблением питания и температурными характеристиками системы было очень сложной для операционной системы задачей. Аппаратное обеспечение контролировалось одним из видов встроенного интерфейса BIOS, таким как: *Plug and Play BIOS (PNPBIOS)*, *Advanced Power Management (APM)* и так далее. Управление питанием и ресурсами это один из ключевых компонентов современной операционной системы. Например, вам может потребоваться, чтобы операционная система следила за температурными ограничениями и возможно, предупреждала при неожиданном росте температуры.

В этом разделе Руководства FreeBSD, мы предоставим исчерпывающую информацию о ACPI. В конце раздела есть ссылки для дальнейшего чтения.

11.15.1. Что такое ACPI?

Advanced Configuration and Power Interface (ACPI) это стандарт, написанный объединением поставщиков в целях предоставления стандартного интерфейса для аппаратных ресурсов и управления питанием (отсюда и название). Это ключевой элемент *Operating System-directed configuration and Power Management*, т.е.: он предоставляет операционной системе (OS) больше контроля и более универсален. Современные системы вышли за пределы ограничений существующих Plug and Play интерфейсов до появления ACPI. ACPI это прямой наследник APM (Advanced Power Management).

11.15.2. Недостатки Advanced Power Management (APM)

Средства *Advanced Power Management (APM)* управляют энергопотреблением системы в зависимости от нагрузки. APM BIOS предоставляется поставщиком системы и специфичен для данной аппаратной платформы. Драйвер APM в OS обеспечивает доступ к *APM Software Interface*, который позволяет управлять уровнями потребления питания.

В APM имеется четыре основных проблемы. Во-первых, управление энергопотреблением осуществляется через зависимый от поставщика BIOS, и OS ничего не знает нем. Один пример: когда пользователь устанавливает время ожидания для жесткого диска в APM BIOS, и это время истекает, BIOS останавливает жесткий диск без согласования с OS. Во-вторых, алгоритм APM встроен в BIOS, и все действия происходят вне контроля OS. Это означает, что пользователи могут решить проблемы с APM BIOS только путем перепрошивки его ROM; это очень опасная процедура, и если она завершится неудачно, система может оказаться в невосстановимом состоянии. В-третьих, реализация технологии APM зависит от поставщика, что означает дублирование усилий и если в BIOS одного из поставщиков будет найдена и исправлена ошибка, ее могли не исправить другие поставщики. Наконец, объем

APM BIOS недостаточно велик для реализации сложной политики управления питанием, или такой политики, которая может хорошо адаптироваться к потребностям компьютера.

Plug and Play BIOS (PNPBIOS) был неудобен во многих ситуациях. PNPBIOS это 16-битная технология, поэтому OS требовалось использовать 16-битную эмуляцию для "взаимодействия" с методами PNPBIOS.

FreeBSD драйвер APM документирован в странице справочника apm(4).

11.15.3. Настройка ACPI

loader(8) загружает драйвер acpi.ko по умолчанию, его *не* надо встраивать в ядро. Причина в том, что с модулями проще работать, например переключиться на другой acpi.ko без пересборки ядра. Преимущество в упрощении тестирования. Другая причина в том, что запуск ACPI после старта системы не очень полезен и при некоторых условиях может приводить к краху. Если вы сомневаетесь, отключите ACPI совсем. Драйвер не должен и не может быть выгружен, поскольку системная шина используется для различных взаимодействий оборудования. ACPI может быть выключен с помощью утилиты acpiconf(8). Фактически большинство взаимодействий с ACPI может быть выполнено через acpiconf(8). В основном это означает, что если в выводе dmesg(8) есть что-то об ACPI, он скорее всего работает.

ACPI и APM не могут сосуществовать и должны использоваться раздельно. Каждый из них прервет загрузку, если обнаружит загруженный драйвер другого.

В простейшей форме, ACPI может использоваться для перевода системы в спящий режим с помощью acpiconf(8), с флагом `-s` и параметром `1-5`. Большинству пользователей нужен только параметр `1`. Параметр `5` сделает "мягкое" завершение работы, так же как и:

```
# halt -p
```

Доступны и другие параметры. Обратитесь к странице справочника acpiconf(8) за дополнительной информацией.

11.16. Использование и отладка FreeBSD ACPI

ACPI это фундаментально новый способ обнаружения устройств, управления энергопотреблением и предоставления стандартизированного доступа к различному оборудованию, ранее управлявшемуся BIOS. Был достигнут определенный прогресс в приспособлении ACPI к работе со всеми системами, но все еще встречаются ошибки в байткоде *ACPI Machine Language* (AML) некоторых материнских плат, незавершенные участки кода в подсистемах ядра FreeBSD и ошибки в интерпретаторе Intel® ACPI-CA.

Этот раздел предназначен для того, чтобы упростить ваше содействие разработчикам FreeBSD ACPI в определении причин наблюдаемых вами проблем, выполнении отладки и выработке решения. Спасибо за помощь и надеемся, что мы сможем помочь в решении

проблем вашей системы.

11.16.1. Отправка отладочной информации

Перед отправкой сообщения об ошибке убедитесь, что у вас последняя версия BIOS, и, если доступна, последняя версия firmware встроенного контроллера.

Те из вас, кто желает составить сообщение о проблеме прямо сейчас, могут воспользоваться адресом freebsd-acpi@FreeBSD.org, отправив на него следующую информацию:

- Описание неправильного поведения, включая тип системы, модель и все, что приводит к появлению ошибки. Кроме того, сообщите настолько точно, насколько возможно, когда появилась ошибка, если ранее вы ее не видели.
- Вывод dmesg(8) после "boot `-v`", включая все сообщения, появившиеся при изучении ошибки.
- Вывод dmesg(8) после "boot `-v`" с выключенным ACPI, если его отключение помогает решить проблему.
- Вывод `sysctl hw.acpi`. Это также хороший способ получения списка возможностей системы.
- URL где можно найти ваш *ACPI Source Language* (ASL). *Не* отправляйте ASL непосредственно в список рассылки, поскольку он может быть очень большим. Копия ASL может быть создана командой:

  ```
  # acpidump -t -d  name-system.asl
  ```

  (Замените вашим логином name и производителем/моделью system. Пример: njl-FooCo6000.asl)

Большинство разработчиков читают Список рассылки, посвящённый обсуждению FreeBSD-CURRENT, но для уверенности, что проблему увидят, отправьте ее в Список рассылки FreeBSD ACPI. Будьте терпеливы, все мы заняты полный рабочий день где-то еще. Если ваше сообщение не заметили сразу, мы возможно попросим вас отправить PR (сообщение о проблеме) через send-pr(1). При вводе PR, включайте ту же информацию, что запрошена выше. Это поможет нам отследить проблему и решить ее. Не отправляйте PR без предварительной отправки письма в Список рассылки FreeBSD ACPI, поскольку мы используем PR в качестве напоминаний о существующих проблемах, а не как механизм сообщений об ошибках. Вероятно, о вашей проблеме кто-то уже сообщал ранее.

11.16.2. Общие сведения

ACPI представлен во всех современных компьютерах, соответствующих архитектурам ia32 (x86), ia64 (Itanium) и amd64 (AMD). Полный стандарт включает множество возможностей, в том числе управление производительностью CPU, уровнем питания, температурой, различными системами аккумуляторов, встроенными контроллерами и опросом шины. В большинстве систем стандарт реализован не полностью. Например, настольные системы

обычно реализуют только опрос шины, а портативные компьютеры кроме того могут поддерживать управление охлаждением и энергопотреблением. Они также поддерживают приостановку и последующий запуск системы различного уровня сложности.

ACPI-совместимые системы состоят из различных компонентов. Производители BIOS и чипсетов предоставляют различные жестко заданные таблицы, (например, FADT), которые определяют функции вроде карты APIC (используется для SMP), регистры настройки и простые значения параметров. Кроме того, предоставляется таблица байткода (*Differentiated System Description Table*, DSDT), определяющая древоподобное пространство имен устройств и методов.

Драйвер ACPI должен прочесть заданные таблицы, реализовать интерпретатор для байткода, модифицировать драйвера устройств и ядро для приема информации от подсистемы ACPI. Для FreeBSD Intel® предоставила интерпретатор (ACPI-CA), тот же что для Linux и NetBSD. Исходный код ACPI-CA находится в каталоге src/sys/contrib/dev/acpica. Код для приспособления ACPI-CA к работе в FreeBSD, находится в src/sys/dev/acpica/Osd. Наконец, драйвера, реализующие различные ACPI устройства, находятся в src/sys/dev/acpica.

11.16.3. Часто встречающиеся проблемы

Для правильной работы ACPI все ее части должны работать правильно. Вот некоторые часто встречающиеся проблемы, в порядке частоты появления, и некоторые обходные пути или исправления.

11.16.3.1. Проблемы с мышью

В некоторых случаях при возобновлении работы после приостановки перестает работать мышь. Известным решением проблемы является добавление строки `hint.psm.0.flags="0x3000"` в файл /boot/loader.conf. Если это не помогло, стоит сообщить о проблеме, как описано выше.

11.16.3.2. Приостановка/возобновление работы

ACPI поддерживает три состояния приостановки в RAM (STR), `S1`-`S3`, и одно состояние приостановки на диск (`STD`), называемое `S4`. `S5` это "мягкое выключение" и это нормальное состояние системы, когда она подключена к сети, но не включена. `S4` может быть реализован двумя различными путями. `S4`BIOS это BIOS-поддерживаемая приостановка на диск. `S4`OS реализуется полностью операционной системой.

Начните с проверки переменных `sysctl hw.acpi`, относящихся к приостановке (suspend). Вот результат для Thinkpad:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

Это означает, что мы можем использовать `acpiconf -s` для тестирования `S3`, `S4`OS, и `S5`. Если `s4bios` был единицей (`1`), это означает поддержку `S4`BIOS вместо `S4`OS.

При тестировании приостановки/возобновления работы, начните с `S1`, если этот режим

поддерживается. Это состояние скорее всего поддерживается, поскольку не требует слишком серьезной поддержки со стороны драйвера. Никто не реализовал `S2`, который похож на `S1`. Следующий режим для тестирования это `S3`. Это наиболее глубокое STR состояние, оно требует существенной поддержки со стороны драйвера, чтобы правильно реинициализировать оборудование. Если у вас возникли проблемы при выходе из этого состояния, отправьте письмо в рассылку Список рассылки FreeBSD ACPI, но не ждите, что проблема будет обязательно решена, поскольку существует множество драйверов/оборудования, нуждающихся в дальнейшем тестировании и разработке.

Для изоляции проблемы удалите из ядра столько драйверов, сколько возможно. Если это работает, вы можете выяснить, какой драйвер вызывает проблему путем загрузки драйверов до тех пор, пока опять не произойдет сбой. Обычно бинарные драйвера, такие как nvidia.ko, драйвера дисплея X11 и USB вызывают большинство проблем, а драйвера Ethernet интерфейсов как правило работают отлично. Если вы можете нормально загрузить/выгрузить драйвера, автоматизируйте этот процесс, поместив соответствующие команды в /etc/rc.suspend и /etc/rc.resume. Это закомментированные примеры выгрузки и загрузки драйверов. Попробуйте установить параметр `hw.acpi.reset_video` в нуль (`0`), если ваш дисплей не включается после возобновления работы. Попробуйте установить большие или меньшие значения для `hw.acpi.sleep_delay`, чтобы проверить, поможет ли это.

Другой способ, который можно попробовать, это запуск последнего дистрибутива Linux с поддержкой ACPI и тестирование поддержки остановки/возобновления работы на том же оборудовании. Если она работает на Linux, проблема скорее всего в драйверах FreeBSD и поиск драйвера, вызывающего проблему, поможет разрешить ситуацию. Имейте ввиду, что разработчики ACPI обычно не поддерживают другие драйверы (звук, ATA, и т.п.), так что все результаты работы по поиску проблемы возможно необходимо отправить в список рассылки Список рассылки, посвящённый обсуждению FreeBSD-CURRENT и человеку, поддерживающему драйвер. Если вы решитесь заняться отладкой, поместите соответствующий код (printf(3)) в вызывающий проблему драйвер для обнаружения места, где прерывается функция восстановления.

Наконец, попробуйте отключить ACPI и включить APM. Если приостановка/возобновление работает с APM, вам возможно лучше подойдет APM, особенно на старом оборудовании (до 2000). Включение корректной поддержки ACPI поставщиками оборудования требует времени и вероятно в старом оборудовании поддержка ACPI в BIOS была некорректна.

11.16.3.3. Система останавливается (временно или постоянно)

Большинство систем останавливаются в результате потери прерываний или "шторма" прерываний. В чипсетах существует много проблем, связанных с тем, как BIOS настраивает прерывания перед загрузкой, правильностью таблицы APIC (MADT), и маршрутизации *System Control Interrupt* (SCI).

"Шторм" прерываний может быть обнаружен по потерянным прерываниям путем проверки вывода строки с `acpi0` команды `vmstat -i`. Если счетчик увеличивается более, чем несколько раз в секунду, это "шторм" прерываний. Если система останавливается, попробуйте войти в DDB (`CTRL` + `ALT` + `ESC` на консоли) и ввести `show interrupts`.

Наиболее надежный способ избавиться от проблемы с прерываниями, это отключение

поддержки APIC с помощью параметра loader.conf`hint.apic.0.disabled="1"`.

11.16.3.4. Паника

Паника, связанная с ACPI, случается довольно редко и имеет наибольший приоритет исправления. Первый шаг это изоляция действий, приводящих к панике (если это возможно) и получение отладки. Следуйте инструкции по включению `options DDB` и настройке последовательной консоли (смотрите Вход в отладчик DDB с последовательной линии) или настройке раздела dump(8). Вы можете получить отладочную информацию DDB с помощью `tr`. Если вы записываете отладку вручную, убедитесь, что переписали как минимум пять (5) строк снизу и пять (5) строк сверху.

Затем попробуйте изолировать проблему, загрузившись с выключенным ACPI. Если это работает, вы можете изолировать подсистему ACPI, используя различные параметры `debug.acpi.disable`. Обратитесь к странице справочника acpi(4) за примерами.

11.16.3.5. Система включается после приостановки или завершения работы

Во-первых, попробуйте установить в loader.conf(5) параметр `hw.acpi.disable_on_poweroff="0"`. Это предотвращает отключение различных событий в ACPI во время завершения работы. В некоторых системах этот параметр необходимо установить в `1` (по умолчанию) по тем же причинам. Обычно это решает проблему, если система неожиданно включается после приостановки или отключения питания.

11.16.3.6. Другие проблемы

Если вы наблюдаете другие проблемы с ACPI (работа с внешним оборудованием, проблемы с обнаружением устройств, и т.д.), отправьте описание проблемы в список рассылки; однако, некоторые из этих проблем могут относиться к незавершенным частям подсистемы ACPI, поэтому может потребоваться время на их реализацию. Будьте терпеливы, и подготовьтесь к тестированию исправлений, которые мы можем вам выслать.

11.16.4. ASL, `acpidump`, и IASL

Наиболее часто встречается проблема, связанная с предоставлением поставщиками BIOS некорректного (или полностью ошибочного!) байткода. Это обычно проявляется появлением консольных сообщений ядра, подобных этому:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
       (Node 0xc3f6d160), AE_NOT_FOUND
```

Зачастую вы можете разрешить эти проблемы путем обновления BIOS до последней ревизии. Большинство консольных сообщений безвредны, но если существуют другие проблемы, такие как не работающий статус батареи, возможно существуют проблемы в AML. Байткод, известный как AML, компилируется из исходного текста на языке ASL. AML находится в таблице, известной как DSDT. Для получения копии ASL, используйте acpidump(8). Вы можете использовать оба параметра `-t` (показывать содержимое постоянных таблиц) и `-d` (дизассемблировать AML в ASL). Обратитесь к разделу Отправка

отладочной информации за примером синтаксиса.

Простейшая первая проверка, которую вы можете провести, это перекомпиляция ASL для поиска ошибок. Предупреждения обычно могут быть проигнорированы, но ошибки обычно не позволяют ACPI работать правильно. Для перекомпиляции ASL, выполните следующую команду:

```
# iasl your.asl
```

11.16.5. Исправление ASL

В дальней перспективе, наша задача состоит в том, чтобы обеспечить поддержку ACPI практически для каждой системы без вмешательства пользователя. Однако, на данный момент мы все еще разрабатываем обходные пути для ошибок, которые часто делают поставщики BIOS. Интерпретатор Microsoft® (acpi.sys и acpiec.sys) не занимается проверкой четкости соблюдения стандартов, поэтому многие поставщики BIOS, проверяющие ACPI только под Windows®, никогда не исправляют ASL. Мы надеемся продолжать обнаружение и документацию нестандартных поведений, позволяемых интерпретатором Microsoft®, и воспроизводить их, чтобы FreeBSD могла работать без необходимости исправления ASL пользователями. В качестве обходного пути для обнаружения неправильного поведения, вы можете исправить ASL вручную. Если исправления будут работать, пожалуйста отправьте diff(1) между старым и новым ASL, чтобы мы могли реализовать обходной путь для неправильного поведения ACPI-CA, чтобы исправление вручную больше не требовалось.

Вот список наиболее часто встречающихся проблем, их причин и способы исправления:

11.16.5.1. OS зависимости

Некоторые AML предполагают, что мир состоит из различных версий Windows®. Вы можете настроить FreeBSD, чтобы она сообщала любое другое имя OS и посмотреть, исправит ли это имеющуюся проблему. Простой способ указания другого имени системы это установка переменной /boot/loader.conf`hw.acpi.osname="Windows 2001"` или в другое подобное значение, имеющееся в ASL.

11.16.5.2. Отсутствие возврата значения

Некоторые методы не возвращают значение явно, как того требует стандарт. Хотя ACPI-CA не обрабатывает эту ситуацию, в FreeBSD существует обходной путь, позволяющей ей явно возвращать значение. Вы можете также добавить явные операторы Return (возврат) там, где требуется, если знаете, что значение должно быть возвращено. Для принудительного компилирования ASL командой `iasl`, используйте флаг `-f`.

11.16.5.3. Перезапись AML по умолчанию

После настройки your.asl для компиляции запустите:

```
# iasl your.asl
```

Вы можете добавить флаг `-f` для создания AML даже при наличии ошибок компиляции. Помните, что некоторые ошибки (например, отсутствующие операторы Return), автоматически обходятся интерпретатором.

Файл DSDT.aml используется `iasl` по умолчанию. Вы можете загрузить его вместо ошибочной копии BIOS (которая остается в постоянной памяти) путем редактирования /boot/loader.conf:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Убедитесь, что скопировали DSDT.aml в каталог /boot.

11.16.6. Получение отладочной информации ACPI

Возможности отладки драйвера ACPI очень гибкие. Они позволяют вам указывать набор подсистем, а также уровень отладки. Подсистемы, которые вы хотите отлаживать, указываются как "слои", и подразделяются на компоненты ACPI-CA (ACPI_ALL_COMPONENTS) и поддержку оборудования ACPI (ACPI_ALL_DRIVERS). Уровень отладки варьируется от ACPI_LV_ERROR (только сообщать об ошибках) до ACPI_LV_VERBOSE (все сообщения). Уровень отладки представляет собой битовую маску, поэтому возможна одновременная установка нескольких параметров, разделенных пробелами. На практике, при использовании для получения отладочной информации последовательной консоли, слишком большое количество информации может переполнить буфер консоли. Полный список отдельных слоев и уровней можно найти на странице справочника acpi(4).

Вывод отладочной информации по умолчанию не включен. Для его включения добавьте параметр `options ACPI_DEBUG` к файлу настройки ядра, если ACPI встроен в ядро. Вы можете добавить параметр `ACPI_DEBUG=1` в файл /etc/make.conf для глобального включения этого параметра. Если вы используете модуль acpi.ko , его можно пересобрать индивидуально:

```
# cd /sys/modules/acpi/acpi
 make clean  make
ACPI_DEBUG=1
```

Установите acpi.ko в /boot/kernel и добавьте предпочитаемый уровень и слой к loader.conf. Этот пример включает отладочные сообщения для всех компонентов ACPI-CA и всех драйверов оборудования ACPI (CPU, LID и т.д.). Будут выводиться только сообщения об ошибках, наименьший уровень отладки.

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

Если требуемая информация получается в результате определенного события (скажем, приостановка и восстановление), вы можете не изменять loader.conf и использовать для указания слоя и уровня `sysctl` после загрузки и подготовки системы к определенному

событию. Имена переменных `sysctl` те же, что и имена параметров настройки в loader.conf.

11.16.7. Ссылки

Дальнейшую информацию о ACPI можно найти по следующим ссылкам:

- Список рассылки FreeBSD ACPI
- Архивы списка рассылки ACPI http://lists.freebsd.org/pipermail/freebsd-acpi/
- Старые архивы списка рассылки ACPI http://home.jp.FreeBSD.org/mail-list/acpi-jp/
- Спецификация ACPI
- Страницы справочника FreeBSD: acpi(4), acpi_thermal(4), acpidump(8), iasl(8), acpidb(8)
- Ресурс по отладке DSDT. (Использует в качестве примера Compaq, но обычно полезен.)

Глава 12. Процесс загрузки FreeBSD

12.1. Описание

Процесс включения компьютера и загрузки операционной системы называется "процессом первоначальной загрузки", или просто "загрузкой". Процесс загрузки FreeBSD предоставляет большие возможности по гибкой настройке того, что происходит при запуске системы, позволяя вам выбирать из различных операционных систем, установленных на одном и том же компьютере, или даже из различных версий той же самой операционной системы или установленного ядра.

Эта глава подробно описывает параметры, которые вы можете изменить для настройки процесса загрузки FreeBSD. Под этим подразумевается все, что происходит до начала работы ядра FreeBSD, обнаружения устройств и запуска init(8). Если вы не совсем уверены, то это происходит, когда выводимый текст меняет цвет с ярко-белого на серый.

После чтения этой главы вы будете знать:

- Из каких частей состоит система начальной загрузки FreeBSD, и как эти части взаимодействуют.
- Параметры, которые вы можете передать компонентам начальной загрузки FreeBSD для управления этим процессом.
- Основы работы device.hints(5)

Только для x86

Эта глава описывает процесс загрузки FreeBSD только для систем на основе архитектуры Intel x86.

12.2. Проблема загрузки

Включение компьютера и запуск операционной системы приводят к интересной дилемме. По определению до запуска операционной системы компьютер не умеет ничего. В том числе и не знает, как запускать программы с диска. Так что компьютер не может запустить программу с диска без операционной системы, но программы операционной системы находятся на диске, но как запустить операционную систему?

Эта проблема имеет параллели с одной проблемой из книги Приключения барона Мюнхгаузена. Герой провалился в болото, и вытащил сам себя, ухватив за волосы и потянув. В эпоху начала компьютеризации термин *начальная загрузка* применялся к механизму, используемому для загрузки операционной системы, и затем был сокращен до просто "загрузки".

На оборудовании архитектуры x86 за загрузку операционной системы отвечает BIOS (Basic Input/Output System). Для этого BIOS ищет на жестком диске MBR (Master Boot Record), которая должна располагаться в определенном месте на диске. BIOS может загрузить и запустить MBR, и предполагается, что MBR может взять на себя остальную работу,

связанную с загрузкой операционной системы.

Выполняемую часть MBR обычно называют *менеджером загрузки (boot manager)*, в особенности если она взаимодействует с пользователем. В этом случае менеджер загрузки, как правило, занимает большее пространство на первом *треке* диска или внутри файловой системы ОС. (Менеджер загрузки иногда называют *загрузчиком (boot loader)*, но во FreeBSD этот термин используется для описания более поздней фазы загрузки). Среди популярных менеджеров загрузки стоит отметить boot0 (он же Boot Easy, стандартный менеджер загрузки FreeBSD), Grub, GAG и LILO. Из перечисленных менеджеров загрузки в MBR помещается только boot0.

Если на вашем диске установлена только одна операционная система, то стандартной MBR будет достаточно. Такая MBR выполняет поиск на диске первого загрузочного (активного) слайса, после чего запускает с этого слайса код загрузки оставшейся части операционной системы. Утилита fdisk(8) по умолчанию устанавливает именно такую MBR, на основе файла /boot/mbr.

Если на ваших дисках установлено несколько операционных систем, то вы можете установить другой менеджер загрузки, который может выдать список различных операционных систем и позволит вам выбрать одну из них для загрузки. Два варианта менеджеров загрузки будут описаны чуть ниже.

Оставшаяся часть системы начальной загрузки FreeBSD разделяется на три этапа. Первый этап запускается из MBR, и он знает достаточно для перевода компьютера в особое состояние и загрузки второго этапа. Второй этап может делать несколько больше до запуска третьего этапа. Третий этап заканчивает работу по загрузке операционной системы. Работа разделена на эти три этапа, потому что стандарты ПК ограничивают размеры программ, которые могут быть запущены на первом и втором этапах. Последовательное выполнение работ позволяет FreeBSD получить более гибкий загрузчик.

Затем стартует ядро, которое начинает опознавать устройства и выполняет их инициализацию. После завершения процесса своей загрузки, ядро передает управление пользовательскому процессу с именем init(8), который выполняет проверку дисков на возможность использования. Затем init(8) запускает пользовательский процесс настройки ресурсов, который монтирует файловые системы, выполняет настройку сетевых адаптеров для работы в сети и вообще осуществляет запуск всех процессов, обычно выполняемых в системе FreeBSD при загрузке.

12.3. Менеджер загрузки и этапы загрузки

12.3.1. Менеджер загрузки

Код MBR или менеджера загрузки время от времени называют *нулевой стадией* процесса загрузки. В этом разделе мы обсудим два из упомянутых ранее менеджеров загрузки: boot0 и LILO.

MBR для FreeBSD находится в /boot/boot0. Это *копия* MBR, так как настоящая MBR должна располагаться в специальном месте диска, вне области FreeBSD.

boot0 очень прост, так как программа в может иметь размер, не превышающий 512 байт. Если вы установили MBR FreeBSD и несколько операционных систем на ваш жесткий диск, то во время загрузки вы увидите нечто похожее на следующее:

Менеджер загрузки boot0: MBR, устанавливаемый программой установки FreeBSD или утилитой boot0cfg(8), основан на /boot/boot0. (boot0 очень прост, так как программа в может иметь размер, не превышающий 446 байт, так как часть первого сектора диска занята таблицей слайсов и сигнатурой `0x55AA`). Если вы установили boot0 и несколько операционных систем на ваш жесткий диск, то во время загрузки вы увидите нечто похожее на следующее:

Пример 19. Образец экрана boot0

```
F1 DOS
F2 FreeBSD
F3 Linux
F4 ??
F5 Drive 1

Default: F2
```

Известно, что другие операционные системы, в частности, Windows® 95, записывают поверх существующей MBR свою собственную. Если так случилось в вашем случае, или же вы хотите заменить существующую MBR на MBR от FreeBSD, то воспользуйтесь следующей командой:

```
# fdisk -B -b /boot/boot0 device
```

Здесь *device* является устройством, с которого вы загружаетесь, таким, как ad0 в случае первого диска IDE, ad2 в случае первого диска IDE на втором контроллере IDE, da0 для первого диска SCSI и так далее. Если вы используете MBR нестандартного вида, воспользуйтесь boot0cfg(8).

Менеджер загрузки LILO: Для того, чтобы этот менеджер загрузки мог загружать FreeBSD, загрузите Linux и добавьте к существующему файлу конфигурации /etc/lilo.conf такие строки:

```
other=/dev/hdXY
table=/dev/hdb
loader=/boot/chain.b
label=FreeBSD
```

Укажите диск с основным разделом FreeBSD в терминах Linux, заменив *X* буквой диска, используемой в Linux, а *Y* - номером основного раздела. Если вы используете диски SCSI, замените */dev/hd* на */dev/sd*. Строка `loader=/boot/chain.b` может быть опущена, если обе

операционные системы находятся на одном диске. Теперь запустите `/sbin/lilo -v` для того, чтобы ваши изменения были восприняты системой, что должно быть подтверждено сообщениями на экране.

12.3.2. Этап первый, /boot/boot1, и этап второй, /boot/boot2

Концептуально первый и второй этапы загрузки являются частями одной и той же программы, в одной области диска. Из-за ограничений на объем дискового пространства они были разделены на две, но вы всегда должны устанавливать их вместе. Они копируются инсталлятором или утилитой bsdlabel (см. ниже) из общего файла /boot/boot.

Они располагаются вне файловых систем, на первом треке загрузочного слайса, то есть там, где boot0 или любой другой менеджер загрузки ожидает найти программу, которую следует запустить для продолжение процесса загрузки. Количество используемых секторов легко может быть вычислено из размера файла /boot/boot.

boot1 очень прост, так как он не может иметь размер, превышающий 512 байт, и знает лишь о *метке диска* FreeBSD, хранящей информацию о слайсе, для того, чтобы найти и запустить boot2.

boot2 устроен несколько более сложно, и умеет работать с файловой системой FreeBSD в объёме, достаточном для нахождения в ней файлов, и может предоставлять простой интерфейс для выбора и передачи управления ядру или загрузчику.

Так как загрузчик устроен гораздо более сложно, и дает удобный и простой способ настройки процесса загрузки, boot2 обычно запускает его, однако раньше его задачей был запуск непосредственно самого ядра.

Пример 20. Образец экрана boot2

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Если вам когда-либо понадобится заменить установленные boot1 и boot2, то используйте утилиту bsdlabel(8):

```
# bsdlabel -B diskslice
```

Здесь *diskslice* являются диском и слайсом, с которых вы загружаетесь, например, ad0s1 в случае первого слайса на первом диске IDE.

Режим Dangerously Dedicated

Если вы используете только имя диска, к примеру, ad0, в команде bsdlabel(8) вы создадите диск в режиме эксклюзивного использования, без слайсов. Это, скорее всего, вовсе не то, что вы хотите сделать, так что дважды проверьте

параметры команды bsdlabel(8), прежде, чем нажать `Return`.

12.3.3. Третий этап, /boot/loader

Передача управления загрузчику является последним, третьим этапом в процессе начальной загрузки, а сам загрузчик находится в файловой системе, обычно как /boot/loader.

Загрузчик являет собой удобный в использовании инструмент для настройки при помощи простого набора команд, управляемого более мощным интерпретатором с более сложным набором команд.

12.3.3.1. Процесс работы загрузчика

Во время инициализации загрузчик пытается произвести поиск консоли, дисков и определить, с какого диска он был запущен. Соответствующим образом он задаёт значения переменных и запускает интерпретатор, которому могут передаваться пользовательские команды как из скрипта, так и в интерактивном режиме.

Затем загрузчик читает файл /boot/loader.rc, который по умолчанию использует файл /boot/defaults/loader.conf, устанавливающий подходящие значения по умолчанию для переменных и читает файл /boot/loader.conf для изменения в этих переменных. Затем с этими переменными работает loader.rc, загружающий выбранные модули и ядро.

И наконец, по умолчанию загрузчик выдерживает 10-секундную паузу, ожидая нажатия клавиши, и загружает ядро, если этого не произошло. Если ожидание было прервано, пользователю выдается приглашение, которое воспринимает простой набор команд, с помощью которых пользователь может изменить значения переменных, выгрузить все модули, загрузить модули и окончательно продолжить процесс загрузки или перезагрузить машину.

12.3.3.2. Встроенные команды загрузчика

Далее следуют наиболее часто используемые команды загрузчика. Полное описание всех имеющихся команд можно найти на странице справки о команде loader(8).

autoboot *секунды*

Продолжает загрузку ядра, если не будет прерван в течение указанного в секундах промежутка времени. Он выводит счетчик, и по умолчанию выдерживается интервал в 10 секунд.

boot [-параметры] [имя ядра]

Продолжить процесс загрузки указанного ядра, если оно было указано, и с указанными параметрами, если они были указаны. Загрузка и использование указанного ядра возможны лишь после выгрузки текущего ядра, а выгрузка текущего ядра производится командой *unload*.

boot-conf

Повторно провести тот же самый процесс автоматической настройки модулей на основе

переменных, что был произведен при загрузке. Это имеет смысл, если до этого вы выполнили команду `unload`, изменили некоторые переменные, например, наиболее часто меняемую `kernel`.

help [тема]

Вывод сообщений подсказки из файла /boot/loader.help. Если в качестве темы указано слово `index`, то выводится список имеющихся тем.

include *имя файла* ...

Выполнить файл с указанным именем. Файл считывается и его содержимое интерпретируется строчка за строчкой. Ошибка приводит к немедленному прекращению выполнения команды include.

load [-t тип] имя файла

Загружает ядро, модуль ядра или файл указанного типа с указанным именем. Все аргументы после имени файла передаются в файл.

ls [-l] [маршрут]

Выводит список файлов по указанному маршруту или в корневом каталоге, если маршрут не был указан. Если указан параметр `-l`, будут выводиться и размеры файлов.

lsdev [-v]

Выводится список всех устройств, с которых могут быть загружены модули. Если указан параметр `-v`, выводится дополнительная информация.

lsmod [-v]

Выводит список загруженных модулей. Если указан параметр `-v`, то выводится дополнительная информация.

more *имя файла*

Вывод указанного файла с паузой при выводе каждой строки `LINES`.

reboot

Выполнить немедленную перезагрузку машины.

set *переменная*

Задает значения переменных окружения загрузчика.

unload

Удаление из памяти всех загруженных модулей.

12.3.3.3. Примеры использования загрузчика

Вот несколько примеров практического использования загрузчика:

- Чтобы просто загрузить ваше ядро обычным образом, но в однопользовательском режиме:

```
boot -s
```

- Для выгрузки обычных ядра и модулей, а потом просто загрузить ваше старое (или другое) ядро:

```
unload
load kernel.old
```

Вы можете использовать kernel.GENERIC для обозначения стандартного ядра, поставляемого на установочном диске, или kernel.old для обращения к ранее установленному ядру (после того, как, например, вы обновили или отконфигурировали новое ядро).

Для загрузки ваших обычных модулей с другим ядром используйте такие команды:

```
unload
set kernel="kernel.old"
boot-conf
```

- Для загрузки скрипта конфигурации ядра (автоматизированный скрипт, который выполняет то, что вы обычно делаете в конфигураторе ядра во время загрузки):

```
load -t userconfig_script /boot/kernel.conf
```

12.3.3.4. Загрузочные экранные заставки

Заставка создает более привлекательный вид процесса загрузки по сравнению с традиционными сообщениями загрузки. Изображение заставки будет отображаться до тех пор, пока не придет очередь приглашения ввода логина на консоли или в менеджере дисплеев.

Есть два базовых окружения во FreeBSD. Первое - это окружение командной строки текстовой виртуальной консоли. По завершении загрузки системы вам предоставляется консольное приглашение ввода логина. Второе окружение - это графическое окружение рабочего стола X11. После установки X11 и одной из графических оболочек, таких как GNOME, KDE или XFce, становится возможным запуск рабочего стола X11 командой `startx`.

Некоторые пользователи предпочитают графический интерфейс входа традиционному текстовому приглашению ввода логина. Менеджеры экранов, наподобие XDM для Xorg, gdm для GNOME, kdm для KDE (а также другие, доступные из коллекции портов), изначально предоставляют графический интерфейс входа. После успешного входа в систему они запускают соответствующий оконный менеджер.

В текстовом окружении экранная заставка скрывает все подробности процесса загрузки и

сообщения стартовых скриптов до момента выдачи приглашения ввода логина. Если используется экранная заставка перед входом в графическое окружение, то пользователи получают визуально более чистый старт системы, чем-то напоминающий опыт работы с Microsoft® Windows® или с иной не unix-подобной системой.

12.3.3.4.1. Экранная заставка в действии

В качестве заставки можно использовать лишь содержащие 256 цветов изображения формата BMP (.bmp) или изображения формата PCX (.pcx) от ZSoft. К тому же, для вывода на стандартный VGA адаптер, файл изображения заставки должен иметь разрешение не более 320 на 200 пикселей.

Чтобы можно было использовать изображения большего размера, вплоть до максимального 1024 на 768, активируйте поддержку VESA. Активация может быть осуществлена либо подключением модуля VESA во время загрузки системы, либо сборкой специализированного ядра с добавленной опцией `VESA` (смотрите Настройка ядра FreeBSD). Поддержка режима VESA дает пользователям возможность отображать заставку, перекрывающую всю видимую область экрана.

Отображаемая во время загрузки заставка может быть убрана нажатием любой клавиши на клавиатуре.

С настройками по умолчанию заставка также становится хранителем экрана в консольном окружении. После некоторого бездействия экран сменится заставкой, яркость которой будет периодически изменяться от её максимального значения к минимальному и обратно. Подобное поведение заставки может быть переопределено добавлением строки `saver=` в /etc/rc.conf. В качестве значения опции `saver=` можно выбрать одно из встроенных имен хранителей экранов, а с полным перечнем можно ознакомиться на странице справочника splash(4). Хранитель экрана, используемый по умолчанию, называется "warp". Заметьте, что установка опции `saver=` в /etc/rc.conf воздействует исключительно на текстовые виртуальные консоли. Она не влияет на менеджеры экранов X11.

Несколько сообщений загрузчика, включая меню загрузки и счетчик, отображаются во время загрузки, даже если экран-заставка активирован.

Файлы-примеры с изображениями для заставок могут быть скачаны из галереи по адресу http://artwork.freebsdgr.org. Установив порт sysutils/bsd-splash-changer, между загрузками вы получите автоматическую смену случайно выбираемых изображений заставок.

12.3.3.4.2. Активация экранной заставки

Файл изображения для заставки (.bmp или .pcx) следует разместить в корневой файловой системе, например в каталоге /boot.

Для работы заставки с разрешением, доступным при загрузке (256 цветов и не более 320x200 точек), отредактируйте /boot/loader.conf, добавив в него следующие строки:

```
splash_bmp_load="YES"
bitmap_load="YES"
```

```
bitmap_name="/boot/splash.bmp"
```

Для получения больших разрешений видео режима (вплоть до максимального 1024x768), внесите в /boot/loader.conf следующие записи:

```
vesa_load="YES"
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Вышеприведённый пример подразумевает, что файл /boot/splash.bmp содержит изображение заставки. Если же требуется выводить файл формата PCX, то используйте следующие строки (в зависимости от необходимого разрешения может также потребоваться строка `vesa_load="YES"`):

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

Возможное имя файла не ограничено одним лишь словом "splash". Оно может выбираться произвольно, например: splash_640x400.bmp или blue_wave.pcx. Важен лишь тип файла: он должен быть либо BMP, либо PCX.

Далее приведены еще две полезные опции loader.conf:

`beastie_disable="YES"`

Эта опция отключит меню загрузчика, но приглашение с обратным отсчетом останется. Даже при не отображаемом меню во время отсчета возможен выбор номера варианта загрузки.

`loader_logo="beastie"`

Эта установка заменит слова "FreeBSD", которые отображаются справа от меню загрузчика, цветным логотипом демона, который занимал это место в предыдущих релизах FreeBSD.

За более детальной информацией обратитесь к следующим страницам справочника: splash(4), loader.conf(5) и vga(4).

12.4. Взаимодействие с ядром во время загрузки

Как только ядро окажется загруженным при помощи загрузчика (обычный способ) или boot2 (минуя загрузчик), оно проверяет флаги загрузки, если они есть, и действует соответствующим образом.

12.4.1. Флаги загрузки ядра

Вот наиболее часто используемые флаги загрузки:

`-a`

во время инициализации ядра запрашивать устройство для его монтирования в качестве корневой файловой системы.

`-C`

загрузка с компакт-диска.

`-c`

запустить UserConfig для конфигурации ядра во время загрузки

`-s`

после загрузки перейти в однопользовательский режим

`-v`

во время запуска ядра выводить более подробную информацию

Есть и другие флаги загрузки, обратитесь к странице справочника по boot(8) для выяснения подробной информации по ним.

12.5. Хинты устройств

Во время начального запуска системы загрузчик loader(8) производит чтение файла device.hints(5). В этом файле хранится необходимая для загрузки ядра информация, задаваемая в виде переменных, которую иногда называют хинтами для устройств ("device hints"). Эти "хинты устройств" используются драйверами устройств для их конфигурации.

Хинты для устройств могут быть также заданы в приглашении начального загрузчика Стадии 3. Переменные могут быть добавлены при помощи команды `set`, удалены посредством `unset` и просмотрены командой `show`. В этот момент могут быть также переопределены переменные, заданные в файле /boot/device.hints. Хинты для устройств, введённые в начальном загрузчике, не сохраняются, и при следующей перезагрузке будут утеряны.

После загрузки системы для выдачи значений всех переменных можно воспользоваться командой kenv(1).

Синтаксически в файле /boot/device.hints в каждой строке определяется по одной переменной, в качестве метки начала комментария используется стандартный символ "#". Строки строятся следующим образом:

```
hint.driver.unit.keyword="value"
```

Синтаксис для начального загрузчика Стадии 3 таков:

```
set hint.driver.unit.keyword=value
```

driver определяет имя драйвера устройства, unit соответствует порядковому номеру модуля устройства, а keyword является ключевым словом хинта. В качестве ключевых слов могут применяться следующие опции:

- at: задаёт шину, к которой подключено устройство.
- port: задаёт начальный адрес используемого диапазона ввода/вывода (I/O).
- irq: задаёт используемый номер запроса на прерывание.
- drq: задаёт номер канала DMA.
- maddr: задаёт физический адрес памяти, занимаемый устройством.
- flags: устанавливает различные битовые флаги для устройства.
- disabled: если установлено в значение "1", то устройство не используется.

Драйверы устройств могут поддерживать (и даже требовать) другие хинты, здесь не перечисленные, поэтому рекомендуется просматривать справочные страницы по этим драйверам. Для получения дополнительной информации обратитесь к страницам справки по device.hints(5), kenv(1), loader.conf(5) и loader(8).

12.6. Init: инициализация управления процессами

После того, как ядро завершит загрузку, оно передает управление пользовательскому процессу init(8), который расположен в файле /sbin/init или в файле, маршрут к которому указан в переменной init_path загрузчика.

12.6.1. Процесс автоматической перезагрузки

Процесс автоматической перезагрузки проверяет целостность имеющихся файловых систем. Если это не так, и утилита fsck(8) не может исправить положение, то init(8) переводит систему в однопользовательский режим для того, чтобы системный администратор сам разобрался с возникающими проблемами.

12.6.2. Однопользовательский режим

В этот режим можно перейти во время процесса автоматической перезагрузки, при ручной загрузке с параметром -s или заданием переменной boot_single для программы loader.

Этот режим может быть также вызван запуском программы shutdown(8) без параметров перезагрузки (-r) или останова (-h) из многопользовательского режима.

Если режим доступа к системной консоли console установлен в файле /etc/ttys в insecure, то система выведет запрос на ввод пароля пользователя root перед переходом в однопользовательский режим.

Пример 21. Незащищённая консоль в /etc/ttys

```
# name  getty               type    status      comments
#
# Если консоль помечена как "insecure", то init будет запрашивать пароль
# пользователя root при переходе в однопользовательский режим.
console none                            unknown off insecure
```

Обозначение консоли как `insecure` означает, что вы считаете физический доступ к консоли незащищённым, и хотите, чтобы только тот, кто знает пароль пользователя `root`, мог воспользоваться однопользовательским режимом, но это не значит, что вы хотите работать с консолью небезопасным способом. Таким образом, если вы хотите добиться защищённости, указывайте `insecure`, а не `secure`.

12.6.3. Многопользовательский режим

Если init(8) определит, что ваши файловые системы находятся в полном порядке, или после того, как пользователь выйдет из однопользовательского режима, система перейдет в многопользовательский режим, работа в котором начинается с настройки ресурсов системы.

12.6.3.1. Настройка ресурсов (rc)

Система настройки ресурсов считывает настройки, применяемые по умолчанию, из файла /etc/defaults/rc.conf, а настройки, специфичные для конкретной системы, из /etc/rc.conf, после чего осуществляется монтирование файловых систем, перечисленных в файле /etc/fstab, запуск сетевых служб, различных системных даемонов и, наконец, выполнение скриптов запуска дополнительно установленных пакетов.

Страница справочника по rc(8) является хорошим источником информации о системе настройки ресурсов, так же, как и самостоятельное изучение скриптов.

12.7. Процесс остановки системы

Во время контролируемого процесса остановки системы через утилиту shutdown(8) программа init(8) будет пытаться запустить скрипт /etc/rc.shutdown, после чего будет посылать всем процессам сигнал `TERM`, а затем и `KILL` тем процессам, которые ещё не завершили свою работу.

Для выключения машины с FreeBSD на аппаратных платформах и системах, которые поддерживают управление электропитанием, просто воспользуйтесь командой `shutdown -p now` для немедленного отключения электропитания. Чтобы просто перезагрузить систему FreeBSD, воспользуйтесь командой `shutdown -r now`. Для запуска команды shutdown(8) вам необходимо быть пользователем `root` или членом группы `operator`. Кроме того, можно также воспользоваться командами halt(8) и reboot(8), пожалуйста, обратитесь к соответствующим

страницам справки и справочной странице по команде shutdown(8) для получения дополнительной информации.

Для управления электропитанием требуется наличие поддержки acpi(4) в ядре или в виде загруженного модуля.

Глава 13. Безопасность

13.1. Краткое описание

Эта глава представляет введение в основные концепции безопасности системы, некоторые эмпирические правила и более подробно обращается к отдельным темам, касающимся FreeBSD. Большая часть затрагиваемых тем может быть применена к безопасности системы и безопасности в интернет вообще. Интернет больше не то "дружественное" место, где каждый хочет быть вам добрым соседом. Защита системы необходима для сохранения ваших данных, интеллектуальной собственности, времени и всего остального от хакеров и им подобных.

FreeBSD предоставляет массу утилит и механизмов для обеспечения целостности и безопасности системы и сети.

После прочтения этой главы вы узнаете:

- Основные концепции безопасности системы, специфику FreeBSD.
- О различных механизмах шифрования в FreeBSD, таких как DES и MD5.
- Как настроить аутентификацию с использованием одноразовых паролей.
- Как настроить TCP Wrappers для использования с `inetd`.
- Как настроить KerberosIV в релизах FreeBSD до 5.0.
- Как настроить Kerberos5 в FreeBSD.
- Как настроить IPsec и создать VPN между компьютерами на FreeBSD/Windows®.
- Как настроить и использовать OpenSSH, реализацию SSH в FreeBSD.
- Что такое ACL и как их использовать.
- Как использовать утилиту Portaudit для проверки пакетов сторонних разработчиков, установленных из Коллекции Портов.
- Как работать с сообщениями безопасности FreeBSD.
- Что такое Process Accounting и как активировать его во FreeBSD.

Перед чтением этой главы вам потребуется:

- Понимание основных концепций FreeBSD и интернет.

В этой книге рассмотрены и другие вопросы безопасности. Например, принудительный контроль доступа (Mandatory Access Control) рассматривается в Принудительный контроль доступа (MAC), а брандмауэры в Межсетевые экраны.

13.2. Введение

Безопасность это первая и основная функция системного администратора. Хотя все многопользовательские системы BSD UNIX® уже снабжены некоторой защитой, работа по

созданию и поддержке дополнительных механизмов безопасности, обеспечивающих защищенную работу пользователей, это одна из самых серьезных задач системного администратора. Компьютеры безопасны настолько, насколько вы сделаете их безопасными и требования безопасности всегда находятся в противоречии с удобством работы пользователей. Системы UNIX® способны одновременно работать с огромным количеством процессов и многие из этих процессов серверные - это означает, что с ними могут взаимодействовать внешние программы. Сегодня десктопы заменили мини-компьютеры и мэйнфрэймы, и поскольку компьютеры в наши дни подключены к сети интернет, безопасность важна как никогда.

Наилучшая реализация системы безопасности представима в виде "послойной" системы. Вообще говоря все, что нужно сделать, это создать столько слоев безопасности, сколько необходимо и затем внимательно следить за вторжениями в систему. Не переусердствуйте в настройке системы безопасности, иначе она сделает невозможной обнаружение вторжений, являющееся одним из наиболее важных аспектов механизма безопасности. Например, нет большого смысла в установке флага `schg` (chflags(1)) на каждый исполняемый файл системы, поскольку хотя таким способом можно временно защитить исполняемые файлы, это помешает обнаружению факта взлома системы.

Безопасность системы также относится к различным формам атак, имеющих своей целью вызвать крах системы, или сделать систему недоступной другим способом, но не пытающихся получить доступ к учётной записи `root` ("break root"). Угрозы безопасности могут быть поделены на несколько категорий:

1. Отказ в обслуживании (Denial of service, DoS).
2. Взлом пользовательских учётных записей.
3. Взлом учётной записи root через доступные сервисы.
4. Взлом учётной записи root через учётные записи пользователей.
5. Создание backdoor.

Атака "отказ в обслуживании" отбирает у машины необходимые ресурсы. Обычно DoS атаки используют грубую силу, чтобы попытаться обрушить систему или сделать ее недоступной другим способом, превысив лимиты ее сервисов или сетевого стека. Некоторые DoS атаки пытаются использовать ошибки в сетевом стеке для обрушения системы одним пакетом. Эту проблему можно решить только исправив ядро системы. Атаки зачастую можно предотвратить правильной установкой параметров, ограничивающих нагрузку на систему в неблагоприятных условиях. С атаками, использующими грубую силу, бороться сложно. Например, атака с использованием пакетов с поддельными адресами, которую почти невозможно остановить, может быстро отключить вашу систему от интернет. Возможно, она не приведет к отказу системы, но сможет переполнить соединение с интернет.

Взлом учётной записи пользователя обычно встречается чаще, чем DoS атаки. Многие системные администраторы все еще используют стандартные сервисы telnetd, rlogind и ftpd на своих серверах. Эти сервисы по умолчанию не работают с зашифрованными соединениям. В результате при среднем количестве пользователей пароль одного или нескольких пользователей, входящих в систему через внешнее соединение (это обычный и наиболее удобный способ входа в систему), будет перехвачен. Внимательный системный

администратор должен анализировать логи удаленного доступа на предмет подозрительных адресов пользователей даже в случае успешного входа.

Кто-то может предположить, что атакующий при наличии доступа к учётной записи пользователя может взломать учётную запись `root`. Однако, реальность такова, что в хорошо защищенной и поддерживаемой системе доступ к учётной записи пользователя не обязательно даст атакующему доступ к `root`. Разница между доступом к обычной учётной записи и к `root` важна, поскольку без доступа к `root` атакующий обычно не способен скрыть свои действия, и в худшем случае сможет лишь испортить файлы пользователя или вызвать крах системы. Взлом пользовательских учётных записей встречается очень часто, поскольку пользователи заботятся о безопасности так, как системные администраторы.

Системные администраторы должны помнить, что существует множество потенциальных способов взлома учётной записи `root`. Атакующий может узнать пароль `root`, найти ошибку в сервисе, работающем с привилегиями и взломать учётную запись `root` через сетевое соединение с этим сервисом, или узнать об ошибке в suid-root программе, позволяющей атакующему взлом `root` с помощью взломанной учётной записи пользователя. Если атакующий нашел способ взлома `root`, ему может не понадобиться установка backdoor. Многие из обнаруженных и закрытых на сегодняшний день брешей в системе, позволяющие взлом `root`, требуют от атакующего серьезной работы по заметанию следов, поэтому большинство атакующих устанавливают backdoor. Backdoor предоставляет атакующему простой способ восстановления доступа к системе с привилегиями `root`, но также дает системному администратору удобный способ обнаружения вторжения. Устранение возможности установки backdoor возможно повредит безопасности системы, поскольку это не устранит брешь, позволившую проникнуть в систему.

Меры безопасности всегда должны реализовываться на нескольких уровнях, которые могут быть классифицированы следующим образом:

1. Защита `root` и служебных учётных записей.
2. Защита работающих под `root` сервисов и suid/sgid исполняемых файлов.
3. Защита учётных записей пользователей.
4. Защита файла паролей.
5. Защита ядра, raw устройств и файловых систем.
6. Быстрое обнаружение несанкционированных изменений в системе.
7. Паранойя.

В следующем разделе этой главы эти темы изложены более подробно.

13.3. Защита FreeBSD

Команда и протокол

В этом документе мы будет использовать выделенный текст, упоминая приложение, и `моноширинный` шрифт, упоминая определенные команды. Для протоколов используется обычный шрифт. Это типографическое отличие

> полезно для таких случаев, как ssh, поскольку это и команда и протокол.

В последующем разделе будут рассмотрены методы защиты системы FreeBSD, упомянутые в предыдущем разделе этой главы.

13.3.1. Защита учётной записи root и служебных учётных записей

Во-первых, не беспокойтесь о защите служебных учётных записей, если не защищена учётная запись root. В большинстве систем у учётной записи root есть пароль. Использование пароля root опасно *всегда*. Это не означает, что вы должны удалить пароль. Пароль почти всегда необходим для доступа по консоли. Но это означает, что вы должны сделать невозможным использование пароля не из консоли или может быть даже с помощью команды su(1). Например, убедитесь, что псевдо-терминалы в файле /etc/ttys перечислены с параметром insecure, что делает невозможным вход на них под root напрямую с помощью telnet или rlogin. При использовании других средств входа, таких как sshd, убедитесь что вход под root напрямую отключен и в них. Сделайте это, открыв файл /etc/ssh/sshd_config, и убедившись, что параметр PermitRootLogin установлен в NO. Проверьте каждый метод доступа - сервис FTP и ему подобные часто подвержены взлому. Прямой вход под root должен быть разрешен только с системной консоли.

Конечно, как системный администратор вы должны иметь доступ root, поэтому потребуется открыть несколько "лазеек". Но убедитесь, что для доступа к ним необходим дополнительный пароль. Одним из способов доступа к root является добавление соответствующих учётных записей к группе wheel (в файле /etc/group). Это позволяет использовать su для доступа к root. Вы никогда не должны давать таким учётным записям доступ к wheel непосредственно, помещая их в группу wheel в файле паролей. Служебные учётные записи должны помещаться в группу staff, а затем добавляться к группе wheel в файле /etc/group. Только те члены группы staff, которым действительно нужен доступ к root, должны быть помещены в группу wheel. При работе с такими методами аутентификации как Kerberos, возможно также использование файла .k5login в каталоге пользователя root для доступа к учётной записи root с помощью ksu(1) без помещения кого-либо в группу wheel. Это решение возможно лучше, поскольку механизм wheel все еще позволяет взлом root, если злоумышленник получил копию файла паролей и смог взломать служебную учётную запись. Хотя использование механизма wheel лучше, чем работа через root напрямую, это не обязательно самый безопасный способ.

Непрямой способ защиты служебных учётных записей и конечно root это использование альтернативных методов доступа и замена зашифрованных паролей на символ “*”. Используя команду vipw(8), замените каждый зашифрованный пароль служебных учётных записей на этот символ для запрета входа с аутентификацией по паролю. Эта команда обновит файл /etc/master.passwd и базу данных пользователей/паролей.

Служебная учётная запись вроде этой:

```
foobar:R9DT/Fa1/LV9U:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Должна быть заменена на такую:

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Это изменение предотвратит обычный вход, поскольку зашифрованный пароль никогда не совпадет с “*”. После этого члены группы staff должны использовать другой механизм аутентификации, например kerberos(1) или ssh(1) с парой ключей: публичным и приватным. При использовании такой системы как Kerberos, потребуется защитить сервер Kerberos и рабочую станцию. При использовании пары публичного/приватного ключей с ssh, потребуется защитить компьютер, *с* которого происходит вход (обычно это рабочая станция). Дополнительных слой защиты может быть добавлен путем защиты пары ключей при создании их с помощью ssh-keygen(1). Возможность заменить пароли служебных учётных записей на “*” гарантирует также, что вход может быть осуществлен только через защищенные методы доступа, которые вы настроили. Это принуждает всех членов staff использовать защищенные, шифрованные соединения для всех входов, что закрывает большую брешь, используемую многими нарушителями: перехват паролей с другого, слабо защищенного компьютера.

Более непрямой механизм безопасности предполагает, что вы входите с более защищенного сервера на менее защищенный. Например, если главный сервер работает со всеми сервисами, рабочая станция не должна работать ни с одним. Для поднятия уровня безопасности до приемлемого уровня, число запущенных на ней сервисов необходимо сократить до минимума, вплоть до отключения их всех, кроме того необходимо использовать защищенный паролем хранитель экрана. Конечно, при наличии физического доступа к рабочей станции атакующий может взломать любую систему безопасности. Это определенно проблема, которую вы должны учитывать, но учтите также тот факт, что большинство взломов совершаются удаленно, через сеть, людьми, которые не имеют физического доступа к вашим рабочим станциям или серверам.

Использование такой системы как Kerberos дает возможность заблокировать или изменить пароль в одном месте, что сразу отразиться на всех компьютерах, где существует служебная учётная запись. Если эта учётная запись будет взломана, возможность немедленно изменить пароль на всех компьютерах нельзя недооценивать. Без этой возможности изменение паролей на N машинах может стать проблемой. Вы можете также наложить ограничения на смену паролей с помощью Kerberos: не только установить значения timeout в Kerberos, но и добавить требование смены пароля пользователем после определенного периода времени (скажем, раз в месяц).

13.3.2. Защита работающих под root сервисов и suid/sgid исполняемых файлов

Предусмотрительный системный администратор запускает только те сервисы, в которых нуждается, ни больше ни меньше. Учитывайте, что сервисы сторонних разработчиков наиболее подвержены ошибкам. К примеру, работа со старыми версиями imapd или popper это все равно что раздача доступа `root` всему миру. Никогда не запускайте сервисы, которые вы не проверили достаточно внимательно. Многим сервисам не требуется работа под `root`. Например, даемоны ntalk, comsat, и finger могут быть запущены в так называемых *песочницах* (*sandboxes*). Песочница это не идеальное решение, поскольку вызывает много проблем, но она подходит под модель послойной безопасности: если кто-то сможет

взломать сервис, работающий в песочнице, ему потребуется взломать еще и саму песочницу. Чем больше уровней ("слоев") потребуется пройти атакующему, тем меньше вероятность его успеха. Ошибки, позволяющие получать root доступ, находили фактически во всех сервисах, запускаемых под `root`, включая основные системные сервисы. Если вы обслуживаете машину, на которую входят только через sshd и никогда не входят через telnetd, rshd или rlogind, отключите эти сервисы!

В FreeBSD сервисы ntalkd, comsat и finger теперь по умолчанию работают в "песочнице". Другая программа, которая может быть кандидатом на запуск в "песочнице" это named(8). /etc/defaults/rc.conf включает необходимые для запуска named в "песочнице" аргументы в закомментированой форме. В зависимости от того, устанавливаете ли вы новую систему, или обновляете старую, учётные записи пользователей, используемые этими "песочницами" могут не быть созданы. Предусмотрительный системный администратор должен узнать о "песочницах" для сервисов и установить их если есть возможность.

Есть множество других сервисов, которые обычно не работают в "песочницах": sendmail, popper, imapd, ftpd, и другие. Некоторым из этих сервисов есть альтернативы, но их установка может потребовать больше работы, чем вы готовы выполнить (фактор удобства). Вы можете запустить эти сервисы под `root` и положиться на другие механизмы обнаружения вторжений, которые могут пройти через них.

Другая большая потенциальная `root` брешь в системе это suid-root и sgid исполняемые файлы. Большинство этих исполняемых файлов, таких как rlogin, установлены в /bin, /sbin, /usr/bin, или /usr/sbin. Хотя ничто не может быть безопасно на 100%, находящиеся по умолчанию в системе suid и sgid исполняемые файлы могут быть признаны достаточно безопасными. Но `root` бреши все еще обнаруживаются в этих исполняемых файлах. `root` брешь, обнаруженная в `Xlib` в 1998 делала xterm (который обычно suid) подверженным взлому. Лучше сразу принять меры предосторожности, чем сожалеть потом. Предусмотрительный системный администратор ограничит права запуска suid исполняемых файлов, которые должны запускаться пользователями группы staff, только этой группой, а также запретит доступ (`chmod 000`) к тем исполняемым файлам suid, которые никем не используются. Серверу без монитора обычно не требуется исполняемый файл xterm. Исполняемые sgid исполняемые файлы могут быть почти так же опасны. Если нарушитель сможет взломать sgid-kmem исполняемый файл, он возможно сможет прочесть /dev/kmem и таким образом получить файл зашифрованных паролей, что потенциально делает возможным взлом любой защищённой паролем учётной записи. Аналогично нарушитель, проникший в группу `kmem`, может отслеживать последовательности клавиш, отправляемые через псевдо-терминалы, включая те, что используют защищённые соединения. Нарушитель, вошедший в группу `tty` может сделать вывод почти на любой пользовательский терминал. Если пользователь работает с терминальной программой или эмулятором с возможностью эмуляции клавиатуры, взломщик может потенциально сгенерировать поток данных, который заставит терминал пользователя ввести команду, и она будет запущена с правами этого пользователя.

13.3.3. Защита учётных записей пользователей

Учетные записи пользователей обычно сложнее всего защитить. Вы можете ввести драконовские ограничения доступа к служебным учётным записям, заменив их пароли на

символ “*”, но возможно не сможете сделать то же с обычными учётными записями пользователей. Если есть такая возможность, вы возможно сможете защитить учётные записи пользователей соответствующим образом. Если нет, просто более бдительно отслеживайте эти учётные записи. Использование ssh и Kerberos для учётных записей пользователей более проблематично, поскольку требует дополнительной административной работы и технической поддержки, но все же это решение лучше, чем файл с шифрованными паролями.

13.3.4. Защита файла паролей

Единственный абсолютно надежный способ это замена на `*` максимально возможного количества паролей и использование ssh или Kerberos для доступа к таким учётным записям. Хотя файл с шифрованными паролями (/etc/spwd.db) доступен для чтения только `root`, возможно, что нарушитель сможет получить доступ на чтение к этому файлу, даже если не получит права `root` на запись.

Ваши скрипты безопасности должны всегда проверять и составлять отчет об изменениях файла паролей (обратитесь к разделу Проверка целостности файлов ниже по тексту).

13.3.5. Защита ядра, raw устройств и файловых систем

Если атакующий взломает `root`, он сможет сделать практически все, но есть способы усложнить его задачу. Например, в большинстве современных ядер встроено устройство перехвата пакетов. В FreeBSD оно называется bpf. Нарушитель обычно пытается запустить перехват пакетов на взломанной машине. Вы не должны предоставлять ему такой возможности, на большинстве систем устройство bpf не должно быть встроено в ядро.

Но даже если вы выключите устройство bpf, все еще остаются проблемы, связанные с устройствами /dev/mem и /dev/kmem. Нарушитель все еще может писать на дисковые raw устройства. Есть также другая возможность ядра, загрузка модулей, kldload(8). Активный нарушитель может использовать KLD модуль для установки собственного устройства bpf или другого перехватывающего устройства на работающее ядро. Для решения этих проблем запускайте ядро с большим уровнем безопасности, как минимум 1. Уровень безопасности может быть установлен с помощью `sysctl` через переменную `kern.securelevel`. После установки уровня безопасности в 1 доступ на запись в raw устройства будет запрещена и полностью заработают специальные флаги `chflags`, такие как `schg`. Убедитесь также, что флаг `schg` установлен на критически важных загрузочных исполняемых файлах, каталогах и файлах скриптов - на всем, что запускается до установке уровня безопасности. Это требует большого объема работы, и обновление системы на более высоком уровне безопасности может стать гораздо сложнее. Вы можете пойти на компромисс и запускать систему на высоком уровне безопасности, но не устанавливать флаг `schg` для каждого существующего системного файла и каталога. Другая возможность состоит в монтировании / и /usr только для чтения. Необходимо заметить, что такие правила слишком жесткие и могут помешать обнаружению вторжения.

13.3.6. Проверка целостности файлов: исполняемые, конфигурационные файлы и т.д.

Вы можете защищать только ядро, файлы настройки и управления системой только до тех пор, пока эта защита не вступит в конфликт с удобством работы в системе. Например, использование `chflags` для установки бита `schg` на большинство файлов в / вероятно может только навредить, поскольку хотя и может защитить файлы, препятствует обнаружению. Последний слой системы безопасности, возможно, наиболее важный - обнаружение. Остальные меры безопасности практически бесполезны (или, что еще хуже, могут дать вам ложное ощущение безопасности) если вы не обнаружите потенциальное вторжение. Половина функций системы безопасности направлена на замедление атакующего, а не на его остановку, для того, чтобы дать системе обнаружения возможность поймать нарушителя на месте преступления.

Лучший способ обнаружения вторжения - отслеживание измененных, отсутствующих, или неожиданно появившихся файлов. Для наблюдения за измененными файлами лучше всего использовать другую (зачастую централизованную) систему с ограниченным доступом. Добавление написанных вами скриптов к этой дополнительно защищенной системе с ограниченным доступом делает ее практически невидимой для потенциальных взломщиков, и это важно. В целях достижения максимального эффекта вам может потребоваться предоставить этой системе доступ к другим машинам в сети, обычно с помощью NFS экспорта только для чтения или сгенерировав пары ключей ssh для доступа к другим машинам по ssh. Помимо большого объема сетевого трафика, NFS более скрытый метод - он позволяет контролировать файловые системы на каждом клиентском компьютере практически незаметно. Если ваш сервер с ограниченным доступом подключен к клиентским компьютерам через коммутатор, NFS метод это зачастую лучший выбор. При соединении через концентратор, или через несколько маршрутизаторов, NFS метод может стать слишком небезопасным и использование ssh может стать лучшим выбором даже несмотря на то, что ssh оставляет следы своей работы.

Как только у вас появился сервер с ограниченным доступом, и как минимум доступ на чтение в клиентских системах, потребуется написать скрипты для выполнения мониторинга. При наличии доступа по NFS вы можете написать скрипты с помощью простых системных утилит, таких как find(1) и md5(1). Лучше всего подсчитывать md5 файлов на клиентском компьютере как минимум один раз в день, а файлы, контролирующие запуск из /etc и /usr/local/etc даже более часто. При обнаружении расхождений в md5, контролирующий компьютер должен просигналить системному администратору проверить изменившиеся файлы. Хороший скрипт безопасности проверит также наличие несоответствующих исполняемых suid файлов и новых или измененных файлов в системных разделах / и /usr.

При использовании ssh вместо NFS, написать скрипты безопасности гораздо сложнее. Вам обязательно потребуется скопировать (`scp`) скрипты на клиентский компьютер, сделать из невидимыми, и для безопасности потребуется также скопировать исполняемые файлы (такие как find), которые будут использоваться скриптом. Приложение ssh на клиентском компьютере может быть уже взломано. В конечном итоге, без ssh не обойтись при работе через небезопасные соединения, но его гораздо сложнее использовать.

Хороший скрипт безопасности проверит также изменения в файлах настройки,

работающих при подключении пользователей и служебных учётных записей: .rhosts, .shosts, .ssh/authorized_keys и так далее... файлы, которые могли не попасть в область проверки `MD5`.

Если для пользователей выделен большой объем дискового пространства, проверка каждого файла на таких разделах может занять слишком много времени. В таком случае установка флагов монтирования для запрета suid исполняемых файлов и устройств на таких разделах это хорошая идея. Примените параметры mount(8) `nodev` и `nosuid`. Проверяйте эти разделы в любом случае, хотя бы раз в неделю, поскольку необходимо обнаруживать попытки взлома, независимо от того, эффективны они или нет.

Учет процессов (accton(8)) это относительно несложная возможность операционной системы, которая может помочь как механизм обнаружения состоявшихся вторжений. Она особенно полезна для обнаружения пути проникновения нарушителя в систему, если файл не был затронут проникновением.

Наконец, скрипты безопасности должны обработать лог файлы, которые необходимо создавать настолько защищенным способом, насколько это возможно - подключение syslog удаленно может быть очень полезным. Злоумышленник попытается уничтожить следы взлома, и лог файлы критически важны для системного администратора, пытающегося отследить время и метод первого проникновения. Один из надежных способов получения лог файлов является подключение системной консоли к последовательному порту и постоянный сбор информации через защищенную машину, отслеживающую консоли.

13.3.7. Паранойя

Немного паранойи никогда не повредит. Как правило, системный администратор может добавлять элементы безопасности в любом количестве, пока это не влияет на удобство, а также некоторое количество элементов безопасности, *влияющих* на удобство. Что даже более важно, системный администратор должен немного изменить их - если вы используете рекомендации, например те, что даны в этом документе, они становятся известны атакующему, который также имеет доступ к этому документу.

13.3.8. Атаки DoS

Этот раздел охватывает DoS атаки. DoS атаки это обычно пакетные атаки. Хотя против современной атаки с подделкой пакетов, которая перегружает сеть, мало что можно сделать, вы можете ограничить повреждения, убедившись, что атака не может обрушить ваши сервера.

1. Ограничение количества порождаемых процессов.
2. Уменьшение последствий springboard атак (ICMP ответ, широковещательный ping и т.д.).
3. Кэш маршрутизации ядра.

Обычная DoS атака против порождающего процессы сервера пытается исчерпать ресурсы сервера по процессам, файловым дескрипторам и памяти до тех пор, пока машина не "повиснет". У inetd (обратитесь к inetd(8)) есть несколько параметров, позволяющих ограничить такие атаки. Необходимо учесть, что хотя можно предотвратить падение

системы, в общем случае невозможно предотвратить прекращение работы сервиса. Внимательно прочтите страницу справочника и обратите особое внимание на параметры `-c`, `-C`, и `-R`. Учтите, что параметр `-C` не работает в случае атак с использованием поддельных IP пакетов, поэтому как правило необходимо использование комбинации параметров. Некоторые standalone сервисы используют собственные параметры, ограничивающие порождение процессов.

У Sendmail есть собственный параметр `-OMaxDaemonChildren`, которая работает гораздо лучше, чем параметр sendmail, ограничивающий нагрузку. Вам необходимо задать параметр запуска sendmail`MaxDaemonChildren` достаточно большим, чтобы обслуживать ожидаемую нагрузку, но так, чтобы компьютер мог обслужить такое количество приложений sendmail без падения системы. Хорошей мерой является запуск sendmail в режиме очереди (`-ODeliveryMode=queued`) и запуск даемона (`sendmail -bd`) отдельно от очереди (`sendmail -q15m`). Если вы все же хотите организовать доставку в режиме реального времени, запускайте очередь с меньшим интервалом `-q1m`, но убедитесь в правильной установке параметра sendmail `MaxDaemonChildren` для предотвращения ошибок.

Syslogd может быть атакован непосредственно, настоятельно рекомендуется использовать параметр `-s` если это возможно и параметр `-a` в остальных случаях.

Вы также должны быть очень осторожны с сервисами, совершающими обратное подключение, например, с TCP Wrapper и его обратным identd-запросом, который может быть атакован напрямую. По этой причине возможность TCP Wrapper генерировать обратный ident обычно не следует использовать.

Правильным будет запрет доступа к внутренним сервисам из внешней сети путем соответствующей настройки брандмауэра на внешнем маршрутизаторе. Идея в том, чтобы предотвратить перегрузку сервисов атаками из внешней сети, а кроме того защитить `root` от взлома через сеть. Всегда настраивайте исключающий брандмауэр, т.е. "закрыть все *кроме* портов A, B, C, D, и M-Z". Этим способом вы можете закрыть все порты нижнего диапазона, кроме явно указанных, таких как named (если вы поддерживаете интернет-зону), ntalkd, sendmail, и других сервисов, доступных из интернет. Если вы попробуете настроить брандмауэр другим способом - включающий, или разрешающий брандмауэр, есть большой шанс забыть "закрыть" пару сервисов, или добавить новый внутрисетевой сервис и забыть обновить брандмауэр. Вы можете открыть диапазон портов с большими номерами для обычных приложений без угрозы портам нижнего диапазона. Учтите также, что FreeBSD позволяет вам контролировать диапазоны портов, используемые для динамической привязки через различные переменные `sysctl``net.inet.ip.portrange` (`sysctl -a | fgrep portrange`), что позволяет упростить настройку брандмауэра. Например, вы можете использовать обычный диапазон портов со значениями от 4000 до 5000, и диапазон портов с большими номерами от 49152 до 65535, а затем заблокировать все до 4000 порта (конечно оставив доступ из интернет к определенным портам.

Другой распространенный тип DoS атак называется springboard - сервер атакуется таким образом, что генерируемые ответы перегружают его, локальную сеть или какие-то другие компьютеры. Наиболее распространенная атака этого вида это *широковещательная ICMP ping атака*. Атакующий подделывает пакеты ping, подставляя IP адрес машины, которую он намеревается атаковать, и отправляет их на широковещательный адрес вашей локальной сети. Если ваш внешний маршрутизатор не настроен на отбрасывание пакетов ping на

широковещательные адреса, ваша сеть начинает генерировать соответствующие ответы на поддельный адрес, что приводит к перегрузке хоста-жертвы, особенно если атакующий использует этот же трюк с множеством широковещательных адресов в множестве сетей одновременно. Были зарегистрированы широковещательные атаки свыше ста двадцати мегабит. Другая распространенная springboard атака направлена на ICMP систему сообщения об ошибках. Конструируя пакеты, вызывающие ICMP сообщения об ошибках, атакующий может нагрузить входящее соединение сервера и вынудить сервер нагрузить исходящее соединение ICMP ответами. Этот тип атаки может также обрушить сервер, когда тот исчерпает mbuf, обычно если сервер не может ограничить число ответов ICMP, когда они генерируются слишком быстро. Используйте переменную sysctl`net.inet.icmp.icmplim`. Последний основной класс springboard атак относится к определенным внутренним сервисам inetd, таким как сервис udp echo. Атакующий просто подделывает адрес источника и адрес назначения UDP пакетов, устанавливая в их качестве соответственно echo порт сервера A и B, оба этих сервера принадлежат вашей локальной сети. Эти два сервера начинают перебрасываться этим пакетом друг с другом. Атакующий может вызвать перегрузку обеих серверов и их сетей, просто отправив несколько пакетов таким способом. Аналогичные проблемы существуют с портом chargen. Компетентный системный администратор должен отключить эти тестовые сервисы inetd.

Атаки с поддельными пакетами могут также использоваться для переполнения кэша маршрутизации ядра. Обратитесь к параметрам `sysctl``net.inet.ip.rtexpire`, `rtminexpire`, и `rtmaxcache`. Атака с поддельными пакетами, использующая произвольный IP адрес источника, заставит ядро сгенерировать временный кэшированный маршрут в таблице маршрутизации, который можно увидеть с помощью `netstat -rna | fgrep W3`. Эти маршруты обычно удаляются через 1600 секунд или около того. Если ядро определит, что кэшированная маршрутная таблица стала слишком большой, оно динамически уменьшит `rtexpire`, но никогда не станет делать его меньше чем `rtminexpire`. С этим связаны две проблемы:

1. Ядро не отреагирует достаточно быстро, когда легко нагруженный сервер будет внезапно атакован.
2. Значение `rtminexpire` недостаточно мало для поддержки работоспособности в условиях продолжительной атаки.

Если ваши серверы подключены к интернет через линию T3 или более быструю, предусмотрительно будет изменить оба значения `rtexpire` и `rtminexpire` с помощью sysctl(8). Никогда не устанавливайте ни один из этих параметров в нуль (если только вы не хотите обрушить систему). Установка обеих параметров в значение 2 секунды должна предотвратить таблицу маршрутизации от атак.

13.3.9. Проблемы, связанные с доступом к Kerberos и SSH

При использовании Kerberos и ssh необходимо учесть несколько возможных проблем. Kerberos V это отличный протокол аутентификации, но в адаптированных к нему приложениях telnet и rlogin есть несколько ошибок, которые могут сделать их непригодными к работе с бинарными потоками. К тому же, по умолчанию Kerberos не шифрует сессию, если вы не используете параметр `-x`. ssh шифрует все по умолчанию.

ssh работает очень хорошо во всех ситуациях, но пересылает ключи по умолчанию. Это означает, что если вы работаете с защищенной рабочей станции, ключи на которой дают доступ к остальной сети, и заходите по ssh на незащищенный компьютер, эти ключи могут быть использованы для взлома. Атакующему не удастся получить сами ключи, но поскольку ssh открывает порт во время входа в систему, то если на незащищенной машине взломан `root`, эти ключи могут быть использованы для доступа к другим компьютерам, на которых они действуют.

Мы рекомендуем использовать ssh в комбинации с Kerberos для служебных учётных записей если это возможно. ssh может быть собран с поддержкой Kerberos. Это уменьшает зависимость от потенциально подверженных взлому ssh ключей, и в то же время защищает пароли через Kerberos. Ключи ssh должны использоваться только для работы скриптов на защищенных компьютерах (там, где Kerberos использовать не получится). Мы также рекомендуем или выключить передачу ключей в настройках ssh, или использовать параметр `from=IP/DOMAIN`, поддерживаемый ssh в файле authorized_keys, который позволяет использовать ключи только с определенных компьютеров.

13.4. DES, MD5, и шифрование

У каждого пользователя UNIX® системы есть пароль, связанный с его учётной записью. Очевидно, что эти пароли должны быть известны только пользователю и соответствующей операционной системе. Для защиты паролей они шифруются способом, известным как "односторонний хэш", то есть их можно легко зашифровать, но нельзя расшифровать. Другими словами, то, что мы сказали чуть раньше было очевидно, но не совсем верно: операционной системе *сам пароль* неизвестен. Ей известен только пароль в *зашифрованной* форме. Единственный способ получить "обычный" пароль это простой перебор всех возможных паролей.

К сожалению, единственный способ шифрования пароля при появлении UNIX® был основан на DES, Data Encryption Standard. Это не было проблемой для пользователей, живущих в США, но поскольку исходный код DES нельзя было экспортировать из США, FreeBSD нашла способ одновременно не нарушать законов США и сохранить совместимость со всеми другими вариантами UNIX®, где все еще использовался DES.

Решение было в разделении библиотек шифрования, чтобы пользователи в США могли устанавливать и использовать библиотеки DES, а у остальных пользователей был метод шифрования, разрешенный к экспорту. Так FreeBSD пришла к использованию MD5 в качестве метода шифрования по умолчанию. MD5 считается более безопасным, чем DES, поэтому установка DES рекомендуется в основном из соображений совместимости.

13.4.1. Определения механизма шифрования

На данный момент библиотека поддерживает хэши DES, MD5 и Blowfish. По умолчанию FreeBSD использует для шифрования паролей MD5.

Довольно легко определить какой метод шифрования используется в FreeBSD. Один из способов это проверка файла /etc/master.passwd. Пароли, зашифрованные в хэш MD5 длиннее, чем те, что зашифрованы с помощью DES и начинаются с символов `$1$`. Пароли,

начинающиеся с символов `$2a$` зашифрованы с помощью Blowfish. Пароли, зашифрованные DES не содержат каких-то определенных идентифицирующих символов, но они короче, чем пароли MD5 и закодированы в 64-символьном алфавите, не содержащем символа `$`, поэтому относительно короткая строка, не начинающаяся с этого символа это скорее всего DES пароль.

Формат паролей, используемых для новых паролей, определяется параметром `passwd_format` в /etc/login.conf, которое может принимать значения `des`, `md5` или `blf`. Обратитесь к странице справочника login.conf(5) за дополнительной информацией о параметрах login.

13.5. Одноразовые пароли

FreeBSD использует для одноразовых паролей OPIE (One-time Passwords In Everything). OPIE по умолчанию использует MD5.

Есть три различных вида паролей, о которых мы поговорим ниже. Первый вид это ваш обычный пароль UNIX® или пароль Kerberos; мы будем называть его "пароль UNIX®". Второй вид это одноразовый пароль, сгенерированный программой OPIE opiekey(1) и принимаемый командой opiepasswd(1) и в приглашении login; мы будем называть их "одноразовыми паролями". Последний вид паролей это защищенные пароли, которые вы передаете программам `opiekey` (и иногда `opiepasswd`), и которые эти программы используют для создания одноразовых паролей; мы будем называть его "защищенными паролями" или просто "паролями".

Защищенный пароль не имеет никакого отношения к вашему паролю UNIX®; они могут быть одинаковыми, но это не рекомендуется. Защищенные пароли OPIE не ограничены 8-ю символами, как старые UNIX® пароли, они могут быть настолько длинными, насколько вы захотите. Очень часто используются пароли длиной в шесть или семь символов. По большей части система OPIE работает полностью независимо от системы паролей UNIX®.

Помимо паролей, есть два других вида данных, важных для OPIE. Первый, известный как "seed" или "ключ", состоит из двух букв и пяти цифр. Другой, называемый "счетчиком цикла", это номер от 1 до 100. OPIE создает одноразовый пароль, соединяя ключ и защищенный пароль, а затем применяя MD4 столько раз, сколько указано счетчиком цикла и выдает результат в виде шести коротких слов на английском. Эти шесть слов на английском и есть ваш одноразовый пароль. Система аутентификации (как правило PAM) хранит последний использованный одноразовый пароль, и пользователь аутентифицируется если хэш вводимого пользователем пароля совпадает с предыдущим паролем. Поскольку используется односторонний хэш, невозможно сгенерировать следующий одноразовый пароль если получен предыдущий; счетчик цикла уменьшается после каждого успешного входа для поддержки синхронизации пользователя с программой login. Когда счетчик цикла уменьшается до 1, набор OPIE должен быть переинициализирован.

В каждой из обсуждаемых ниже систем задействованы три программы. Программа `opiekey` получает счетчик цикла, ключ и защищенный пароль и создает одноразовый пароль или последовательный список одноразовых паролей. Программа `opiepasswd` используется для инициализации OPIE соответственно, и для смены паролей, счетчиков цикла, или ключей;

она принимает защищенный пароль или счетчик цикла, ключ и одноразовый пароль. Программа opieinfo проверяет соответствующий файл (/etc/opiekeys) и печатает текущий счетчик цикла и ключ вызывающего пользователя.

Мы рассмотрим четыре вида операций. Первая это использование opiepasswd через защищенное соединение для первоначальной настройки системы одноразовых паролей, или для изменения пароля или ключа. Вторая операция это использование в тех же целях opiepasswd через незащищенное соединение, в сочетании с opiekey через защищенное соединение. Третья это использование opiekey для входа через незащищенное соединение. Четвертая это использование opiekey для генерации набора ключей, которые могут быть записаны или распечатаны для соединения из места, где защищенное соединение недоступно.

13.5.1. Защищенная установка соединения

Для первоначальной настройки OPIE используется команда opiepasswd:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:
ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

В приглашениях Enter new secret pass phrase: или Enter secret password:, введите пароль или фразу. Запомните, это не тот пароль, с которым вы будете входить, он используется для генерации одноразовых паролей. Строка "ID" содержит информацию для вашего конкретного случая: имя пользователя, счетчик цикла и ключ. При входе система запомнит эти параметры и отправит их вам, поэтому их не надо запоминать. В последней строке находится одноразовый пароль, соответствующий этим параметрам и секретному паролю; если вы войдете в систему сразу, используйте этот одноразовый пароль.

13.5.2. Незащищенная установка соединения

Для инициализации или изменения защищенного пароля через незащищенное соединение, вам потребуется существующее защищенное соединение куда-то, где вы сможете запустить opiekey; это может быть shell на компьютере, которому вы доверяете. Вам потребуется также установить значение счетчика цикла (100 возможно подойдет), и задать ключ или использовать сгенерированный. Через незащищенное соединение (к компьютеру, на котором производится настройка), используйте команду opiepasswd:

```
% opiepasswd
```

```
Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
        otp-md5 498 to4268 ext
        Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
        otp-md5 499 to4269
        Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Чтобы принять ключ по умолчанию нажмите `Enter`. Затем, перед вводом пароля доступа введите те же параметры в вашем защищенном соединении или средстве доступа OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Теперь переключитесь на незащищенное соединение и скопируйте одноразовый пароль, сгенерированный соответствующей программой.

13.5.3. Создание одного одноразового пароля

Как только вы настроите OPIE, во время входа появится приглашение вроде этого:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
otp-md5 498 gr4269 ext
Password:
```

Кроме того, у OPIE есть полезная особенность (не показанная здесь): если вы нажмете `Enter` в приглашении на ввод пароля, включится эхо, и вы сможете увидеть то, что вводите. Это может быть очень полезно, если вы пытаетесь ввести пароль вручную, например с распечатки.

В этот момент вам потребуется сгенерировать одноразовый пароль, чтобы ввести его в приглашение. Это должно быть выполнено на защищенной системе, в которой вы можете запустить `opiekey` (есть версии для DOS, Windows® и Mac OS®). Им требуются значения

счетчика цикла и ключ в качестве параметров командной строки. Вы можете скопировать и вставить их прямо из приглашения login компьютера, на который входите.

В защищенной системе:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Теперь, когда у вас есть одноразовый пароль, можете продолжить вход в систему.

13.5.4. Создание нескольких одноразовых паролей

Иногда вы отправляетесь туда, где нет доступа к защищенному компьютеру или защищенному соединению. В этом случае, можно использовать команду `opiekey` для создания нескольких одноразовых паролей, которые вы сможете распечатать и забрать с собой. Например:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

Параметр `-n 5` запрашивает пять паролей, `30` указывает значение последнего счетчика цикла. Обратите внимание, что пароли печатаются в *обратном* по сравнению с обычным использованием порядке. Если вы действительно параноик, перепишите результат вручную; иначе скопируйте и передайте его `lpr`. Обратите внимание, что каждая линия содержит как счетчик цикла, так и одноразовый пароль; вам может показаться удобным отрывать пароль после использования.

13.5.5. Ограничение использования UNIX® паролей

OPIE может ограничивать использование паролей UNIX® на основе IP адреса. Соответствующий файл называется /etc/opieaccess, он существует по умолчанию. Обратитесь к opieaccess(5) за более подробной информацией об этом файле и о предосторожностях, которые вы должны предпринять при использовании этого файла.

Вот пример файла opieaccess:

```
permit 192.168.0.0 255.255.0.0
```

Эта строка позволяет пользователям, чей IP адрес (который подвержен подделке) соответствует указанному значению и маске, входить с паролем UNIX®.

Если ни одно из правил в opieaccess не сработало, поведением по умолчанию является запрет всех не-OPIE входов.

13.6. TCP Wrappers

Каждый, кто знаком с inetd(8), возможно когда-то слышал о TCP Wrappers. Но немногие полностью понимают их полезность в сетевой среде: большинство используют брандмауэр. Хотя его применимость очень широка, есть вещи, с которыми брандмауэр не может работать, такие как отправка текста обратно вызывающей стороне. Программное обеспечение уровня TCP может делать это и многое другое. В следующих нескольких разделах обсуждаются многие возможности TCP Wrappers, и, когда это необходимо, даются примеры настроек.

Программное обеспечение TCP Wrappers расширяет возможность `inetd` по поддержке каждого даемона. С ним становится возможным протоколирование, возврат сообщений вызывающей стороне, ограничение подключений внутренней сетью и т.п. Хотя некоторые из этих возможностей могут быть реализованы брандмауэром, TCP Wrappers не только предоставляют дополнительный уровень защиты, но и дают больше контроля над системой, чем это возможно с брандмауэром.

Расширенная функциональность обработчиков TCP не может заменить хороший сетевой экран. Тем не менее, обработчики TCP могут использоваться совместно с сетевым экраном и другими средствами обеспечения информационной безопасности, обеспечивая тем самым дополнительный уровень защиты системы.

Поскольку рассматривается расширение к настройкам `inetd`, предполагается, что читатель ознакомился с разделом о настройке inetd.

> Хотя программы, запускаемые из inetd(8), на самом деле не соответствуют термину "даемоны", существует традиция называть их именно так. Этот термин и используется в данном разделе.

13.6.1. Начальная настройка

Единственное требование для использования TCP Wrappers в FreeBSD это наличие в rc.conf параметров запуска `inetd`-Ww`; это настройки по умолчанию. Конечно, ожидается также наличие правильной настройки /etc/hosts.allow, но syslogd(8) отправит сообщения в системный протокол если что-то не так.

> В отличие от других реализаций TCP Wrappers, использование hosts.deny не поддерживается. Все параметры настройки должны быть помещены в /etc/hosts.allow.

В простейшей конфигурации, политика подключения сводится к разрешению или блокированию в зависимости от параметров в /etc/hosts.allow. Настройка в FreeBSD по умолчанию заключается в разрешении подключения к любому даемону, запущенному из `inetd`. Изменение этого поведения будет обсуждаться только после рассмотрения базовой настройки.

Базовая настройка обычно принимает форму `daemon : address : action`, где `daemon` это имя даемона, который запускается `inetd`. В поле `address` может находиться имя хоста, IP адрес, или IPv6 адрес, заключенный в квадратные скобки ([ ]). Поле action может принимать значения allow или deny, чтобы соответственно разрешать или запрещать доступ. Помните, что поиск правил производится до первого совпадения. При обнаружении совпадения применяется соответствующее правило и поиск прерывается.

Существуют и другие параметры, но они будут описаны в следующих разделах. Простая конфигурация может быть, например, такой: для разрешения соединений по протоколу POP3 к даемону mail/qpopper, в hosts.allow необходимо добавить следующие строки:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

После добавления этой строки, `inetd` необходимо перезапустить. Это можно выполнить командой kill(1) или скриптом /etc/rc.d/inetd с параметром restart.

13.6.2. Расширенная конфигурация

У TCP Wrappers имеются дополнительные параметры; они дают дополнительные возможности контроля над соединениями. Иногда бывает полезно возвращать комментарий определенным хостам или при подключении к определенным даемонам. В других случаях может быть необходимо добавить запись в лог файл, или отправить письмо администратору. В определенных ситуациях сервис должен использоваться только для локальных соединений. Все это возможно с использованием параметров с шаблонами, символами подстановки и путем выполнения внешних команд. Следующие два раздела посвящены этим типам настроек.

13.6.2.1. Внешние команды

Предположим ситуацию, в которой соединение должно быть запрещено, а о причине необходимо сообщить вызывающей стороне. Как это можно сделать? Соответствующую возможность предоставляет параметр `twist`. При попытке подключения выполняется команда или скрипт, заданный этим параметром. Пример дан в файле hosts.allow:

```
# The rest of the daemons are protected.
ALL : ALL \
        : severity auth.info \
        : twist /bin/echo "You are not welcome to use %d from %h."
```

В этом примере сообщение, "You are not allowed to use `daemon` from `hostname`." будет

возвращено от всех даемонов, которые не были предварительно настроены в файле доступа. Обратите внимание, что возвращаемое сообщение *должно* быть заключено в кавычки; из этого правила нет исключений.

Возможна реализация DoS атаки, когда группа атакующих производит множество запросов на подключение.

Возможно также использование параметра `spawn`. Как и параметр `twist`, параметр `spawn` подразумевает запрет соединения и может использоваться для запуска команд или скриптов. В отличие от `twist`, `spawn` не отправляет ответ вызывающей стороне. Например, следующая конфигурация:

```
# We do not allow connections from example.com:
ALL : .example.com \
        : spawn (/bin/echo %a from %h attempted to access %d >> \
          /var/log/connections.log) \
        : deny
```

отклонит все попытки соединения из домена `*.example.com`; имя хоста, IP адрес и даемон протоколируются в файл /var/log/connections.log.

Помимо приведенных выше символов подстановки, например %a, существует еще несколько символов. Обратитесь к странице hosts_access(5) справочной системы за полным списком.

13.6.2.2. Параметры - шаблоны

До этого момента в примерах использовался шаблон `ALL`. Существуют и другие параметры, функциональность которых в дальнейшем может быть расширена. `ALL` соответствует любому даемону, домену или IP адресу. Другой доступный шаблон это `PARANOID`, который соответствует хосту, IP адрес которого может быть подделан. Другими словами, `paranoid` может быть использован для определения действия с хостами, IP адрес которых не соответствует имени хоста. Вот пример применения этого параметра:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```

В этом примере все запросы на подключения к `sendmail` от хостов, IP адрес которых не соответствует имени хоста, будут отклонены.

Использование `PARANOID` невозможно, если у клиента или сервера неправильно настроен DNS. В таких случаях необходимо вмешательство администратора.

Более подробная информация о шаблонах и их возможностях дана на странице hosts_access(5) справочной системы.

Для того, чтобы любая выбранная конфигурация заработала, в hosts.allow необходимо закомментировать первую строку настройки. В начале раздела об этом не упоминалось.

13.7. KerberosIV

Kerberos это сетевая дополнительная система/протокол, которая делает возможной аутентификацию пользователей через сервисы на защищенном сервере. Такие сервисы, как удаленный вход, удаленное копирование, защищенное копирование файлов между системами и другие задачи с высоким риском становятся допустимо безопасными и более контролируемыми.

Последующие инструкции могут использоваться в качестве руководства по настройке поставляемого с FreeBSD Kerberos. Тем не менее, вам могут потребоваться страницы справочника полного дистрибутива.

13.7.1. Установка KerberosIV

Kerberos это опциональный компонент FreeBSD. Простейший способ установки этой программы это выбор `krb4` или `krb5` из sysinstall во время первой установки FreeBSD. Будет установлен "eBones" (KerberosIV) или "Heimdal" (Kerberos5) вариант Kerberos. Включение этих реализаций объясняется тем, что они разработаны вне США/Канады и доступны вне этих стран, поскольку на них не влияют ограничения на экспорт криптографического кода из США.

Кроме того, реализация MIT Kerberos доступна из Коллекции Портов в виде пакета security/krb5.

13.7.2. Создание базы данных

Это необходимо сделать только на сервере Kerberos. Во-первых, убедитесь что не осталось старой базы данных Kerberos. Войдите в каталог /etc/kerberosIV и убедитесь, что в нем находятся только эти файлы:

```
# cd /etc/kerberosIV
# ls
README          krb.conf        krb.realms
```

Если присутствуют еще какие-то файлы (такие как principal.* или master_key), используйте команду `kdb_destroy` для удаления старой базы данных Kerberos, или, если Kerberos не запущен, просто удалите эти файлы.

Затем отредактируйте файлы krb.conf и krb.realms, введя ваши данные. В этом примере уникальный идентификатор `EXAMPLE.COM`, сервер `grunt.example.com`. Отредактируем или создадим файл krb.conf:

```
# cat krb.conf
EXAMPLE.COM
```

```
EXAMPLE.COM grunt.example.com admin server
CS.BERKELEY.EDU okeeffe.berkeley.edu
ATHENA.MIT.EDU kerberos.mit.edu
ATHENA.MIT.EDU kerberos-1.mit.edu
ATHENA.MIT.EDU kerberos-2.mit.edu
ATHENA.MIT.EDU kerberos-3.mit.edu
LCS.MIT.EDU kerberos.lcs.mit.edu
TELECOM.MIT.EDU bitsy.mit.edu
ARC.NASA.GOV trident.arc.nasa.gov
```

В этом примере другие идентификаторы введены для иллюстрации настройки с несколькими хостами. С целью упрощения настройки вы можете не включать их.

Первая строка содержит идентификатор, под которым работает эта система. Остальные строки связывают идентификаторы с именами хостов. Сначала указывается идентификатор, затем хост под этим идентификатором, работающий как "центр распространения ключей". Слова `admin server` с последующим именем хоста означают, что этот хост также является сервером администрирования базы данных. За дальнейшей информацией об этих терминах обратитесь к страницам справочника по Kerberos.

Мы добавили `grunt.example.com` к идентификатору `EXAMPLE.COM` и кроме того сопоставили всем хостам в домене `.example.com` идентификатор `EXAMPLE.COM`. Файл krb.realms будет выглядеть так:

```
# cat krb.realms
grunt.example.com EXAMPLE.COM
.example.com EXAMPLE.COM
.berkeley.edu CS.BERKELEY.EDU
.MIT.EDU ATHENA.MIT.EDU
.mit.edu ATHENA.MIT.EDU
```

Как и в предыдущем примере, другие идентификаторы добавлены только для примера. С целью упрощения настройки вы можете не включать их.

В первой строке *определенная* система сопоставляется с идентификатором. В остальных строках показано, сопоставить идентификатору остальные системы определенного поддомена.

Теперь мы готовы к созданию базы данных. Потребуется всего лишь запустить сервер Kerberos (или центр распространения ключей). Используйте для этого `kdb_init`:

```
# kdb_init
Realm name [default  ATHENA.MIT.EDU ]: EXAMPLE.COM
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.

Введите главный ключ Kerberos:
```

Теперь мы должны сохранить ключ, чтобы сервера на локальных компьютерах могли его взять. Используйте для этого команду `kstash`:

```
# kstash

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
```

Этой командой зашифрованный главный пароль сохранен в /etc/kerberosIV/master_key.

13.7.3. Запуск Kerberos

Для каждой системы, защищаемой Kerberos, в базу данных должны быть добавлены две записи. Это `kpasswd` и `rcmd`. Они добавляются вместе с именем системы.

Эти даемоны, kpasswd и rcmd позволяют другим системам изменять пароли Kerberos и запускать такие команды как rcp(1), rlogin(1), rsh(1).

Теперь добавим эти записи:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: passwd
Instance: grunt

<Not found>, Create [y] ? y

Principal: passwd, Instance: grunt, kdc_key_ver: 1
New Password:                    <---- enter RANDOM here
Verifying password

New Password: <---- enter RANDOM here

Random password [y] ? y

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
```

```
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name: rcmd
Instance: grunt

<Not found>, Create [y] ?

Principal: rcmd, Instance: grunt, kdc_key_ver: 1
New Password:       <---- enter RANDOM here
Verifying password

New Password:           <---- enter RANDOM here

Random password [y] ?

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name:         <---- null entry here will cause an exit
```

13.7.4. Создание файла настройки сервера

Теперь необходимо создать все записи сервисов, которые были определены для каждого компьютера. Используем для этого команду `ext_srvtab`. Будет создан файл, который должен быть скопирован или перемещен *безопасным способом* в каталог /etc/kerberosIV каждого Kerberos клиента. Этот файл должен присутствовать на каждом сервере и клиенте, он необходим для работы Kerberos.

```
# ext_srvtab grunt
Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Generating 'grunt-new-srvtab'....
```

Эта команда создаст временный файл, который должен быть переименован в srvtab, чтобы серверы смогли обратиться к нему. Используйте команду mv(1) для перемещения его в исходной системе:

```
# mv grunt-new-srvtab srvtab
```

Если файл предназначен для клиентской системы, и сеть не безопасна, скопируйте client-new-srvtab на съемный носитель и перенесите файл с его помощью. Убедитесь, что

переименовали его в srvtab в каталоге /etc/kerberosIV клиента, и что режим доступа к нему 600:

```
# mv grumble-new-srvtab srvtab
# chmod 600 srvtab
```

13.7.5. Пополнение базы данных

Теперь необходимо добавить в базу данных пользователей. Во-первых, создадим запись для пользователя `jane`. Используйте команду `kdb_edit`:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
Instance:

<Not found>, Create [y] ? y

Principal: jane, Instance: , kdc_key_ver: 1
New Password:                <---- enter a secure password here
Verifying password

New Password:                <---- re-enter the password here
Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name:        <---- null entry here will cause an exit
```

13.7.6. Тестирование всей системы

Во-первых, запустите даемоны Kerberos. При правильном редактировании файла /etc/rc.conf они запустятся автоматически при перезагрузке. Это необходимо только на сервере Kerberos. Клиенты Kerberos получат все необходимые данные из каталога /etc/kerberosIV.

```
# kerberos &
Kerberos server starting
```

```
Sleep forever on error
Log file is /var/log/kerberos.log
Current Kerberos master key version is 1.

Master key entered. BEWARE!

Current Kerberos master key version is 1
Local realm: EXAMPLE.COM
# kadmind -n &
KADM Server KADM0.0A initializing
Please do not use 'kill -9' to kill this job, use a
regular kill instead

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
```

Теперь для получения доступа через созданного пользователя `jane` используйте `kinit`:

```
% kinit jane
MIT Project Athena (grunt.example.com)
Kerberos Initialization for "jane"
Password:
```

Попробуйте просмотреть имеющиеся данные с помощью `klist`:

```
% klist
Ticket file:    /tmp/tkt245
Principal:      jane@EXAMPLE.COM

  Issued           Expires          Principal
Apr 30 11:23:22  Apr 30 19:23:22  krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

Теперь попробуйте изменить пароль с помощью passwd(1), чтобы убедиться, что даемон kpasswd может получить информацию из базы данных Kerberos:

```
% passwd
realm EXAMPLE.COM
Old password for jane:
New Password for jane:
Verifying password
New Password for jane:
Password changed.
```

13.7.7. Включение su

Kerberos позволяет назначить *каждому* пользователю, который нуждается в привилегиях root, свой *собственный* пароль su(1). Необходимо добавить учётную запись, которой разрешено получать root доступ через su(1). Это делается путем связывания учётной записи root с пользовательской учётной записью. Создадим в базе данных Kerberos запись jane.root с помощью kdb_edit:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
Instance: root

<Not found>, Create [y] ? y

Principal: jane, Instance: root, kdc_key_ver: 1
New Password:                    <---- enter a SECURE password here
Verifying password

New Password:            <---- re-enter the password here

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ? 12 <--- Keep this short!
Attributes [ 0 ] ?
Edit O.K.
Principal name:              <---- null entry here will cause an exit
```

Теперь проверим работоспособность этой записи:

```
# kinit jane.root
MIT Project Athena (grunt.example.com)
Kerberos Initialization for "jane.root"
Password:
```

Необходимо добавить пользователя к root файлу .klogin:

```
# cat /root/.klogin
```

```
jane.root@EXAMPLE.COM
```

Теперь попробуйте выполнить su(1):

```
% su
Password:
```

и посмотрите на имеющиеся данные:

```
# klist
Ticket file:    /tmp/tkt_root_245
Principal:      jane.root@EXAMPLE.COM

  Issued           Expires          Principal
May  2 20:43:12  May  3 04:43:12  krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

13.7.8. Использование других команд

В примере выше мы создали запись (principal) jane с доступом к root (instance). Она основана на пользователе с таким же именем, как и идентификатор, что принято Kerberos по умолчанию; <principal>.<instance> в форме <username>.``root позволяет использовать su(1) для доступа к root, если соответствующие записи находятся в файле .klogin домашнего каталога root:

```
# cat /root/.klogin
jane.root@EXAMPLE.COM
```

Подобно этому, если в файле .klogin из домашнего каталога пользователя есть строки в форме:

```
% cat ~/.klogin
jane@EXAMPLE.COM
jack@EXAMPLE.COM
```

это позволит любому с идентификатором EXAMPLE.COM, кто аутентифицировался как jane или jack (с помощью команды kinit, см. выше) получить доступ к учётной записи пользователя jane или файлам этой системы (grunt) через rlogin(1), rsh(1) или rcp(1).

Например, jane может входить в другую систему используя Kerberos:

```
% kinit
MIT Project Athena (grunt.example.com)
Password:
% rlogin grunt
```

```
Last login: Mon May  1 21:14:47 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
        The Regents of the University of California.   All rights reserved.

FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

Или `jack` входит в учётную запись `jane's на этом же компьютере (файл .klogin`jane` настроен как показано выше, и в Kerberos настроена учётная запись *jack*):

```
% kinit
% rlogin grunt -l jane
MIT Project Athena (grunt.example.com)
Password:
Last login: Mon May  1 21:16:55 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
        The Regents of the University of California.   All rights reserved.
FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

13.8. Kerberos5

Все релизы FreeBSD после FreeBSD-5.1 включают поддержку только Kerberos5. Таким образом, Kerberos5 это единственная включаемая в поставку версия и его конфигурация похожа на KerberosIV во многих аспектах. Эта информация применима только к Kerberos5 из релизов после FreeBSD-5.0. Пользователи, желающие использовать пакет KerberosIV, могут установить его из порта security/krb4.

Kerberos это дополнительная сетевая система/протокол, позволяющая пользователям авторизоваться через защищенные сервисы на защищенном сервере. Такие сервисы как удаленный вход, удаленное копирование, защищенное копирование файлов между системами и другие задачи с высоким риском становятся допустимо безопасными и более контролируемыми.

Kerberos может быть описана как прокси система идентификации-проверки. Она также может быть описана как защищенная внешняя система аутентификации. Kerberos предоставляет только одну функцию - защищенную аутентификацию пользователей сети. Он не предоставляет функций авторизации (что разрешено делать пользователям) или функций аудита (какой пользователь что делает). После того, как клиент и сервер использовали Kerberos для идентификации, они могут зашифровать все соединения для гарантирования собственной безопасности и целостности данных.

Следовательно крайне рекомендуется использовать Kerberos с другими методами безопасности, предоставляющими сервисы авторизации и аудита.

Последующие инструкции могут использоваться в качестве руководства по настройке Kerberos, поставляемого с FreeBSD. Тем не менее, вам потребуется обратиться к соответствующим страницам справочника за полным описанием.

В целях демонстрации установки Kerberos, будут применены следующие обозначения:

- DNS домен ("зона") example.org.
- Уникальный идентификатор Kerberos EXAMPLE.ORG.

Используйте действующие имена доменов при настройке Kerberos даже если вы будете использовать его во внутренней сети. Это позволит избежать проблем с DNS и гарантирует возможность связи с Kerberos под другими идентификаторами.

13.8.1. История

Kerberos был создан MIT в качестве решения проблем с безопасностью сети. Протокол Kerberos использует стойкую криптографию, так что клиент может идентифицироваться на сервере (и обратно) через незащищенное сетевое соединение.

Kerberos это и имя сетевого протокола аутентификации и общий термин для описания программ, где он реализован (например, Kerberos telnet). Текущая версия протокола 5 описана в RFC 1510.

Доступно несколько свободных реализаций этого протокола, работающих на множестве операционных систем. Massachusetts Institute of Technology (MIT), где Kerberos был первоначально разработан, продолжает разрабатывать собственный пакет Kerberos. Он обычно использовался в США как криптографический продукт, и в этом качестве попадал под действие ограничений на экспорт. MITKerberos доступен в виде порта (security/krb5). Heimdal Kerberos это другая реализация версии 5, которая разрабатывалась исключительно вне США для обхода экспортных ограничений (и поэтому часто включалась в некоммерческие реализации UNIX®). Heimdal Kerberos доступен в виде порта (security/heimdal), его минимальный комплект включен в базовую установку FreeBSD.

В целях получения наибольшей аудитории, в этих инструкциях предполагается использование Heimdal включаемого в FreeBSD.

13.8.2. Настройка Heimdal KDC

Центр распространения ключей (Key Distribution Center, KDC) это централизованный сервис аутентификации, предоставляемый Kerberos - это компьютер, который предоставляет доступ через Kerberos. KDC считается доверяемым всеми другими компьютерами с определенным идентификатором Kerberos и поэтому к нему предъявляются высокие требования безопасности.

Имейте ввиду, что хотя работа сервера Kerberos требует очень немного вычислительных ресурсов, из соображений безопасности для него рекомендуется отдельный компьютер, работающий только в качестве KDC.

Перед началом настройки KDC, убедитесь что в файле /etc/rc.conf содержатся правильные настройки для работы в качестве KDC (вам может потребоваться изменить пути в соответствии с собственной системой):

```
kerberos5_server_enable="YES"
```

```
kadmind5_server_enable="YES"
```

Затем приступим к редактированию файла настройки Kerberos, /etc/krb5.conf:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
        kdc = kerberos.example.org
        admin_server = kerberos.example.org
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Обратите внимание что в файле /etc/krb5.conf подразумевается наличие у KDC полного имени `kerberos.example.org`. Вам потребуется добавить CNAME (синоним) к файлу зоны, если у KDC другое имя.

Для больших сетей с правильно настроенным сервером BINDDNS пример выше может быть урезан до:

```
[libdefaults]
      default_realm = EXAMPLE.ORG
```

Со следующими строками, добавленными в файл зоны `example.org`:

```
_kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
_kerberos._tcp      IN  SRV     01 00 88 kerberos.example.org.
_kpasswd._udp       IN  SRV     01 00 464 kerberos.example.org.
_kerberos-adm._tcp  IN  SRV     01 00 749 kerberos.example.org.
_kerberos           IN  TXT     EXAMPLE.ORG
```

Чтобы клиенты могли найти сервисы Kerberos, *необходимо* наличие или полностью настроенного /etc/krb5.conf или минимально настроенного /etc/krb5.conf*и* правильно настроенного DNS сервера.

Создадим теперь базу данных Kerberos. Эта база данных содержит ключи всех основных хостов, зашифрованных с помощью главного пароля. Вам не требуется помнить этот пароль, он хранится в файле (/var/heimdal/m-key). Для создания главного ключа запустите `kstash` и введите пароль.

Как только будет создан главный ключ, вы можете инициализировать базу данных с помощью программы `kadmin` с ключом `-l` (означающим "local"). Этот ключ сообщает `kadmin` обращаться к файлам базы данных непосредственно вместо использования сетевого сервиса `kadmind`. Это помогает решить "проблему курицы и яйца", когда обращение идет к

еще не созданной базе данных. Как только вы увидите приглашение kadmin, используйте команду init для создания базы данных идентификаторов.

Наконец, оставаясь в приглашении kadmin, создайте первую запись с помощью команды add. Оставьте неизменными параметры по умолчанию, вы всегда сможете изменить их позже с помощью команды modify. Обратите внимание, что вы всегда можете использовать команду ? для просмотра доступных параметров.

Пример создания базы данных показан ниже:

```
# kstash
Master key: xxxxxxxx
Verifying password - Master key: xxxxxxxx

# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Теперь пришло время запустить сервисы KDC. Выполните команды /etc/rc.d/kerberos start и /etc/rc.d/kadmind start для запуска сервисов. Заметьте, что ни один из поддерживающих Kerberos даемонов на этот момент запущен не будет, но у вас должна быть возможность убедиться в том, что KDC функционирует путем получения списка доступа для пользователя, которого вы только что самостоятельно создали из командной строки самого KDC:

```
% k5init tillman
tillman@EXAMPLE.ORG's Password:

% k5list
Credentials cache: FILE:/tmp/krb5cc_500
        Principal: tillman@EXAMPLE.ORG

  Issued           Expires          Principal
Aug 27 15:37:58  Aug 28 01:37:58  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

13.8.3. Сервер Kerberos с сервисами Heimdal

Для начала нам потребуется копия файла настройки Kerberos, /etc/krb5.conf. Просто скопируйте его с KDC на клиентский компьютер безопасным способом (используя сетевые утилиты, такие как scp(1), или физически, с помощью дискеты).

Затем вам понадобится файл /etc/krb5.keytab. Это основное различие между сервером,

поддерживающим Kerberos и рабочими станциями - на сервере должен быть файл keytab. В этом файле находится центральный ключ сервера, который позволяет KDC проверять все другие идентификаторы. Он должен быть помещен на сервер безопасным способом, поскольку безопасность сервера может быть нарушена, если ключ станет общедоступен. Это означает, что его передача через прозрачный канал, такой как FTP - очень плохая идея.

Обычно перенос файла keytab на сервер производится с помощью программы `kadmin`. Это удобно, поскольку вам потребуется также создать запись хоста (KDC часть krb5.keytab) с помощью `kadmin`.

Обратите внимание, что должны быть уже зарегистрированы в системе и необходимо наличие прав на использование интерфейса `kadmin` в файле kadmind.acl. Обратитесь к разделу "Remote administration" в info страницах Heimdal (`info heimdal`) за деталями по составлению списка доступа. Если вы не хотите включать удаленный доступ `kadmin`, можете просто подключиться к KDC через защищенное соединение (локальную консоль, ssh(1) или Kerberos telnet(1)) и выполнять администрирование локально с помощью `kadmin -l`.

После добавления файла /etc/krb5.conf, вы можете использовать `kadmin` с сервера Kerberos. Команда `add --random-key` позволит вам добавить запись для сервера, а команда `ext` позволит перенести эту запись в собственный keytab файл сервера. Например:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
kadmin> ext host/myserver.example.org
kadmin> exit
```

Обратите внимание, что команда `ext` (сокращение от "extract") сохраняет полученный ключ в файле /etc/krb5.keytab по умолчанию.

Если на KDC не запущен `kadmind` (возможно по соображениям безопасности) и вы не можете получить доступ к `kadmin` удаленно, возможно добавление записи хоста (`host/myserver.EXAMPLE.ORG`) непосредственно на KDC с последующим извлечением ее во временный файл (и перезаписью /etc/krb5.keytab на KDC) примерно так:

```
# kadmin
kadmin> ext --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

Затем вы можете скопировать keytab на сервер защищенным способом (например, используя `scp` или дискету). Убедитесь, что используемое имя keytab не совпадает с именем по умолчанию во избежание перезаписывания keytab на KDC.

Теперь ваш сервер может связываться с KDC (добавлен файл krb5.conf) и идентифицировать себя (добавлен файл krb5.keytab). Теперь вы готовы к включению некоторых сервисов Kerberos. В этом примере мы включим сервис `telnet`, поместив в /etc/inetd.conf

нижеприведенную строку и перезапустив сервис inetd(8) командой /etc/rc.d/inetd restart:

```
telnet    stream  tcp     nowait  root    /usr/libexec/telnetd  telnetd -a user
```

Очень важно установить ключ -a (тип аутентификации) в user. Обратитесь к странице справочника telnetd(8) за подробной информацией.

13.8.4. Клиент Kerberos с Heimdal

Настройка клиентского компьютера почти тривиально проста. Как только настройка Kerberos закончена, вам потребуется только файл настройки Kerberos, /etc/krb5.conf. Просто скопируйте его безопасным способом на клиентский компьютер с KDC.

Протестируйте клиентский компьютер, попытавшись использовать kinit, klist, и kdestroy для получения, отображения и удаления списка доступа. Соединитесь с Kerberos севером используя клиент Kerberos, если соединение не работает и получение доступа является проблемой, это скорее всего проблема сервера, а не клиента или KDC.

При тестировании приложения вроде telnet, попробуйте использовать программу перехвата пакетов (такую как tcpdump(1)), чтобы убедиться, что ваш пароль не передается незашифрованным. Попробуйте использовать telnet с параметром -x, чтобы зашифровать весь поток данных (подобно ssh).

Основные клиентские приложения Kerberos (традиционно называющиеся kinit, klist, kdestroy, и kpasswd) находятся в базовой установке FreeBSD. Обратите внимание, что в FreeBSD версий до 5.0 они были переименованы в k5init, k5list, k5destroy, k5passwd, и k5stash (хотя их обычно использовали лишь однократно).

Различные неосновные клиентские приложения Kerberos также устанавливаются по умолчанию. Здесь проявляется "минимальность" базовой установки Heimdal: telnet это единственное приложение, поддерживающее Kerberos.

Порт Heimdal добавляет некоторые отсутствующие клиентские приложения: поддерживающие Kerberos версии ftp, rsh, rcp, rlogin, и некоторые другие реже используемые программы. Порт MIT также содержит полный пакет клиентских приложений Kerberos.

13.8.5. Пользовательские файлы настройки: .k5login и .k5users

Учётные записи пользователя в Kerberos (например tillman@EXAMPLE.ORG) обычно связаны с локальными учётными записями (например с локальной учётной записью6 tillman). Клиентские приложения, такие как telnet, обычно не требуют указания имени пользователя или учётной записи.

Тем не менее, время от времени вам может потребоваться дать доступ к локальной учётной записи кому-то, у кого нет соответствующей учётной записи Kerberos. Например, пользователю tillman@EXAMPLE.ORG может потребоваться доступ к локальной учётной записи webdevelopers. Другим учётным записям также может потребоваться доступ к этой

локальной учётной записи.

Файлы .k5login и .k5users, помещенные в домашний каталог пользователя, могут быть использованы подобно действенной комбинации .hosts и .rhosts для решения этой проблемы. Например, файл .k5login со следующим содержанием:

```
tillman@example.org
jdoe@example.org
```

помещен в домашний каталог локального пользователя `webdevelopers`, то обе упомянутые учётные записи получат доступ к этой учётной записи без необходимости наличия общего пароля.

Рекомендуется прочитать страницу справочника по этим командам. Обратите внимание, что страница справочника о `ksu` содержит информацию по .k5users.

13.8.6. Подсказки, советы и решение проблем с Kerberos

- При использовании портов как Heimdal так и MITKerberos убедитесь, что в `PATH` версии Kerberos клиентов указаны перед их версиями в базовой системе.
- Все ли компьютеры в пределах данного realm синхронизированы по времени? Если нет, аутентификация может завершиться неудачно. Синхронизация часов через NTP описывает как синхронизировать часы с использованием NTP.
- MIT и Heimdal успешно взаимодействуют. За исключением `kadmin`, протокол для которого не стандартизован.
- Если вы изменяете hostname, потребуется также изменить учётную запись `host/` и обновить keytab. Это также необходимо для специальных записей в keytab, таких как `www/` запись модуля Apache www/mod_auth_kerb.
- Все хосты под общим идентификатором должны разрешаться DNS (прямое и обратное разрешение), или как минимум через /etc/hosts. Записи CNAME будут работать, но записи A и PTR должны быть корректны и находиться на своем месте. Сообщение об ошибке не всегда интуитивно понятно: `Kerberos5 refuses authentication because Read req failed: Key table entry not found`.
- Некоторые операционные системы, способные работать в качестве клиентов KDC не устанавливают права для `ksu` в setuid `root`. Это означает, что `ksu` не работает, что хорошо является хорошей идеей для безопасности, но неудобно. Это не ошибка KDC.
- С MITKerberos, если вы хотите продлить действие доступа до значения большего, чем десять часов по умолчанию, используйте команду `modify_principal` в `kadmin` для изменения maxlife доступа к самой учётной записи и к учётной записи `krbtgt`. Затем возможно использование `kinit` с параметром `-l` для запроса доступа с большим временем действия. *

Если вы запускаете перехватчик пакетов на KDC для разрешения проблем, а затем запускаете `kinit` с рабочей станции, то увидите, что TGT посылается непосредственно при запуске `kinit` - даже до того, как вы введете пароль! Объяснение в том, что сервер

Kerberos свободно распространяет TGT (Ticket Granting Ticket) на каждый неавторизованный запрос; однако, каждый TGT зашифрован ключом, полученным из пароля пользователя. Следовательно, когда пользователь вводит свой пароль, он не отправляется на KDC, а используется для расшифровка TGT, который уже получен `kinit`. Если в процессе расшифровки получается правильный билет с правильным значением времени, у пользователя есть действующее "удостоверение". Это удостоверение содержит ключ сессии для установления безопасного соединения с сервером Kerberos, как и действующий TGT, зашифрованный ключом сервера Kerberos. Второй уровень шифрования недоступен пользователю, но позволяет серверу Kerberos проверять правильность каждого TGT.

- Если вы хотите установить большое время жизни доступа (например, неделю), и используете OpenSSH для соединения с компьютером, где хранится "билет", убедитесь, что параметр Kerberos`TicketCleanup` установлен в `no` в файле sshd_config, или билеты будут уничтожены при выходе из сеанса.
- Запомните, что время жизни билетов хостов больше. Если время жизни билета для учётной записи пользователя составляет неделю, а время жизни учётной записи хоста, к которому вы подсоединяетесь девять часов, учётная запись хоста в кэше устареет и кэш билетов будет работать не так, как ожидается.
- При настройке файла krb5.dict на предотвращение использования определенных плохих паролей (страница справочника для `kadmind` кратко рассказывает об этом), запомните, что это применимо только к учётным записям, для которых действует политика паролей. Формат файла krb5.dict прост: одно слово на строку. Может помочь создание символической ссылки на /usr/shared/dict/words.

13.8.7. Отличия от порта MIT

Основное различие между установками MIT и Heimdal относится к программе `kadmin`, которая имеет другой (но эквивалентный) набор команд и использует другой протокол. Если ваш KDC работает на MIT, вы не сможете использовать `kadmin` для удаленного администрирования KDC (и наоборот, по этой же причине).

Опции командной строки клиентов также могут немного отличаться для одинаковых задач. Рекомендуется следование инструкциям на MITKerberos Web-сайте (http://web.mit.edu/Kerberos/www/). Будьте внимательны при определении `PATH`: порт MIT устанавливается по умолчанию в /usr/local/, и если в `PATH` вначале указаны системные каталоги, вместо приложений MIT могут быть запущены системные приложения.

С портом MITsecurity/krb5, предоставляемым FreeBSD, убедитесь что файл /usr/local/shared/doc/krb5/README.FreeBSD установлен портом, если вы хотите понять почему вход через `telnetd` и `klogind` иногда происходит так странно. Наиболее важно, исправление "incorrect permissions on cache file" требует использования бинарного файла `login.krb5` для аутентификации, чтобы права на переданное удостоверение передавались правильно.

13.8.8. Преодоление ограничений, обнаруженных в Kerberos

13.8.8.1. Kerberos это все или ничего

Каждый сервис, работающий в сети, должен быть модифицирован для работы с Kerberos (или другим способом защищен от атак по сети) или удостоверения пользователей могут быть украдены или использованы повторно. В качестве примера может быть приведено использование Kerberos версий оболочек для удаленной работы (например через `rsh` и `telnet`), при наличии POP3 сервера, получающего пароли в незашифрованном виде.

13.8.8.2. Kerberos предназначен для однопользовательских рабочих станций

В многопользовательской среде Kerberos менее безопасен. Это потому, что он хранит билеты в каталоге /tmp, которая доступна для чтения всем. Если пользователь работает с несколькими другими пользователями одновременно на одном компьютере (т.е. в многопользовательской среде), возможна кража (копирование) билета другим пользователем.

Решить проблему можно с помощью параметра командной строки `-c` или (предпочтительно) с помощью переменной окружения `KRB5CCNAME`, но это делается редко. Для преодоления ограничения достаточно сохранять билет в домашнем каталоге пользователя и использовать простые ограничения на доступ к файлам.

13.8.8.3. От KDC зависит вся система

Архитектура системы такова, что KDC должен быть максимально защищен, поскольку главный пароль базы данных содержится в нем. На KDC не должно быть запущено никаких других сервисов и он должен быть защищен физически. Опасность велика, поскольку Kerberos хранит все пароли зашифрованными одним ключом ("главным" ключом), который хранится в файле на KDC.

Хорошей новостью является то, что кража главного ключа не станет такой проблемой, как может показаться. Главный ключ используется только для шифрования базы данных Kerberos и в качестве seed для генератора случайных чисел. Поскольку доступ к KDC защищен, атакующий мало что сможет сделать с главным ключом.

Кроме того, если KDC станет недоступен (возможно по причине атак DoS или проблем в сети) сетевые сервисы будет невозможно использовать, поскольку аутентификация не может быть выполнена. Уменьшить последствия можно при наличии нескольких KDC (один главный и один или несколько резервных) и с аккуратно реализованной резервной аутентификацией (отлично подойдет PAM).

13.8.8.4. Недостатки Kerberos

Kerberos позволяет пользователям, хостам и сервисам производить аутентификацию друг друга. В нем нет механизма аутентификации KDC для пользователей, хостов или сервисов. Это означает, что поддельный `kinit` (например) может записывать все имена пользователей и паролей. Помочь решить проблему может security/tripwire или другой инструмент проверки целостности файловой системы.

13.8.9. Ресурсы и информация для дальнейшего изучения

- Kerberos FAQ
- Разработка системы аутентификации: диалог в четырех сценах
- RFC 1510, Kerberos Network Authentication Service (V5)
- Домашняя страница MIT Kerberos
- Домашняя страница Heimdal Kerberos

13.9. OpenSSL

Одной из программ, требующих особого внимания пользователей, является набор программ OpenSSL, включенный в FreeBSD. OpenSSL предоставляет уровень шифрования поверх обычных уровней соединения; следовательно, он может быть использован многими сетевыми приложениями и сервисами.

OpenSSL может использоваться для шифрования соединений почтовых клиентов, транзакций через интернет, например для кредитных карт, и многого другого. Многие порты, такие как www/apache13-ssl и mail/sylpheed-claws собираются с OpenSSL.

> В большинстве случаев в Коллекции Портов будет сделана попытка построения порта security/openssl, если только переменная `WITH_OPENSSL_BASE` не установлена явно в "yes".

Версия OpenSSL, включаемая в FreeBSD, поддерживает сетевые протоколы безопасности Secure Sockets Layer v2/v3 (SSLv2/SSLv3), Transport Layer Security v1 (TLSv1) и может быть использована в качестве основной криптографической библиотеки.

> Хотя OpenSSL поддерживает алгоритм IDEA, по умолчанию он отключен из-за патентных ограничений Соединенных Штатов. Для его использования необходимо ознакомиться с лицензией, и, если ограничения приемлемы, установить в make.conf переменную `MAKE_IDEA`.

Наиболее часто OpenSSL используется для создания сертификатов, используемых программными пакетами. Эти сертификаты подтверждают, что данные компании или частного лица верны и не подделаны. Если рассматриваемый сертификат не был проверен одним из нескольких сертификационных центров ("Certificate Authorities" - CA), обычно выводится предупреждение. Центр сертификации представляет собой компанию, такую, как VeriSign, которая подписывает сертификаты для подтверждения данных частных лиц или компаний. Эта процедура не бесплатна и не является абсолютно необходимой для использования сертификатов; однако может успокоить некоторых особо осторожных пользователей.

13.9.1. Генерирование сертификатов

Для генерирования сертификатов доступна следующая команда:

```
# openssl req -new -nodes -out req.pem -keyout cert.pem
Generating a 1024 bit RSA private key
................++++++
.......................................++++++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SOME PASSWORD
An optional company name []:Another Name
```

Ввод после приглашения "Common Name" содержит имя домена. Здесь вводится имя сервера для верификации; помещение в это поле чего-либо кроме этого имени приведет к созданию бесполезного сертификата. Доступны и другие параметры, например срок действия, альтернативные алгоритмы шифрования и т.д. Полный список находится на странице справочного руководства openssl(1).

В текущем каталоге, из которого была вызвана вышеуказанная команда, должны появиться два файла. Файл req.pem с запросом на сертификацию может быть послан в центр выдачи сертификатов, который проверит введённые вами подтверждающие данные, подпишет запрос и возвратит сертификат вам. Второй созданный файл будет иметь название cert.pem и содержать приватный сертификационный ключ, который необходимо тщательно защищать; если он попадёт в руки посторонних лиц, то может быть использован для имитации лично вас (или вашего сервера).

Когда подпись CA не требуется, может быть создан самоподписанный сертификат. Сначала создайте ключ RSA:

```
# openssl dsaparam -rand -genkey -out myRSA.key 1024
```

Теперь создайте ключ CA:

```
# openssl gendsa -des3 -out myca.key myRSA.key
```

Используйте этот ключ при создании сертификата:

```
# openssl req -new -x509 -days 365 -key myca.key -out new.crt
```

В каталоге должно появиться два новых файла: подпись сертификата, myca.key и сам сертификат, new.crt. Они должны быть помещены в каталог, доступный для чтения только `root`, желательно внутри /etc. Права на каталог можно изменить `chmod` с параметрами 0700.

13.9.2. Использование сертификатов, пример

Итак, что могут сделать эти файлы? Хорошим применением может стать шифрование соединений для SendmailMTA. Это сделает ненужным использование простой текстовой аутентификации для тех, кто отправляет почту через локальный MTA.

Это не лучшее из возможных использований, поскольку некоторые MUA выдадут ошибку, если сертификат не установлен локально. Обратитесь к поставляемой с программой документации за информацией по установке сертификата.

Следующие строки должны быть помещены в локальный файл .mc:

```
dnl SSL Options
define(`confCACERT_PATH',`/etc/certs')dnl
define(`confCACERT',`/etc/certs/new.crt')dnl
define(`confSERVER_CERT',`/etc/certs/new.crt')dnl
define(`confSERVER_KEY',`/etc/certs/myca.key')dnl
define(`confTLS_SRV_OPTIONS', `V')dnl
```

Где /etc/certs/ это каталог для локального хранения сертификата и ключей. После настройки необходимо собрать локальный файл .cf. Это легко сделать, набрав `make`install в каталоге /etc/mail. Затем выполните команду `make`restart, которая должна запустить даемон Sendmail.

Если все пройдет нормально, в файле /var/log/maillog не появятся сообщения об ошибках и запустится процесс Sendmail.

Для проведения простого теста подключитесь к почтовому серверу программой telnet(1):

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.12.10/8.12.10; Tue, 31 Aug 2004 03:41:22 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
```

```
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

Если в выводе появилась строка "STARTTLS", все работает правильно.

13.10. VPN через IPsec

Создание VPN между двумя сетями, соединенными через интернет, с использованием шлюзов FreeBSD.

13.10.1. Принципы работы IPsec

Этот раздел послужит вам руководством по настройке IPsec и его использованию в среде FreeBSD и Microsoft® Windows® 2000/XP, соединяемых безопасным способом. Для настройки IPsec необходимо ознакомиться с процессом сборки ядра (Настройка ядра FreeBSD).

IPsec это протокол, расположенный поверх слоя Internet Protocol (IP). Он позволяет двум или более хостам связываться защищенным способом (отсюда и название протокола). "Сетевой стек" FreeBSD IPsec основан на реализации KAME, поддерживающей оба семейства протоколов, IPv4 и IPv6.

FreeBSD содержит "аппаратно поддерживаемый" стек IPsec, известный как "Fast IPsec", заимствованный из OpenBSD. Для оптимизации производительности IPsec он задействует криптографическое оборудование (когда оно доступно) через подсистему crypto(4). Это новая подсистема и она не поддерживает всех возможностей, доступных в KAME версии IPsec. Для включения IPsec с аппаратной поддержкой необходимо добавить в файл настройки ядра следующий параметр:

```
options   FAST_IPSEC  # new IPsec (cannot define w/ IPSEC)
```

Обратите внимание, что на данный момент невозможно использовать подсистему "Fast IPsec" вместе с KAME реализацией IPsec. Обратитесь к странице справочника fast_ipsec(4) за дальнейшей информацией.

Для того, чтобы применять к туннелям gif(4) межсетевые экраны, вам

потребуется включить в ядро опцию IPSEC_FILTERGIF:

```
options   IPSEC_FILTERGIF  #filter ipsec packets from a tunnel
```

IPsec состоит из двух подпротоколов:

- *Encapsulated Security Payload (ESP)*, защищающей данные IP пакета от вмешательства третьей стороны путем шифрования содержимого с помощью симметричных криптографических алгоритмов (таких как Blowfish,3DES).
- *Authentication Header (AH)*, защищающий заголовок IP пакета от вмешательства третьей стороны и подделки путем вычисления криптографической контрольной суммы и хеширования полей заголовка IP пакета защищенной функцией хеширования. К пакету добавляется дополнительный заголовок с хэшем, позволяющий аутентификацию информации пакета.

ESP и AH могут быть использованы вместе или по отдельности, в зависимости от обстоятельств.

IPsec может быть использован или для непосредственного шифрования трафика между двумя хостами (*транспортный режим*); или для построения "виртуальных туннелей" между двумя подсетями, которые могут быть использованы для защиты соединений между двумя корпоративными сетями (*туннельный режим*). Последний обычно называют *виртуальной частной сетью* (Virtual Private Network, VPN). За детальной информацией о подсистеме IPsec в FreeBSD обратитесь к странице справочника ipsec(4).

Для включения поддержки IPsec в ядре, добавьте следующие параметры к файлу настройки ядра:

```
options   IPSEC        #IP security
options   IPSEC_ESP    #IP security (crypto; define w/ IPSEC)
```

Если желательна поддержка отладки IPsec, должна быть также добавлена следующая строка:

```
options   IPSEC_DEBUG  debug for IP security
```

13.10.2. Проблема

Не существует стандарта VPN. Они могут быть реализованы множеством различных технологий, каждая из которых имеет свои сильные и слабые стороны. Этот раздел представляет сценарий и стратегию реализации VPN для этого сценария.

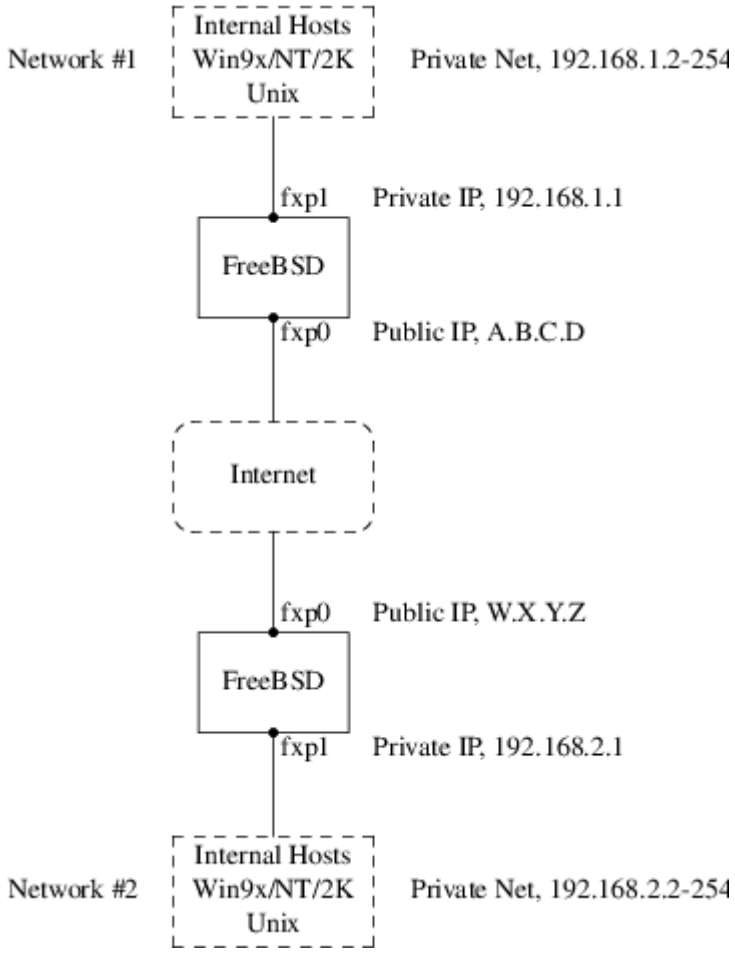
13.10.3. Сценарий: Две сети, подключенных к интернет, работающие как одна

Исходные условия таковы:

- Существует как минимум две сети
- Внутри обеих сетей используется IP
- Обе сети соединены через интернет через шлюз, работающий на FreeBSD.
- У шлюза каждой из сетей есть как минимум один публичный IP адрес.
- Внутренние IP адреса двух сетей могут быть публичными или приватными, не имеет значения. На шлюзе может работать NAT, если это необходимо.
- Внутренние IP адреса двух сетей *не должны пересекаться*. Хотя вероятно теоретически возможно использование комбинации VPN технологии и NAT для настройки такой конфигурации, эта конфигурация будет кошмарна.

Если две сети, которые вы пытаетесь соединить, используют один и тот же диапазон приватных адресов (например, обе используют `192.168.1.x`), номера в одной из сетей необходимо изменить.

Топология сети может выглядеть примерно так:



Заметьте, что здесь присутствуют два публичных IP-адреса. В дальнейшем для их обозначения будут использоваться буквы. Если вы увидите эти буквы, замените их на свои публичные IP адреса. Также обратите внимание, что у обеих шлюзов внутренний адрес

заканчивается на .1 и диапазоны приватных адресов двух сетей различны (192.168.1.x и 192.168.2.x соответственно). Все компьютеры локальных сетей настроены на использование в качестве шлюза по умолчанию компьютера с адресом, оканчивающимся на .1.

С сетевой точки зрения замысел в том, чтобы каждая сеть видела компьютеры из другой сети так, как если бы они были непосредственно подключены к тому же самому маршрутизатору - хотя и немного медленному маршрутизатору, иногда теряющему пакеты.

Это означает, что (например) компьютер 192.168.1.20 может запустить

```
ping 192.168.2.34
```

и это будет прозрачно работать. Компьютеры с Windows® должны видеть компьютеры в другой сети, просматривать сетевые ресурсы, и так далее, точно так же, как и для компьютеров в локальной сети.

И все это безопасным способом. Это означает, что трафик между сетями зашифрован.

Создание VPN между этими двумя сетями это многошаговый процесс. Этапы создания VPN таковы:

1. Создание "виртуального" сетевого подключения между двумя сетями через интернет. Тестирование подключения с помощью таких инструментов как ping(8), чтобы убедиться, что оно работает.
2. Применение политики безопасности чтобы убедиться, что трафик между двумя сетями прозрачно шифруется и расшифровывается если необходимо. Тестирование с помощью таких инструментов как tcpdump(1), чтобы убедиться, что трафик шифруется.
3. Настройка дополнительных программ на шлюзах FreeBSD, чтобы компьютеры Windows® из одной сети видели компьютеры в другой через VPN.

13.10.3.1. Шаг 1: Создание и тестирование "виртуального" сетевого подключения

Предположим, что вы работаете на шлюзе сети #1 (с публичным адресом A.B.C.D, приватным адресом 192.168.1.1) и запускаете ping 192.168.2.1, т.е. на приватный адрес машины с IP адресом W.X.Y.Z. Что должно произойти, чтобы это сработало?

1. Шлюз должен знать, как достичь 192.168.2.1. Другими словами, у него должен быть маршрут к 192.168.2.1.
2. Приватные IP адреса, такие как диапазон 192.168.x не адресуются в интернет. Каждый пакет, отправляемый на 192.168.2.1 должен быть "завернут" в другой пакет. Исходным адресом пакета должен быть A.B.C.D, а адресом назначения W.X.Y.Z. Этот процесс называется *инкапсуляцией*.
3. Как только этот пакет достигнет W.X.Y.Z, необходимо будет "декапсулировать" его и доставить к 192.168.2.1.

Как вы можете увидеть, это требует "туннеля" между двумя сетями. Два конца "туннеля" это IP адреса A.B.C.D и W.X.Y.Z. Туннель используется для передачи трафика с приватными IP

адресами через интернет.

В FreeBSD этот туннель создается с помощью устройства generic interface, или gif. Как вы можете догадаться, интерфейс gif на каждом хосте должен быть настроен с четырьмя IP адресами; два для публичных IP адресов и два для приватных IP адресов.

В ядро обеих компьютеров FreeBSD должна быть встроена поддержка устройства gif. Вы можете сделать это, добавив строку:

```
device gif
```

к файлу настройки ядра на обеих компьютерах, с последующей компиляцией, установкой и перезагрузкой.

Настройка туннеля это двухшаговый процесс. Во-первых, необходимо задать сведения о внешнем (или публичном) IP адресе с помощью ifconfig(8). Затем о приватном IP адресе, также с помощью ifconfig(8).

На шлюзе сети #1 для настройки туннеля вам потребуется запустить следующие две команды.

```
ifconfig gif0 A.B.C.D W.X.Y.Z
ifconfig gif0 inet 192.168.1.1 192.168.2.1 netmask 0xffffffff
```

На другом шлюзе подобные команды, но с IP адресами в обратном порядке.

```
ifconfig gif0 W.X.Y.Z A.B.C.D
ifconfig gif0 inet 192.168.2.1 192.168.1.1 netmask 0xffffffff
```

Затем вы можете запустить:

```
ifconfig gif0
```

для просмотра настройки. Например, на шлюзе сети #1 вы увидите:

```
# ifconfig gif0
gif0: flags=8011<UP,POINTTOPOINT,MULTICAST> mtu 1280
inet 192.168.1.1 --> 192.168.2.1 netmask 0xffffffff
physical address inet A.B.C.D --> W.X.Y.Z
```

Как вы можете видеть, был создан туннель между физическими адресами `A.B.C.D` и `W.X.Y.Z`, для туннелирования разрешен трафик между `192.168.1.1` и `192.168.2.1`.

Это также добавляет запись к таблице маршрутизации на обеих машинах, вы можете проверить запись командой `netstat -rn`. Вот вывод этой команды на шлюзе сети #1.

```
# netstat -rn
Routing tables

Internet:
Destination      Gateway       Flags    Refs    Use    Netif  Expire
...
192.168.2.1      192.168.1.1   UH        0        0    gif0
...
```

Как показывает значение поля "Flags", это маршрут к хосту, что означает, что каждый шлюз знает, как достичь другого шлюза, но не знает как достичь остальной части соответствующей сети. Эта проблема будет быстро решена.

Вероятно, на обеих машинах запущен брандмауэр. VPN должен обходить его. Вы можете разрешить весь трафик между двумя сетями, или включить правила, защищающие каждый конец соединения от другого.

Это сильно упрощает тестирование настройки брандмауэра, если вы разрешаете весь трафик через VPN. Вы всегда можете усилить защиту позже. Если вы используете на шлюзах ipfw(8), команда вроде этой

```
ipfw add 1 allow ip from any to any via gif0
```

разрешит весь трафик между двумя концами VPN без влияния на другие правила брандмауэра. Очевидно, вам потребуется запустить эту команду на обеих шлюзах.

Этого достаточно для включения ping с одного шлюза на другой. На `192.168.1.1`, вы сможете запустить

```
ping 192.168.2.1
```

и получить ответ, и аналогично на другом шлюзе.

Однако, машины в другой сети пока недоступны. Это из-за маршрутизации - хотя шлюзы знают, как связаться друг с другом, они не знают, как связаться с сетью за другим шлюзом.

Для решения этой проблемы вы должны добавить статический маршрут на каждом шлюзе. Команда на первом шлюзе будет выглядеть так:

```
route add 192.168.2.0 192.168.2.1 netmask 0xffffff00
```

Она говорит "Для достижения хостов в сети `192.168.2.0`, отправляйте пакеты хосту `192.168.2.1`". Вам потребуется запустить похожую команду на другом шлюзе, но с адресами `192.168.1.x`.

IP трафик с хостов в одной сети теперь может достичь хосты в другой сети.

Теперь создано две трети VPN между двумя сетями, поскольку это "виртуальная (virtual)""сеть (network)". Она еще не приватная (private). Вы можете протестировать ее с помощью ping(8) и tcpdump(1). Войдите на шлюз и запустите

```
tcpdump dst host 192.168.2.1
```

В другой сессии на этом же хосте запустите

```
ping 192.168.2.1
```

Вы увидите примерно такие строки:

```
16:10:24.018080 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:24.018109 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:25.018814 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:25.018847 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:26.028896 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:26.029112 192.168.1.1 > 192.168.2.1: icmp: echo reply
```

Как вы видите, ICMP сообщения пересылаются вперед и назад незашифрованными. Если вы использовали с tcpdump(1) параметр `-s` для получения большего объема данных пакета, то увидите больше информации.

Конечно же это неприемлемо. В следующем разделе мы обсудим защиту соединения между двумя сетями, так что весь трафик будет автоматически шифроваться.

Резюме:

- Настройте оба ядра с "device gif".
- Отредактируйте /etc/rc.conf на шлюзе #1 и добавьте следующие строки (подставляя IP адреса где необходимо).

  ```
  gifconfig_gif0="A.B.C.D W.X.Y.Z"
  ifconfig_gif0="inet 192.168.1.1 192.168.2.1 netmask 0xffffffff"
  static_routes="vpn"
  route_vpn="192.168.2.0 192.168.2.1 netmask 0xffffff00"
  ```

- Отредактируйте скрипт брандмауэра (/etc/rc.firewall, или подобный) на обеих хостах и добавьте

  ```
  ipfw add 1 allow ip from any to any via gif0
  ```

- Выполните соответствующие изменения в /etc/rc.conf на шлюзе #2, меняя порядок IP адресов.

13.10.3.2. Шаг 2: Защита соединения

Для защиты соединения мы будем использовать IPsec. IPsec предоставляет хостам механизм определения ключа для шифрования и для последующего использования этого ключа для шифрования данных между двумя хостами.

Здесь будут рассмотрены два аспекта настройки.

1. У хостов должен быть способ согласования используемого алгоритма шифрования. Как только хосты договорятся об этом, можно говорить об установленном между ними "безопасном соединении".
2. Должен быть механизм определения, какой трафик необходимо шифровать. Конечно, вам не требуется шифровать весь исходящий трафик - достаточно шифровать только трафик, идущий через VPN. Правила, определяющие то, какой трафик необходимо шифровать, называются "политикой безопасности".

Безопасное соединение и политика безопасности поддерживаются ядром, и могут быть изменены программами пользователя. Однако перед тем, как вы сможете сделать это, необходимо настроить поддержку протоколов IPsec и Encapsulated Security Payload (ESP) в ядре. Это делается добавлением в настройку ядра параметров:

```
options IPSEC
options IPSEC_ESP
```

с последующим перекомпилированием, переустановкой и перезагрузкой. Как и прежде вам потребуется сделать это с ядрами на обеих шлюзах.

При настройке параметров безопасности (security associations) у вас есть два варианта. Вы можете настроить их вручную для обеих хостов, задав алгоритм шифрования, ключи для шифрования и так далее, или использовать даемоны, реализующие Internet Key Exchange protocol (IKE), который сделает это за вас.

Рекомендуется последнее. Помимо прочего, этот способ более прост.

Редактирование и отображение политики безопасности выполняется с помощью setkey(8). По аналогии, `setkey` используется для настройки таблиц политики безопасности ядра так же, как route(8) используется для настройки таблиц маршрутизации ядра. `setkey` также может отображать текущие параметры безопасности, и продолжая аналогию дальше, это соответствует `netstat -r`.

Существует множество даемонов для управления параметрами безопасности в FreeBSD. Здесь будет описано использование одного из них, racoon - он доступен в составе порта security/ipsec-tools в Коллекции Портов FreeBSD.

Даемон racoon должен работать на обеих шлюзах. На каждом из хостов он настраивается с IP адресом другого конца VPN, и секретным ключом (по вашему выбору, должен быть одним и тем же на обеих шлюзах).

Эти два даемона подключаются друг к другу, подтверждают, что они именно те, за кого себя

выдают (используя секретный ключ, заданный вами). Затем даемоны генерируют новый секретный ключ и используют его для шифрования трафика через VPN. Они периодически изменяют этот ключ, так что даже если атакующий сломает один из ключей (что теоретически почти невозможно) это не даст ему слишком много - он сломал ключ, который два даемона уже сменили на другой.

Настройки racoon сохраняются в файле ${PREFIX}/etc/racoon. Этот файл не требует слишком больших изменений. Другим компонентом настройки racoon, который потребуется изменить, является "предварительный ключ".

В настройке по умолчанию racoon ищет его в файле ${PREFIX}/etc/racoon/psk.txt. Необходимо отметить, что предварительный ключ *не* используется для шифрования трафика через VPN соединение это просто маркер, позволяющий управляющим ключами даемонам доверять друг другу.

psk.txt содержит строку для каждого удаленного сервера, с которым происходит соединение. В этом примере два сервера, каждый файл psk.txt будет содержать одну строку (каждый конец VPN общается только с другим концом.

На шлюзе #1 эта строка будет выглядеть примерно так:

```
W.X.Y.Z            secret
```

То есть *публичный* IP-адрес противоположной стороны, пробел и текстовая строка с секретной фразой. Конечно, вам не стоит использовать в качестве ключевой фразы слово "secret" — здесь применяются обычные правила выбора паролей.

На шлюзе #2 строка будет выглядеть примерно так:

```
A.B.C.D            secret
```

То есть публичный IP адрес удаленной стороны и та же секретная фраза. Перед запуском racoon режим доступа к файлу psk.txt должен быть установлен в `0600` (т.е. запись и чтение только для `root`).

Вы должны запустить racoon на обоих шлюзах. Вам также потребуется добавить правила для включения IKE трафика, передающегося по UDP через порт ISAKMP (Internet Security Association Key Management Protocol). Опять же, они должны быть расположены насколько возможно ближе к началу набора правил.

```
ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
```

Как только racoon будет запущен, вы можете попробовать выполнить ping с одного шлюза на другой. Соединение все еще не зашифровано, но racoon установит параметры безопасности между двумя хостами - это может занять время и вы можете заметить небольшую задержку перед началом ответа команды ping.

Как только параметры безопасности установлены, вы можете просмотреть их используя setkey(8). Запустите

```
setkey -D
```

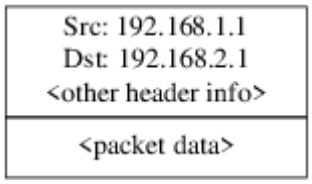
на любом из хостов для просмотра информации о параметрах безопасности.

Это одна сторона проблемы. Другая сторона это настройка политики безопасности.

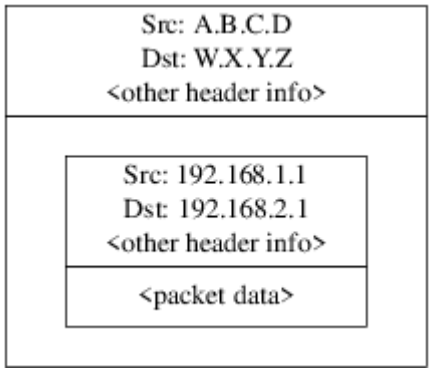
Для создания разумной политики безопасности давайте вспомним, что уже было настроено. Это рассмотрение относится к обеим концам соединения.

Каждый отправляемый IP пакет имеет заголовок, содержащий информацию о пакете. Заголовок включает IP адреса источника и назначения. Как мы уже знаем, приватные IP адреса, такие как `192.168.x.y`, не могут появиться в интернет. Они должны быть сначала включены внутрь другого пакета. В этом пакете приватные IP адреса источника и назначения заменяются публичными IP адресами.

То есть исходящий пакет, который выглядит примерно так:



будет инкапсулирован в другой пакет, выглядящий примерно так:



Этой инкапсуляцией занимается устройство gif. Как вы можете видеть, теперь у пакета есть реальный IP адрес, исходный пакет был включен в этот пакет в виде данных, которые передаются через интернет.

Конечно, мы хотим зашифровать весь трафик между VPN. Вы можете сформулировать это на словах так:

"Если пакет отправляется с `A.B.C.D`, и предназначен для `W.X.Y.Z`, расшифровать его, используя необходимые параметры безопасности."

"Если пакет отправляется с `W.X.Y.Z`, и предназначен для `A.B.C.D`, расшифровать его, используя необходимые параметры безопасности."

Это похоже на желаемое, но не совсем то. Если вы сделаете это, весь трафик от и к `W.X.Y.Z`, даже если он не является частью VPN, будет зашифрован. Правильная политика такова:

"Если пакет отправляется с `A.B.C.D`, в нем инкапсулирован другой пакет и адрес назначения `W.X.Y.Z`, зашифровать его, используя необходимые параметры безопасности."

"Если пакет отправляется с `W.X.Y.Z`, в нем инкапсулирован другой пакет и адрес назначения `A.B.C.D`, зашифровать его, используя необходимые параметры безопасности."

Тонкое, но необходимое различие.

Политика безопасности также устанавливается с использованием setkey(8). В setkey(8) предусмотрен язык определения политики setkey(8). Вы можете или ввести инструкции по настройке со стандартного ввода, или использовать параметр `-f` для задания файла, содержащего эти инструкции.

Настройка на шлюзе #1 (где есть публичный IP адрес `A.B.C.D`) для включения шифрования всего предназначенного `W.X.Y.Z` трафика:

```
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Поместите эти команды в файл (например, /etc/ipsec.conf) и запустите

```
# setkey -f /etc/ipsec.conf
```

`spdadd` указывает setkey(8) добавить правило к базе данных политики безопасности. Остальная часть строки указывает какие пакеты будут соответствовать политике. `A.B.C.D/32` и `W.X.Y.Z/32` это IP адреса и сетевые маски, определяющие сети или хосты, к которым будет применяться данная политика. В данном случае мы хотим применить их к трафику между этими двумя хостами. Параметр `ipencap` сообщает ядру, что эта политика должна применяться только к пакетам, инкапсулирующим другие пакеты. Параметр `-P out` сообщает, что эта политика применяется к исходящим пакетам, и `ipsec` - то, что пакеты будут зашифрованы.

Оставшаяся часть строки определяет, как эти пакеты будут зашифрованы. Будет использоваться протокол `esp`, а параметр `tunnel` показывает, что пакет в дальнейшем будет инкапсулирован в IPsec пакет. Повторное использование `A.B.C.D` и `W.X.Y.Z` предназначено для выбора используемых параметров безопасности, и наконец параметр `require` разрешает шифрование пакетов, попадающих под это правило.

Это правило соответствует только исходящим пакетам. Вам потребуется похожее правило, соответствующее входящим пакетам.

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec esp/tunnel/W.X.Y.Z-A.B.C.D/require;
```

Обратите внимание, что вместо `in` используется `out` и IP адреса переставлены.

Другому шлюзу (с публичным IP адресом `W.X.Y.Z`) потребуются похожие правила.

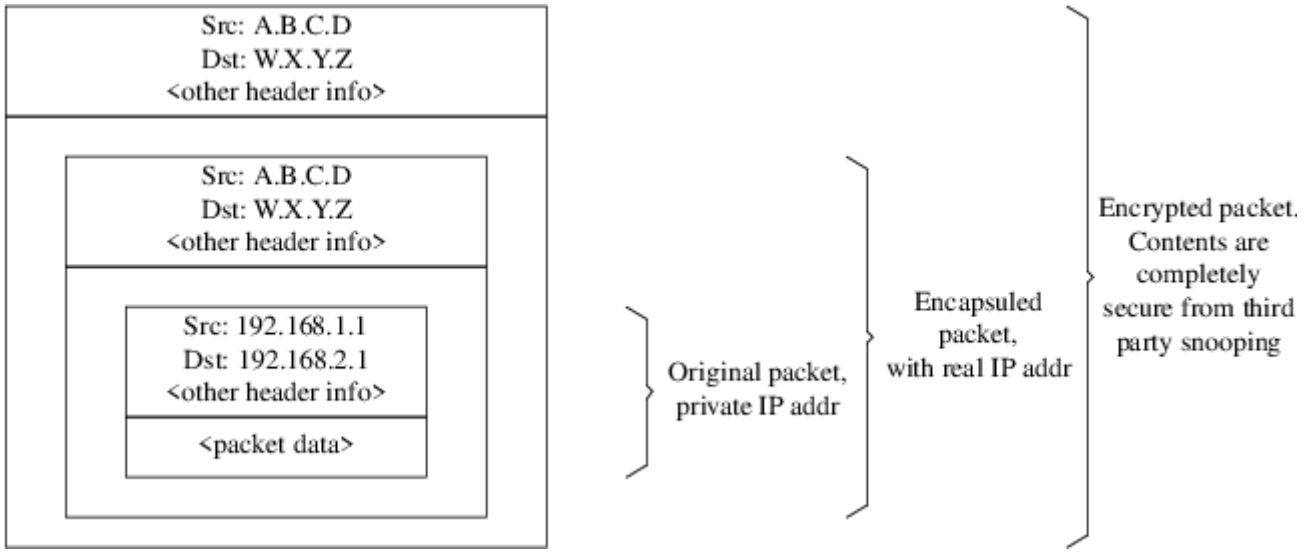
```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec esp/tunnel/W.X.Y.Z-A.B.C.D/require;
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Наконец, вам потребуется добавить правила к брандмауэру для включения прохождения пакетов ESP и IPENCAP в обе стороны. На обеих хостах потребуется добавить следующие правила:

```
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

Поскольку правила симметричны, можно использовать их без изменения на обеих хостах

Исходящие пакеты теперь будут выглядеть примерно так:



Когда эти пакеты будут получены на удаленном конце VPN соединения, они будут расшифрованы (используя параметры безопасности, о которых договорился racoon). Затем они будут переданы интерфейсу gif, который "развернет" второй слой, оставив пакет с внутренними адресами, который сможет попасть во внутреннюю сеть.

Вы можете проверить безопасность тем же ping(8), который использовался ранее. Сначала войдите на шлюз `A.B.C.D` и запустите:

```
tcpdump dst host 192.168.2.1
```

В другой сессии на том же хосте запустите

```
ping 192.168.2.1
```

В этот момент вы должны увидеть примерно это:

```
XXX tcpdump output
```

Теперь, как видите, tcpdump(1) показывает ESP пакеты. Если вы попытаетесь просмотреть их с параметром `-s`, то вероятно увидите нечто непонятное, поскольку применяется шифрование.

Поздравляем. Вы только что настроили VPN между двумя удаленными сетями.

Резюме

- Настройте оба ядра с:

  ```
  options IPSEC
  options IPSEC_ESP
  ```

- Установите security/ipsec-tools. Отредактируйте ${PREFIX}/etc/racoon/psk.txt на обеих шлюзах, добавив запись для каждого IP адреса удаленного хоста и секретный ключ, который будет известен им обеим. Убедитесь, что режим доступа к файлу 0600.
- Добавьте к /etc/rc.conf на каждом хосте следующие строки:

  ```
  ipsec_enable="YES"
  ipsec_file="/etc/ipsec.conf"
  ```

- Создайте /etc/ipsec.conf на каждом хосте с необходимыми строками spdadd. На шлюзе #1 он будет таким:

  ```
  spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec
    esp/tunnel/A.B.C.D-W.X.Y.Z/require;
  spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec
    esp/tunnel/W.X.Y.Z-A.B.C.D/require;
  ```

  А на шлюзе #2 таким:

  ```
  spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec
    esp/tunnel/W.X.Y.Z-A.B.C.D/require;
  spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec
    esp/tunnel/A.B.C.D-W.X.Y.Z/require;
  ```

- Добавьте правила к брандмауэрам обеих хостов для включения IKE, ESP и IPENCAP трафика:

  ```
  ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
  ```

```
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

Двух приведенных шагов должно быть достаточно для настройки и включения VPN. Машины в каждой сети смогут обращаться друг к другу по IP адресам, и весь трафик через соединение будет автоматически надежно зашифрован.

13.11. OpenSSH

OpenSSH это набор сетевых инструментов, используемых для защищенного доступа к удаленным компьютерам. Он может быть использован в качестве непосредственной замены `rlogin`, `rsh`, `rcp` и `telnet`. Кроме того, через SSH могут быть безопасно туннелированы и/или перенаправлены произвольные TCP/IP соединения. OpenSSH шифрует весь трафик, эффективно предотвращая кражу данных, перехват соединения и другие сетевые атаки.

OpenSSH поддерживается проектом OpenBSD, он основан на SSH v1.2.12 со всеми последними исправлениями и обновлениями, совместим с протоколами SSH версий 1 и 2.

13.11.1. Преимущества использования OpenSSH

Обычно при использовании telnet(1) или rlogin(1) данные пересылаются по сети в незашифрованной форме. Перехватчик пакетов в любой точке сети между клиентом и сервером может похитить информацию о пользователе/пароле или данные, передаваемые через соединение. Для предотвращения этого OpenSSH предлагает различные методы шифрования.

13.11.2. Включение sshd

В FreeBSD даемон sshd должен быть разрешен в процессе инсталляции. За запуск ответственна следующая строка в файле rc.conf:

```
sshd_enable="YES"
```

При следующей загрузке системы будет запущен sshd(8), даемон для OpenSSH. Вы можете также воспользоваться скриптом /etc/rc.d/sshd системы rc(8) для запуска OpenSSH:

```
/etc/rc.d/sshd start
```

13.11.3. SSH клиент

Утилита ssh(1) работает подобно rlogin(1).

```
# ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *******
```

Вход продолжится так же, как если бы сессия была инициирована с использованием `rlogin` или `telnet`. SSH использует систему опознавательных ключей для проверки подлинности сервера при подключении клиента. Пользователю предлагается `yes` только при первом подключении. Дальнейшие попытки входа предваряются проверкой сохраненного ключа сервера. SSH клиент сообщит вам, если сохраненный ключ будет отличаться от только что полученного. Ключи серверов сохраняются в ~/.ssh/known_hosts, или в ~/.ssh/known_hosts2 для SSH v2.

По умолчанию современные серверы OpenSSH настроены на приём только соединений SSH v2. Клиент будет использовать версию 2 там, где это возможно, а затем версию 1. Также, клиент можно заставить использовать конкретную версию при помощи опций `-1` и `-2` для указания соответствующей версии протокола. Версия 1 поддерживается ради совместимости со старыми серверами.

13.11.4. Безопасное копирование

Команда scp(1) работает подобно rcp(1); она копирует файл с удаленного компьютера, но делает это безопасным способом.

```
#  scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password: *******
COPYRIGHT            100% |****************************|  4735       00:00
#
```

Поскольку в предыдущем примере ключ сервера уже был сохранен, в этом примере он проверяется при использовании scp(1).

Параметры, передаваемые scp(1), похожи на параметры cp(1), с файлом или файлами в качестве первого аргумента и приемником копирования во втором. Поскольку файлы передаются по сети через SSH, один или более аргументов принимают форму `user@host:<path_to_remote_file>`.

13.11.5. Настройка

Системные файлы настройки для даемона и клиента OpenSSH расположены в каталоге /etc/ssh.

Файл ssh_config используется для настройки клиента, а sshd_config для даемона.

Кроме того, параметры `sshd_program` (по умолчанию /usr/sbin/sshd), и `sshd_flags`rc.conf дают дополнительные возможности настройки.

13.11.6. ssh-keygen

Вместо использования паролей, с помощью ssh-keygen(1) можно создать ключи DSA или RSA, которыми пользователи могут аутентифицироваться:

```
% ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_dsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_dsa.
Your public key has been saved in /home/user/.ssh/id_dsa.pub.
The key fingerprint is:
bb:48:db:f2:93:57:80:b6:aa:bc:f5:d5:ba:8f:79:17 user@host.example.com
```

ssh-keygen(1) создаст пару публичного и приватного ключей, используемых для аутентификации. Приватный ключ сохраняется в ~/.ssh/id_dsa или ~/.ssh/id_rsa, а публичный в ~/.ssh/id_dsa.pub или ~/.ssh/id_rsa.pub (для ключей DSA и RSA соответственно). Для включения аутентификации по ключам публичный ключ должен быть помещен в файл ~/.ssh/authorized_keys на удаленном компьютере.

Это позволяет соединяться с удаленным компьютером с помощью SSH-ключей вместо паролей.

Если при генерации ключей был использован пароль, каждый раз при использовании приватного ключа он будет запрашиваться у пользователя. Для того, чтобы избежать непрерывного набора кодовой фразы, можно использовать утилиту ssh-agent(1), как описано в разделе Утилиты ssh-agent и ssh-add ниже.

Параметры и имена файлов могут различаться для разных версий OpenSSH, установленных в системе, для решения проблем обратитесь к странице справочника ssh-keygen(1).

13.11.7. Утилиты ssh-agent и ssh-add

Утилиты ssh-agent(1) и ssh-add(1) позволяют сохранять ключи SSH в памяти, чтобы не набирать кодовые фразы при каждом использовании ключа.

Утилита ssh-agent(1) обеспечивает процесс аутентификации загруженными в нее секретными ключами; для этого утилита ssh-agent(1) должна запустить внешний процесс. В самом простом случае это может быть шелл-процесс; в чуть более продвинутом - оконный менеджер.

Для использования ssh-agent(1) совместно с шеллом, ssh-agent(1) должен быть запущен с именем этого шелла в качестве аргумента. После этого в его память при помощи утилиты ssh-add(1) могут быть добавлены необходимые ключи; при этом будут запрошены соответствующие кодовые фразы. Добавленные ключи могут затем использоваться для

ssh(1) на машины, на которых установлены соответствующие публичные ключи:

```
% ssh-agent csh
% ssh-add
Enter passphrase for /home/user/.ssh/id_dsa:
Identity added: /home/user/.ssh/id_dsa (/home/user/.ssh/id_dsa)
%
```

Для того чтобы использовать ssh-agent(1) в X11, вызов ssh-agent(1)должен быть помещен в файл ~/.xinitrc. Это обеспечит поддержкой ssh-agent(1) все программы, запущенные в X11. Файл ~/.xinitrc может выглядеть, например, так:

```
exec ssh-agent startxfce4
```

При этом будет запущен ssh-agent(1), который, в свою очередь, вызовет запуск XFCE, при каждом старте X11. После запуска X11, выполните команду ssh-add(1) для добавления ваших SSH-ключей.

13.11.8. Туннелирование SSH

OpenSSH поддерживает возможность создания туннеля для пропуска соединения по другому протоколу через защищенную сессию.

Следующая команда указывает ssh(1) создать туннель для telnet:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

Команда `ssh` используется со следующими параметрами:

`-2`

Указывает `ssh` использовать версию 2 протокола (не используйте этот параметр, если работаете со старыми SSH серверами).

`-N`

Означает использование в не-командном режиме, только для туннелирования. Если этот параметр опущен, `ssh` запустит обычную сессию.

`-f`

Указывает `ssh` запускаться в фоновом режиме.

`-L`

Означает локальный туннель в стиле *localport:remotehost:remoteport*.

`user@foo.example.com`

Удаленный сервер SSH.

Туннель SSH создается путем создания прослушивающего сокета на определенном порту `localhost`. Затем все принятые на локальном хосту/порту соединения переправляются через SSH на определенный удаленный хост и порт.

В этом примере, порт *5023* на `localhost` перенаправляется на порт *23* на `localhost` удаленного компьютера. Поскольку *23* это порт telnet, будет создано защищенное соединение telnet через туннель SSH.

Этот метод можно использовать для любого числа небезопасных протоколов, таких как SMTP, POP3, FTP, и так далее.

Пример 22. Использование SSH для создания защищенного туннеля на SMTP

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTP
```

Этот метод можно использовать вместе с ssh-keygen(1) и дополнительными пользовательскими учётными записями для создания более удобного автоматического SSH туннелирования. Ключи могут быть использованы вместо паролей, и туннели могут запускаться от отдельных пользователей.

13.11.8.1. Практические примеры SSH туннелирования

13.11.8.1.1. Защищенный доступ к серверу POP3

На работе находится SSH сервер, принимающий соединения снаружи. В этой же офисной сети находится почтовый сервер, поддерживающий протокол POP3. Сеть или сетевое соединение между вашим домом и офисом могут быть или не быть полностью доверяемыми. По этой причине вам потребуется проверять почту через защищенное соединение. Решение состоит в создании SSH соединения к офисному серверу SSH и туннелирование через него к почтовому серверу.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: ******
```

Когда туннель включен и работает, вы можете настроить почтовый клиент для отправки запросов POP3 на `localhost`, порт 2110. Соединение будет безопасно переправлено через туннель на `mail.example.com`.

13.11.8.1.2. Прохождение через Драконовский Брандмауэр

Некоторые сетевые администраторы устанавливают на брандмауэрах драконовские

правила, фильтруя не только входящие соединения, но и исходящие. Вам может быть разрешен доступ к удаленным компьютерам только по портам 22 и 80, для SSH и просмотра сайтов.

Вам может потребоваться доступ к другому (возможно, не относящемуся к работе) сервису, такому как Ogg Vorbis для прослушивания музыки. Если этот сервер Ogg Vorbis выдает поток не с портов 22 или 80, вы не сможете получить к нему доступ.

Решение состоит в создании SSH соединения с компьютером вне брандмауэра и использование его для туннелирования сервера Ogg Vorbis.

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org's password: *******
```

Клиентскую программу теперь можно настроить на `localhost` порт 8888, который будет перенаправлен на `music.example.com` порт 8000, успешно обойдя брандмауэр.

13.11.9. Параметр ограничения пользователей `AllowUsers`

Зачастую хорошие результаты даёт ограничение того, какие именно пользователи и откуда могут регистрироваться в системе. Задание параметра `AllowUsers` является хорошим способом добиться этого. К примеру, для разрешения регистрации только пользователю `root` с машины `192.168.1.32`, в файле /etc/ssh/sshd_config нужно указать нечто вроде следующего:

```
AllowUsers root@192.168.1.32
```

Для разрешения регистрации пользователя `admin` из любой точки, просто укажите имя пользователя:

```
AllowUsers admin
```

Несколько пользователей должны перечислять в одной строке, как здесь:

```
AllowUsers root@192.168.1.32 admin
```

Важно, чтобы бы перечислили всех пользователей, которые должны регистрироваться на этой машине; в противном случае они будут заблокированы.

После внесения изменений в /etc/ssh/sshd_config вы должны указать sshd(8) на повторную загрузку конфигурационных файлов, выполнив следующую команду:

```
# /etc/rc.d/sshd reload
```

13.11.10. Дополнительная литература

OpenSSH

ssh(1) scp(1) ssh-keygen(1) ssh-agent(1) ssh-add(1) ssh_config(5)

sshd(8) sftp-server(8) sshd_config(5)

13.12. Списки контроля доступа файловой системы (ACL)

В дополнение к другим расширениям файловой системы, таким как снимки (snapshots), FreeBSD 5.0 и более поздние версии системы предлагают защиту с помощью списков контроля доступа файловой системы (File System Access Control Lists, ACLs).

Списки контроля доступа расширяют стандартную модель прав UNIX® высоко совместимым (POSIX®.1e) способом. Эта возможность позволяет администратору получить преимущество от использования более интеллектуальной модели безопасности.

Для включения поддержки ACL в файловой системе UFS, следующая строка:

```
options UFS_ACL
```

должна быть добавлена в файл настройки ядра. Если параметр не добавлен, при попытке монтирования систем, поддерживающих ACL, появится предупреждающее сообщение. Этот параметр включен в ядро GENERIC. ACL основывается на дополнительных атрибутах, встроенных в файловую систему. Дополнительные атрибуты поддерживаются по умолчанию следующим поколением файловых систем UNIX®, UFS2.

> Для включения дополнительных атрибутов в UFS1 требуется больше усилий по сравнению с UFS2. Производительность дополнительных атрибутов в UFS2 также существенно выше. По этим причинам для работы с списками контроля доступа предпочтительно использование UFS2

ACL включаются во время монтирования флагом `acls`, который добавляется к /etc/fstab. Этот флаг также можно сделать постоянным с помощью tunefs(8), изменив флаг ACL в заголовке файловой системы. Вообще говоря, использование флага в суперблоке предпочтительно по нескольким причинам:

- Постоянный ACL флаг не может быть изменен путем перемонтирования системы (mount(8) `-u`), а только через umount(8) и mount(8). Это означает, что ACL нельзя включить на корневой файловой системе после загрузки. Это также означает, что вы не можете изменить флаг на используемой файловой системе.
- Установка флага в суперблоке приводит к постоянному монтированию файловой системы с включенным ACL, даже если нет записи в fstab или при смене порядка устройств. Это предотвращает случайное монтирование файловой системы без ACL, которое может повлечь за собой проблемы с безопасностью.

Мы можем изменить поведение ACL для включения флага без полного перемонтирования, но считаем, что желательно исключить случайное монтирование без ACL, поскольку вы можете попасть в неприятную ситуацию, если включите ACL, затем выключите их, затем опять включите без сброса расширенных атрибутов. Обычно, как только вы включили ACL в файловой системе, они не должны быть выключены, поскольку получающаяся защита файлов может быть не совместима с той, что применяется пользователями системы, и повторное включение ACL может подключить предыдущие списки контроля доступа к файлам, права на которые изменены, что приведет к непредсказуемому поведению.

Файловые системы с включенными ACLs показывают знак + при просмотре прав на файлы. Например:

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

Здесь мы видим, что каталоги directory1, directory2, и directory3 используют преимущества ACL. Каталог public_html их не использует.

13.12.1. Использование ACL

ACL файловой системы можно просмотреть с помощью утилиты getfacl(1). Например, для просмотра настроек ACL файла test, может использоваться команда:

```
% getfacl test
        #file:test
        #owner:1001
        #group:1001
        user::rw-
        group::r--
        other::r--
```

Для изменения ACL этого файла, вызовите утилиту setfacl(1). Выполните:

```
% setfacl -k test
```

Параметр -k удалит все установленные на данный момент ACL из файла или файловой системы. Более предпочтительный метод это использование параметра -b, который оставит необходимые для работы ACL поля.

```
% setfacl -m u:trhodes:rwx,group:web:r--,o::--- test
```

В вышеприведенной команде параметр `-m` использован для изменения записей ACL по умолчанию. Поскольку предустановленных записей не было (они были удалены предыдущей командой), эта команда восстановит параметры по умолчанию и задаст приведенные параметры. Имейте ввиду, при добавлении пользователя или группы, которых нет в системе, на stdout будет выведена ошибка `Invalid argument`.

13.13. Мониторинг вопросов безопасности в ПО сторонних разработчиков

В последние годы в области информационной безопасности произошло много улучшений, касающихся выработки оценки уязвимости. Угроза проникновения в систему увеличивается вместе с установкой и настройкой утилит сторонних разработчиков, какой бы современной операционной системы это ни касалось.

Оценка уязвимости является ключевым фактором обеспечения защиты, и хотя для базового комплекта FreeBSD выпускаются бюллетени безопасности, но делать это для каждой сторонней утилиты выше возможностей участников Проекта FreeBSD. Существует способ смягчения уязвимостей программного обеспечения сторонних разработчиков и предупреждения администраторов об известных проблемах с безопасностью. Во FreeBSD существует утилита под названием Portaudit, которая служит исключительно этой цели.

Порт security/portaudit обращается к базе данных, обновляемой и поддерживаемой Группой информационной безопасности FreeBSD и разработчиками портов, для получения информации об известных проблемах с защитой.

Для того, чтобы приступить к использованию Portaudit, необходимо установить его из Коллекции Портов:

```
# cd /usr/ports/security/portaudit && make install clean
```

В процессе установки будут обновлены конфигурационные файлы для periodic(8), в которые будет добавлена выдача Portaudit при ежедневном её запуске. Проверьте, что ежедневные сообщения электронной почты, касающиеся безопасности, которые посылаются на адрес `root`, прочитываются. Другой дополнительной настройки больше не понадобится.

После установки администратор может обновить базу данных и посмотреть список известных уязвимостей в установленных пакетах при помощи команды

```
# portaudit -Fda
```

База данных будет автоматически обновлена при запуске periodic(8); таким образом, предыдущая команду можно полностью опустить. Она требуется

только для следующих примеров.

Для аудита утилит сторонних разработчиков, установленных как часть Коллекции Портов, администратору достаточно запускать только следующую команду:

```
# portaudit -a
```

Утилита portaudit выдаст примерно следующее:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <http://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-
0001020eed82.html>

1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

Перейдя в Web-браузере по показанному URL, администратор может получить более подробную информацию о показанной уязвимости. В неё войдёт перечисление версий, затронутых соответствующей версией порта FreeBSD, а также другие Web-сайты, которые могут содержать бюллетени безопасности.

Если описывать вкратце, то Portaudit является мощной и, при использовании вместе с портом Portupgrade, чрезвычайно полезной утилитой.

13.14. Сообщения безопасности FreeBSD

Как многие и высококачественные операционные системы, FreeBSD публикует "Сообщения безопасности" ("Security Advisories"). Эти сообщения обычно отправляются по почте в списки рассылки, посвященные безопасности и публикуются в списке проблем только после выхода исправлений к соответствующим релизам. В этом разделе разъясняется, что такое сообщения безопасности, как их читать и какие меры принимать для исправления системы.

13.14.1. Как выглядит сообщение?

Сообщение безопасности FreeBSD выглядит подобно сообщению ниже, взятому из списка рассылки Список рассылки FreeBSD, посвящённый срочным сообщениям, связанным с безопасностью.

```
=============================================================================
FreeBSD-SA-XX:XX.UTIL                                       Security Advisory
                                                            The FreeBSD Project

Topic:          denial of service due to some problem①
```

```
Category:       core②
Module:         sys③
Announced:      2003-09-23④
Credits:        Person@EMAIL-ADDRESS⑤
Affects:        All releases of FreeBSD⑥
                FreeBSD 4-STABLE prior to the correction date
Corrected:      2003-09-23 16:42:59 UTC (RELENG_4, 4.9-PRERELEASE)
                2003-09-23 20:08:42 UTC (RELENG_5_1, 5.1-RELEASE-p6)
                2003-09-23 20:07:06 UTC (RELENG_5_0, 5.0-RELEASE-p15)
                2003-09-23 16:44:58 UTC (RELENG_4_8, 4.8-RELEASE-p8)
                2003-09-23 16:47:34 UTC (RELENG_4_7, 4.7-RELEASE-p18)
                2003-09-23 16:49:46 UTC (RELENG_4_6, 4.6-RELEASE-p21)
                2003-09-23 16:51:24 UTC (RELENG_4_5, 4.5-RELEASE-p33)
                2003-09-23 16:52:45 UTC (RELENG_4_4, 4.4-RELEASE-p43)
                2003-09-23 16:54:39 UTC (RELENG_4_3, 4.3-RELEASE-p39)⑦
CVE Name:   CVE-XXXX-XXXX⑧

For general information regarding FreeBSD Security Advisories,
including descriptions of the fields above, security branches, and the
following sections, please visit
http://www.FreeBSD.org/security/.

I.   Background⑨

II.  Problem Description⑩

III. Impact⑪

IV.  Workaround⑫

V.   Solution⑬

VI.  Correction details⑭

VII. References⑮
```

① Поле `Topic` показывает в чем именно заключается проблема. Это обычно введение в сообщение безопасности, упоминающее утилиту, в которой возникла ошибка.

② Поле `Category` относится к затронутой части системы и может быть выбрана из `core`, `contrib`, или `ports`. Категория `core` означает, что уязвимость затрагивает основной компонент операционной системы FreeBSD. Категория `contrib` означает, что уязвимость затрагивает программы, предоставленные проекту FreeBSD, например sendmail. Наконец, категория `ports` означает, что уязвимость затрагивает программное обеспечение, доступное из Коллекции Портов.

③ Поле `Module` указывает на местоположение компонента, например `sys`. В этом примере мы видим, что затронут модуль `sys`, следовательно, эта уязвимость относится к компоненту, используемому в ядре.

④ Поле `Announced` отражает дату публикации сообщения безопасности, или его анонсирования. Это означает, что команда обеспечения безопасности убедилась, что проблема существует и что патч помещён в хранилище исходных текстов FreeBSD.

⑤ Поле `Credits` упоминает частное лицо или организацию, обнаружившую уязвимость и сообщившую о ней.

⑥ Поле `Affects` дает информацию о релизах FreeBSD, к которым относится данная уязвимость. Для базовой системы, просмотр вывода команды `ident` для файлов, затронутых уязвимостью, поможет определить ревизию. Номер версии портов приведен после имени порта в каталоге /var/db/pkg. Если система не синхронизируется с CVS-хранилищем FreeBSD и не пересобирается ежедневно, высок шанс, что она затронута уязвимостью.

⑦ Поле `Corrected` показывает дату, время, смещение во времени и релиз, в котором исправлена ошибка.

⑧ Зарезервировано для идентификации уязвимости в общей базе данных CVD (Common Vulnerabilities Database).

⑨ Поле `Background` дает информацию именно о той утилите, для которой выпущено сообщение. Как правило информация о том, зачем утилита присутствует в FreeBSD, для чего она используется, и немного информации о том, как появилась эта утилита.

⑩ Поле `Problem Description` дает более глубокие разъяснения возникшей проблемы. Оно может включать информацию об ошибочном коде, или даже о том, как утилита может быть использована для создания бреши в системе безопасности.

⑪ Поле `Impact` описывает тип воздействия, который проблема может оказать на систему. Это может быть все, что угодно, от атаки на отказ в обслуживании до получения пользователями дополнительных привилегий, или даже получения атакующим прав суперпользователя.

⑫ Поле `Workaround` предлагает тем, системным администраторам, которые не могут обновить систему, обходной путь решения проблемы. Он может пригодиться при недостатке времени, отсутствии подключения к сети или по массе других причин. В любом случае, к безопасности нельзя относиться несерьезно, и необходимо либо применить указанный обходной путь, либо исправить систему.

⑬ Поле `Solution` предлагает инструкции по исправлению затронутой системы. Это пошаговое руководство, протестированный метод восстановления безопасности системы.

⑭ Поле `Correction Details` показывает ветвь CVS (имя релиза с точками, замененными на символы подчеркивания). Здесь также показан номер ревизии каждого файла из каждой ветви.

⑮ Поле `References` обычно упоминает другие источники информации. Это могут быть Web-страницы, книги, списки рассылки и группы новостей.

13.15. Учёт используемых ресурсов

Учёт используемых процессами ресурсов представляет собой метод защиты, при котором администратор может отслеживать использование системных ресурсов и их распределение

между пользователями для нужд системного мониторинга и минимального отслеживания команд пользователей.

На самом деле здесь есть свои положительный и отрицательные моменты. Положительной стороной является то, что проникновение может быть отслежено до первоначальной точки входа. Отрицательной стороной является объём протоколов, который генерируется при мониторинге, и соответствующие требования к дисковому пространству. В этом разделе администратору даются основы учёта ресурсов процессов.

13.15.1. Активация и использование учёта ресурсов

Прежде чем использовать систему учёта ресурсов, её необходимо активировать. Для этого выполните следующие команды:

```
# touch /var/account/acct

# accton /var/account/acct

# echo 'accounting_enable="YES"' >> /etc/rc.conf
```

После активации система учёта ресурсов начнёт отслеживать статистику CPU, команд и так далее. Все протоколы учёта ведутся в формате, недоступном для чтения человеком, и могут просматриваться при помощи утилиты sa(8). Запущенная без параметров, `sa` выдаст информацию, относящуюся к количеству вызовов в расчёте на каждого пользователя, общее затраченное время в минутах, общее время CPU и пользователя в минутах, среднее количество операций ввода/вывода и так далее.

Для просмотра информации о запущенных командах, необходимо воспользоваться утилитой lastcomm(1). Команду `lastcomm` можно использовать, например, для выдачи списка директив, выданных пользователями определённого терминала ttys(5):

```
# lastcomm ls trhodes ttyp1
```

Эта команда выдаст все зафиксированные использования команды `ls` пользователем `trhodes` на терминале ttyp1.

Существует многие другие полезные параметры, которые описаны на соответствующих справочных страницах lastcomm(1), acct(5) и sa(8).

Глава 14. Принудительный контроль доступа (MAC)

14.1. Краткий обзор

FreeBSD 5.X представляет новые расширения системы безопасности от проекта TrustedBSD, основанные на документах POSIX®.1e. Два из наиболее важных нововведений в механизмах безопасности это списки контроля доступа файловой системы (Access Control Lists, ACLs) и принудительный контроль доступа Mandatory Access Control, MAC). Инфраструктура позволяет загружать новые модули контроля доступа, реализуя новые политики безопасности. Некоторые из них предоставляют защиту ключевых подсистем, защищая определенный сервис, в то время как другие предоставляют исчерпывающую систему безопасности с метками на всех субъектах и объектах. Контроль называется принудительным, поскольку применение контроля производится администраторами и системой, и не зависит от решения пользователей, как это происходит при обычном контроле доступа (Discretionary Access Control, DAC, стандартные файловые и System V IPC права в FreeBSD).

Вся эта глава фокусируется на инфраструктуре принудительного контроля доступа и настройке подключаемых модулей, реализующих различные политики безопасности.

После прочтения этой главы вы узнаете:

- Какие модули MAC включены в настоящее время в FreeBSD, какие политики с ними связаны.
- Что способны реализовать политики MAC, различие между политиками с метками (label) и без меток.
- Как эффективно настроить систему для использования инфраструктуры MAC.
- Как настроить различные политики, используемые модулями MAC.
- Как реализовать более защищенную среду, используя инфраструктуру MAC и приведенные примеры.
- Как протестировать настройку MAC, чтобы убедиться, что инфраструктура была реализована правильно.

Перед прочтением этой главы вам потребуется:

- Понимание основ UNIX® и FreeBSD (Основы UNIX).
- Ознакомиться с основами настройки/компилирования ядра (Настройка ядра FreeBSD).
- Иметь некоторые понятия о безопасности и как она относится к FreeBSD (Безопасность).

Неправильное использование информации этой главы может вызвать потерю доступа к системе, проблемы у пользователей, или невозможность запуска XFree86™. Что более важно, MAC не должен восприниматься как полная защита системы. Инфраструктура MAC лишь усиливает имеющуюся

> систему безопасности: без применения методов защиты и регулярных проверок, система никогда не станет полностью защищенной.
>
> Необходимо также учесть, что примеры, приведенные в этой главе, это всего лишь примеры. Не рекомендуется дублирование этих настроек для реальных задач. Реализация политик безопасности требует вдумчивого планирования. Тот, кто не понял полностью как все это на самом деле работает, может столкнуться с необходимостью повторной полной перенастройки системы безопасности для многих файлов и каталогов.

14.1.1. Что не будет затронуто

В этой главе охвачен широкий спектр вопросов безопасности, относящихся к инфраструктуре MAC. однако разработка политик MAC не будет затронута. Несколько модулей, включенных в инфраструктуру MAC, имеют особые характеристики, которые предназначены для тестирования и разработки новых модулей. Это относится к модулям/политикам mac_test(4), mac_stub(4) и mac_none(4). За дальнейшей информацией по этим модулям и различным предоставляемым ими механизмам, обратитесь к соответствующим страницам справочника.

14.2. Ключевые термины этой главы

Перед чтением этой главы необходимо ознакомиться с некоторыми ключевыми терминами. Это возможно разрешит возникающие вопросы и предотвратит перегрузку материала новыми терминами и информацией.

- *отдел* (compartment): Отдел это набор программ и данных, которые необходимо отделить, и где пользователи получают явный доступ к отдельным компонентам системы. Отдел представляет группирование, например рабочую группу, департамент, проект или тему. С использованием отделов возможна реализация политики с явно определенным доступом.
- *целостность* (integrity): Целостность, как ключевая концепция, это уровень доверия, который может быть присвоен данным. Поскольку целостность данных повышается, это дает возможность доверять данным.
- *метка* (label): Метка является инструментом безопасности, она может быть применена к файлам, каталогам и другим сущностям системы. Ее можно представить как штамп конфиденциальности; метка, помещенная на файл, описывает уровень секретности данного файла и разрешит доступ только файлам, пользователям, ресурсам и т.д. с теми же или меньшими установками безопасности. Некоторые из политик могут обрабатывать метки различными способами; это будет обсуждаться в разделе политик ниже.
- *multilabel* (множественные метки): свойство `multilabel` это параметр файловой системы, который может быть установлен в однопользовательском режиме с помощью утилиты tunefs(8), во время загрузки через файл fstab(5), или при создании новой файловой системы. Этот параметр позволяет администратору помещать различные метки MAC на различные объекты. разрешает помещение множественных MAC меток на файлы и

каталоги файловой системы. Этот параметр применим только к политикам с метками.

- *объект* (object): Объект или системный объект это сущность, через которую информация проходит к *субъекту*. Это могут быть каталоги, файлы, поля, экраны, клавиатуры, память, магнитные накопители, принтеры или любые другие устройства хранения/перемещения данных. В сущности это контейнер данных или ресурс системы; доступ к *объекту* фактически означает доступ к данным.
- *политика* (policy): Набор правил, определяющих как достичь объекта. *Политика* обычно документирует обращение с определенными элементами. В этой главе *политика* будет означать *политику безопасности*; т.е. коллекцию правил, которые будут контролировать поток данных и определять кто будет иметь доступ к этим данным.
- *чувствительность* (sensitivity): Обычно используется при обсуждении MLS. Уровень чувствительности это термин, используемый для описания того, насколько важны или секретны данные. Увеличение уровня чувствительности означает важность данных.
- *одиночная метка* (single label): означает, что вся файловая система использует одну метку для определения доступа всего потока данных. Когда файловая система использует эту установку, что происходит всегда, если не установлен параметр `multilabel`, ко всем файлам будет применяться одна и та же установка метки.
- *субъект* (subject): субъект это любая активная сущность, вызывающая перемещение информации между *объектами*; т.е. пользователь, пользовательский обработчик, системный процесс и т.д. В FreeBSD это почти всегда поток, работающий в процессе или представляющий пользователя.

14.3. Описание MAC

Усвоив все эти термины, рассмотрим как MAC повышает безопасность системы в целом. Различные политики, предоставляемые инфраструктурой MAC, могут быть использованы для защиты сети и файловых систем, блокирования доступа пользователей к определенным портам и сокетам, и так далее. Возможно, наилучшее использование политик это сочетание их вместе путем загрузки нескольких модулей одновременно, для создания многослойной защищенной среды. В многослойной среде безопасности несколько политик обеспечивают контролируемость защиты. Это отличается от усиления защиты, когда обычно усиливаются элементы системы, используемой в определенных целях. Единственным недостатком является дополнительная административная нагрузка в случае множественных меток файловой системы, установки сетевого доступа по пользователям, и т.д.

Эти недостатки минимальны по сравнению с длительным эффектом функционирования инфраструктуры; например, возможность выбора политик, необходимых для определенных конфигураций, уменьшает потерю производительности. Возможность удаления поддержки не используемых политик может увеличить общую производительность системы, а также дает гибкость выбора. Хорошая реализация удовлетворит общие требования безопасности и будет эффективно использовать различные политики, предоставляемые инфраструктурой.

Система, использующая возможности MAC, должна как минимум гарантировать, что пользователю не разрешается самостоятельно изменять атрибуты безопасности; все

утилиты пользователя, программы и скрипты должны работать с ограничениями доступа, налагаемыми выбранной политикой; весь контроль правил доступа MAC находится в руках системного администратора.

Право выбора правильных политик безопасности принадлежит только системному администратору. В некоторых случаях может потребоваться ограничение доступа через сеть; для этого могут пригодиться mac_portacl(4), mac_ifoff(4) и даже mac_biba(4). В других случаях может быть необходима строгая конфиденциальность объектов в файловой системе. Для этого существуют политики mac_bsdextended(4) и mac_mls(4).

Выбор политики может быть сделан на основе конфигурации сети. Возможно только определенным пользователям можно разрешить доступ через ssh(1) к сети или интернет. В таких ситуациях подойдет политика mac_portacl(4). Но что необходимо сделать для файловых систем? Должен ли доступ к определенным каталогам быть запрещен для других групп или определенных пользователей? Или мы должны ограничить доступ пользователей или утилит к определенным файлам путем классификации определенных объектов?

В случае файловой системы, доступ может считаться конфиденциальным для отдельных пользователей, но не для всех. Например, большая команда разработчиков может быть разбита на небольшие группы. Разработчикам проекта A может быть не разрешен доступ к объектам, написанным разработчиками из проекта B. Хотя им может понадобиться доступ к объектам, созданным разработчиками проекта C; это реально встречающаяся ситуация. С помощью различных политик, предоставляемых инфраструктурой MAC, пользователи могут быть разделены на эти три группы и затем получить доступ к соответствующим областям без опасности утечки информации.

Таким образом, каждая политика имеет уникальный способ взаимодействия с общей безопасностью системы. Выбор политики должен быть основан на хорошо продуманной политике безопасности. Во многих случаях политика должна быть полностью пересмотрена и реализована заново для всей системы. Понимание различных политик, предоставляемых инфраструктурой MAC, поможет администраторам выбрать лучшую политику в своей ситуации.

Стандартное ядро FreeBSD не включает параметр MAC; необходимо добавить следующий параметр ядра перед тем, как пробовать какие-либо из примеров или применять информацию этой главы:

```
options MAC
```

Затем необходимо пересобрать и переустановить ядро.

Хотя различные страницы справочника для модулей MAC сообщают, что они могут быть встроены в ядро, возможна блокировка доступа системы к сети и другие побочные эффекты. Включение MAC очень похоже на включение брандмауэра, но необходимо быть внимательным, чтобы полностью не заблокировать систему. Необходимо предусмотреть возможность возврата к предыдущей конфигурации, а реализация MAC

удаленно должна производиться с особой осторожностью.

14.4. Метки MAC

Метка MAC это атрибут безопасности, который может быть применен к субъектам и объектам всей системы.

При установке метки пользователь должен в точности понимать, что именно она делает. Атрибуты, доступные для объекта, зависят от загруженной политики, а политики интерпретируют свои атрибуты совершенно различным образом. Результатом недостаточного понимания настроек может стать их неправильная реализация, что может привести к неожиданному, и возможно нежелательному поведению системы.

Метка безопасности на объекте используется политикой для определения правил доступа. Для некоторых политик метка сама по себе содержит всю необходимую для этого информацию, в других моделях метки могут обрабатываться как часть большого набора правил, и т.д.

Например, установка метки в `biba/low` на файле присвоит этому файлу метку, обрабатываемую политикой Biba со значением "low".

Несколько политик, поддерживающих метки в FreeBSD, предоставляют три определенные предустановленные метки. Это низкая, высокая и равная метки. Хотя они устанавливают контроль различными способами для каждой политики, вы можете быть уверены, что низкая метка задаст минимальные установки, равная метка означает отмену или недействительность, а высокая метка установит максимально возможные настройки в политиках Biba и MLS.

При применении в файловой системе одиночной метки, только одна метка может быть использована для объектов. Это вызовет установку одних и тех же прав доступа для всей системы, и во многих случаях это все, что необходимо. Тем не менее, существует несколько ситуаций, в которых на объекты и субъекты файловой системы могут быть установлены множественные метки. В этих ситуациях необходимо с помощью tunefs(8) установить параметр `multilabel`.

В случае Biba и MLS может быть установлена числовая метка для указания точного уровня иерархического контроля. Этот числовой уровень используется для разделения или сортировки информации по различным группам классификации, разрешающей доступ только к этой группе или группе с более высоким уровнем.

В большинстве случаев системный администратор использует только одну метку на всю файловую систему.

Постойте, но это же похоже на DAC! Я думал, что MAC дает контроль только администратору. Это утверждение все еще верно, только `root` контролирует и настраивает политики, так что пользователи помещаются в соответствующие категории/уровни доступа. Многие политики могут ограничить также и пользователя `root`. Базовый контроль над объектами затем передается группе, но пользователь `root` может отменить или изменить эти настройки в любое время. Данная иерархическая модель соответствует таким

политикам как Biba и MLS.

14.4.1. Настройка меток

Практически все действия по настройке политики с метками могут быть выполнены с использованием утилит базовой системы. Эти команды обеспечивают простой интерфейс для настройки объектов или субъектов, или для изменения и проверки настроек.

Все настройки могут быть выполнены с использованием утилит setfmac(8) и setpmac(8). Команда `setfmac` используется для установки меток MAC на системные объекты, а команда `setpmac` используется для установки меток на системные субъекты. Выполните:

```
# setfmac biba/high test
```

Если не произойдет ошибок, будет возвращено приглашение командной строки, как и после команд chmod(1) и chown(8). В некоторых случаях может появиться ошибка `Permission denied`, и она обычно появляется при установке или изменении метки на объект с ограничениями. Системный администратор для обхода этой проблемы может использовать следующие команды:

```
# setfmac biba/high test
Permission denied
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

Как видно из примера выше, команда `setpmac` может быть использована для изменения установок политики путем присвоения иной метки вызывающему процессу. Утилита `getpmac` обычно используется с существующим на данный момент процессом, таким как sendmail, хотя она принимает PID вместо команды, ее действие аналогично. Если пользователи попытаются манипулировать файлами, к которым у них нет доступа в соответствии с правилами загруженных политик, функцией `mac_set_link` будет выдано сообщение об ошибке `Operation not permitted`.

14.4.1.1. Пользователи и установки меток

Пользователям необходимо иметь метки, чтобы их файлы и процессы могли правильно взаимодействовать с определенной в системе политикой безопасности. Это настраивается через файл login.conf путем использования классов. Каждая политика, использующая метки, реализует установку класса пользователя.

Пример записи, содержащей все политики, приведенные ниже:

```
default:\
    :copyright=/etc/COPYRIGHT:\
    :welcome=/etc/motd:\
    :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
```

```
    :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
    :manpath=/usr/shared/man /usr/local/man:\
    :nologin=/usr/sbin/nologin:\
    :cputime=1h30m:\
    :datasize=8M:\
    :vmemoryuse=100M:\
    :stacksize=2M:\
    :memorylocked=4M:\
    :memoryuse=8M:\
    :filesize=8M:\
    :coredumpsize=8M:\
    :openfiles=24:\
    :maxproc=32:\
    :priority=0:\
    :requirehome:\
    :passwordtime=91d:\
    :umask=022:\
    :ignoretime@:\
    :label=partition/13,mls/5,biba/10(5-15),lomac10[2]:
```

Параметр `label` используется для установки метки MAC по умолчанию для класса пользователя. Пользователи не смогут изменять это значение, поэтому его можно признать не опциональным. В реальной ситуации администратору никогда не потребуется включать каждую политику. Рекомендуется прочесть главу полностью перед реализацией любой из этих настроек.

Пользователи могут изменить свою метку после входа; однако политика накладывает ограничение на это изменение. В примере выше политике Biba указано, что минимальная целостность процесса 5, максимальная 15, а эффективная целостность по умолчанию 10. Процесс будет работать на уровне 10, пока метка не будет изменена, например если пользователь использует команду setpmac, которую Biba ограничит диапазоном, установленным при входе.

Во всех случаях после изменения login.conf, база данных "login class capability" должна быть пересобрана с использованием команды `cap_mkdb` и это будет отражено в каждом последующем примере главы.

Полезно отметить, что количество пользователей, которым требуются различные классы, во многих сетях может быть велико. Необходимо тщательное планирование, поскольку управление такой сетью может серьезно усложниться.

В будущих версиях FreeBSD появится новый способ связывания пользователей с метками; однако, он будет доступен только через некоторое время после выхода FreeBSD 5.3.

14.4.1.2. Сетевые интерфейсы и установка меток

Метки также могут быть установлены на сетевые интерфейсы, для контроля потока данных в сети. Во всех случаях они функционируют аналогично тому, как политики по отношению

к объектам. Пользователи с высокими установками, например, biba, не смогут получить доступ к сетевым интерфейсам с низкими установками.

Для установки MAC меток на сетевых интерфейсах параметр maclabel может быть передан ifconfig. Например:

```
# ifconfig bge0 maclabel biba/equal
```

установит MAC метку biba/equal на интерфейс bge(4). При использовании метки, подобной biba/high(low-high) вся метка должна быть взята в кавычки, иначе будет выдано сообщение об ошибке.

Каждая политика, использующая метки, снабжена переменной sysctl, которая может быть использована для отключения MAC меток на сетевых интерфейсах. Установка метки в equal будет иметь подобный эффект. Просмотрите вывод команды sysctl, страницы справочника для политик, или дальнейшую информацию из этой главы по этим переменным.

14.4.2. Одиночные или множественные метки?

По умолчанию система будет использовать параметр singlelabel. Но что это означает для администратора? Существуют несколько различий между политиками, каждая из которых правильна сама по себе, но имеет свои доводы за и против относительно гибкости модели безопасности системы.

singlelabel (одиночная метка) разрешает использование только одной метки, например biba/high, для каждого объекта или субъекта. Ее преимущество в меньшей нагрузке на системного администратора, а недостаток в малой гибкости политик, поддерживающих метки. Многие администраторы в своих политиках безопасности могут предпочесть использование параметра multilabel.

С параметром multilabel каждый субъект или объект может иметь собственную метку MAC, в то время как со стандартным параметром singlelabel возможна только одна метка на весь раздел. Параметры multilabel и singlelabel требуются только для политик, реализующих метки, включая Biba, Lomac, MLS и SEBSD.

Во многих случаях multilabel может вообще не потребоваться. Предположим следующую ситуацию и модель безопасности:

- FreeBSD веб-сервер, использующий инфраструктуру MAC и набор различных политик.
- Этому компьютеру потребуется лишь одна метка, biba/high, для всей системы. Файловой системе не нужен параметр multilabel, поскольку по умолчанию работает одиночная метка.
- Но поскольку этот компьютер будет веб сервером, процесс веб сервера должен быть запущен с biba/low для предотвращения записи. Политика Biba и то, как она работает, будет обсуждаться позже, поэтому предыдущий комментарий сложно интерпретировать; просто продолжайте чтение. Сервер может использовать дисковый раздел с установленной меткой biba/low для большинства, если не для всех своих операций. В этом примере отсутствуют многие детали, такие как ограничения на

данные, конфигурация системы и установки пользователей; однако, это лишь предварительный пример.

Если используется любая из политик, не поддерживающих метки, параметр `multilabel` не требуется. Сюда включаются политики `seeotheruids`, `portacl` и `partition`.

Необходимо также отметить, что использование параметра `multilabel` на разделе и установление модели безопасности, основанной на функциональности `multilabel`, может повлечь за собой множество дополнительной административной работы, поскольку всему в файловой системе должны быть присвоены метки. Это включает каталоги, файлы, и даже файлы устройств.

Следующая команда установит параметр `multilabel` на файловых системах. Это может быть сделано только в однопользовательском режиме:

```
# tunefs -l enable /
```

Это не требуется для файловой системы подкачки.

Некоторые пользователи сталкиваются с проблемами при установке флага `multilabel` на корневой раздел. В данном случае обратитесь к Решение проблем с инфраструктурой MAC.

14.4.3. Настройка MAC переменными sysctl

Независимо от загрузки модулей, существует несколько частей MAC, которые могут быть настроены с использованием интерфейса `sysctl`. Эти переменные описаны ниже и во всех случаях значение 1 означает включение, а 0 - отключение:

- `security.mac.enforce_fs` по умолчанию установлена в 1 и включает политики MAC на файловых системах.
- `security.mac.enforce_kld` по умолчанию 1 и включает линкование политик MAC в ядре (см. kld(4)).
- `security.mac.enforce_network` по умолчанию 1 и включает сетевые политики MAC.
- `security.mac.enforce_pipe` по умолчанию 1 и включает политики MAC для каналов (pipe).
- `security.mac.enforce_process` по умолчанию 1 и включает политики MAC для процессов, использующих средства межпроцессного взаимодействия.
- `security.mac.enforce_socket` по умолчанию 1 и включает политики MAC на сокетах (см. страницу справочника socket(2)).
- `security.mac.enforce_system` по умолчанию 1 и включает политики MAC для действий системы, таких как учет (accounting) и перезагрузка.
- `security.mac.enforce_vm` по умолчанию 1 и включает политики MAC для системы виртуальной памяти.

Каждая политика MAC поддерживает переменные `sysctl`. Они обычно

попадают в дерево security.mac.<policyname>. Для просмотра всех переменных MAC, используйте следующую команду:

```
# sysctl -da | grep mac
```

Это должно быть интерпретировано так, что все основные политики MAC включены по умолчанию. Если модули встроены в ядро, система будет заблокирована, и скорее всего не сможет связаться с локальной сетью или с интернет, и т.д. Поэтому встраивание модулей в ядро не рекомендуется. Не потому, что это ограничит возможность отключения командой sysctl, а потому, что включение политик в виде модулей позволит администратору переключать политики системы без необходимости пересборки и переустановки новой системы.

14.5. Настройка модулей

Каждый модуль, включенный в инфраструктуру MAC, может быть или встроен в ядро, как упоминалось выше, или загружен в виде модуля ядра. Рекомендуется добавление имени модуля в файл /boot/loader.conf, этот модуль будет активирован в самом начале загрузки.

В последующих разделах будут обсуждаться различные модули MAC и их возможности. Реализация этих возможностей в определенных ситуациях также будет обсуждаться в этой главе. Некоторые модули поддерживают использование меток, которые контролируют доступ путем применения правил вида "это разрешено, а это нет". Настройка меток может контролировать доступ к файлам, сетевым коммуникациям и т.д. В предыдущем разделе было показано как флаг multilabel может быть установлен на файловые системы для включения контроля доступа по файлам или по разделам.

Конфигурация с одной меткой не допускает применение нескольких меток в системе, поэтому параметр tunefs называется multilabel.

14.5.1. Модуль MAC seeotheruids

Имя модуля: mac_seeotheruids.ko

Строка в конфигурации ядра: options MAC_SEEOTHERUIDS

Параметр загрузки: mac_seeotheruids_load="YES"

Модуль mac_seeotheruids(4) копирует и расширяет переменные sysctl security.bsd.see_other_uids и security.bsd.see_other_gids. Он не требует установки меток и может прозрачно работать с другими модулями.

После загрузки модуля, для управления им могут быть использованы следующие переменные sysctl:

- security.mac.seeotheruids.enabled включит модуль с настройками по умолчанию. Эти настройки запрещают пользователям просмотр процессов и сокетов, принадлежащих другим пользователям.

- `security.mac.seeotheruids.specificgid_enabled` позволит исключить определенные группы из этой политики. Для исключения определенной группы, используйте переменную `sysctl security.mac.seeotheruids.specificgid=XXX`. В примере выше необходимо заменить *XXX* на числовой ID группы.
- `security.mac.seeotheruids.primarygroup_enabled` используется для исключения определенной основной группы из этой политики. При использовании этой переменной `security.mac.seeotheruids.specificgid_enabled` может быть не установлена.

Необходимо отметить, что пользователь `root` не является исключением из этой политики. Это одно из самых существенных различий между MAC версией и обычными переменными, существующими по умолчанию: `security.bsd.seeotheruids`.

14.6. Модуль MAC bsdextended

Имя модуля: mac_bsdextended.ko

Строка конфигурации ядра: `options MAC_BSDEXTENDED`

Параметр загрузки: `mac_bsdextended_load="YES"`

Модуль mac_bsdextended(4) включает брандмауэр файловой системы. Политика этого модуля является расширением стандартной модели разрешений файловой системы, позволяя администратору создавать набор правил для защиты файлов, утилит и каталогов иерархии файловой системы в стиле брандмауэра.

Политика может быть создана с помощью утилиты, ugidfw(8), синтаксис которой похож на синтаксис ipfw(8). Другие инструменты могут быть написаны с использованием функций библиотеки libugidfw(3).

При работе с этим модулем необходимо соблюдать особую осторожность; некорректное его использование может заблокировать доступ к отдельным частям файловой системы.

14.6.1. Примеры

После загрузки модуля mac_bsdextended(4) для просмотра текущей настройки правил может быть использована следующая команда:

```
# ugidfw list
0 slots, 0 rules
```

Как и можно было ожидать, правила не определены. Это означает, что доступ полностью открыт. Для создания правила, которое заблокирует доступ всех пользователей, но не повлияет на `root`, просто запустите следующую команду:

```
# ugidfw add subject not uid root new object not uid root mode n
```

В релизах FreeBSD до 5.3, параметр add не существует. Вместо него

необходимо использовать set. Пример дан ниже.

Это очень плохая идея, поскольку такое правило запретит пользователям использовать даже самые простые команды, такие как `ls`. Более патриотический список правил может быть таким:

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

Эти команды запретят весь и любой доступ пользователя `user1`, включая просмотр подкаталогов, к домашнему каталогу пользователя `user2`.

Вместо `user1` может быть задано `not uid user2`. Это включит те ограничения, о которых говорилось выше, для всех пользователей кроме одного.

На пользователя `root` эти изменения не повлияют.

Материал выше должен дать общую идею как модуль mac_bsdextended(4) может быть использован в качестве средства защиты файловой системы. За дальнейшей информацией обращайтесь к страницам справочника mac_bsdextended(4) и ugidfw(8).

14.7. Модуль MAC ifoff

Имя модуля: mac_ifoff.ko

Строка конфигурации ядра: `options MAC_IFOFF`

Параметр загрузки: `mac_ifoff_load="YES"`

Модуль mac_ifoff(4) существует только для отключения сетевых интерфейсов в работающей системе и удержания их от отправки пакетов во время начальной загрузки. Это не требует установления в системе каких-либо меток, нет и зависимости от других модулей MAC.

Большая часть управления может быть выполнена через переменные `sysctl`.

- `security.mac.ifoff.lo_enabled` включает/выключает весь трафик на loopback (lo(4)) интерфейсе.
- `security.mac.ifoff.bpfrecv_enabled` включает/выключает весь трафик на интерфейсе Berkeley Packet Filter (bpf(4)).
- `security.mac.ifoff.other_enabled` включает/выключает весь трафик на всех других интерфейсах.

Одно из наиболее частых использований mac_ifoff(4) это сетевой мониторинг в среде, где сетевой трафик не должен быть разрешен во время загрузки. Другое предлагаемое применение это написание скрипта, использующего security/aide для автоматического блокирования сетевого трафика, если будут обнаружены новые или измененные файлы в защищаемых каталогах.

14.8. Модуль MAC portacl

Имя модуля: mac_portacl.ko

Строка конфигурации ядра: `MAC_PORTACL`

Параметр загрузки: `mac_portacl_load="YES"`

Модуль mac_portacl(4) используется для ограничения привязки (binding) к локальным портам TCP и UDP, используя различные переменные `sysctl`. По сути mac_portacl(4) делает возможной привязку к привилегированным портам, т.е. к портам с номерами меньше 1024 для не-`root` пользователей.

После загрузки этот модуль включит политику MAC на всех сокетах. Доступны следующие переменные `sysctl`:

- `security.mac.portacl.enabled` включает/отключает политику целиком.
- `security.mac.portacl.port_high` установит наибольший номер порта, для которого mac_portacl(4) включает защиту.
- `security.mac.portacl.suser_exempt`, если установлена в ненулевое значение, исключает пользователя `root` из этой политики.
- `security.mac.portacl.rules` задает действующую политику mac_portacl: см. ниже.

Действующая политика `mac_portacl`, указанная в `security.mac.portacl.rules`, это текстовая строка в форме `rule[,rule,…]` с таким количеством правил, которое требуется. Каждое правило задается в формате: `idtype:id:protocol:port`. Параметр idtype может принимать значения `uid` или `gid` и используется для интерпретации параметра id, в качестве id пользователя или группы соответственно. Параметр protocol используется для определения применимости этого правила к протоколу TCP или UDP, он может принимать значения `tcp` или `udp`. Последний параметр, port, задает номер порта, к которому разрешается привязка указанного пользователя или группы.

Поскольку набор правил интерпретируется непосредственно ядром, для ID пользователя, группы и номера порта могут быть использованы только числовые значения. Т.е. имена пользователей, групп и сервисов портов не могут быть использованы.

По умолчанию в UNIX®-подобных системах порты с номерами менее чем 1024 могут быть использованы только привилегированными процессами, т.е. теми, что запущены от `root`. С mac_portacl(4) для разрешения привязки непривилегированных процессов к портам с номерами ниже 1024 эти стандартные ограничения UNIX® должны быть отменены. Это может быть выполнено путем установки переменных sysctl(8) `net.inet.ip.portrange.reservedlow` и `net.inet.ip.portrange.reservedhigh` в ноль.

Обратитесь к примерам ниже или к странице справочника mac_portacl(4) за дальнейшей информацией.

14.8.1. Примеры

Следующие примеры должны осветить обсуждение выше чуть лучше:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0 net.inet.ip.portrange.reservedhigh=0
```

Сначала мы настраиваем mac_portacl(4) для работы со стандартными привилегированными портами и отмены обычных ограничений UNIX® на привязку.

```
# sysctl security.mac.portacl.suser_exempt=1
```

Пользователь `root` должен быть исключен из этой политики, для этого переменная `security.mac.portacl.suser_exempt` установлена в ненулевое значение. Модуль mac_portacl(4) теперь настроен на то поведение UNIX®-подобных систем по умолчанию.

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Разрешает пользователю с UID 80 (обычно это пользователь `www`) привязку к порту 80. Теперь пользователь `www` сможет запустить веб сервер даже без привилегии `root`.

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

Разрешит пользователю с UID 1001 привязку к TCP портам 110 ("pop3") и 995 ("pop3s"). Это позволит данному пользователю запустить сервер, принимающий соединения на портах 110 и 995.

14.9. Политики MAC, использующие метки

В следующих нескольких разделах будут обсуждаться политики MAC, использующие метки.

С этого момента обсуждение будет сфокусировано на возможностях mac_biba(4), mac_lomac(4), mac_partition(4), и mac_mls(4).

> Это лишь примерные настройки, они не должны использоваться непосредственно в реальных задачах. Цель изложения в том, чтобы документировать и показать синтаксис, а также примеры реализации и тестирования.

Для правильной работы этих политик необходимо выполнить некоторые приготовления.

14.9.1. Приготовления к использованию политик с метками

В файл login.conf необходимо внести следующие изменения:

- Должен быть добавлен класс `insecure`, или другой подобный класс. Наличие класса `insecure` не обязательно, он приводится здесь в качестве примера; другие конфигурации могут использовать другое имя класса.
- Класс `insecure` должен использовать приведенные ниже настройки и определения. Некоторые из них могут быть изменены, но строка, определяющая метку по умолчанию, необходима и должна быть оставлена.

```
insecure:\
    :copyright=/etc/COPYRIGHT:\
    :welcome=/etc/motd:\
    :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
    :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
    :manpath=/usr/shared/man /usr/local/man:\
    :nologin=/usr/sbin/nologin:\
    :cputime=1h30m:\
    :datasize=8M:\
    :vmemoryuse=100M:\
    :stacksize=2M:\
    :memorylocked=4M:\
    :memoryuse=8M:\
    :filesize=8M:\
    :coredumpsize=8M:\
    :openfiles=24:\
    :maxproc=32:\
    :priority=0:\
    :requirehome:\
    :passwordtime=91d:\
    :umask=022:\
    :ignoretime@:\
    :label=partition/13,mls/5,biba/low:
```

Перед тем, как переключать пользователей на новый класс, необходимо запустить команду cap.mkdb(1) на login.conf(5).

Пользователю `root` также необходимо присвоить класс; иначе, почти любой команде, выполняемой от `root`, потребуется использование `setpmac`.

- Убедитесь, что все разделы, на которых будут установлены метки MAC, поддерживают параметр `multilabel`. Нам необходимо сделать это, поскольку многие из примеров здесь содержат различные метки в целях тестирования. Просмотрите вывод команды `mount` в качестве необходимой предосторожности.
- Переключите всех пользователей, которые будут использовать новые механизмы безопасности, на этот класс. Информация по этой процедуре находится в pw(8) или vipw(8).

14.10. Модуль MAC partition

Имя модуля: mac_partition.ko

Строка настройки ядра: `options MAC_PARTITION`

Параметр загрузки: `mac_partition_load="YES"`

Политика mac_partition(4) распределяет процессы по "разделам" на основе их MAC меток. Это может быть представлено как особый тип jail(8), хотя такое сравнение едва ли подходит.

Этот модуль должен быть добавлен в loader.conf(5), чтобы политика была загружена и включена при загрузке системы.

Большая часть настройки этой политики выполняется с помощью утилиты setpmac(8), которая будет описана ниже. Для данной политики имеется также следующая переменная `sysctl`:

- `security.mac.partition.enabled` включит MAC разделение процессов.

Когда эта политика включена, пользователям разрешено просматривать только собственные процессы, но не разрешено пользоваться определенными утилитами. Например, пользователю из класса `insecure` выше не будет разрешено использование команды `top`, а также многих других команд, которые должны порождать процесс.

Для присвоения утилитам меток partition используйте утилиту `setpmac`:

```
# setpmac partition/13 top
```

Команда `top` будет добавлена к метке, установленной для пользователей класса `insecure`. Обратите внимание, что все процессы, порожденные пользователями класса `insecure`, останутся с меткой `partition/13`.

14.10.1. Примеры

Следующая команда покажет вашу метку раздела и список процессов:

```
# ps Zax
```

Следующей командой можно просмотреть метку раздела процессов других пользователей и их запущенные процессы:

```
# ps -ZU trhodes
```

Пользователи могут могут увидеть процессы `root`, если не загружена политика mac_seeotheruids(4).

Действительно "продвинутая" реализация должна отключать все сервисы через /etc/rc.conf и запускать их через скрипт, который установит правильный набор меток.

Следующие политики поддерживают целочисленные установки вместо трех меток по умолчанию. Эти опции, включая их ограничения, описываются более подробно в страницах справочника модулей.

14.11. Модуль многоуровневой безопасности MAC (MLS)

Имя модуля: mac_mls.ko

Строка конфигурации ядра: `options MAC_MLS`

Параметр загрузки: `mac_mls_load="YES"`

Политика mac_mls(4) контролирует взаимодействие субъектов и объектов системы путем применения строгой политики к потоку информации.

В среде MLS, для каждого субъекта или объекта внутри отдела (compartment) устанавливается "уровень допуска". Поскольку количество уровней допуска может превышать шесть тысяч, для любого системного администратора задача настройки каждого субъекта или объекта может быть слишком сложной. К счастью, существуют "постоянные" метки, которые уже включены в эту политику.

Эти метки `mls/low`, `mls/equal` и `mls/high`. Поскольку эти метки подробно описываются в справочнике, здесь мы дадим только краткое описание:

- Метка `mls/low` содержит минимальную настройку, что позволяет доминирование над ней всех других объектов. Все, что помечено с `mls/low`, находится на низком уровне доступа и доступ к более высоким уровням будет запрещен. Кроме того, эта метка предотвратит запись или передачу информации объектам с более высоким уровнем доступа.
- Метка `mls/equal` должна быть помещена на объекты, являющиеся исключением из политики.
- Метка `mls/high` это наибольший возможный уровень доступа. Объекты с этой меткой будут доминировать над всеми другими объектами системы; однако, утечка информации от них к объектам более низкого класса невозможна.

MLS представляет собой:

- Иерархические уровни безопасности с набором не иерархических категорий;
- Фиксированные правила: нет чтения сверху, нет записи вниз (субъект может иметь доступ на чтение объектов собственного уровня или ниже, но не выше. Аналогично, субъект может иметь доступ на запись в объекты своего уровня или выше, но не наоборот.);
- Секретность (предотвращение неавторизованного раскрытия данных);

- Основа для разработки систем, одновременно работающих с данными на нескольких уровнях секретности (без утечки информации).

Для настройки специальных сервисов и интерфейсов доступны следующие переменные `sysctl`:

- `security.mac.mls.enabled` используется для включения/отключения политики MLS.
- `security.mac.mls.ptys_equal` пометит все устройства pty(4) как `mls/equal` во время создания.
- `security.mac.mls.revocation_enabled` используется для запрета доступа к объектам после того, как их метка изменится в меньшую сторону.
- `security.mac.mls.max_compartments` используется для установки максимального количества уровней отделов на объекты; обычно это максимальное количество отделов, разрешенных в системе.

Для управления метками MLS существует команда setfmac(8). Для присвоения метки объекту, выполните следующую команду:

```
# setfmac mls/5 test
```

Для получения метки MLS файла test, выполните следующую команду:

```
# getfmac test
```

Выше представлен краткий обзор возможностей политики MLS. Существует метод, связанный с созданием основного файла политики в каталоге /etc, где будет определена необходимая для политики MLS информация, которая будет передана команде `setfmac`. Этот метод будет описан после рассмотрения всех политик.

Итоги: объект с низким уровнем доступа не может прочесть данные объекта с высоким уровнем доступа. Базовая политика должна устанавливать `mls/high` на всем, что не должно быть прочитано, даже если туда необходимо записывать. На всем, куда нельзя писать, должна быть установлена метка `mls/low`, даже если это необходимо читать. Наконец, на всем остальном установите `mls/equal`. Все пользователи, помеченные как `insecure`, должны иметь метку `mls/low`.

14.12. Модуль MAC Biba

Имя модуля: mac_biba.ko

Строка конфигурации ядра: `options MAC_BIBA`

Параметр загрузки: `mac_biba_load="YES"`

Модуль mac_biba(4) загружает MAC политику Biba. Эта политика работает в основном так же, как и MLS, за исключением того, что правила потока информации изменены на противоположные. Они предназначены для предотвращения передачи потока секретной

информации вверх, в то время как политика MLS предотвращает передачу потока секретной информации вниз; таким образом, большая часть этого раздела применима к обеим политикам.

В среде Biba, каждому субъекту или объекту присваивается метка "целостности". Эти метки состоят из иерархических уровней и не-иерархических компонентов. При возрастании уровня объекта или субъекта это повышает его целостность.

Поддерживаемые метки `biba/low`, `biba/equal`, и `biba/high`; описаны ниже:

- Метка `biba/low` обеспечивает наименьшую целостность объекта или субъекта. Установка ее на объект или субъект заблокирует их доступ к объектам или субъектам, имеющим более высокую метку. Тем не менее, у них остается доступ на чтение.
- Метка `biba/equal` должна помещаться только на объекты, исключающиеся из политики.
- Метка `biba/high` разрешит запись в объекты с более низкой меткой, но не разрешит чтение из этих объектов. Рекомендуется помещать такую метку на объекты, влияющие на целостность всей системы.

Biba представляет собой:

- Иерархические уровни целостности с набором не иерархических категорий;
- Фиксированные правила: нет записи наверх, нет чтения снизу (обратно MLS). Субъект может иметь доступ на запись к объектам своего уровня или ниже, но не выше. Аналогично, субъект может иметь доступ на чтение к объектам своего уровня или выше, но не ниже;
- Целостность (предотвращение неавторизованного изменения данных);
- Уровни целостности (вместо уровней секретности MLS).

Для управления политикой Biba могут быть использованы следующие переменные `sysctl`:

- `security.mac.biba.enabled` может использоваться для включения/выключения политики Biba.
- `security.mac.biba.ptys_equal` может использоваться для отключения политики Biba на устройствах pty(4).
- `security.mac.biba.revocation_enabled` включит отмену доступа к объектам, если метка изменена на более высокую, чем у субъекта.

Для выполнения настроек политики Biba на системных объектах, применяются команды `setfmac` и `getfmac`:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

Итоги: субъект с низким уровнем целостности не может писать в субъект с высоким уровнем целостности; субъект с высоким уровнем целостности не может читать из

субъекта с низким уровнем целостности.

14.13. Модуль MAC LOMAC

Имя модуля: mac_lomac.ko

Строка конфигурации ядра: `options MAC_LOMAC`

Параметр загрузки: `mac_lomac_load="YES"`

В отличие от политики MAC Biba, политика mac_lomac(4) разрешает доступ к объектам с более низким уровнем целостности только после уменьшения уровня целостности, чтобы не нарушать каких-либо правил целостности.

MAC версия политики целостности Low-watermark, чтобы не пересекаться со старой реализацией lomac(4), работает почти так же, как и Biba, за исключением использования плавающих меток для поддержки понижения метки субъекта через отдел для вспомогательной градации (auxiliary grade compartment). Этот вспомогательный отдел принимает вид `[auxgrade]`. При включении политики lomac с вспомогательной градацией метка должна выглядеть приблизительно так: `lomac/10[2]`, где номер 2 это вспомогательная градация.

Политика MAC LOMAC основана на тотальной пометке всех системных объектов метками целостности, разрешая субъектам читать из объектов с более низкой степенью целостности и с уменьшением метки субъекта для предотвращения последующей записи в объекты с более высокой степенью целостности. Параметр `[auxgrade]` обсуждался выше, таким образом политика может быть более совместимой и требовать меньшей первоначальной настройки, чем Biba.

14.13.1. Примеры

Как и для политик Biba и MLS, для установки меток на системные объекты и субъекты могут быть использованы утилиты `setfmac` и `setpmac`:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Обратите внимание, что вспомогательная градация здесь `low`, эта возможность предоставляется только политикой MAC LOMAC policy.

14.14. Реализация защищенной среды с MAC

Нижеследующая демонстрация реализует защищенную среду с использованием различных MAC модулей с соответственно настроенными политиками. Используйте этот пример только для тестирования, он не предназначен для удовлетворения всех требований к защите. Реализация этих политик без понимания принципа их работы неприменима в реальных задачах.

Перед началом процесса настройки, на каждую файловую систему необходимо установить параметр `multilabel`, который упоминался в начале этой главы. Невыполнение этого требования приведет к ошибкам.

14.14.1. Создание insecure класса пользователя

Начните процедуру добавлением следующего класса пользователя к файлу /etc/login.conf:

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
:manpath=/usr/shared/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=partition/13,mls/5:
```

и добавлением следующей строки к default классу пользователя:

```
:label=mls/equal,biba/equal,partition/equal:
```

После завершения этих действий, для пересборки базы данных должна быть выполнена следующая команда:

```
# cap_mkdb /etc/login.conf
```

14.14.2. Загрузка с необходимыми модулями

Добавьте к /boot/loader.conf следующие строки, чтобы необходимые модули были загружены при старте системы:

```
mac_biba_load="YES"
mac_mls_load="YES"
mac_seeotheruids_load="YES"
mac_partition_load="YES"
```

14.14.3. Установка всех пользователей в insecure

Всем учетным записям, кроме root или системных пользователей теперь потребуется присвоить класс (login class). При отсутствии класса пользователи не смогут получить доступа к обычным командам, таким как vi(1). Следующий скрипт sh сделает все необходимое:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
    /etc/passwd`; do pw usermod $x -L insecure; done;
```

После этого изменения необходимо запустить команду cap_mkdb на файле /etc/master.passwd.

14.14.4. Завершение настройки

Должен быть создан файл контекста; следующий пример взят из примера политики от Robert Watson, он может быть помещен в /etc/policy.contexts:

```
# This is the default BIBA/MLS policy for this system.

.*                              biba/high,mls/high
/sbin/dhclient                  biba/high(low),mls/high(low)
/dev(/.*)?                      biba/equal,mls/equal
# This is not an exhaustive list of all "privileged" devices.
/dev/mdctl                      biba/high,mls/high
/dev/pci                        biba/high,mls/high
/dev/k?mem                      biba/high,mls/high
/dev/io                         biba/high,mls/high
/dev/agp.*                      biba/high,mls/high
(/var)?/tmp(/.*)?               biba/equal,mls/equal
/tmp/\.X11-unix                 biba/high(equal),mls/high(equal)
/tmp/\.X11-unix/.*              biba/equal,mls/equal
/proc(/.*)?                     biba/equal,mls/equal
/mnt.*                          biba/low,mls/low
(/usr)?/home                    biba/high(low),mls/high(low)
(/usr)?/home/.*                 biba/low,mls/low
/var/mail(/.*)?                 biba/low,mls/low
/var/spool/mqueue(/.*)?         biba/low,mls/low
(/mnt)?/cdrom(/.*)?             biba/high,mls/high
(/usr)?/home/(ftp|samba)(/.*)?  biba/high,mls/high
/var/log/sendmail\.st           biba/low,mls/low
/var/run/utmp                   biba/equal,mls/equal
```

```
/var/log/(lastlog|wtmp)         biba/equal,mls/equal
```

Эта политика обеспечит безопасность путем применения ограничений на нисходящий и восходящий потоки информации в применении к каталогам и утилитам, приведенным в левой части файла.

Он может быть внесен в систему следующими командами:

```
# setfsmac -ef /etc/policy.contexts /
# setfsmac -ef /etc/policy.contexts /usr
```

Раскладка вышеприведенной файловой системы может быть различной для разных систем.

Файл /etc/mac.conf требует следующих изменений в основном разделе:

```
default_labels file ?biba,?mls
default_labels ifnet ?biba,?mls
default_labels process ?biba,?mls,?partition
default_labels socket ?biba,?mls
```

14.14.5. Тестирование настройки

Добавьте пользователя с помощью команды `adduser` и поместите его в класс `insecure` для этих тестов.

В примерах ниже тестирование `root` и обычных пользователей будет смешиваться; форма приглашения поможет различить этих пользователей.

14.14.5.1. Основное тестирование меток

```
% getpmac
biba/15(15-15),mls/15(15-15),partition/15
# setpmac partition/15,mls/equal top
```

Процесс top будет уничтожен перед тем, как мы запустим другой процесс top.

14.14.5.2. Тестирование MAC seeotheruids

```
% ps Zax
biba/15(15-15),mls/15(15-15),partition/15  1096 #C:  S      0:00.03 -su (bash)
biba/15(15-15),mls/15(15-15),partition/15  1101 #C:  R+     0:00.01 ps Zax
```

Просмотр процессов всех других пользователей должен быть запрещен.

14.14.5.3. Тестирование MAC partition

Отключите политику MAC `seeotheruids` для остальных тестов:

```
# sysctl security.mac.seeotheruids.enabled=0
% ps Zax
LABEL                                                PID  TT  STAT      TIME
COMMAND
  biba/equal(low-high),mls/equal(low-high),partition/15  1122 #C:  S+     0:00.02 top
  biba/15(15-15),mls/15(15-15),partition/15              1096 #C:  S      0:00.05 -su
(bash)
  biba/15(15-15),mls/15(15-15),partition/15              1123 #C:  R+     0:00.01 ps
Zax
```

Все пользователи должны видеть каждый процесс в своем разделе (partition).

14.14.5.4. Тестирование меток Biba и MLS

```
# setpmac partition/15,mls/equal,biba/high\(high-high\) top
% ps Zax
LABEL                                                PID  TT  STAT    TIME
COMMAND
  biba/high(high-high),mls/equal(low-high),partition/15   1251 #C:  S+     0:00.02 top
  biba/15(15-15),mls/15(15-15),partition/15               1096 #C:  S      0:00.06 -su
(bash)
  biba/15(15-15),mls/15(15-15),partition/15               1157 #C:  R+     0:00.00 ps
Zax
```

Политика Biba позволяет чтение объектов с более высокими метками.

```
# setpmac partition/15,mls/equal,biba/low top
% ps Zax
LABEL                                        PID  TT  STAT      TIME COMMAND
  biba/15(15-15),mls/15(15-15),partition/15  1096 #C:  S      0:00.07 -su (bash)
  biba/15(15-15),mls/15(15-15),partition/15  1226 #C:  R+     0:00.01 ps Zax
```

Политика Biba не позволяет чтение объектов с более низкими метками; тем не менее, MLS разрешает это.

```
% ifconfig bge0 | grep maclabel
maclabel biba/low(low-low),mls/low(low-low)
% ping -c 1 192.0.34.166
PING 192.0.34.166 (192.0.34.166): 56 data bytes
ping: sendto: Permission denied
```

Пользователи не могут выполнить ping на example.com, или на любой домен по этой причине.

Для устранения этой ошибки, запустите следующую команду:

```
# sysctl security.mac.biba.trust_all_interfaces=1
```

Она устанавливает метку интерфейса по умолчанию в незащищенный режим, так что политика Biba по умолчанию не будет применена.

```
# ifconfig bge0 maclabel biba/equal\(low-high\),mls/equal\(low-high\)
% ping -c 1 192.0.34.166
PING 192.0.34.166 (192.0.34.166): 56 data bytes
64 bytes from 192.0.34.166: icmp_seq=0 ttl=50 time=204.455 ms
--- 192.0.34.166 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 204.455/204.455/204.455/0.000 ms
```

Установив более корректную метку, мы можем использовать ping.

Теперь создадим файлы для процедуры тестирования чтения и записи:

```
# touch test1 test2 test3 test4 test5
# getfmac test1
test1: biba/equal,mls/equal
# setfmac biba/low test1 test2; setfmac biba/high test4 test5; \
  setfmac mls/low test1 test3; setfmac mls/high test2 test4
# setfmac mls/equal,biba/equal test3 && getfmac test?
test1: biba/low,mls/low
test2: biba/low,mls/high
test3: biba/equal,mls/equal
test4: biba/high,mls/high
test5: biba/high,mls/equal
# chown testuser:testuser test?
```

Все эти файлы должны принадлежать пользователю testuser. Тесты на чтение:

```
% ls
test1   test2   test3   test4   test5
% ls test?
ls: test1: Permission denied
ls: test2: Permission denied
ls: test4: Permission denied
test3   test5
```

Доступ на чтение не должен быть разрешен для пар: (biba/low,mls/low), (biba/low,mls/high) и

(biba/high,mls/high). Теперь несколько тестов на запись:

```
% for i in `echo test*`; do echo 1 > $i; done
-su: test1: Permission denied
-su: test4: Permission denied
-su: test5: Permission denied
```

Подобно тестам на чтение, доступ на запись должен быть запрещен для пар: (biba/low,mls/high) и (biba/equal,mls/equal).

```
% cat test?
cat: test1: Permission denied
cat: test2: Permission denied
1
cat: test4: Permission denied
```

А теперь от root:

```
# cat test2
1
```

14.15. Другой пример: Использование MAC для защиты веб сервера

Будет создано отдельное хранилище для веб данных, к которому пользователи должны иметь доступ. Это позволит biba/high управлять доступом к веб данным.

Начните с создания каталога для хранения веб данных:

```
# mkdir /usr/home/cvs
```

Теперь инициализируйте его командой cvs:

```
# cvs -d /usr/home/cvs init
```

Для начала необходимо включить политику biba, добавив mac_biba_enable="YES" в /boot/loader.conf. Предполагается, что ядро скомпилировано с поддержкой MAC.

Далее установите метку biba/high для всей системы по умолчанию.

В файл login.conf, класс default, необходимо внести следующие изменения:

```
:ignoretime@:\
```

```
    :umask=022:\
    :label=biba/high:
```

Каждого пользователя необходимо поместить в класс по умолчанию; такая команда:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
    /etc/passwd`; do pw usermod $x -L default; done;
```

быстро решит эту задачу.

Теперь создадим другой класс, web, копию класса default с меткой, установленной в `biba/low`.

Создайте пользователя для работы с основными веб данными, хранящимися в репозитории cvs. Этого пользователя необходимо поместить в новый класс, `web`.

Поскольку метка по умолчанию `biba/high`, на репозитории она будет той же. Веб данные должны иметь ту же метку, чтобы у пользователей был доступ к ним на чтение/запись. Веб сервер должен иметь доступ к тем же данным, к которым есть доступ у пользователей с меткой `biba/high`, для этого необходимо понизить метку данных.

Все, что потребуется, это следующий sh(1) скрипт, который может быть запущен из cron(8):

```
PATH=/bin:/usr/bin:/usr/local/bin; export PATH;
CVSROOT=/home/repo; export CVSROOT;
cd /home/web;
cvs -qR checkout -P htdocs;
exit;
```

Во многих случаях в веб файлы `cvs` необходимо поместить теги Id.

Этот скрипт теперь может быть помещен в домашний каталог каталог пользователя `web`, необходимо также добавить следующую запись crontab(1):

```
# Выполнять checkout web данных под меткой biba/low каждые 12 часов:
0       */12       *       *       *       web    /home/web/checkout.sh
```

Эта запись будет извлекать HTML страницы каждые двенадцать часов.

Метод запуска веб сервера по умолчанию также должен быть изменен для запуска процесса с меткой `biba/low`. Это может быть сделано путем следующего изменения в скрипте /usr/local/etc/rc.d/apache.sh:

```
command="setpmac biba/low /usr/local/sbin/httpd"
```

Настройки Apache должны быть изменены для работы с политикой `biba/low`. В этом случае

необходимо указать для хранения лог файлов каталог с меткой `biba/low`, иначе будут возвращены ошибки `access denied`.

В этом примере необходимо указать в директиве `docroot` каталог /home/web/htdocs; или, Apache не сможет найти каталог с документами.

Необходимо также изменить другие параметры конфигурации, включая PID файл, `Scoreboardfile`, `DocumentRoot`, или любые другие настройки для каталогов, где необходим доступ на запись. При использовании `biba` будет запрещен доступ на запись во все каталоги сервера, на которых нет метки `biba/low`.

14.16. Решение проблем с инфраструктурой MAC

На стадии разработки несколько пользователей сообщали о проблемах при обычных настройках. Некоторые из этих проблем приведены ниже:

14.16.1. Параметр `multilabel` не может быть включен на /

Параметр `multilabel` не включается на моем корневом (/) разделе!

Похоже, что каждый пятидесятый пользователь сталкивается с этой проблемой; на самом деле, и у нас была эта проблема в первых настройках. Дальнейшие наблюдения за этой так называемой "ошибкой" привели меня к мнению, что это результат или некорректной документации, или неправильной интерпретации этой документации. Независимо от того, почему это случилось, для решения этой проблемы могут быть предприняты следующие шаги:

1. Отредактируйте /etc/fstab и установите для корневого раздела параметр только для чтения (`ro`).
2. Перегрузитесь в однопользовательский режим.
3. Запустите команду `tunefs -l enable` на /.
4. Перегрузите систему в нормальный режим.
5. Запустите `mount -urw/` и измените параметр `ro` обратно на `rw` в /etc/fstab; перегрузите систему опять.
6. Дважды проверьте вывод `mount`, чтобы убедиться, что параметр `multilabel` был установлен на корневой файловой системе.

14.16.2. Не могу запустить XFree86™ после MAC

После настройки системы безопасности MAC, я больше не могу запускать XFree86™!

Это может быть вызвано политикой MAC `partition` или путем неправильной установки меток одной из политик MAC. Для отладки попробуйте следующее:

1. Просмотрите сообщение об ошибке; если пользователь находится в классе `insecure`, проблема может быть в политике `partition`. Попробуйте установить класс пользователя обратно в `default` и пересобрать базу данных командой `cap_mkdb`. Если это не решит проблемы, попробуйте шаг два.
2. Дважды проверьте политики с метками. Убедитесь, что политики настроены правильно для рассматриваемого пользователя, приложения XFree86™, и устройств в /dev.
3. Если проблема не решена, отправьте сообщение об ошибке и описание вашей системы в список рассылки TrustedBSD, находящийся на веб сайте TrustedBSD или в Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD.

14.16.3. Error: _secure_path(3) cannot stat .login_conf

При попытке переключения от `root` на другого пользователя системы, появляется сообщение об ошибке `_secure_path: unable to state .login_conf`.

Это сообщение обычно показывается, когда у пользователя более высокая метка, чем у пользователя, которым он пытается стать. Например, у пользователя системы `joe` метка по умолчанию `biba/low`. Пользователь `root`, метка которого `biba/high`, не может просматривать домашний каталог пользователя `joe`. Это не зависит от того, использует ли пользователь `root` команду `su joe` или нет. В этом сценарии модель целостности Biba не позволит `root` просматривать объекты с низким уровнем целостности.

14.16.4. Пользователя `root` нет!

В нормальном или даже однопользовательском режиме `root` не обнаруживается. Команда `whoami` возвращает 0 (нуль) и `su` возвращает `who are you?`. Что можно сделать?

Это может произойти, если политика с метками была отключена, или через sysctl(8), или путем выгрузки модуля политики. Если политика была постоянно или временно отключена, базу данных login необходимо перенастроить. Дважды проверьте login.conf, чтобы убедиться, что все параметры `label` были удалены и пересоберите базу данных командой `cap_mkdb`.

Глава 15. Аудит событий безопасности

15.1. Краткий обзор

Операционная система FreeBSD включает в себя поддержку аудита событий безопасности. Аудит позволяет выполнять надежное, детальное и гибко настраиваемое протоколирование различных событий, связанных с безопасностью, включая входы в систему, изменения конфигурации, доступ к файлам и сети. Эти записи могут быть незаменимы для мониторинга функционирующей системы, обнаружения вторжений и для анализа событий, приведших к краху системы. В FreeBSD реализован опубликованный Sun™ интерфейс прикладного программирования (Application Programming Interface, API), называемый Basic Security Module (BSM), и формат файла, который совместим с реализациями аудита в Solaris™ и Mac OS® X.

В этой главе описывается процесс установки и конфигурирования системы аудита. В том числе, приводится разъяснение политик аудита, а также даются примеры конфигурационных файлов.

После прочтения этой главы вы будете знать:

- Что такое система аудита и как она работает.
- Как настроить аудит во FreeBSD для мониторинга пользователей и процессов.
- Как просматривать журнал аудита при помощи инструментов просмотра и фильтрации (reduction).

Перед прочтением этой главы вы должны:

- Понимать основы UNIX® и FreeBSD (Основы UNIX).
- Уметь конфигурировать и компилировать ядро (Настройка ядра FreeBSD).
- Понимать основные принципы безопасности в применении к операционной системе FreeBSD (Безопасность).

Реализация аудита имеет известные ограничения. Не все события в настоящий момент протоколируемые. Также, некоторые механизмы входа в систему, такие как оконные менеджеры X11 или демоны от сторонних производителей, не настраивают аудит пользовательских сессий должным образом.

Использование системы аудита может привести к генерированию изобилующих подробностями журнальных файлов. Их размер на загруженных серверах в некоторых конфигурациях может превышать несколько гигабайт в неделю. Администраторы должны принимать во внимание требования к дисковому пространству для нагруженных конфигураций системы аудита. Например, желательно выделить отдельный раздел для файловой системы аудита /var/audit, чтобы заполнение раздела аудита не влияло на другие файловые системы.

15.2. Ключевые понятия

Следующие термины относятся к аудиту событий безопасности:

- *событие* (event): событие, которое может быть занесено в журнал. Примерами событий, относящихся к безопасности системы, являются: создание файла, инициализацию сетевого соединения, вход пользователя в систему. События разделяются на "приписываемые" (attributable) - те, которые могут быть отнесены к конкретному пользователю - и "неприписываемые" (non-attributable). Пример неприписываемого события - любое событие, произошедшее до аутентификации пользователя, например, неверно набранный пароль.
- *класс* (class): именованные наборы однотипных событий, которые используются в выражениях выбора. Часто используемые классы событий включают "создание файла" (fc), "выполнение файла" (ex) и "события входа в систему и выхода из нее" (lo).
- *запись* (record): единичная запись в журнале, описывающая то или иное событие. Записи содержат информацию о типе события, информацию о субъекте события (пользователе), который выполнил некоторое действие, дату и время события, информацию об объектах и аргументах события, а также информацию об успешности или неуспешности выполнения операции.
- *журнал* (trail): файл, содержащий последовательность записей аудита, описывающих события безопасности (security events). Журнал содержит записи в ориентировочно хронологическом порядке по времени завершения события. Только авторизованные процессы могут добавлять записи в журнал.
- *выражение выбора* (selection expression): строка, содержащая список префиксов и имен классов, используемая для выбора группы событий.
- *предварительный выбор* (preselection): процесс, с помощью которого система определяет, какие события имеют важность для администратора. Предварительный выбор использует ряд выражений выбора, задающих какие именно классы событий и для какого пользователя необходимо вносить в журнал, а также - глобальные настройки, которые будут применяться как для авторизованных, так и для неавторизованных процессов.
- *фильтрация* (reduction): процесс, в результате которого записи из существующего журнала выделяются для хранения, распечатки или анализа. Также, это процесс, в результате которого нежелательные записи удаляются из журнала аудита. Используя фильтрацию, администраторы могут реализовывать различные политики хранения данных аудита. Например, детализированный журнал может храниться месяц, но после этого он может быть сокращен чтобы хранить только информацию о входе в систему и выходе из нее.

15.3. Настройка системы аудита

Пользовательская часть системы аудита входит в базовую систему FreeBSD, системная часть включена в ядро GENERIC, старт демона auditd(8) активируется включением следующей записи в /etc/rc.conf:

```
auditd_enable="YES"
```

Затем нужно запустить демон аудита:

```
# service auditd start
```

Пользователям, предпочитающим строить специализированное ядро, необходимо включить следующую запись в файл конфигурации ядра:

```
options AUDIT
```

15.3.1. Выражения выбора событий

Выражения выбора используются в нескольких местах конфигурации для отбора событий, подлежащих аудиту. Выражения содержат перечень классов событий, с которым сравнивается происшедшее событие. Выражения выбора рассматриваются слева направо, и два выражения объединяются добавлением первого выражения ко второму.

Классы событий системы аудита перечисляет имеющиеся по умолчанию записи:

Таблица 6. Классы событий системы аудита

Имя класса	Расшифровка	Действие
all	all	Соответствует всем классам событий.
aa	authentication and authorization	
ad	administrative	Аудит административных действий, произошедших в системе.
ap	application	События, определяемые каким-либо приложением.
cl	file close	Аудит вызовов системной функции `close`.
ex	exec	Аудит запуска приложения. Аудит аргументов командной строки и переменных окружения контролируется через audit_control(5) используя параметры `argv` и `envv` в опции `policy`.

Имя класса	Расшифровка	Действие
fa	file attribute access	Аудит доступа к атрибутам объектов, например таких как stat(1), pathconf(2).
fc	file create	Аудит событий, в результате которых создаются файлы.
fd	file delete	Аудит событий, в результате которых удаляются файлы.
fm	file attribute modify	Аудит событий, в результате которых изменяются атрибуты файлов, например, chown(8), chflags(1), flock(2).
fr	file read	Аудит событий, в результате которых происходит чтение данных или открываются файлы на чтение.
fw	file write	Аудит событий, в результате которых происходит запись данных, запись или изменение файлов.
io	ioctl	Аудит вызовов системной функции ioctl(2).
ip	ipc	Аудит различных видов взаимодействия процессов, включая создание неименованных каналов (POSIX pipe) и взаимодействие процессов в стиле System V IPC.
lo	login_logout	Аудит событий login(1) и logout(1).
na	non attributable	Аудит неприписываемых событий.
no	invalid class	Не соответствует никаким событиям аудита.
nt	network	Аудит событий, связанных с сетевыми подключениями, например connect(2) и accept(2).
ot	other	Аудит различных событий.
pc	process	Аудит действий процессов, таких как exec(3) и exit(3).

Эти классы событий могут быть настроены изменением конфигурационных файлов audit_class и audit_event.

Каждый класс аудита можно скомбинировать с префиксом, показывающим, какие операции будут учитываться - удачные или неудачные, а также то, включает ли данная запись аудит для данного класса и типа, либо отключает его. Префиксы классов аудита событий обобщает доступные префиксы:

Таблица 7. Префиксы классов аудита событий

Префикс	Действие
+	Аудит успешных событий в данном классе.
-	Аудит ошибочных событий в данном классе.
^	Отключение аудита как успешных, так и ошибочных событий в данном классе.
^+	Отключение аудита успешных событий в данном классе.
^-	Отключение аудита ошибочных событий в данном классе.

Если префикс не указан, то аудиту подлежат как успешные, так и неуспешные события.

Следующий пример выбирает успешные и неуспешные события входа в систему и выхода из нее, и только успешные события выполнения приложения:

```
lo,+ex
```

15.3.2. Конфигурационные файлы

В каталоге /etc/security находятся следующие конфигурационные файлы системы аудита:

- audit_class: содержит определения классов аудита.
- audit_control: контроллирует некоторые аспекты системы аудита, такие как классы по умолчанию, минимальное дисковое пространство, которое должно оставаться на разделе журнала аудита, максимальный размер журнала аудита. *

  audit_event: связывает идентификаторы событий (eventnum) с их текстовыми именами, описаниями и классами событий.

- audit_user: уточняет настройки аудита для конкретных пользователей; они комбинируются с глобальными настройками при входе пользователя в систему.
- audit_warn: настраиваемый скрипт командного интерпретатора, который вызывается auditd(8) для генерации предупреждений в исключительных ситуациях, таких как исчерпание дискового пространства записями аудита или при ротации журнала аудита.

Файлы конфигурации аудита должны редактироваться и изменяться с

> осторожностью, так как ошибки в конфигурации могут привести к сохранению бесполезных записей.

В большинстве случаев администратору придется вносить изменения только в два конфигурационных файла системы аудита: audit_control и audit_user. Первый из них содержит общие настройки системы аудита, второй может использоваться для уточнения настроек аудита для конкретных пользователей.

15.3.2.1. Файл audit_control

Ниже приведен перечень настроек по умолчанию, содержащихся в audit_control:

```
dir:/var/audit
dist:off
flags:lo,aa
minfree:5
naflags:lo,aa
policy:cnt,argv
filesz:2M
expire-after:10M
```

Запись `dir` используется для установки одного или более каталогов, в которых будет сохраняться журнал системы аудита. Если указан более чем один каталог, то указанные каталоги будут использоваться по очереди, по мере заполнения. Как правило, система аудита настраивается на хранение журнала аудита на отдельном разделе, чтобы предотвратить взаимное влияние подсистемы аудита и остальных подсистем в случае исчерпания свободного места на разделе.

Если опция `dist` имеет значение `on` или `yes`, то для всех журналов аудита будут создаваться жесткие ссылки, сохраняемые в /var/audit/dist.

Запись `flags` используется для установки глобальной маски предварительного выбора для приписываемых событий. В примере выше аудиту будут подвергаться как успешные, так и неудачные попытки входа в систему и выхода из нее, а также - аутентификация и авторизация для всех пользователей.

Запись `minfree` определяет минимальное количество свободного дискового пространства на разделе, в который сохраняются файлы журналов аудита.

Запись `naflags` определяет классы аудита для неприписываемых событий, например, процессов входа в систему и системных демонов.

Запись `policy` определяет разделяемый запятыми список флагов политики, определяющей различные аспекты поведения аудита. Флаг `cnt` указывает, что система должна продолжать работать, несмотря на ошибки аудита (данный флаг настоятельно рекомендуется). Второй флаг, `argv`, заставляет подвергать аудиту аргументы командной строки при вызове системного вызова execve(2).

Запись `filesz` определяет максимальный размер журнала событий аудита, по достижении

которого журнал будет автоматически закончен и подвергнут ротации. Значение `0` запрещает автоматическую ротацию логов. Если указанный размер ниже минимального значения 512K, то он будет проигнорирован, и будет сгенерировано предупреждающее сообщение в логах.

Поле `expire-after` определяет момент времени, при достижении которого журнальные файлы считаются неактуальными и удаляются.

15.3.2.2. Файл audit_user

Администратор может определить дополнительные требования к аудиту для конкретных пользователей в файле audit_user. Каждая строка позволяет уточнить настройки аудита для пользователя при помощи двух полей: `alwaysaudit` - определяющее набор событий, которые должны всегда подвергаться аудиту для данного пользователя, и `neveraudit` - перечисляющее набор событий, которые никогда не должны подвергаться аудиту для пользователя.

Нижеследующий пример настраивает аудит всех событий входа в систему, выхода из системы, а также аудит всех успешных выполнений команд для пользователя `root`, а также - аудит всех событий, связанных с созданием файлов и успешным выполнением команд пользователем `www`. С настройками по умолчанию в audit_control запись `lo` для `root` является избыточной, кроме того, события входа в систему и выхода из системы будут подвергаться аудиту и для пользователя `www`.

```
root:lo,+ex:no
www:fc,+ex:no
```

15.4. Работа с журналами аудита

Так как журнал аудита хранится в бинарном формате BSM, то для его изменения или перевода в текстовый формат предоставляются встроенные утилиты. Утилита `praudit` преобразует журнал аудита в текстовый формат. Утилита `auditreduce` применяется для фильтрации журнальных записей с целью анализа, архивирования или распечатки. Последняя утилита поддерживает разнообразие параметров, позволяющих выбирать записи по типу события, по классу события, по пользователю, по дате или времени события, по пути к файлу или по объекту, над которым производилось действие.

Например, для отображения всего содержимого журнала аудита в текстовом формате выполните:

```
# praudit /var/audit/AUDITFILE
```

В данном примере *AUDITFILE* - журнал, который будет выведен в текстовом формате.

Журнал аудита состоит из серии записей, которые, в свою очередь состоят из элементов, которые команда `praudit` выводит последовательно - по одному на строку. Каждый элемент имеет определенный тип, например `header` (содержит заголовок записи) или `path` (полный

путь к файлу). Следующий пример показывает запись для события `execve`:

```
header,133,10,execve(2),0,Mon Sep 25 15:58:03 2006, + 384 msec
exec arg,finger,doug
path,/usr/bin/finger
attribute,555,root,wheel,90,24918,104944
subject,robert,root,wheel,root,wheel,38439,38032,42086,128.232.9.100
return,success,0
trailer,133
```

Эта запись отражает результат успешного выполнения системного вызова `execve`, который стал результатом выполнения команды `finger doug`. В элементе записи `exec arg` есть командная строка, которую оболочка передала ядру. Элемент `path` содержит путь к исполняемому файлу в представлении ядра. Элемент `attribute` описывает исполняемый файл, а также права доступа файла. Элемент `subject` описывает ID аудируемого пользователя, исполняющие (effective) UID и GID, реальные ID пользователя и группы, идентификатор процесса, идентификатор сессии, порт и адрес, с которого был осуществлен вход в систему. Обратите внимание: идентификатор аудируемого пользователя и реальный идентификатор пользователя отличаются, так как пользователь `robert` повысил привилегии до пользователя `root` перед выполнением команды, но система аудита занесла его действия в журнал используя изначальный идентификатор. Элемент `return` описывает успешное выполнение операции, а элемент `trailer` завершает запись.

Указав аргумент `-x` можно получить вывод в формате XML.

Поскольку логи системы аудита могут иметь огромный размер, возможно выделить только часть записей при помощи `auditreduce`. В следующем примере из AUDITFILE выбираются все записи, касающиеся пользователя `trhodes`:

```
# auditreduce -u trhodes /var/audit/AUDITFILE | praudit
```

Члены группы `audit` имеют доступ на чтение к журналу аудита, находящемуся в /var/audit. По умолчанию эта группа пуста, и только `root` имеет к ним доступ. Для того, чтобы дать пользователю права на чтение журнала, его необходимо добавить в группу `audit`. Право на чтение журнала аудита позволяет получить множество информации о поведении пользователей и процессов, поэтому рекомендуется делегировать права на чтение журнала аудита с большой осторожностью.

15.4.1. Мониторинг системы в реальном времени с использованием потоков аудита

Потоки системы аудита - клонирующиеся псевдоустройства, позволяющие приложениям просматривать в реальном времени поток событий аудита. В первую очередь, это должно заинтересовать авторов программ определения вторжений и мониторинга системы. Тем не менее, для администратора поток системы аудита предоставляет возможность организовать наблюдение за системой, избежав проблем с правами доступа на журнал аудита или с прерыванием потока событий из-за ротации журнала. Для отслеживания

потока событий аудита в реальном времени, выполните:

```
# praudit /dev/auditpipe
```

По умолчанию, потоки доступны только пользователю `root`. Чтобы сделать их доступными членам группы `audit`, добавьте правило `devfs` в файл /etc/devfs.rules:

```
add path 'auditpipe*' mode 0440 group audit
```

Обратитесь к devfs.rules(5) за более полной информацией о настройке файловой системы `devfs`.

Довольно легко создать зацикленный поток событий аудита, в котором просмотр каждого события порождает несколько событий аудита. Например, если аудиту подвергаются все операции сетевого ввода-вывода, и команда `praudit` запущена во время SSH-сессии, то будет сгенерирован интенсивный поток сообщений аудита, так как каждое печатаемое событие вызовет еще одно событие. По этой причине рекомендуется запускать `praudit` на устройстве потока только из сессий, для которых нет детального аудита ввода-вывода.

15.4.2. Ротация и сжатие журнальных файлов аудита

Журнал аудита пишется ядром и управляется демоном аудита auditd(8). Администраторам не следует пытаться использовать newsyslog.conf(5) или другие инструменты для прямой ротации логов. Вместо этого, для прекращения аудита, реконфигурации и ротации журнальных файлов должна использоваться команда `audit`. Следующая команда приведет к созданию нового журнального файла и даст указание ядру переключиться на запись в этот файл. Протоколирование в старый файл будет прекращено, а сам файл - переименован, в результате чего с ним можно будет работать администратору:

```
# audit -n
```

Если auditd(8) не запущен, то эта команда окончится неудачей, и будет выведено сообщение об ошибке.

Добавление следующей строки в файл /etc/crontab приведет к ротации каждые двенадцать часов:

```
0     */12       *       *       *       root    /usr/sbin/audit -n
```

Изменения вступят в силу после сохранения файла /etc/crontab.

Автоматическая ротация журнальных файлов на основании их размера возможна при использовании опции `filesz` в файле audit_control, которая описана в Файл audit_control.

Поскольку журнальные файлы могут достигать очень больших размеров, может возникнуть необходимость сжимать их в целях хранения сразу же после закрытия их демоном аудита. Для выполнения определенных пользователем действий, соответствующих разнообразным событиям системы аудита, включая нормальное завершение журналов аудита при их ротации, может быть использован скрипт audit_warn. Например, добавление следующих строк в файл /etc/security/audit_warn приведет к сжатию файла аудита после его закрытия:

```
#
# Compress audit trail files on close.
#
if [ "$1" = closefile ]; then
    gzip -9 $2
fi
```

Примерами других действий могут быть копирование файлов аудита на централизованный сервер, удаление старых журнальных файлов, фильтрация журнальных файлов для удаления ненужных записей. Скрипт будет запущен только при корректном закрытии журнала системой аудита и не запустится для журнальных файлов, запись в которые была прекращена в результате некорректного завершения.

Глава 16. Устройства хранения

16.1. Краткий обзор

В этой главе описывается использование дисков во FreeBSD. К ним относятся диски в памяти, диски, подключенные по сети, обычные устройства хранения SCSI/IDE и устройства, использующие интерфейс USB.

После чтения этой главы вы будете знать:

- Терминологию, используемую во FreeBSD для описания организации данных на физическом диске (разделы и слайсы).
- Как добавить дополнительные винчестеры к вашей системе.
- Как настроить FreeBSD для использования дисковых устройств USB.
- Как настроить виртуальные файловые системы, такие, как диски в оперативной памяти.
- Как использовать квоты для ограничения использования дискового пространства.
- Как зашифровать диски, чтобы защитить их от взлома.
- Как создавать и записывать CD и DVD во FreeBSD.
- Различные варианты использования устройств хранения для резервных копий.
- Как использовать программы резервного копирования, имеющиеся для FreeBSD.
- Как выполнять резервное копирование на дискеты.
- Что такое мгновенные копии файловых систем и как их эффективно использовать

Перед прочтением этой главы вам потребуется:

- Узнать как настраивать и устанавливать новое ядро FreeBSD (Настройка ядра FreeBSD).

16.2. Имена устройств

Далее приводится список физических устройств хранения информации, которые поддерживаются во FreeBSD, и имена устройств, которые им соответствуют.

Таблица 8. Соглашения по именованию физических дисков

Тип диска	Имя дискового устройства
Винчестеры IDE	`ad`
Приводы IDE CDROM	`acd`
Винчестеры SCSI и дисковые устройства USB	`da`
Приводы SCSI CDROM	`cd`
Различные нестандартные приводы CDROM	`mcd` для Mitsumi CD-ROM, `scd` для Sony CD-ROM
Дискеты	`fd`

Тип диска	Имя дискового устройства
Ленточные приводы SCSI	`sa`
Ленточные приводы IDE	`ast`
Флэш-диски	`fla` для флэш-устройств DiskOnChip®
Диски RAID	`aacd` для Adaptec® AdvancedRAID, `mlxd` и `mlyd` для Mylex®, `amrd` для AMI MegaRAID®, `idad` для Compaq Smart RAID, `twed` для 3ware® RAID.

16.3. Добавление дисков

В этом разделе будет описан процесс добавления нового SCSI диска на машину, имеющую в данный момент только один диск. Сначала выключим компьютер и установим диск в компьютер согласно инструкциям к компьютеру, контроллеру и от производителя диска. Из-за большого разнообразия этих процедур их рассмотрение выходит за рамки этого документа..

Войдите в систему как пользователь `root`. После того, как вы установили диск, просмотрите файл /var/run/dmesg.boot, чтобы убедиться, что новый диск был найден. Продолжая наш пример, только что добавленный диск будет называться da1 и мы хотим смонтировать его в каталог /1 (если вы добавляете диск IDE, то устройство будет называться ad1).

FreeBSD работает на IBM-PC совместимых компьютерах, поэтому она должна уметь работать с разделами PC BIOS. Однако они отличаются от традиционных разделов BSD. Диск ПК может иметь до четырёх записей разделов BIOS. Если диск на самом деле будет использоваться исключительно под FreeBSD, вы можете использовать режим *dedicated*. В противном случае FreeBSD будет располагаться в одном из разделов PC BIOS. Во FreeBSD разделы PC BIOS называются *слайсами*, чтобы не путать их с традиционными разделами BSD. Вы также можете использовать слайсы и с диском, предназначенным исключительно для FreeBSD, однако используемым в компьютере, на котором имеется дополнительная операционная система. Это является хорошим способом избежать путаницы в утилите `fdisk` других операционных систем, не связанных с FreeBSD.

В случае слайсов диск будет добавлен как /dev/da1s1e. Это интерпретируется следующим образом: диск SCSI, устройство номер 1 (второй диск SCSI), слайс 1 (раздел PC BIOS 1), и раздел BSD e. В случае использования в выделенном режиме диск будет добавлен просто как /dev/da1e.

Вследствие использования 32-разрядных целых чисел для адресации секторов, bsdlabel(8) ограничен 2^32-1 секторами на диск, или 2TB в большинстве случаев. Формат fdisk(8) позволяет наличие первого сектора со смещением не более 2^32-1 и длину не более 2^32-1, что ограничивает размер раздела до 2TB, а размер диска до 4TB в большинстве случаев. Формат sunlabel(8) ограничен 2^32-1 секторами на раздел и 8 разделами, что составляет 16TB. Для дисков большего раздела могут быть использованы разделы gpt(8).

16.3.1. Использование утилиты sysinstall(8)

1. Использование Sysinstall

   Вы можете использовать простые меню утилиты `sysinstall` для разбиения на разделы и разметки нового диска. Войдите как пользователь `root` или воспользуйтесь командой `su`. Запустите команду `sysinstall` и войдите в меню `Configure`. Внутри `FreeBSD Configuration Menu`, пролистайте и выберите пункт `Fdisk`.

2. Редактор разделов fdisk

   При работе с утилитой fdisk нажатие `A` используется для выделения под FreeBSD полностью всего диска. Когда будет задан вопрос о том, хотите ли вы "сохранить совместимость с другими возможными операционными системами в будущем", ответьте `YES`. Запишите изменения на диск при помощи команды `W`. А теперь выйдите из редактора FDISK, нажав `Q`. В этот момент вам будет задан вопрос о "Master Boot Record" (главной загрузочной записи). Так как вы добавляете диск к уже работающей системе, выберите `None`.

3. Редактор метки диска

   Теперь вам нужно выйти из sysinstall и запустить эту утилиту снова. Следуйте указаниям выше, но на этот раз выберите пункт `Label`. Вы перейдёте к меню `Disk Label Editor`. Здесь вы создадите традиционные разделы BSD. На диске может быть до восьми разделов, имеющих метки `a-h`. Некоторые из меток разделов имеют особый смысл. Раздел `a` используется для размещения корневого раздела (/). По этой причине только ваш системный диск (например, тот, с которого происходит загрузка), должен иметь раздел `a`. Раздел `b` используется под раздел подкачки, и вы можете иметь много дисков с разделами подкачки. Раздел `c` используется для доступа ко всему диску в режиме эксклюзивного использования или ко всему слайсу FreeBSD при работе в режиме с использованием слайсов. Остальные разделы имеют обычное предназначение.

   Редактор метки диска программы sysinstall использует раздел `e` для некорневого раздела и не для раздела подкачки. Внутри редактора метки диска создайте отдельную файловую систему, нажав `C`. Когда будет задан вопрос о том, будет ли это раздел с файловой системой (FS) или это будет раздел подкачки, выберите `FS` и наберите точку монтирования (например, /mnt). При добавлении диска после установки системы, программа sysinstall не будет автоматически создавать записи в файле /etc/fstab, поэтому точка монтирования не так уж и важна.

   Теперь вы готовы записать новую метку на диск и создать на нем файловую систему. Сделайте это, нажав `W`. Проигнорируйте сообщения об ошибках от sysinstall о невозможности смонтировать новый раздел. Полностью выйдите из редактора метки диска и из программы sysinstall.

4. Завершение

   Последний шаг заключается в редактировании файла /etc/fstab и добавлении записи

для вашего нового диска.

16.3.2. Использовании утилит командной строки

16.3.2.1. Работа со слайсами

Следующая настройка позволит вашему диску корректно работать с другими операционными системами, которые могут быть установлены на вашем компьютере, и не вызовет конфликта с утилитами `fdisk` других операционных систем. Этот способ рекомендуется использовать для установок новых дисков. Используйте `эксклюзивный` режим, только если у вас есть реальные причины делать это!

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# fdisk -BI da1 # Инициализируем новый диск.
# bsdlabel -B -w da1s1 auto # Размечаем его.
# bsdlabel -e da1s1 # Редактируем только что созданную метку диска и добавляем
разделы.
# mkdir -p /1
# newfs /dev/da1s1e # Повторяем этот шаг для всех созданных разделов.
# mount /dev/da1s1e /1 # Монтируем раздел(ы)
# vi /etc/fstab # Добавляем соответствующую запись/записи в файл /etc/fstab.
```

Если у вас установлен диск IDE, подставьте ad вместо da.

16.3.2.2. Эксклюзивный режим

Если вы не будете использовать новый диск совместно с другой операционной системой, то вы можете использовать режим `эксклюзивного` использования. Отметьте, что этот режим может ввести в заблуждение операционные системы от Microsoft; однако информацию они не разрушат. А вот OS/2® компании IBM будет "забирать себе" любой раздел, который она найдет и не сможет распознать.

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# bsdlabel -Bw da1 auto
# bsdlabel -e da1                          # create the `e' partition
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab                             # add an entry for /dev/da1e
# mount /1
```

Альтернативный метод заключается в следующем:

```
# dd if=/dev/zero of=/dev/da1 count=2
# bsdlabel /dev/da1 | bsdlabel -BR da1 /dev/stdin
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab                             # add an entry for /dev/da1e
```

```
# mount /1
```

16.4. RAID

16.4.1. Программный RAID

16.4.1.1. Конфигурация драйвера объединённого диска (CCD)

При выборе решения для организации хранилища самыми важными характеристиками являются скорость, надежность и стоимость. Редко все эти характеристики наличествуют одновременно; обычно быстрое и надёжное устройство хранения стоит дорого, а при уменьшении стоимости в жертву приносятся скорость работы или надёжность.

При проектировании описываемой далее системы в качестве самого важного фактора была выбрана её стоимость, затем быстродействие и надёжность. Скорость передачи данных для этой системы ограничивалась только пропускной способностью сети. И, хотя надёжность очень важна, CCD-диск, описываемый ниже, обслуживал работу с данными, полные копии которых уже хранились на дисках CD-R, так они могли быть с лёгкостью обновлены.

При выборе решения для массового хранения данных первым шагом является определение ваших требований к нему. Если в ваших требованиях главными являются скорость или надёжность, а не стоимость, то ваш выбор будет отличаться от описываемой в этом разделе системы.

16.4.1.1.1. Установка оборудования

Кроме системного IDE-диска, основу описываемого далее CCD-диска общим объёмом примерно в 90 Гбайт составили три IDE-диска Western Digital 30GB, 5400 RPM. В идеальном случае каждый диск IDE имеет собственный контроллер и кабель, но для минимизации стоимости дополнительные контроллеры IDE не использовались. Вместо этого диски были настроены при помощи переключателей так, что на каждом IDE-контроллере находилось по одному ведущему и одному ведомому диску.

До перезагрузки BIOS системы была настроена на автоматическое распознавание подключенных дисков. Более важно то, что при перезагрузке их распознала FreeBSD:

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```

Если FreeBSD не распознала все диски, проверьте корректность положения переключателей на них. На большинстве IDE-дисков имеется также переключатель "Cable Select". Он *не имеет* отношения к выбору ведущего и ведомого устройств. Для получения помощи по правильному положению переключателей обратитесь к документации по устройствам.

16.4.1.1.2. Настройка CCD

Драйвер ccd(4) позволяет вам взять несколько идентичных дисков и объединить их в одну логическую файловую систему. Для использования ccd(4) нужно ядро со встроенной поддержкой ccd(4). Добавьте такую строку в файл конфигурации ядра, перестройте и установите новое ядро:

```
device   ccd
```

Поддержка ccd(4) также может быть обеспечена загрузкой подгружаемого модуля ядра.

Для настройки ccd(4) сначала вам нужно воспользоваться утилитой bsdlabel(8) для разметки дисков:

```
bsdlabel -w ad1 auto
bsdlabel -w ad2 auto
bsdlabel -w ad3 auto
```

При этом создаются метки для ad1c, ad2c и ad3c, которые занимают диск полностью.

Следующим шагом является изменение типа метки диска. Для редактирования дисков можно использовать утилиту bsdlabel(8):

```
bsdlabel -e ad1
bsdlabel -e ad2
bsdlabel -e ad3
```

При этом в редакторе, задаваемом переменной окружения `EDITOR` (обычно это vi(1)), открывается текущая метка каждого диска.

Не модифицированная метка диска будет выглядеть примерно следующим образом:

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    0 - 59597)
```

Добавьте новый раздел `e` для использования драйвером ccd(4). Как правило, он может быть скопирован с раздела `c`, но поле `fstype` *должно* иметь значение `4.2BSD`. Теперь метка диска должна выглядеть примерно так:

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    0 - 59597)
  e: 60074784        0    4.2BSD        0     0     0   # (Cyl.    0 - 59597)
```

16.4.1.1.3. Построение файловой системы

Теперь, когда все диски размечены, вы должны построить ccd(4). Для этого используйте утилиту ccdconfig(8) с параметрами, подобными следующим:

```
ccdconfig ccd0 32 0 /dev/ad1e /dev/ad2e /dev/ad3e
```

Использование и значение каждого параметра описывается ниже: * Первым аргументом является конфигурируемое устройство, в нашем случае /dev/ccd0c. Часть /dev/ является необязательной. * Чередование для файловой системы. Оно определяет размер единицы блока данных в количестве дисковых блоков, каждый из которых обычно имеет объём в 512 байт. Таким образом, при чередовании в 32 это будет составлять 16384 байт. * Опции для ccdconfig(8). Если вы хотите включить зеркалирование диска, то можете задать это здесь. В нашей конфигурации зеркалирование для ccd(4) не предусмотрено, поэтому здесь задан 0 (ноль). * Последним параметром для ccdconfig(8) является список устройств для объединения в массив. Для каждого устройства нужно задавать полное имя.

После запуска ccdconfig(8) устройство ccd(4) будет отконфигурировано. Может будет построить файловую систему. Обратитесь к справке по команде newfs(8) для выяснения требуемых параметров, или просто запустите:

```
newfs /dev/ccd0c
```

16.4.1.1.4. Автоматическое выполнение

Вообще говоря, вам потребуется монтировать ccd(4) при каждой перезагрузке. Для этого сначала вы должны отконфигурировать это устройство. Запишите вашу текущую конфигурацию в файл /etc/ccd.conf при помощи такой команды:

```
ccdconfig -g > /etc/ccd.conf
```

При перезагрузке скрипт `/etc/rc` запускает команду `ccdconfig -C`, если существует файл /etc/ccd.conf. При этом ccd(4) автоматически конфигурируется так, чтобы он мог быть смонтирован.

Если при загрузке вы входите в однопользовательский режим, то перед тем, как выполнять монтирование ccd(4) по команде mount(8), вам нужно для конфигурации массива запустить следующую команду:

```
ccdconfig -C
```

Для автоматического монтирования ccd(4) поместите запись о ccd(4) в файл /etc/fstab, чтобы он мог быть смонтирован во время загрузки системы:

```
/dev/ccd0c              /media       ufs     rw      2       2
```

16.4.2. Аппаратный RAID

FreeBSD поддерживает также целый ряд аппаратных контроллеров RAID. Эти устройства самостоятельно управляют RAID-подсистемой, без необходимости иметь специфичное для FreeBSD программное обеспечения управления массивом.

При помощи встроенной в адаптер BIOS, он сам управляет большинством дисковых операций. Далее следует краткое описание установки при помощи контроллера Promise IDERAID. После установки адаптера и запуска системы, выдаётся запрос на ввод. Следуйте указаниям для входа в настройку адаптера. Отсюда вы можете объединить все подключенные диски. После этого во FreeBSD диск(и) будут выглядеть как один диск. Аналогично могут быть настроены и другие уровни RAID.

16.4.3. Перестроение массивов ATA RAID1

FreeBSD позволяет вам выполнять горячую замену вышедшего из строя диска. При этом требуется, чтобы вы заметили это до перезагрузки.

Вероятно, в файле /var/log/messages или в выдаче команды dmesg(8) вы увидите примерно следующее:

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn 1107 tn 4 sn 11)\\
status=59 error=40
ar0: WARNING - mirror lost
```

При помощи atacontrol(8) получите дополнительную информацию:

```
# atacontrol list
ATA channel 0:
        Master:      no device present
        Slave:       acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
        Master:      no device present
        Slave:       no device present

ATA channel 2:
        Master:      ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
        Slave:       no device present

ATA channel 3:
```

```
        Master:      ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
        Slave:       no device present

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. Сначала вам нужно отключить канал контроллера ATA, содержащий отказавший диск, чтобы его можно было без последствий извлечь:

   ```
   # atacontrol detach ata3
   ```

2. Замените диск.
3. Повторно подключите канал дискового контроллера:

   ```
   # atacontrol attach ata3
   Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
   Slave:   no device present
   ```

4. Добавьте новый диск к массиву в качестве резервного:

   ```
   # atacontrol addspare ar0 ad6
   ```

5. Перестройте массив:

   ```
   # atacontrol rebuild ar0
   ```

6. Проверить состояние дел можно при помощи следующей команды:

   ```
   # dmesg | tail -10
   [выдача удалена]
   ad6: removed from configuration
   ad6: deleted from ar0 disk1
   ad6: inserted into ar0 disk1 as spare

   # atacontrol status ar0
   ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
   ```

7. Дождитесь завершения этой операции.

16.5. USB устройства хранения

Множество современных устройств хранения используют Universal Serial Bus (USB): жесткие диски, брелоки USB, CD-R приводы, и т.д. FreeBSD предоставляет поддержку этих устройств.

16.5.1. Настройка

Драйвер umass(4) предоставляет поддержку устройств хранения USB. Если вы используете GENERIC ядро, изменять что-либо в настройках не потребуется. Если вы используете настроенное ядро, убедитесь, что в файле настройки присутствуют следующие строки:

```
device scbus
device da
device pass
device uhci
device ohci
device ehci
device usb
device umass
```

Для доступа к устройствам хранения USB драйвер umass(4) использует подсистему SCSI, ваши устройства USB будут видны системе как SCSI устройства. В зависимости от чипсета USB на материнской плате, для включения поддержки USB 1.X вам потребуется только один из параметров `device uhci` или `device ohci`. Однако, наличие обоих этих параметров не помешает. Поддержка контроллеров USB 2.0 предоставляется драйвером ehci(4) (строка `device ehci`). Не забудьте скомпилировать и установить новое ядро после добавления каких-либо строк.

Если ваше USB устройство это пишущий привод CD-R или DVD, необходимо добавить в ядро SCSI CD-ROM драйвер, cd(4), следующей строкой:

```
device cd
```

Поскольку устройство записи видно как SCSI диск, драйвер atapicam(4) не должен использоваться в файле настройки.

16.5.2. Тестирование конфигурации

Конфигурация готова к тестированию, подключите устройство USB, и в буфере системных сообщений (dmesg(8)), диск должен отобразиться примерно так:

```
umass0: USB Solid state disk, rev 1.10/1.00, addr 2
GEOM: create disk da0 dp=0xc2d74850
da0 at umass-sim0 bus 0 target 0 lun 0
da0: <Generic Traveling Disk 1.11> Removable Direct Access SCSI-2 device
da0: 1.000MB/s transfers
```

```
da0: 126MB (258048 512 byte sectors: 64H 32S/T 126C)
```

Конечно, производитель, имя устройства (da0) и другие детали могут отличаться в зависимости от конфигурации.

Поскольку устройство USB видится как SCSI, команда `camcontrol` может быть использована для вывода списка устройств хранения USB, подключенных к системе:

```
# camcontrol devlist
<Generic Traveling Disk 1.11>      at scbus0 target 0 lun 0 (da0,pass0)
```

Если на диске есть файловая система, у вас должна быть возможность смонтировать ее. Добавление дисков поможет вам создать и отформатировать разделы на диске USB если потребуется.

> Ниже описанный механизм (`vfs.usermount`), допускающий монтирование случайных носителей пользователями, не являющимися доверенными, считается небезопасным. Большинство файловых систем во FreeBSD никак не ограждено от возможности *несанкционированного* монтирования устройств.

Чтобы это устройство мог смонтировать обычный пользователь, необходимо выполнить определенные действия. Для начала, необходимо дать обычным пользователям доступ к устройствам, создаваемым при подключении USB устройства. Решение состоит во включении всех пользователей данных устройств в группу `operator`. Это делается утилитой pw(8). Затем, когда устройства созданы, у группы `operator` должен быть доступ на чтение и запись для этих устройств. Это выполняется путем добавления следующих строк в /etc/devfs.rules:

```
[localrules=5]
add path 'da*' mode 0660 group operator
```

> Если к системе подключены SCSI диски, это должно быть сделано немного иначе. Так, если в системе уже есть диски с da0 по da2, вторая строка должна выглядеть так:
>
> ```
> add path 'da[3-9]*' mode 0660 group operator
> ```
>
> Это исключит уже существующие диски из группы `operator`.

Вам также потребуется включить набор правил devfs.rules(5) в файл /etc/rc.conf:

```
devfs_system_ruleset="localrules"
```

Затем, ядро необходимо настроить так, чтобы оно позволяло обычным пользователям монтировать файловые системы. Простейший способ сделать это - добавить в файл /etc/sysctl.conf следующую строку:

```
vfs.usermount=1
```

Этот параметр установится только после последующей перезагрузки. Для установки этой переменной можно также использовать sysctl(8).

Последний шаг - создание каталога, куда будет монтироваться файловая система. Каталог должен принадлежать пользователю, монтирующему файловую систему. Один из способов сделать это под пользователем `root` - создать каталог /mnt/username (замените *username* именем пользователя, а *usergroup* - именем главной группы пользователя):

```
# mkdir /mnt/username
# chown username:usergroup /mnt/username
```

Предположим, что USB брелок подключен, и появилось устройство /dev/da0s1. Поскольку эти устройства обычно поставляются форматированными с файловой системой FAT, их можно смонтировать так:

```
% mount -t msdosfs -o -m=644,-M=755 /dev/da0s1 /mnt/username
```

Если вы отключите устройство (диск должен быть сначала размонтирован), вы должны увидеть в буфере системных сообщений что-то подобное:

```
umass0: at uhub0 port 1 (addr 2) disconnected
(da0:umass-sim0:0:0:0): lost device
(da0:umass-sim0:0:0:0): removing device entry
GEOM: destroy disk da0 dp=0xc2d74850
umass0: detached
```

16.5.3. Дополнительная информация

Помимо разделов Добавление дисков и Монтирование и размонтирование файловых систем, также может быть полезно чтение различных страниц справочника: umass(4), camcontrol(8), и usbconfig(8) для FreeBSD 8.X или usbdevs(8) для более ранних версий FreeBSD.

16.6. Запись и использование оптических носителей (CD)

16.6.1. Введение

Компакт-диски (CD) имеют несколько особенностей, отличающих их от обычных дисков. Во-первых, на них невозможно производить запись. Они спроектированы с расчетом на то, что их можно читать последовательно без задержек на перемещение головки между дорожками. К тому же их гораздо проще переносить от системы к системе, чем носители близкого объема.

У CD имеются дорожки, но они представляют собой последовательность данных, читаемую последовательно, и не являются физической характеристикой диска. Для записи CD во FreeBSD вы готовите файлы данных, которые будут формировать дорожки на компакт-диске, а затем записываете дорожки на CD.

Файловая система ISO 9660 была разработана с учетом этих отличий, К сожалению, она унаследовала ограничения файловых систем, которые были тогда. К счастью, она дает механизм расширений, которые позволяют правильно записанным дискам обходить эти ограничения и при этом продолжать работать с системами, которые не поддерживают эти расширения.

Для создания файла данных, содержащего файловую систему ISO 9660, используется программа mkisofs(8), которая включена в порт sysutils/cdrtools. Она имеет опции, поддерживающие различные расширения, и описана ниже.

Какой инструмент использовать для записи CD, зависит от того, является ли ваше устройство для записи CD устройством ATAPI или каким-либо другим. С устройствами для записи стандарта ATAPI используется программа `burncd`, которая является частью комплекта поставки системы. С устройствами SCSI и USB нужно использовать `cdrecord` из порта sysutils/cdrtools. Утилиту `cdrecord` и другие инструменты для SCSI-приводов также можно использовать при работе с ATAPI-оборудованием через модуль ATAPI/CAM.

Если для записи CD вам нужна программа с графическим интерфейсом пользователя, взгляните на X-CD-Roast или K3b. Они доступны в виде пакетов или из портов sysutils/xcdroast и sysutils/k3b. Программам X-CD-Roast и K3b для работы с оборудованием ATAPI требуется модуль ATAPI/CAM.

16.6.2. mkisofs

Программа mkisofs(8), поставляемая с портом sysutils/cdrtools создаёт файловую систему ISO 9660, которая является образом дерева каталогов в пространстве имён файловой системы UNIX®. В самом простом случае она используется так:

```
# mkisofs -o imagefile.iso /path/to/tree
```

Эта команда создаст файл *imagefile.iso*, содержащий файловую систему ISO 9660, которая является копией дерева каталогов */path/to/tree*. Во время работы она будет преобразовывать имена файлов в имена, которые удовлетворяют ограничениям файловой системы ISO 9660, и исключит файлы, которые носят имена, неподходящие для файловой системы ISO.

Для того, чтобы обойти эти ограничения, имеется несколько опций. В частности, `-R` включает использование расширений Rock Ridge, распространенных в UNIX®-системах, с `-J` будут применены расширения Joliet, используемые в системах от Microsoft, а `-hfs` может использоваться для создания файловых систем HFS, используемых в Mac OS®.

Для CD, которые будут использоваться только с системами FreeBSD, может использоваться опция `-U`, отменяющая все ограничения на имена файлов. При использовании с опцией `-R` генерируется образ файловой системы, идентичный начальному дереву FreeBSD, хотя при этом стандарт ISO 9660 может нарушаться в нескольких местах.

Последней часто используемой опцией является `-b`. Она используется для указания загрузочного образа для использования при создании загрузочного CD в стандарте "El Torito". Этой опции указывается аргумент, который является маршрутом к загрузочному образу из корня дерева, записываемого на CD. По умолчанию, mkisofs(8) создает образ ISO в так называемом режиме "эмуляции флоппи-диска", и потому ожидает загрузочный образ размера строго 1200, 1440 или 2880 KB. Некоторые загрузчики, в том числе и тот, что используется на дистрибутивных дисках FreeBSD, не используют режим эмуляции; в этом случае должна использоваться опция `-no-emul-boot`. Так что, если /tmp/myboot содержит загрузочную систему FreeBSD с загрузочным образом в /tmp/myboot/boot/cdboot, вы можете создать образ файловой системы ISO 9660 в /tmp/bootable.iso следующим образом:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Сделав это, и имея в ядре отконфигурированное устройство md, вы можете смонтировать файловую систему, выполнив:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

В этот момент вы можете проверить, что /mnt и /tmp/myboot идентичны.

Имеется много других опций, которые можно использовать с программой mkisofs(8) для тонкой настройки её поведения. В частности: модификации в размещении ISO 9660 и создание дисков в форматах Joliet и HFS. Обратитесь к справочным страницам по mkisofs(8) для получения более подробной информации.

16.6.3. burncd

Если ваше устройство для записи CD соответствует стандарту ATAPI, то для записи ISO-образа на компакт-диск вы можете воспользоваться командой `burncd`. `burncd` входит в базовый комплект операционной системы и установлена как /usr/sbin/burncd. Использовать её очень просто, так как параметров у ней немного:

```
# burncd -f cddevice data imagefile.iso fixate
```

По этой команде файл *imagefile.iso* будет скопирован на *cddevice*. По умолчанию

используется устройство /dev/acd0. Для получения информации о параметрах, задающих скорость записи, выброс диска после записи и запись звуковых данных, обратитесь к burncd(8).

16.6.4. cdrecord

Если ваше устройство для записи CD не соответствует стандарту ATAPI, то для записи компакт-дисков вам нужно пользоваться программой `cdrecord`. `cdrecord` не входит в комплект поставки системы; вы должны установить её из порта sysutils/cdrtools или из соответствующего пакета. Изменения в системе могут приводить к тому, что откомпилированные версии этой программы работать не будут, или приводить к порче дисков. Поэтому вы должны при обновлении системы либо обновить порт, либо, если вы следуете -STABLE, обновить порт при появлении его новой версии.

Хотя `cdrecord` имеет много опций, в основном использовать её ещё проще, чем `burncd`. Запись образа ISO 9660 делается такой командой:

```
# cdrecord dev=device imagefile.iso
```

Тонким моментом при использовании `cdrecord` является определение правильного устройства `dev`. Чтобы задать параметр правильно, воспользуйтесь флагом `-scanbus` команды `cdrecord`, в результате чего может получиться примерно такой результат:

```
# cdrecord -scanbus
Cdrecord-Clone 2.01 (i386-unknown-freebsd7.0) Copyright (C) 1995-2004 J"org Schilling
Using libscg version 'schily-0.1'
scsibus0:
    0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
    0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
    0,2,0     2) *
    0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable Disk
    0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
    0,5,0     5) *
    0,6,0     6) *
    0,7,0     7) *
scsibus1:
    1,0,0   100) *
    1,1,0   101) *
    1,2,0   102) *
    1,3,0   103) *
    1,4,0   104) *
    1,5,0   105) 'YAMAHA  ' 'CRW4260         ' '1.0q' Removable CD-ROM
    1,6,0   106) 'ARTEC   ' 'AM12S           ' '1.06' Scanner
    1,7,0   107) *
```

Здесь приведены соответствующие значения параметров `dev` для имеющихся устройств. Найдите здесь ваше устройство для записи CD, а в качестве параметров для `dev` задавайте три числа через запятые. В нашем случае CRW-устройству соответствуют числа 1,5,0, так что

правильным параметром будет `dev=1,5,0`. Имеется более простой способ задать эти значения; обратитесь к справочной информации о cdrecord(1) для выяснения подробностей. Там же находится информация о записи звуковых дорожек, управлении скоростью и другим вещам.

16.6.5. Копирование аудио CD

Вы можете копировать музыкальные CD, извлекая данные аудио с CD в набор файлов, а затем записывая эти файлы на чистый CD. Процесс несколько различен в случаях использования устройств ATAPI и SCSI.

Procedure: Устройства SCSI

1. Используйте `cdda2wav` для извлечения данных аудио.

```
% cdda2wav -vall -D2,0 -B -Owav
```

2. Воспользуйтесь `cdrecord` для записи файлов .wav.

```
% cdrecord -v dev=2,0 -dao -useinfo  *.wav
```

   Значение, соответствующее *2,0*, должно быть установлено правильно, как это описано в cdrecord.

Procedure: Устройства ATAPI

> На приводах ATAPI также можно использовать утилиту `cdda2wav`. Для её функционирования потребуется драйвер ATAPI/CAM. Следует отметить, что данная утилита предназначена для корректного извлечения и обработки аудио данных, в отличие от утилиты, приведенной в нижеследующем примере.

1. Драйвер устройств ATAPI CD делает каждую дорожку доступной как /dev/acddtnn, где *d* является номером привода, а *nn* соответствует номеру дорожки, который записывается двумя десятичными цифрами с нулём в начале, если это нужно. Таким образом, первая дорожка на первом диске будет носить имя /dev/acd0t01, вторая будет именоваться /dev/acd0t02, третья будет носить имя /dev/acd0t03 и так далее.

   Удостоверьтесь, что соответствующий файл имеется в каталоге /dev. При его отсутствии следует принудительно перечитать оглавление диска:

```
# dd if=/dev/acd0 of=/dev/null count=1
```

2. Извлеките каждую дорожку при помощи команды dd(1). При извлечении файлов вы должны также использовать специфическое значение для размера блока.

```
# dd if=/dev/acd0t01 of=track1.cdr bs=2352
# dd if=/dev/acd0t02 of=track2.cdr bs=2352
...
```

3. Запишите извлечённые файлы на диск при помощи утилиты `burncd`. Вы должны указать, что это файлы с аудио, и что `burncd` должна зафиксировать диск по окончании работы.

```
# burncd -f /dev/acd0 audio track1.cdr track2.cdr ... fixate
```

16.6.6. Копирование компакт-дисков с данными

Вы можете скопировать CD с данными в файл образа, который функционально эквивалентен файлу образа, созданному командой mkisofs(8), и вы можете использовать его для копирования любого CD с данными. В приводимом здесь примере предполагается, что ваш привод CDROM называется acd0. Подставьте название вашего привода CDROM.

```
# dd if=/dev/acd0 of=file.iso bs=2048
```

Теперь, когда вы имеете образ, вы можете записать его на CD так, как это описано выше.

16.6.7. Использование компакт-диски с данными

Теперь, после того, как вы создали стандартный CDROM с данными, вы, наверное, захотите смонтировать его и считать с него данные. По умолчанию mount(8) предполагает, что файловая система имеет тип `ufs`. Если вы попытаетесь выполнить что-то вроде:

```
# mount /dev/cd0 /mnt
```

вы получите сообщение `Incorrect super block`, и диск не смонтируется. CDROM не является файловой системой `UFS`, поэтому попытки смонтировать его таким образом будут терпеть неудачу. Вам просто нужно указать команде mount(8), что файловая система имеет тип `ISO9660`, и всё должно заработать. Сделайте это, задав параметр `-t cd9660` при вызове mount(8). К примеру, если вы хотите смонтировать устройство CDROM, /dev/cd0, в каталог /mnt, вы должны выполнить:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Заметьте, что имя вашего устройства (/dev/cd0 в этом примере) может быть другим, в зависимости от интерфейса, используемого в CDROM. Кроме того, параметр `-t cd9660` всего

лишь задаёт выполнение утилиты mount_cd9660(8). Пример выше может быть упрощён до:

```
# mount_cd9660 /dev/cd0c /mnt
```

Таким способом, вообще говоря, вы можете использовать компакт-диски любого производителя. Диски с некоторыми расширениями ISO 9660 могут, однако, работать со странностями. К примеру диски Joliet хранят все имена файлов в виде последовательностей двухбайтовых символов Unicode. Ядро FreeBSD не может работать с Unicode, но CD9660 драйвер способен преобразовывать Unicode символы на лету. Если некоторые символы не английского алфавита выглядят, как знаки вопроса, то вам нужно указать используемую вами кодировку с помощью опции `-C`. За дополнительной информацией, обращайтесь к странице справочника mount_cd9660(8).

Чтобы смочь произвести преобразование символов посредством опции `-C`, ядру понадобится загрузить модуль cd9660_iconv.ko. Это может быть сделано либо добавлением ниже представленной строчки в loader.conf:

```
cd9660_iconv_load="YES"
```

с последующей перезагрузкой машины, либо загрузкой модуля вручную с помощью kldload(8).

Время от времени вы можете получать сообщения `Device not configured` при попытке смонтировать CDROM. Это обычно означает, что привод CDROM полагает, что в нём нет диска, или что привод не виден на шине. Приводу CDROM может понадобиться несколько секунд, чтобы понять, что он был закрыт, так что будьте терпеливы.

Иногда SCSI CDROM может потеряться из-за того, что у него не было достаточно времени, чтобы ответить на сброс шины. Если у вас имеется SCSI CDROM, то, пожалуйста, добавьте следующий параметр в конфигурацию вашего ядра и перестройте его.

```
options SCSI_DELAY=15000
```

Это укажет вашей шине SCSI выдерживать 15-секундную паузу во время загрузки, чтобы дать вашему приводу CDROM шанс ответить на сброс шины.

16.6.8. Запись необработанных данных на компакт-диски

Вы можете предпочесть запись файла непосредственно на CD без создания файловой системы ISO 9660. Некоторые поступают так при создании резервных копий. Это выполняется гораздо быстрее. чем запись стандартного компакт-диска:

```
# burncd -f /dev/acd1 -s 12 data archive.tar.gz fixate
```

Для извлечения данных, записанных так на компакт-диск, вы должны считывать данные

из файла непосредственного доступа к устройству:

```
# tar xzvf /dev/acd1
```

Вы не можете монтировать этот диск как обычный CDROM. Такой компакт-диск не может быть прочитан ни в какой другой операционной системе, кроме FreeBSD. Если вы хотите монтировать CD или обменяться данными с другой операционной системой, то вы должны использовать mkisofs(8) так, как это было описано выше.

16.6.9. Использование драйвера ATAPI/CAM

Этот драйвер позволяет работать с ATAPI-устройствами (приводы CD-ROM, CD-RW, DVD и так далее) через подсистему SCSI, таким образом расширяя использование таких приложений, как sysutils/cdrdao или cdrecord(1).

Для использования этого драйвера вам необходимо добавить в файл /boot/loader.conf следующую строку:

```
atapicam_load="YES"
```

с последующей перезагрузкой машины.

Если для вас предпочтительнее статически скомпилировать поддержку atapicam(4) в ядро, то добавьте эту строчку в файл конфигурации ядра:

```
device atapicam
```

Кроме того, в файле конфигурации ядра должны быть следующие строки:

```
device ata
device scbus
device cd
device pass
```

которые уже должны там присутствовать. Затем пересоберите, установите новое ядро и перезагрузите компьютер.

В процессе загрузки ваш пишущий привод должен появиться примерно следующим образом:

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at ata1-master PIO4
cd0 at ata1 bus 0 target 0 lun 0
cd0: <MATSHITA CDRW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
```

```
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

Теперь с ним можно работать через устройство /dev/cd0, например, чтобы смонтировать CD-ROM в каталог /mnt, просто наберите следующую команду:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Для получения SCSI-адреса пишущего привода, вы можете, работая как пользователь `root`, запустить такую команду:

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00>   at scbus1 target 0 lun 0 (pass0,cd0)
```

Таким образом, `1,0,0` будет SCSI-адресом для использования с cdrecord(1) и другими приложениями для работы со SCSI.

Для получения дополнительной информации об ATAPI/CAM и системе SCSI, обратитесь к страницам справочной системы по atapicam(4) и cam(4).

16.7. Создание и использование оптических носителей (DVD)

16.7.1. Введение

DVD это следующее после CD поколение оптических носителей. DVD может вмещать больше данных чем любой CD и является современным стандартом распространения видео.

Для записываемых DVD существует пять физических форматов записи:

- DVD-R: Был первым форматом записываемых DVD. Стандарт DVD-R был создан DVD Forum. Это формат для однократной записи.
- DVD-RW: Это перезаписываемая версия стандарта DVD-R. Носители DVD-RW могут быть перезаписаны около 1000 раз.
- DVD-RAM: Это также перезаписываемый формат, поддерживаемый DVD Forum. DVD-RAM может быть виден как съемный жесткий диск. Однако, этот носитель не совместим с большинством приводов DVD-ROM и проигрывателями DVD-Video; лишь некоторые пишущие DVD поддерживают формат DVD-RAM. Более подробно о работе с DVD-RAM можно прочитать в разделе Использование DVD-RAM.
- DVD+RW: Это перезаписываемый формат, созданный DVD+RW Alliance. Носитель DVD+RW может быть перезаписан около 1000 раз.
- DVD+R: Этот формат - однократно записываемая версия формата DVD+RW.

Однослойный записываемый DVD может хранить до 4,700,000,000 байт, что равно 4.38 Гбайт, или 4485 Мбайт (1 килобайт это 1024 байт).

Необходимо различать физический носитель и приложение. Например, DVD-Video это определенная файловая раскладка, которая может быть помещена на записываемый DVD любого физического формата: DVD-R, DVD+R, DVD-RW и т.д. Перед выбором типа носителя вы должны убедиться, что и устройство записи и DVD-Video проигрыватель (отдельный или DVD-ROM привод компьютера) совместимы с данным носителем.

16.7.2. Настройка

Для записи DVD будет использоваться программа growisofs(1). Эта команда входит в набор утилит dvd+rw-tools (sysutils/dvd+rw-tools), который поддерживает все типы носителей DVD.

Эти утилиты используют подсистему SCSI для доступа к устройствам, следовательно необходимо добавить в ядро поддержку ATAPI/CAM. Если пишущий привод использует USB интерфейс, это добавление бесполезно и необходимо прочесть более подробную информацию по настройке устройств USB в USB устройства хранения

Вам также потребуется включить DMA доступ для устройств ATAPI, это можно сделать добавив в /boot/loader.conf следующую строку:

```
hw.ata.atapi_dma="1"
```

Перед использованием dvd+rw-tools вы должны свериться со списком совместимого оборудования dvd+rw-tools с информацией по устройствам для записи DVD.

Если вам нужен графический интерфейс пользователя, взгляните на K3b (sysutils/k3b), который предоставляет дружественный пользователю интерфейс к growisofs(1) и многим другим программам записи.

16.7.3. Запись DVD с данными

Команда growisofs(1) является оболочкой для mkisofs, она вызовет mkisofs(8) для создания файловой системы и запишет DVD. Это означает, что вам не потребуется создавать образ с данными перед началом процесса записи.

Для записи данных из каталога /path/to/data на DVD+R или DVD-R, используйте следующую команду:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

Параметры `-J -R` передаются mkisofs(8) для создания файловой системы (в данном случае: файловая система ISO 9660 с расширениями Joliet и Rock Ridge), обратитесь к странице справочника mkisofs(8) за более подробной информацией.

Параметр `-Z` используется для первой сессии записи в любом случае: для одной или нескольких сессий. Устройство DVD, */dev/cd0*, должно быть изменено в соответствии с

имеющимися настройками. Параметр `-dvd-compat` закроет диск и дозапись станет невозможна. Это должно улучшить совместимость с приводами DVD-ROM.

Возможна также запись предварительного (pre-mastered) образа, например, для записи *imagefile.iso* запустим:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

Скорость записи должна быть определена и автоматически установлена в соответствии с носителем и приводом. Если вы хотите явно указать скорость записи, используйте параметр `-speed=`. За дальнейшей информацией обратитесь к странице справочника growisofs(1).

Если размер файлов внутри набора превышает 4.38Гб, то необходимо будет создать гибридную файловую систему UDF/ISO-9660, для чего потребуется передать параметры `-udf -iso-level 3` в mkisofs(8) и в остальные соответствующие программы (например, growisofs(1)). Указание параметров обязательно лишь во время создания файла образа или во время непосредственной записи на диск. Созданный таким способом диск должен монтироваться утилитой mount_udf(8). Диск будет доступен лишь тем операционным системам, которые поддерживают UDF; в противном случае носитель будет отображаться как поврежденный.

Для того, чтобы создать такой образ, выполните:

```
% mkisofs -R -J -udf -iso-level 3 -o imagefile.iso /path/to/data
```

Для того, чтобы записать файлы прямо на диск, наберите:

```
# growisofs -dvd-compat -udf -iso-level 3 -Z /dev/cd0 -J -R
/path/to/data
```

Если у вас в распоряжении уже имеется образ, содержащий в себе файлы большого размера, то для записи образа на диск никаких дополнительных опций для growisofs(1) не потребуется.

Также удостоверьтесь, что у вас установлена последняя версия sysutils/cdrtools (mkisofs(8) принадлежит к этому порту), поскольку предыдущие версии утилит не поддерживают работу с большими файлами. Если с этим портом возникают проблемы, то установите sysutils/cdrtools-devel и прочитайте страницу справочника mkisofs(8).

16.7.4. Запись DVD-Video

DVD-Video это особая файловая система, базирующаяся на ISO 9660 и спецификациях micro-UDF (M-UDF). DVD-Video также представляет определенную иерархию структуры данных,

поэтому для создания DVD потребуется особая программа, такая как multimedia/dvdauthor.

Если у вас уже есть образ файловой системы DVD-Video, просто запишите его как любой другой образ, примеры находятся в предыдущем разделе. Если вы создали DVD и результат находится в каталоге /path/to/video, для записи DVD-Video должна быть использована следующая команда:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

Параметр `-dvd-video` будет передан mkisofs(8) и укажет создать файловую систему DVD-Video. Помимо этого, параметр `-dvd-video` подразумевает параметр growisofs(1) `-dvd-compat`.

16.7.5. Использование DVD+RW

В отличие от CD-RW, новый DVD+RW необходимо отформатировать перед первым использованием. Программа growisofs(1) позаботится об этом сама при необходимости, и это *рекомендованный* способ. Тем не менее, для форматирования DVD+RW вы можете использовать команду `dvd+rw-format`:

```
# dvd+rw-format /dev/cd0
```

Эту операцию необходимо выполнить лишь однажды, помните, что только новые носители DVD+RW необходимо форматировать. Затем запишите DVD+RW тем способом, который описан в предыдущем разделе.

Если вы хотите записать новые данные (полностью новую файловую систему, а не дописать данные) на DVD+RW, его не нужно очищать, просто запишите поверх предыдущей записи (создав новую начальную сессию) примерно так :

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

Формат DVD+RW делает возможным легко дописать данные к предыдущей записи. Операция состоит в присоединении предыдущей сессии к существующей, это не мультисессионная запись, growisofs(1) *расширит* (grow) файловую систему ISO 9660, существующую на носителе.

Например, для дозаписи данных к предыдущей сессии на DVD+RW, используется следующая команда:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

При последующих записях mkisofs(8) необходимо передавать те же параметры, что и при первой записи.

Вы можете использовать параметр `-dvd-compat` для улучшения

совместимости с приводами DVD-ROM. В случае DVD+RW это не помешает добавлению данных.

Если по какой-либо причине вам потребуется очистить носитель, используйте следующую команду:

```
# growisofs -Z /dev/cd0=/dev/zero
```

16.7.6. Использование DVD-RW

Существует два формата дисков DVD-RW: последовательно дополняемый и с ограниченной перезаписью. По умолчанию формат дисков DVD-RW последовательный.

Новый DVD-RW может быть записан непосредственно без необходимости форматирования, однако DVD-RW с данными в последовательном формате необходимо очистить перед созданием новой начальной сессии.

Для очистки DVD-RW в последовательном формате, запустите:

```
# dvd+rw-format -blank=full /dev/cd0
```

Полная очистка (`-blank=full`) займет около одного часа на скорости 1x. Быструю очистку можно выполнить с параметром `-blank`, если DVD-RW будет записан в режиме Disk-At-Once (DAO). Для записи DVD-RW в режиме DAO, используйте команду:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
```

Параметр `-use-the-force-luke=dao` не должен потребоваться, поскольку growisofs(1) попытается определить был ли носитель быстро очищен и включить DAO запись.

Фактически, лучше использовать режим с ограниченной перезаписью с любым DVD-RW, этот формат более гибкий, чем формат по умолчанию с последовательной записью.

Для записи данных на последовательный DVD-RW, используйте ту же команду, что и для других форматов DVD:

```
# growisofs -Z /dev/cd0 -J -R /path/to/data
```

Если вы хотите добавить данные к предыдущей записи, используйте параметр growisofs(1) `-M`. Однако при добавлении данных на DVD-RW в последовательном режиме, на диске будет создана новая сессия и в результате получится мультисессионный диск.

В формате DVD-RW с ограниченной перезаписью не требуется очищать носитель перед созданием новой начальной сессии, вам всего лишь нужно переписать диск с параметром `-Z`, подобно DVD+RW. Возможно также увеличение существующей файловой системы ISO 9660, записанной на диск тем же способом, как для DVD+RW с параметром `-M`. В результате получится односессионный DVD.

Для перевода DVD-RW в формат с ограниченной перезаписью, необходимо использовать следующую команду:

```
# dvd+rw-format /dev/cd0
```

Для перевода обратно в последовательный формат, выполните:

```
# dvd+rw-format -blank=full /dev/cd0
```

16.7.7. Мультисессия

Лишь несколько DVD-ROM и проигрывателей поддерживают мультисессионные DVD, в основном они в лучшем случае прочтут только первую сессию. DVD+R, DVD-R и DVD-RW в последовательном формате могут работать с несколькими сессиями, и это не относится к форматам DVD+RW и DVD-RW в формате ограниченной перезаписи.

Использование следующей команды после первой (не закрытой) сессии для DVD+R, DVD-R, или DVD-RW в последовательном формате, добавит на диск новую сессию:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Использование этой командной строки с DVD+RW или DVD-RW в режиме ограниченной перезаписи добавит данные, объединив новую сессию с предыдущей. В результате получится односессионный диск. Такой способ используется для добавления данных после первой записи на эти носители.

Некоторый объем носителя используется между сессиями для завершения и начала сессии. Следовательно, для оптимизации объема хранения сессии должны быть большими. Количество сессий ограничено 154 для DVD+R, около 2000 для DVD-R и 127 для DVD+R Double Layer.

16.7.8. Дополнительная информация

Для получения дополнительной информации о DVD, можно запустить команду `dvd+rw-mediainfo /dev/cd0`, диск должен находиться в приводе.

Дополнительная информация о dvd+rw-tools может быть найдена на странице справочника growisofs(1), на Web-сайте dvd+rw-tools и в архивах списка рассылки cdwrite.

Вывод `dvd+rw-mediainfo` при записи или проблемный носитель необходимы для любого сообщения о проблеме. Без этого вывода будет совершенно невозможно помочь вам.

16.7.9. Использование DVD-RAM

16.7.9.1. Конфигурация

Записывающие устройства DVD-RAM поставляются с интерфейсами SCSI и ATAPI. В последнем случае вы должны убедиться, что для них включен режим DMA, добавив в файл /boot/loader.conf строку

```
hw.ata.atapi_dma="1"
```

16.7.9.2. Подготовка носителя

Как указывалось ранее, DVD-RAM представляется съемным жестким диском. Как и другие дисковые устройства, DVD-RAM должен быть "подготовлен" к первому использованию. В нашем примере мы займём все пространство диска одной файловой системой UFS2:

```
# dd if=/dev/zero of=/dev/acd0 bs=2k count=1
# bsdlabel -Bw acd0
# newfs /dev/acd0
```

Имя устройства DVD device, acd0, должно соответствовать вашей конфигурации.

16.7.9.3. Использование носителя

После выполнения указанных выше команд, DVD-RAM может быть смонтирован как обычный жесткий диск:

```
# mount /dev/acd0 /mnt
```

После этого вы можете читать и писать на DVD-RAM.

16.8. Дискеты

Хранение данных на дискетах иногда бывает полезным, например, когда нет других съёмных носителей или когда необходимо перенести небольшой объём данных на другой компьютер.

В этом разделе будет описано, как использовать дискеты во FreeBSD. В основном речь пойдёт о форматировании и использовании дискет DOS размером 3.5 дюйма, однако общие принципы применимы и для других форматов гибких дисков.

16.8.1. Форматирование дискет

16.8.1.1. Устройство

Доступ к гибким дискам, как, впрочем, и к остальным устройствам, осуществляется через соответствующие файлы в каталога /dev. Чтобы обратиться к дискете, просто используйте /dev/fdN.

16.8.1.2. Форматирование

Перед тем, как дискетой можно будет воспользоваться, её необходимо отформатировать на низком уровне. Обычно это выполняется производителем, однако форматирование является хорошим способом проверить целостность носителя. Большинство гибких дисков предназначены для использования с размером 1440kB, однако возможно задать меньший или больший размер.

Для низкоуровневого форматирования дискет вам нужно использовать fdformat(1). В качестве параметра этой утилите передаётся имя устройства.

Обратите внимание на появление сообщений об ошибках, так как они могут помочь определить, хорошая это дискета или плохая.

16.8.1.2.1. Форматирование гибких дисков

Для форматирования гибких дисков используйте устройства /dev/fdN. Вставьте новую 3.5-дюймовую дискету в дисковод и введите команду:

```
# /usr/sbin/fdformat -f 1440 /dev/fd0
```

16.8.2. Метка диска

После низкоуровневого форматирования диска вам нужно поместить на него метку диска. Эта метка будет потом разрушена, но она будет нужна системе для определения размера диска и его характеристик.

Новая метка диска будет касаться диска в целом, и будет содержать полную информацию о параметрах дискеты. Значения геометрии для метки диска перечислены в файле /etc/disktab.

Теперь вы можете запустить bsdlabel(8) примерно так:

```
# /sbin/bsdlabel -B -w /dev/fd0 fd1440
```

16.8.3. Файловая система

Теперь ваша дискета готова к высокоуровневому форматированию. При этом на неё будет помещаться новая файловая система, которая позволит FreeBSD читать и записывать информацию на диск. После создания новой файловой системы метка диска уничтожается,

так что если вы захотите переформатировать диск, вам придётся создавать метку диска повторно.

Файловой системой для дискеты может служить UFS или FAT. Вообще говоря, FAT для дискет походит лучше.

Для размещения на дискете новой файловой системы, выполните:

```
# /sbin/newfs_msdos /dev/fd0
```

Теперь диск готов к работе.

16.8.4. Использование дискет

Для работы с гибким диском смонтируйте его при помощи утилит mount_msdosfs(8). Можно также использовать пакет emulators/mtools из коллекции портов.

16.9. Создание и использование архивных копий на магнитной ленте

К наиболее часто используемым носителям на магнитной ленте следует отнести ленты шириной 4мм и 8мм, а также типа QIC, мини-картриджи и DLT.

16.9.1. 4мм (DDS: Digital Data Storage)

Ленты шириной 4мм заменяют QIC в качестве наиболее предпочтительного носителя для создания резервных копий. Эта тенденция значительно усилилась после покупки компанией Conner фирмы Archive, ведущего производителя накопителей QIC и последующего прекращения их выпуска. Накопители 4мм малы по размеру и мало шумят, но у них нет репутации носителя, обладающего надежностью приводов 8мм. Картриджи более дешевы и меньше по размеру (3 x 2 x 0.5 дюймов; 76 x 51 x 12 мм), чем 8мм-картриджи. Накопители для лент шириной 4мм, как и 8мм, имеют сравнительно малый срок службы головок, по причине использования в обоих случаях технологии спирального сканирования (helical scan).

Пропускная способность у таких накопителей начинается с цифры ~150 kB/s, пиковая достигает ~500 kB/s. Ёмкость накопителей начинается с 1.3 GB и может достигать 2.0 GB. Аппаратное сжатие, имеющееся на большинстве таких накопителей, даёт увеличение ёмкости примерно вдвое. Блоки многоприводных ленточных библиотек могут иметь до 6 накопителей в одном модуле с автоматической сменой ленты. Ёмкость библиотек может достигать 240 Гбайт.

Стандарт DDS-3 в настоящее время поддерживает ёмкости лент вплоть до 12 Гбайт (или 24 Гбайт сжатой информации).

В накопителях 4мм, как и в приводах 8мм, используется технология спирального сканирования. Все плюсы и минусы этой технологии относятся как к 4мм, так и 8мм

приводам.

Не следует использовать ленты после того, как они были подвергнуты 2000 проходов, или были использованы для создания 100 полных копий.

16.9.2. 8мм (Exabyte)

Ленты шириной 8мм являются самым распространённым типом для ленточных SCSI-накопителей; они же являются наиболее удачным выбором при выборе типа носителей для обмена лентами. Наверное, каждый сервер имеет привод Exabyte шириной 8мм и объёмом 2 Гбайт. Эти приводы удобны, они работают надёжно и тихо. Картриджи дешевы и малы по размеру (4.8 x 3.3 x 0.6 дюймов; 122 x 84 x 15 мм). Одним минусом лент шириной 8мм является сравнительно малое время службы головок и лент из-за высокой скорости движения ленты вдоль головок.

Скорость передачи данных варьируется от ~250 kB/s до ~500 kB/s. Объём хранимых данных начинается с 300 Мбайт и может достигать 7 Гбайт. Аппаратное сжатие, имеющееся практически на всех таких приводах, увеличивает емкость примерно вдвое. Эти приводы существуют как в виде отдельных модулей, так и в виде многоприводных ленточных библиотек с 6 приводами и 120 лентами в одном отсеке. Ленты сменяются автоматически модулем. Емкости библиотек достигают величин, превышающих 840 Гбайт.

Модель Exabyte "Mammoth" поддерживает ёмкость ленты в 12 Гбайт (24 Гбайт со сжатием) и стоит примерно вдвое больше, чем обычный ленточный накопитель.

Данные на ленту записываются по технологии спирального сканирования, головки позиционируются под углом к носителю (примерно в 6 градусов). Лента оборачивается на 270 градусов вокруг шпульки, которая держит головки. Во время скольжения ленты вокруг шпульки последняя вращается. В результате достигается высокая плотность записи данных с очень близко лежащими дорожками, расположенными под наклоном по всей ленте.

16.9.3. QIC

Ленты и накопители формата QIC-150, наверное, являются наиболее распространенным типом носителей. Приводы лент формата QIC являются самыми дешёвыми "серьёзными" накопителями для резервного копирования. Минусом является стоимость носителей. Ленты формата QIC по сравнению с лентами шириной 8мм или 4мм являются дорогими, превосходя их по стоимости хранения одного гигабайта в пять раз. Однако если вам будут достаточно половины ленты, QIC может оказаться правильным выбором. QIC является *самым* распространенным типом привода. Каждый сайт имеет привод QIC какой-либо емкости. QIC имеет большое количество плотностей на физически похожих (иногда даже идентичных) лентах. Приводы QIC работают вовсе не тихо. Эти накопители громко осуществляют поиск перед тем, как начать запись данных и достаточно шумны в процессе чтения, записи или поиска. Ленты QIC имеют размеры (6 x 4 x 0.7 дюймов; 152 x 102 x 17 мм).

Скорость обмена данными лежит в границах от ~150 kB/s до ~500 kB/s. Ёмкость накопителей варьируется от 40 Мбайт до 15 Гбайт. Аппаратное сжатие присутствует во многих современных накопителях QIC. Приводы QIC устанавливаются менее часто; они вытесняются накопителями DAT.

На ленту данные записываются в виде дорожек. Дорожки располагаются в длину вдоль всей ленты. Количество дорожек, и, в свою очередь, их ширина, меняется вместе с емкостью ленты. Большинство, если не все современные накопители обеспечивают обратную совместимость по крайней мере для чтения (однако зачастую и для режима записи). Формат QIC имеет хорошую репутацию в области надежности хранения данных (механика устроена проще и более надежна, чем в случае накопителей, построенных по технологии спирального сканирования).

Ленты не следует больше использовать после создания 5,000 резервных копий.

16.9.4. DLT

Формат DLT обладает самой высокой скоростью передачи данных среди всех перечисленных здесь накопителей. Лента шириной 1/2" (12.5мм) помещена в один картридж с катушкой (4 x 4 x 1 дюймов; 100 x 100 x 25 мм). Вдоль одной из сторон картриджа расположена сдвигающаяся крышечка. Механизм накопителя открывает эту крышку, чтобы вытащить конец ленты. На этом конце имеется овальное отверстие, которое используется для "захвата" ленты. Принимающая катушка размещена внутри накопителя. Все другие типы картриджей, перечисленные здесь (за исключением 9-дорожечных лент), имеют как подающий, так и принимающий барабаны внутри самого картриджа.

Скорость передачи данных равна примерно 1.5 MB/s, что в три раза больше скорости передачи данных для накопителей 4мм, 8мм или QIC. Ёмкость картриджей варьируется от 10 Гбайт до 20 Гбайт для одного накопителя. Приводы могут компоноваться как многоленточные роботизированные, так и многоленточные, многоприводные библиотеки лент, вмещающие от 5 до 900 лент и от 1 до 20 приводов, что даёт ёмкость хранилища от 50 Гбайт до 9 Тбайт.

Формат DLT Type IV поддерживает емкость до 70 Гбайт со сжатием.

Данные на ленту записываются в виде дорожек, параллельных направлению движения (точно также, как и для лент QIC). Одновременно записываются две дорожки. Срок жизни головок чтения/записи сравнительно велик; как только лента перестает двигаться, одновременно прекращается трение между головками и лентой.

16.9.5. AIT

AIT - это новый формат фирмы Sony, который позволяет хранить до 50 Гбайт (со сжатием) информации на одной ленте. Ленты содержат микросхемы памяти, на которых размещается каталог содержимого ленты. Этот каталог может быть быстро считан накопителем для определения расположения файлов на ленте, вместо того, чтобы тратить несколько минут на поиск, как это происходит с другими форматами. Такое программное обеспечение, как SAMS:Alexandria, может управлять сорока или большим количеством ленточных библиотек AIT, связываясь непосредственно с памятью лент для вывода их содержимого, определения того, какие файлы были скопированы на какую ленту, выбора нужной ленты, её загрузки и восстановления данных с ленты.

Библиотеки с такими функциями стоят в районе $20,000, выводя их из ниши любительского рынка.

16.9.6. Использование новой ленты первый раз

Если вы попытаетесь прочитать или записать новую, абсолютно чистую ленту, в первый раз, то вам это не удастся. Выводимые на консоль сообщения будут выглядеть примерно так:

```
sa0(ncr1:4:0): NOT READY asc:4,1
sa0(ncr1:4:0):  Logical unit is in process of becoming ready
```

На ленте отсутствует идентификационный блок (блок номер 0). Со времен принятия стандарта QIC-525 все накопители формата QIC записывают на ленту идентификационный блок (Identifier Block). Здесь имеется два решения:

- По команде `mt fsf 1` ленточный накопитель записывает идентификационный блок на ленту.
- Воспользуйтесь кнопкой на передней панели для выброса ленты.

  Вставьте ленту повторно и по команде `dump` сбросьте данные на ленту.

  Программа `dump` выдаст `DUMP: End of tape detected`, а на консоли будет выведено: `HARDWARE FAILURE info:280 asc:80,96`.

  перемотайте ленту такой командой: `mt rewind`.

  Последующие операции с лентой будут успешными.

16.10. Создание резервных копий на дискетах

16.10.1. Можно ли использовать дискеты для создания резервных копий моих данных?

На самом деле дискеты не подходят для создания резервных копий, потому что:

- Носитель ненадёжен, особенно если речь идет о больших сроках хранения.
- Создание резервных копий и восстановление данных происходит очень медленно.
- Дискеты имеют весьма ограниченную емкость (дни, когда весь винчестер копировался на десяток или около того дискет, давно прошли).

Несмотря на все это, если у вас нет другого способа сделать резервную копию ваших данных, то дискеты все же лучше, чем ничего.

Если вы используете дискеты, то проверьте, что они должны быть хорошего качества. Дискеты, которые валялись по всему офису в течении нескольких лет, не подойдут. Идеально использовать новые от известного производителя.

16.10.2. Итак, как же сделать резервную копию данных на дискетах?

Самым лучшим методом создания резервной копии на дискете является использование утилиты tar(1) с опцией `-M` (многотомные архивы), которая позволяет размещать архивы на нескольких дискетах.

Для копирования всех файлов в текущем каталоге и подкаталогах выполните следующее (работая как пользователь `root`):

```
# tar Mcvf /dev/fd0 *
```

Когда первая дискета окажется полностью заполненной, программа tar(1) выдаст запрос на следующий том (так как работа утилиты tar(1) не зависит от носителя, она имеет дело с томами; здесь это означает дискету).

```
Prepare volume 2 for /dev/fd0 and hit return:
```

Это сообщение будет повторяться (со все увеличивающимся номером тома) до тех пор, пока все указанные файлы не будут заархивированы.

16.10.3. Можно ли резервные копии подвергнуть компрессии?

К сожалению, tar(1) при создании многотомных архивов не позволяет использовать опцию `-z`. Вы конечно же, можете скомпрессировать все файлы утилитой gzip(1), программой gzip(1) скопировать их на дискеты, а затем распаковать файлы снова утилитой gunzip(1)!

16.10.4. Как восстановить данные из моих резервных копий?

Для полного восстановления архива воспользуйтесь такой командой:

```
# tar Mxvf /dev/fd0
```

Есть два подхода к восстановлению только нужных вам файлов. В первом вы можете начать с первой дискеты и выдать такую команду:

```
# tar Mxvf /dev/fd0 filename
```

Программа tar(1) будет выдавать запрос на подачу последующих дискет до тех пор, пока не найдет требуемый файл.

Как альтернатива, если вы знаете, на какой дискете расположен файл, то вы можете просто подать ее и дать ту же самую команду, что и выше. Заметьте, что если первый файл на дискете является продолжением предыдущего, то tar(1) выдаст предупреждение о том, что не может его восстановить, хотя вы этого и не просили делать!

16.11. Стратегии резервного копирования

При разработке плана резервного копирования первым делом надо продумать методы защиты от следующих проблем:

- Отказ жесткого диска
- Случайное удаление файлов
- Повреждение содержимого файлов
- Полное уничтожение компьютера (например, при пожаре), при котором погибнут также резервные копии, физически находящиеся рядом.

Вполне возможно, что для ваших нужд нет единой стратегии, наилучшим образом покрывающей все описанные проблемы; более того, скорее всего, ее и не может быть (разве что для персональных систем, где ценность данных очень низка).

Вот несколько наиболее распространенных технологий, применяемых для резервного копирования:

- Архивация системы целиком с копированием на какой-либо надежный внешний носитель и размещение его вдалеке от основной системы. При этом вы защищены от всех перечисленных проблемы, однако этот метод требует много времени и неудобен в процессе восстановления. Вы можете хранить резервные копии рядом или даже смонтированными, однако все равно столкнетесь с неудобствами при восстановлении, в особенности для непривилегированных пользователей.
- Снэпшоты файловых систем. Помогают только от случайного удаления файлов, но как раз в этом случае *очень* полезны и эффективны.
- Полные копии файловых систем или дисков (например, периодический запуск программы rsync(1) для машины целиком). Для защиты от отказа жестких дисков этот способ обычно несколько уступает RAID; для восстановления случайно удаленных файлов может быть сравним по удобству со снэпшотами UFS, в зависимости от вашей ситуации.
- RAID. Минимизирует или исключает вовсе простои при отказе жестких дисков. При этом средняя частота таких отказов увеличивается (поскольку количество дисков больше), но разбираться с ними становится много спокойнее.
- Проверка отпечатков файлов (fingerprints). Для этого весьма полезна утилита mtree(8). Не являясь собственно технологией резервного копирования, этот метод помогает выяснять, когда вам пока обращаться к резервным копиям. В особенности это важно для "оффлайновых" резервных копий.

Довольно легко придумать и другие стратегии резервного копирования; многие из них будут композициями уже упомянутых. Наличие специальных требований, как правило, приводит к специализированным же технологиям (например, резервное копирование базы данных, как правило, требует использования методов, специфичных для соответствующей СУБД). Главным остается знание опасностей потери данных, от которых вы хотите себя оградить, и методов защиты от них.

16.12. Основы технологии резервного копирования

Тремя основными программами резервного копирования являются dump(8), tar(1) и cpio(1).

16.12.1. Dump и Restore

Для UNIX® традиционными программами резервного копирования являются `dump` и `restore`. Они работают с приводом как с набором дисковых блоков, которые расположены ниже понятий файлов, связей и каталогов, создаваемых файловыми системами. В отличие от других программ для резервного копирования, программа `dump` выполняет резервное копирование всей файловой системы, располагающейся на устройстве. Невозможно выполнить резервное копирование части файловой системы или дерева каталогов, которые располагаются более чем в одной файловой системе. Утилита `dump` не записывает на ленту файлы и каталоги, она записывает блоки данных, из которых строятся файлы и каталоги. Утилита `restore` по умолчанию настроена так, что в процессе своей работы она сохраняет временные файлы в каталог /tmp/. В некоторых ситуациях доступного в /tmp/ места бывает недостаточно: например, если вы работаете с диска восстановления (recovery disk). В таких случаях для успешной работы утилиты необходимо указать в переменной окружения `TMPDIR` путь к каталогу на менее заполненной файловой системе.

> Если вы используете программу `dump` для работы с корневым каталогом, при этом не будет выполняться резервное копирование /home, /usr и многих других каталогов, так как они обычно являются точками монтирования других файловых систем или символическими ссылками на эти файловые системы.

В программе `dump` имеются некоторые неудобства, оставшиеся от её ранних дней в составе Version 6 операционной системы AT&T UNIX (примерно 1975). Параметры, используемые по умолчанию, подходят для 9-дорожечных лент (6250 bpi), но не для современных носителей с высокой плотностью записи информации (до 62,182 ftpi). Для использования ёмкостей нынешних накопителей на магнитной ленте эти параметры могут быть заданы в командной строке.

При помощи `rdump` и `rrestore` возможно резервное копирование данных по сети на накопитель, подключенный к другому компьютеру. Обе программы используют в работе rcmd(3) и ruserok(3) для доступа к накопителю на магнитной ленте на удалённом компьютере. Поэтому пользователь, выполняющий резервное копирование, должен быть указан в файле .rhosts на удалённом компьютере. Аргументы для `rdump` и `rrestore` должны подходить для использования на другом компьютере. При выполнении копирования по команде `rdump` на компьютере с FreeBSD на накопитель Exabyte, подключенный к машине Sun по имени `komodo`, используйте такую команду:

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /dev/da0a 2>&1
```

Будьте осторожны: есть проблемы с обеспечением безопасности при аутентификации посредством .rhosts. Внимательно рассмотрите вашу ситуацию.

Программы dump и restore можно использовать в более защищённом режиме посредством ssh.

Пример 23. Использование dump через ssh

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-l0.gz
```

Либо воспользуйтесь встроенной в dump возможностью, задав переменную окружения RSH:

Пример 24. Использование dump при работе через ssh с заданием RSH

```
# RSH=/usr/bin/ssh /sbin/dump -0uan -f
targetuser@targetmachine.example.com:/dev/sa0 /usr
```

16.12.2. tar

Утилита tar(1) также восходит корнями к Version 6 системы AT&T UNIX (около 1975). tar работает с файловой системой, записывая на ленту файлы и каталоги. Эта утилита поддерживает не полный набор опций, имеющихся в cpio(1), однако не требует необычного перенаправления в командной строке, которое используется в утилите cpio.

Чтобы скопировать данные на накопитель Exabyte, подключенный к машине Sun по имени komodo, используйте такую команду:

```
# tar cf - . | rsh komodo dd of=tape-device obs=20b
```

Если вы беспокоитесь о безопасности создания резервных копий по сети, то вместо rsh вам нужно использовать ssh.

16.12.3. cpio

cpio(1) является оригинальной программой UNIX® для обмена файлами на магнитных носителях. В утилите cpio имеются опции (кроме всего прочего), позволяющие выполнять изменение порядка следования байтов, поддерживающие различные форматы архивов и выполняющие перенаправление данных другим программам. Последняя возможность делает cpio прекрасным выбором для целей установки. cpio не знает о том, как работать с каталогами, список файлов должен даваться через stdin.

cpio не поддерживает создание резервных копий по сети. Вы можете воспользоваться перенаправлением вывода и программой rsh для посылки данных на удалённый накопитель.

```
# for f in directory_list; do
```

```
find $f >> backup.list
done
# cpio -v -o --format=newc < backup.list | ssh user@host "cat > backup_device"
```

Где directory_list это список директорий, с которых Вы хотите создать резервные копии, *user@host* это комбинация пользователь/хост которая описывает того кто занимается резервированием, и *backup_device* это устройство куда копии должны быть записаны (например, /dev/nsa0).

16.12.4. pax

pax(1) является ответом IEEE/POSIX® на утилиты `tar` и `cpio`. В течение многих лет различные версии программ `tar` и `cpio` получались не совсем совместимыми. Так что вместо того, чтобы попытаться полностью их стандартизировать, POSIX® создал новую утилиту для работы с архивами. `pax` пытается читать и писать различные форматы `cpio` и `tar`, и, кроме того, свои собственные новые форматы. Набор команд этой утилиты больше напоминает `cpio`, чем `tar`.

16.12.5. Amanda

Amanda (Advanced Maryland Network Disk Archiver) является целой клиент/серверной системой резервного копирования, а не отдельной программой. Сервер Amanda сможет осуществлять резервное копирование на единственный накопитель любого количества компьютеров, на которых имеется клиент Amanda и которые могут связываться по сети с сервером Amanda. Общей проблемой систем с большим количеством больших дисков является то, что время, требуемое для непосредственной записи данных на ленту, превышает лимит времени, выделенный на эту задачу. Amanda решает эту проблему. Amanda может использовать "промежуточный диск" для резервного копирования нескольких файловых систем одновременно. Amanda создаёт "наборы архивов": группа лент, используемых в некоторый период времени для создания полных копий всех файловых систем, перечисленных в конфигурационном файле системы Amanda. "Архивный набор" содержит также создаваемый каждую ночь инкрементальные (или дифференциальные) резервные копии всех файловых систем. Восстановление повреждённой файловой системы требует наличия самой последней полной копии и инкрементальных резервных копий.

Конфигурационный файл даёт прекрасный механизм для управления процессом резервного копирования и объёмом трафика, генерируемого системой Amanda. Amanda сможет использовать любую из перечисленных выше программ для записи данных на ленту. Amanda имеется в виде как порта, так и пакета, и по умолчанию она не установлена.

16.12.6. Не делать ничего

"Не делать ничего" - это не программа для компьютера, и в то же время это наиболее широко используемая стратегия резервного копирования. Здесь нет никаких первоначальных затрат. Здесь нет расписания, которому нужно следовать. Просто скажите нет. Если что-то случится с вашими данными, улыбнитесь и забудьте о них!

Если ваше время и данные практически ничего не стоят, то "не делать ничего" является самой подходящей программой для вашего компьютера. Но будьте осторожны, POSIX® является весьма полезным инструментом, и через полгода вы можете обнаружить, что у вас есть набор файлов, представляющих для вас определенную ценность.

"Ничего не делать" является правильным методом резервного копирования для /usr/obj и других деревьев каталогов, которые могут быть в точности перегенерированы вашим компьютером. Примером являются файлы, представляющие страницы этого Руководства в форматах HTML или PostScript®. Они генерируются из входных файлов в формате SGML. Создавать резервные копии файлов в форматах HTML и PostScript® не нужно. Исходные файлы в формате SGML копируются регулярно.

16.12.7. Какая программа резервного копирования самая лучшая?

dump(8) *Точка.* Elizabeth D. Zwicky протестировала все программы резервного копирования, обсуждаемые здесь. Беспроигрышным вариантом для сохранения всех ваших данных и особенностей файловых систем UNIX® является `dump`. Элизабет создала файловые системы, содержащие большое количество необычных элементов (и некоторых не так уж необычных) и тестировала каждую из программ, выполняя резервное копирование и последующее восстановление этих файловых систем. В число необычных элементов входили: файлы с дырами, файлы с дырами и блоком пустого места, файлы с необычными символами в их именах, нечитаемые и незаписываемые файлы, устройства, меняющие свой размер во время резервного копирования, файлы, создаваемые и удаляемые во время копирования и тому подобное. Она представила результаты на конференции LISA V в октябре 1991 года. Посмотрите ссылку на сайте torture-testing Backup and Archive Programs.

16.12.8. Процедура восстановления при сбое

16.12.8.1. До того, как случится катастрофа

Вам нужно выполнить всего лишь четыре шага для того, чтобы быть готовым к любому сбою.

Во-первых, распечатайте разметку диска для всех ваших дисков (к примеру, `bsdlabel da0 | lpr`), таблицу файловых систем (/etc/fstab) и все сообщения, выводимые при загрузке, каждого по два экземпляра.

Во-вторых, запишите CD диск с "livefs". Этот диск позволяет загружаться в режим аварийного восстановления FreeBSD, давая возможность пользователю выполнять ряд утилит, среди которых dump(8), restore(8), fdisk(8), bsdlabel(8), newfs(8), mount(8) и т.д. Образ CD с "livefs" для FreeBSD/i386 12.0-RELEASE находится по адресу ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/12.0/FreeBSD-12.0-RELEASE-i386-livefs.iso.

В-третьих, регулярно создавайте резервные копии на ленте. Любые изменения, которые вы делали после последнего резервного копирования, могут быть безвозвратно потеряны. На лентах включайте защиту от записи.

В-четвертых, проверяйте работу CD диска (который вы сделали при выполнении второго

шага) и лент с резервными копиями. Ведите журнал выполняемых действий. Храните эти записи вместе с загрузочным CD диском, распечатками и лентами. Вы просто обезумеете при восстановлении данных, если окажется, что записи помогли бы избежать разрушения ваших резервных копий (Каким образом? Вместо команды `tar xvf /dev/sa0` вы могли случайно набрать `tar cvf /dev/sa0` и тем самым перезаписать вашу резервную копию).

Для дополнительной страховки, каждый раз создавайте загрузочный CD диск с "livefs" и две резервные копии на ленте. Храните одну из копий в каком-то удаленном месте и НЕ в том же здании, где находится ваш офис. Достаточно большое количество компаний во Всемирном Торговом Центре изучило это на своей шкуре. Это удаленное хранилище должно быть физически отделено на большое расстояние от ваших компьютеров и дисковых устройств.

16.12.8.2. После сбоя

Главный вопрос: выжило ли ваше оборудование? Вы регулярно делали резервные копии, так что нет нужды беспокоиться о программном обеспечении.

Если оборудование было повреждено, должны быть заменены неисправные компоненты.

Если с оборудованием всё в порядке, вставьте CD диск с "livefs" в привод и загрузите компьютер. На экран будет выведено оригинальное меню установки. Выберите требуемую страну, потом - пункт меню Fixit — Repair mode with CDROM/DVD/floppy or start a shell., а в нём выберите пункт CDROM/DVD — Use the live filesystem CDROM/DVD. Утилита `restore` и другие нужные вам программы находятся в каталоге /mnt2/rescue.

Восстановите по отдельности каждую файловую систему.

Попробуйте выполнить команду `mount` (например, `mount /dev/da0a /mnt`) по отношению к корневому разделу вашего первого диска. Если метка диска была испорчена, то воспользуйтесь командой `bsdlabel` для переразбиения на разделы и разметки диска так, чтобы получившаяся метка совпала с той, которая вами была распечатана и сохранена. Для повторного создания файловых систем используйте утилиту `newfs`. Повторно смонтируйте корневой раздел диска в режиме чтения-записи (`mount -u -o rw /mnt`). Воспользуйтесь вашей программой резервного копирования и резервными копиями на лентах для восстановления данных для этой файловой системы (например. `restore vrf /dev/sa0`). Размонтируйте файловую систему (например, `umount /mnt`). Повторите эту процедуру для каждой файловой системы, которая была повреждена.

Как только ваша система заработает, сделайте резервную копию на новые ленты. Что бы ни вызвало сбой или потерю данных, это может случиться снова. Ещё один час, потраченный в этот момент, может спасти вас от неприятностей в будущем.

16.13. Сетевые файловые системы, файловые системы в памяти и с отображением в файл

Кроме дисков, которые вы физически устанавливаете в ваш компьютер; дискеты, компакт-диски, винчестеры и так далее, FreeBSD воспринимает и другие типы дисков - *виртуальные*

диски.

Сюда могут быть отнесены сетевые файловые системы, такие, как Network File System и Coda, а также файловые системы с организацией в памяти и создаваемые в файлах.

В зависимости от версии FreeBSD, которую вы используете, для создания и работы с файловыми системами, отображаемыми в оперативную память или файлы, вам нужно будет пользоваться разными инструментами.

Пользователи FreeBSD 4.X для создания требуемых устройств должны использовать MAKEDEV(8). Во FreeBSD 5.0 и более поздних версиях для создания файлов устройств используется devfs(5), которая выполняет это прозрачно для пользователей.

16.13.1. Файловая система в файле во FreeBSD 4.X

Утилита vnconfig(8) конфигурирует и позволяет использовать дисковые устройства на основе псевдо-устройств vnode. *vnode* представляет собой файл и отвечает за работу с файлом. Это означает, что vnconfig(8) использует файлы для создания и работы с файловой системой. Одним из возможных способов использования является монтирование образов дискет или образов компакт-дисков, сброшенных в файлы.

Для использования vnconfig(8) в конфигурационном файле ядра вам нужно включить поддержку vn(4):

```
pseudo-device vn
```

Чтобы смонтировать имеющийся образ файловой системы:

Пример 25. Использование vnconfig для монтирования имеющегося образа файловой системы во FreeBSD 4.X

```
# vnconfig vn0 diskimage
# mount /dev/vn0c /mnt
```

Для создания нового образа файловой системы с помощью vnconfig(8):

Пример 26. Создание нового диска в файле с помощью `vnconfig`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# vnconfig -s labels -c vn0 newimage
# bsdlabel -r -w vn0 auto
# newfs vn0c
Warning: 2048 sector(s) in last cylinder unallocated
```

```
/dev/vn0c:  10240 sectors in 3 cylinders of 1 tracks, 4096 sectors
    5.0MB in 1 cyl groups (16 c/g, 32.00MB/g, 1280 i/g)
super-block backups (for fsck -b #) at:
 32
# mount /dev/vn0c /mnt
# df /mnt
Filesystem  1K-blocks     Used    Avail Capacity  Mounted on
/dev/vn0c    4927        1     4532     0%    /mnt
```

16.13.2. Файловые системы, отображаемые в файлы

Во FreeBSD 5.X и более поздних для конфигурации и подключения дисков md(4), отображаемых в оперативную память, используется утилита mdconfig(8). Для работы с mdconfig(8) вам нужно подгрузить модуль md(4) или добавить поддержку этих устройств в файл конфигурации ядра:

```
device md
```

Утилита mdconfig(8) поддерживает три типа виртуальных дисков, отображаемых в память: диски в памяти, которая выделяется запросами malloc(9) и диски в памяти, использующие в качестве устройств хранения файлы или раздел подкачки. Одним из возможных использований таких дисков является монтирование файлов с образами дискет или CD.

Для монтирования образа существующей файловой системы:

Пример 27. Использование `mdconfig` *для монтирования файла с образом существующей файловой системы*

```
# mdconfig -a -t vnode -f diskimage -u 0
# mount /dev/md0 /mnt
```

Для создания образа новой файловой системы при помощи mdconfig(8):

Пример 28. Создание нового диска, отображаемого в файл, при помощи `mdconfig`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f newimage -u 0
# bsdlabel -w md0 auto
# newfs md0a
/dev/md0c: 5.0MB (10224 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
super-block backups (for fsck -b #) at:
 160, 2720, 5280, 7840
```

```
# mount /dev/md0a /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0a       4710    4  4330     0%    /mnt
```

Если в параметре `-u` вы не задали номер устройства, то mdconfig(8) для выбора неиспользуемого устройства будет использовать функцию автоматическое выделения в md(4). Имя выделенного устройства будет выдано на стандартное устройство выводы в виде, например, md4. Для получения более полной информации о mdconfig(8), пожалуйста, обратитесь к соответствующей странице справочной системы.

Утилита mdconfig(8) весьма полезна, однако для создания файла с файловой системой требуется произвести много действий. Вместе с FreeBSD 5.0 поставляется утилита под названием mdmfs(8), которая создаёт диск md(4) при помощи mdconfig(8), размещает на нём файловую систему UFS при помощи newfs(8) и монтирует её командой mount(8). Например, если вы хотите создать и смонтировать такой же образ файловой системе, как выше, просто наберите такую команду:

Пример 29. Настройка и монтирование диска, отображаемого в файл, при помощи команды `mdmfs`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdmfs -F newimage -s 5m md0 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0        4718    4  4338     0%    /mnt
```

Если вы используете параметр `md` без номера устройства, то mdmfs(8) будет использовать автоматическую нумерацию md(4) для автоматического выбора неиспользуемого устройства. Более полную информацию о mdmfs(8) можно найти на страницах справочной системы.

16.13.3. Файловая система в памяти во FreeBSD 4.X

Драйвер md(4) является простым и эффективным способом создания файловых систем в памяти во FreeBSD 4.X. Для выделения памяти используется malloc(9).

Просто возьмите файловую систему, которую вы приготовили при помощи, скажем, vnconfig(8) и:

Пример 30. Диск md в памяти во FreeBSD 4.X

```
# dd if=newimage of=/dev/md0
5120+0 records in
```

```
5120+0 records out
# mount /dev/md0c /mnt
# df /mnt
Filesystem  1K-blocks     Used    Avail Capacity  Mounted on
/dev/md0c    4927        1     4532     0%    /mnt
```

Для получения более полной информации, пожалуйста, обратитесь к страницам справочной системы по md(4).

16.13.4. Файловые системы с отображением в память

При работе с файловыми системами, отображаемыми в файл или память, используются одни и те же утилиты: mdconfig(8) или mdmfs(8). Обычно для отображаемых в память файловых систем следует использовать опцию "хранение на области подкачки". Это не означает, что такая файловая система будет сразу сброшена на диск: место под нее будет выделено из общего пула памяти, и при необходимости может перемещаться в область подкачки. Также, возможно выделение места под файловую систему в основной памяти (через malloc(9)); однако, следует помнить, что использование таких файловых систем, в особенности большого размера, может привести к панике системы от исчерпания ядерной памяти.

Пример 31. Создание нового диска с отображением в память при помощи `mdconfig`

```
# mdconfig -a -t swap -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 1.27MB, 81 blks, 192 inodes.
    with soft updates
super-block backups (for fsck -b #) at:
 160, 2752, 5344, 7936
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4718    4  4338     0%    /mnt
```

Пример 32. Создание нового диска с отображением в память при помощи `mdmfs`

```
# mdmfs -s 5m md2 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md2        4846    2  4458     0%    /mnt
```

16.13.5. Отключение диска, отображаемого в память, от системы

Если файловые системы, отображаемые в память или файл, больше не используются, вам нужно высвободить все ресурсы для системы. Первым делом нужно размонтировать файловую систему, затем воспользоваться mdconfig(8) для отключения диска от системы и освободить ресурсы.

К примеру, чтобы отключить и освободить все ресурсы, используемые /dev/md4:

```
# mdconfig -d -u 4
```

Для выдачи информации об отконфигурированных устройствах md(4) используется команда `mdconfig -l`.

Во FreeBSD 4.X для отключения устройства используется команда vnconfig(8). Например, для отключения и освобождения всех ресурсов, используемых /dev/vn4:

```
# vnconfig -u vn4
```

16.14. Мгновенные копии файловых систем

Во FreeBSD 5.0 вместе с технологией Отложенных обновлений представлена новая возможность: генерация мгновенных копий файловых систем.

Мгновенные копии позволяют пользователю создавать образы заданных файловых систем и работать с ними как с файлами. Файлы мгновенных копий должны создаваться в той файловой системе, над которой производится действие, и пользователь может создавать не более 20 мгновенных копий для каждой файловой системы. Активные копии записываются в суперблок, так что они остаются в силе между операциями монтирования и размонтирования в процессе системных перезагрузок. Если мгновенная копия больше не нужна, она может быть удалена стандартной командой rm(1). Мгновенные копии могут удаляться в любом порядке, однако всё использованное пространство не может быть использовано, так как другая мгновенная копия может претендовать на некоторые блоки из освобождённых.

Неизменяемый флаг `snapshot` устанавливается на файл при помощи mksnap_ffs(8) после первоначального создания файла мгновенной копии. Команда unlink(1) делает исключение для файлов мгновенных копий, позволяя их удалять.

Мгновенные копии создаются при помощи утилиты mount(8). Чтобы создать мгновенную копию /var в файле /var/snapshot/snap, воспользуйтесь такой командой:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

В качестве альтернативного средства создания мгновенных копий вы можете использовать утилиту mksnap_ffs(8):

```
# mksnap_ffs /var /var/snapshot/snap
```

Файлы мгновенных копий файловых систем (к примеру, /var) можно найти при помощи команды find(1):

```
# find /var -flags snapshot
```

После создания мгновенной копии есть несколько способов её использования:

- Некоторые администраторы будут использовать файл мгновенной копии для целей создания резервной копии, так как мгновенная копия может быть перенесена на CD или магнитную ленту.
- Утилита проверка целостности файловой системы, fsck(8), может быть запущена над мгновенной копией. Полагая, что файловая система была в порядке, когда она была смонтирована, вы всегда должны получать нормальный (и неизменный) результат. Это именно то, что выполняет фоновый процесс fsck(8).
- Запустить утилиту dump(8) с мгновенной копией. Будет создаваться дамп, соответствующий файловой системе на момент создания мгновенной копии. Утилита dump(8) при использовании опции `-L` тоже может работать с мгновенными копиями, создавать их дампы, а затем удалять за один проход.
- Смонтировать командой mount(8) мгновенную копию как замороженный образ файловой системы. Чтобы смонтировать командой mount(8) мгновенную копию /var/snapshot/snap, запустите:

  ```
  # mdconfig -a -t vnode -f /var/snapshot/snap -u 4
  # mount -r /dev/md4 /mnt
  ```

Теперь вы можете пройтись по иерархии вашей зафиксированной файловой системы /var, смонтированной в каталог /mnt. Первоначально всё будет в том же самом состоянии, в каком это было во время создания мгновенной копии. Единственным исключением будет то, что любые ранее сделанные мгновенные копии будут видны как файлы нулевой длины. Когда использование мгновенной копии закончено, она может быть удалена командой:

```
# umount /mnt
# mdconfig -d -u 4
```

Для получения более полной информации о `softupdates` и мгновенных копиях файловых систем, включая технической описание, вы можете посетить сайт Маршалла Кёрка МакКузика (Marshall Kirk McKusick) по адресу http://www.mckusick.com/.

16.15. Квотирование файловых систем

Квоты - это опциональная возможность операционной системы, которая позволяет

ограничивать объем дискового пространства и/или количество файлов для конкретного пользователя или членов определенной группы в рамках одной файловой системы. Чаще всего эта возможность используется в системах разделения времени, когда желательно ограничить количество ресурсов, которые может использовать один пользователь или группа пользователей. Это позволит не допустить ситуации, когда один пользователь или группа пользователей заполняют всё доступное дисковое пространство.

16.15.1. Настройка вашей системы на использование дисковых квот

Перед тем, как попытаться использовать дисковые квоты, необходимо убедиться, что квоты включены в вашем ядре. Это делается добавлением следующей строки в конфигурационный файл вашего ядра:

```
options QUOTA
```

В стандартном ядре GENERIC это по умолчанию не включено, так что для использования дисковых квот вам нужно будет настроить, откомпилировать и установить собственное ядро. Пожалуйста, обратитесь к Настройка ядра FreeBSD за дополнительной информацией о настройке ядра.

Затем вам потребуется включить квотирование дисков в файле /etc/rc.conf. Это делается добавление такой строчки:

```
enable_quotas="YES"
```

Для более полного контроля над запуском квотирования имеется дополнительная переменная для настройки. Как правило, при загрузке целостность квот каждой файловой системы проверяется программой quotacheck(8). При работе программы quotacheck(8) проверяется точное соответствие данных в базе данных квот данным в файловой системе. Это весьма долгий процесс, что отражается на времени загрузки системы. Если вам захочется пропустить этот шаг, то для этого предназначена специальная переменная в файле /etc/rc.conf:

```
check_quotas="NO"
```

Наконец, вам потребуется отредактировать файл /etc/fstab для включения дисковых квот на уровне файловых систем. Это то место, где вы можете включить квоты для пользователей, для групп или для обеих этих категорий для всех ваших файловых систем.

Для включения пользовательских квот для файловой системы, добавьте параметр `userquota` в поле параметров файловой системы, на которой вы хотите включить квотирование, в файле /etc/fstab. Например:

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

Подобным же образом для включения квотирования на уровне групп, воспользуйтесь параметром `groupquota` вместо `userquota`. Чтобы включить квотирование как для пользователей, так и для групп, измените строчку следующим образом:

```
/dev/da1s2g   /home    ufs rw,userquota,groupquota 2 2
```

По умолчанию файлы квот хранятся в корневом каталоге файловой системы в файлах с именами quota.user и quota.group соответственно для пользовательских и групповых квот. Для получения подробной информации обратитесь к команде fstab(5). Хотя справочная страница по fstab(5) утверждает, что вы можете указать другое местоположение файлов с квотами, этого делать не рекомендуется, потому что различные утилиты для работы с квотами не могут нормально работать в такой ситуации.

На этом этапе вы должны перезагрузить вашу систему с новым ядром. Скрипт /etc/rc автоматически запустит соответствующие команды для создания начальных файлов для всех квот, которые вы создали в файле /etc/fstab, так что нет нужды вручную создавать никаких файлов квот нулевой длины.

При нормальной работе вам не потребуется вручную запускать программы quotacheck(8), quotaon(8) или quotaoff(8). Однако вам нужно хотя бы прочесть страницы справочника по этим командам, просто чтобы ознакомиться с их функциями.

16.15.2. Установка квот

Как только вы настроили вашу систему на использование квот, проверьте, что они действительно были задействованы. Простым способом сделать это является запуск такой команды:

```
# quota -v
```

Вы должны увидеть однострочную информацию, отражающую использование диска и текущие ограничения для каждой файловой системы, на которой включено квотирование.

Теперь вы действительно готовы задавать ограничения при помощи команды edquota(8).

У вас есть несколько вариантов того, как приводить в действие ограничения по объему дискового пространства, который могут занимать пользователь или группа, а также по количеству файлов, которые они могут создать. Вы можете ограничивать размещение ресурсов на основе объема дискового пространства (квотирование блоков), количества файлов (квотирование inode) или их комбинации. Каждое из этих ограничений, в свою очередь, делится на две категории: мягкие и жёсткие ограничения.

Жёсткое ограничение не может быть превышено. Как только пользователь достиг своих ограничений, ресурсы соответствующей файловой системы ему больше выделяться не будут. Например, если пользователь имеет жесткое ограничение в 500 Кбайт на файловой системе и в текущий момент использует 490 Кбайт, то пользователь может получить дополнительно ещё 10 Кбайт. Попытка занять ещё 11 Кбайт окончится неудачно.

С другой стороны, мягкие ограничения могут быть превышены в течении некоторого периода времени. Этот период времени также называют периодом отсрочки, который по умолчанию равен одной неделе. Если пользователь превышает своё мягкое ограничение в течение периода времени, превышающего отсрочку, то это мягкое ограничение становится жестким и последующее выделение ресурсов будет запрещено. Когда пользователь вернётся обратно к отметке, меньшей, чем мягкое ограничение, то период отсрочки будет сброшен.

Далее приводится пример того, что вы можете наблюдать при запуске команды edquota(8). Когда вызывается команда edquota(8), вы оказываетесь в редакторе, заданном переменной переменной окружения `EDITOR`, или в редакторе vi, если переменная `EDITOR` не задана, и можете редактировать квоты.

```
# edquota -u test
```

```
Quotas for user test:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
    inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
    inodes in use: 0, limits (soft = 50, hard = 60)
```

Для каждой файловой системы, на которой включено квотирование, вы должны увидеть две строки. В одной строке приведены ограничения на блоки, а в другой на количество inode. Например, чтобы увеличить ограничения на количество блоков для пользователя с мягкого ограничения в 50 и жёсткого ограничения в 75, на мягкое ограничение в 500 и жёсткое ограничение в 600, измените:

```
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
```

на:

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

Новые ограничения вступят в силу после выхода из редактора.

Иногда желательно установить ограничения квот на некоторый диапазон UID (идентификаторов пользователей). Это можно сделать при помощи параметра `-p` в команде edquota(8). Во-первых, установите желаемое ограничение для пользователя, а затем запустите команду `edquota -p protouser startuid-enduid`. Например, если пользователь `test` имеет желаемые ограничения, то для дублирования этих ограничений на пользователей с UID от 10000 до 19999 может быть использована такая команда:

```
# edquota -p test 10000-19999
```

Дополнительную информацию можно получить из справочной страницы по команде edquota(8).

16.15.3. Проверка ограничений и использования диска

Для проверки квот и использования дисков вы можете использовать команды quota(1) или repquota(8). Команда quota(1) может быть использована для проверки квот отдельных пользователей, групп, а также использования дисков. Пользователь может только проверить собственную квоту и квоту той группы, к которой он принадлежит. Только администратор системы может проверить квоты всех пользователей и групп. Команду repquota(8) можно использовать для получения суммарной статистики всех квот и использования дисков для файловых систем с включенными квотами.

Далее приведен пример вывода команды `quota -v` для пользователя, который имеет ограничения на двух файловых системах.

```
Disk quotas for user test (uid 1002):
     Filesystem  usage    quota   limit   grace   files   quota   limit   grace
           /usr      65*     50      75   5days       7      50      60
       /usr/var       0      50      75           0      50      60
```

В этом примере для файловой системы /usr пользователь превысил свое мягкое ограничение в 50 Кбайт на 15 Кбайт и имеет 5 дней до истечения отсрочки. Отметьте знак звездочки *, который указывает на превышение пользователем своего ограничения.

Как правило, файловые системы, на которых пользователь не занимает дискового пространства, не показываются в выводе команды quota(1), даже если ему выделена квота на этой файловой системе. При использовании параметра `-v` эти файловые системы выводятся, как, например, файловая система /usr/var в примере выше.

16.15.4. Квоты в NFS

Квоты определяются подсистемой квот на сервере NFS. Даемон rpc.rquotad(8) предоставляет информацию о квотах для программы quota(1) на клиентах NFS, позволяя пользователям на этих машинах смотреть свою статистику о квотах.

Включите `rpc.rquotad` в файле /etc/inetd.conf следующим образом:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Теперь перезапустите `inetd`:

```
# /etc/rc.d/inetd restart
```

16.16. Шифрование дисковых разделов

FreeBSD предоставляет прекрасную возможность по защите от несанкционированного доступа к данным. Права на доступ к файлам и технология принудительного контроля доступа MAC (Mandatory Access Control) (смотрите see Принудительный контроль доступа (MAC)) помогают предотвратить несанкционированный доступ посторонних лиц к данным, при условии работы операционной системы и компьютера. Однако права доступа, контролируемые операционной системой, не имеют значения, если нападающий получает физический доступ к компьютеру и может просто перенести жёсткий диск на другую машину для копирования и дальнейшего анализа важных данных.

Вне зависимости от того, как атакующий завладел жёстким диском или выключенным компьютером, технологии gbde (GEOM Based Disk Encryption - шифрование диска на уровне GEOM) и криптографическая подсистема `geli` FreeBSD могут защитить данные файловой системы компьютера даже против очень заинтересованной атакующей стороны с достаточными ресурсами. В отличие от громоздких систем шифрования, которые шифруют отдельные файлы, `gbde` и `geli` шифруют в прозрачном режиме файловую систему в целом, при этом данные в открытом виде на диск никогда не записываются.

16.16.1. Шифрование диска при помощи gbde

1. Получите права пользователя `root`

   Настройка gbde требует права доступа администратора системы.

   ```
   % su -
   Password:
   ```

2. Включите поддержку gbde(4) в конфигурационный файл ядра

   Добавьте следующую строку в файл конфигурации вашего ядра:

   `options GEOM_BDE`

   Перестройте ядро FreeBSD. Этот процесс описан в Настройка ядра FreeBSD.

   Перезагрузитесь, запустив новое ядро.

3. Альтернативой пересборке ядра является использование `kldload` для загрузки модуля gbde(4):

   ```
   # kldload geom_bde
   ```

16.16.1.1. Подготовка зашифрованного жёсткого диска

В следующем примере предполагается, что в вашу систему вы добавляете новый винчестер,

на котором будет располагаться единственный раздел с зашифрованными данными. Этот раздел будет монтироваться в каталог /private. gbde может также использоваться для шифрования /home и /var/mail, но это требует более сложной последовательности действий, что выходит за рамки этого вводного материала.

1. Подключите новый жёсткий диск

   Установите новый диск в систему, как это описано в Добавление дисков. В рамках этого примера раздел, соответствующий новому жёсткому диску, будет называться /dev/ad4s1c. Устройства /dev/ad0s1* представляют существующие стандартные разделы FreeBSD нашей тестовой системы.

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   /dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
   /dev/ad0s1a     /dev/ad0s1d     /dev/ad4
   ```

2. Создайте каталог для размещения файлов блокировок GBDE

   ```
   # mkdir /etc/gbde
   ```

   Файл блокировки gbde содержит информацию, которая нужна gbde для доступа к зашифрованному разделу. Не имея доступа к файлу блокировки, gbde не сможет расшифровать данные, хранимые в зашифрованном разделе, без значительного ручного вмешательства, что программно не поддерживается. Каждый зашифрованный раздел использует отдельный файл блокировки.

3. Инициализируйте раздел gbde

   Перед началом работы с разделом gbde его необходимо проинициализировать. Эта инициализация производится только один раз:

   ```
   # gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c.lock
   ```

   gbde(8) запустит редактор, что позволит вам задать в шаблоне различные конфигурационные параметры. При работе с файловыми системами UFS1 и UFS2 задайте значение sector_size равным 2048:

   ```
   $FreeBSD: src/sbin/gbde/template.txt,v 1.1 2002/10/20 11:16:13 phk Exp $
   #
   # Sector size is the smallest unit of data which can be read or written.
   # Making it too small decreases performance and decreases available space.
   # Making it too large may prevent filesystems from working.  512 is the
   # minimum and always safe.  For UFS, use the fragment size
   #
   sector_size     =       2048
   ```

```
[...]
```

gbde(8) дважды запросит ввод пароля, который будет использоваться для защиты данных. Пароль в обоих случаях должен вводиться одинаковый. Возможности gbde по защите ваших данных полностью зависят от качества выбранной вами ключевой фразы.

По команде `gbde init` создаётся файл блокировок для вашего раздела gbde, который в нашем случае будет иметь имя /etc/gbde/ad4s1c.lock. Для того, чтобы файлы блокировок корректно распознавались стартовым скриптом /etc/rc.d/gbde, их имена должны заканчиваться на ".lock".

Резервные копии файлов блокировок gbde *должны* храниться вместе с содержимым шифруемых разделов. Хотя удаление только блокировочного файла не сможет противостоять дешифрации атакующим раздела gbde, без этого файла даже легитимный пользователь не сможет получить доступ к данным без определённых и значительных усилий, что не поддерживается gbde(8) и его разработчиком.

4. Подключите зашифрованный раздел к системе

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

Будет выдан запрос на ввод ключевой фразы, которую вы выбирали во время инициализации зашифрованного раздела. Новое защищённое устройство будет видно в каталоге /dev под названием /dev/device_name.bde:

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Создайте файловую систему на зашифрованном устройстве

Как только защищённое устройство будет подключено к системе, вы сможете создать на нём файловую систему. Для этого используется утилита newfs(8). Так как инициализация новой файловой системы UFS2 происходит быстрее, чем инициализация файловой системы старого формата UFS1, то рекомендуется использовать newfs(8) с параметром `-O2`.

```
# newfs -U -O2 /dev/ad4s1c.bde
```

Запуск команды newfs(8) должен выполняться над подключенном разделе gbde, который идентифицируется по расширению *.bde в

имени устройства.

6. Смонтируйте зашифрованный раздел

   Создайте точку монтирования для зашифрованной файловой системы.

```
# mkdir /private
```

   Смонтируйте защищённую файловую систему.

```
# mount /dev/ad4s1c.bde /private
```

7. Проверьте доступность зашифрованной файловой системы

   Защищённая файловая система теперь должна быть доступна утилите df(1) и доступной для использования.

```
% df -H
Filesystem        Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a      1037M    72M   883M     8%
/devfs            1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f       8.1G    55K   7.5G     0%    /home
/dev/ad0s1e      1037M   1.1M   953M     0%    /tmp
/dev/ad0s1d       6.1G   1.9G   3.7G    35%    /usr
/dev/ad4s1c.bde   150G   4.1K   138G     0%    /private
```

16.16.1.2. Монтирование имеющихся зашифрованных файловых систем

После каждой загрузки для каждой защищённой файловой системы перед их использованием должны выполняться повторное подключение к системе, проверка на наличие ошибок и монтирование. Требуемые для этого команды должны выполняться пользователем `root`.

1. Подключение gbde-раздела к системе

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

   Будет выдан запрос на ввод ключевой фразы, выбранной на этапе инициализации зашифрованного раздела gbde.

2. Проверка файловой системы на наличие ошибок

   Так как защищаемая файловая система не может пока быть указана в файле /etc/fstab для автоматического монтирования, то она должны проверяться на

наличие ошибок посредством ручного запуска fsck(8) до её монтирования.

```
# fsck -p -t ffs /dev/ad4s1c.bde
```

3. Монтирование зашифрованной файловой системы

```
# mount /dev/ad4s1c.bde /private
```

Теперь защищённая файловая система доступна для работы.

16.16.1.2.1. Автоматическое монтирование зашифрованных разделов

Для автоматического подключения, проверки и монтирования зашифрованного раздела можно создать скрипт, но по соображениям безопасности в этом скрипте пароля для gbde(8) быть не должно. Поэтому рекомендуется запускать такие скрипты вручную, а пароль задавать с консоли или сеанса ssh(1).

Кроме того, базовая система содержит скрипт rc.d для автоматического монтирования шифрованных разделов. Его аргументы могут быть указаны в файле rc.conf(5):

```
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
gbde_lockdir="/etc/gbde"
```

При этом ключевая фраза для gbde должна быть введена на этапе загрузки. После введения ключевой фразы зашифрованный раздел будет смонтирован автоматически. Такой подход может быть очень удобным для использования gbde на ноутбуках.

16.16.1.3. Криптографическая защита, применяемая в gbde

gbde(8) шифрует содержимое секторов при помощи 128-битного AES в режиме CBC. Каждый сектор диска шифруется различным ключом AES. Более полная информацию о системе шифрования gbde, включая алгоритм генерации ключей для секторов из ключевой фразы, вводимой пользователем, можно найти на страницах справочной системы о gbde(4).

16.16.1.4. Вопросы совместимости

sysinstall(8) несовместим с устройствами, зашифрованными gbde. Все устройства *.bde перед запуском sysinstall(8) должны быть отключены от системы, или эта утилита аварийно завершит работу на этапе обнаружения устройств. Для отключения защищённого устройства, используемого в нашем примере, воспользуйтесь такой командой:

```
# gbde detach /dev/ad4s1c
```

16.16.2. Шифрование дисков при помощи geli

Во FreeBSD имеется альтернативный криптографический класс GEOM - geli. В настоящий момент он поддерживается Paweł Jakub Dawidek <pjd@FreeBSD.org>. Утилита geli отличается от gbde; она предоставляет другой комплекс возможностей и использует иную схему криптования.

Наиболее значимыми особенностями geli(8) являются:

- Использование инфраструктуры crypto(9): при наличии аппаратной криптографической поддержки, geli автоматически использует ее.
- Поддержка разнообразных криптоалгоритмов (в настоящее время AES, Blowfish и 3DES).
- Поддержка шифрованного корневого раздела. Для загрузки в такой ситуации потребуется ввести ключевую фразу.
- Поддержка двух независимых ключей шифрования (например, "основного ключа" и "ключа компании").
- Высокая скорость работы geli за счет простого криптования сектор-сектор.
- Поддержка архивирования основных ключей. При необходимости текущие ключи могут быть уничтожены, а в дальнейшем доступ к данным восстановлен при помощи архивированных ключей.
- Поддержка криптования файловых систем случайным одноразовым ключом - например, для разделов подкачки или временных файловых систем.

Другие возможности класса geli описаны в его странице справочника: geli(8).

Несколько следующих страниц будут посвящены описанию процесса конфигурации geli в ядре FreeBSD, а также объяснят, как создавать и использовать криптографический провайдер geli.

Поскольку в процессе настройки возникнет необходимость внесения изменений в конфигурацию ядра, потребуются также привилегии суперпользователя.

1. Добавление поддержки geli в ядро

   Добавьте в конфигурационный файл ядра следующие строки:

   ```
   options GEOM_ELI
   device crypto
   ```

   Перестройте ядро, как описано в разделе Настройка ядра FreeBSD.

   Помимо этого, поддержка geli может быть активирована модулем ядра на этапе загрузки. Для этого добавьте в файл /boot/loader.conf строку:

   ```
   geom_eli_load="YES"
   ```

Теперь ядро должно поддерживать geli(8).

2. Генерация главного ключа

   Предлагаемый пример описывает процесс генерации ключевого файла, который послужит частью главного ключа для шифрованного провайдера, монтируемого в каталог /private. При помощи содержимого ключевого файла создается набор случайных данных, которым зашифровывается главный ключ. Кроме того, он будет защищен кодовой фразой. Размер сектора провайдера будет составлять 4kB. Наконец, мы обсудим, как присоединиться к провайдеру `geli`, создать на базе его файловую систему, как ее смонтировать и работать с ней, и, в заключение, как корректно завершить работу.

   Больший чем обычно размер сектора (как в нашем примере, 4 кБ) рекомендуется для увеличения производительности.

   Главный ключ будет защищен кодовой фразой; данные для ключевого файла берутся из /dev/random. Размер сектора создаваемого нами шифрованного провайдера /dev/da2.eli - 4кБ.

```
# dd if=/dev/random of=/root/da2.key bs=64 count=1
# geli init -s 4096 -K /root/da2.key /dev/da2
Enter new passphrase:
Reenter new passphrase:
```

   Использование одновременно кодовой фразы и ключевого файла не обязательно: любой из этих методов защиты главного ключа может применяться независимо.

   Если в качестве имени ключевого файла указан "-", используется стандартный ввод. Это позволяет использовать более одного ключевого файла:

```
# cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
```

3. Свяжите сгенерированный ключ с провайдером

```
# geli attach -k /root/da2.key /dev/da2
Enter passphrase:
```

   Созданный при этом файл дискового устройства будет называться /dev/da2.eli.

```
# ls /dev/da2*
/dev/da2  /dev/da2.eli
```

4. Создайте новую файловую систему

```
# dd if=/dev/random of=/dev/da2.eli bs=1m
# newfs /dev/da2.eli
# mount /dev/da2.eli /private
```

Зашифрованная файловая система будет видна в выводе утилиты df(1) и готова к использованию:

```
# df -H
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a    248M    89M   139M    38%    /
/devfs         1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
/dev/ad0s1d    989M   1.5M   909M     0%    /tmp
/dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
/dev/da2.eli   150G   4.1K   138G     0%    /private
```

5. Размонтирование и деактивация провайдера

   После завершения работы с шифрованным разделом, когда содержимое каталога /private больше не нужно, будет разумным отключить раздел от системы.

```
# umount /private
# geli detach da2.eli
```

Дополнительную информацию о geli(8) можно найти на соответствующей странице справочника.

16.16.2.1. Использование стартового скрипта rc.d geli

Для удобства использования подсистемы `geli` в комплект базовой системы FreeBSD входит стартовый скрипт, работой которого можно управлять из rc.conf(5):

```
geli_devices="da2"
geli_da2_flags="-p -k /root/da2.key"
```

При этом дисковый раздел /dev/da2 будет сконфигурирован как провайдер `geli`, связан с ключевым файлом /root/da2.key, а кодовая фраза не будет использоваться (отметим, что это возможно только в том случае, если при инициализации `geli init` был указан ключ `-P`). Шифрованный провайдер `geli` будет отсоединен перед выключением системы.

Дополнительную информацию о конфигурации скриптов rc.d можно найти в соответствующей главе Руководства.

16.17. Шифрование области подкачки

Шифрование области подкачки в FreeBSD достаточно легко конфигурируется. Варианты конфигурации слегка различаются в зависимости от версии системы. Для шифрования разделов подкачки можно использовать утилиты gbde(8) или geli(8);. В обоих случаях используется скрипт rc.d encswap.

Предыдущий раздел, Шифрование дисковых разделов, кратко описывает различные методы криптования.

16.17.1. Зачем шифровать область подкачки?

Как и в случае дисковых разделов, шифрование области подкачки применяется для защиты важной информации. Возьмем, к примеру, приложение, которому требуется работать с паролями. До тех пор, пока пароли хранятся в физической памяти, все в порядке. Если же операционная система начинает выгружать отдельные участки памяти в область подкачки, чтобы освободить память для других приложений, пароли могут быть записаны на диск в открытом виде и тем самым оказаться легко доступными злоумышленнику (имеющему физический доступ к диску - прим. пер.). В таких ситуациях решением может стать шифрование раздела подкачки.

16.17.2. Подготовка

В данном разделе мы будем считать, что разделом подкачки является ad0s1b.

До настоящего момента раздел подкачки не был зашифрован. Таким образом, на нем могут содержаться пароли или какая-либо иная важная информация в открытом виде. Чтобы избавиться от этого, заполним раздел подкачки случайными данными:

```
# dd if=/dev/random of=/dev/ad0s1b bs=1m
```

16.17.3. Шифрование раздела подкачки при помощи gbde(8)

В строку файла /etc/fstab, описывающую раздел подкачки, необходимо добавить суффикс `.bde`:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1b.bde         none            swap    sw              0       0
```

16.17.4. Шифрование раздела подкачки при помощи geli(8)

Процедура при использовании geli(8) для шифрования раздела подкачки сходна с использованием gbde(8). В строку файла /etc/fstab, описывающую раздел подкачки, нужно добавить суффикс `.eli`:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1b.eli         none            swap    sw              0       0
```

По умолчанию, geli(8) использует алгоритм криптования AES с длиной ключа 256 бит.

При необходимости эти параметры могут быть изменены в опции `geli_swap_flags` файла конфигурации /etc/rc.conf. Приведенная ниже строка указывает, что скрипт rc.d encswap должен использовать для криптования алгоритм Blowfish с ключом длиной 128 бит, размером сектора 4 килобайта и включенной опцией "отсоединиться при последнем закрытии":

```
geli_swap_flags="-e blowfish -l 128 -s 4096 -d"
```

За списком возможных опций обращайтесь к описанию команды `onetime` в странице справочника geli(8).

16.17.5. Окончательная проверка

После перезагрузки системы правильность работы шифрованного раздела подкачки может быть проверена при помощи команды `swapinfo`.

В случае использования gbde(8):

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.bde    542720        0   542720     0%
```

При использовании geli(8):

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.eli    542720        0   542720     0%
```

Глава 17. GEOM: Модульная инфраструктура преобразования дисковых запросов

17.1. Краткий обзор

Эта глава описывает использование дисков, управляемых инфраструктурой GEOM во FreeBSD. Среди прочего, здесь описывается большая часть утилит управления RAID, использующих GEOM для настройки. В этой главе мы не будем вдаваться в подробности взаимодействия GEOM с подсистемой ввода/вывода или с программным кодом, эту информацию вы можете получить на странице справочника geom(4). Эта глава также не является подробным руководством по настройке RAID. Мы обсудим только типы RAID, поддерживаемые GEOM.

После прочтения этой главы вы будете знать:

- Какие типы RAID поддерживает GEOM.
- Как использовать стандартные утилиты для настройки, обслуживания и управления различными уровнями RAID.
- Как с помощью GEOM создавать зеркальные, последовательные и шифрованные дисковые последовательности, а так же последовательности из дисков, присоединённых удалённо.
- Как решать проблемы с дисками, присоединёнными к инфраструктуре GEOM.

Перед чтением этой главы вы должны:

- Понимать, как FreeBSD работает с дисками (Устройства хранения).
- Уметь сконфигурировать и установить новое ядро FreeBSD (Настройка ядра FreeBSD).

17.2. Введение в GEOM

GEOM позволяет классам - MBR, BSD labels, и так далее - получить доступ к устройству и управлять им, используя поставщиков GEOM (providers) или специальные файлы устройств, расположенные в каталоге /dev. GEOM поддерживает различные программные конфигурации RAID, и прозрачно предоставляет доступ к дискам системе и системным приложениям.

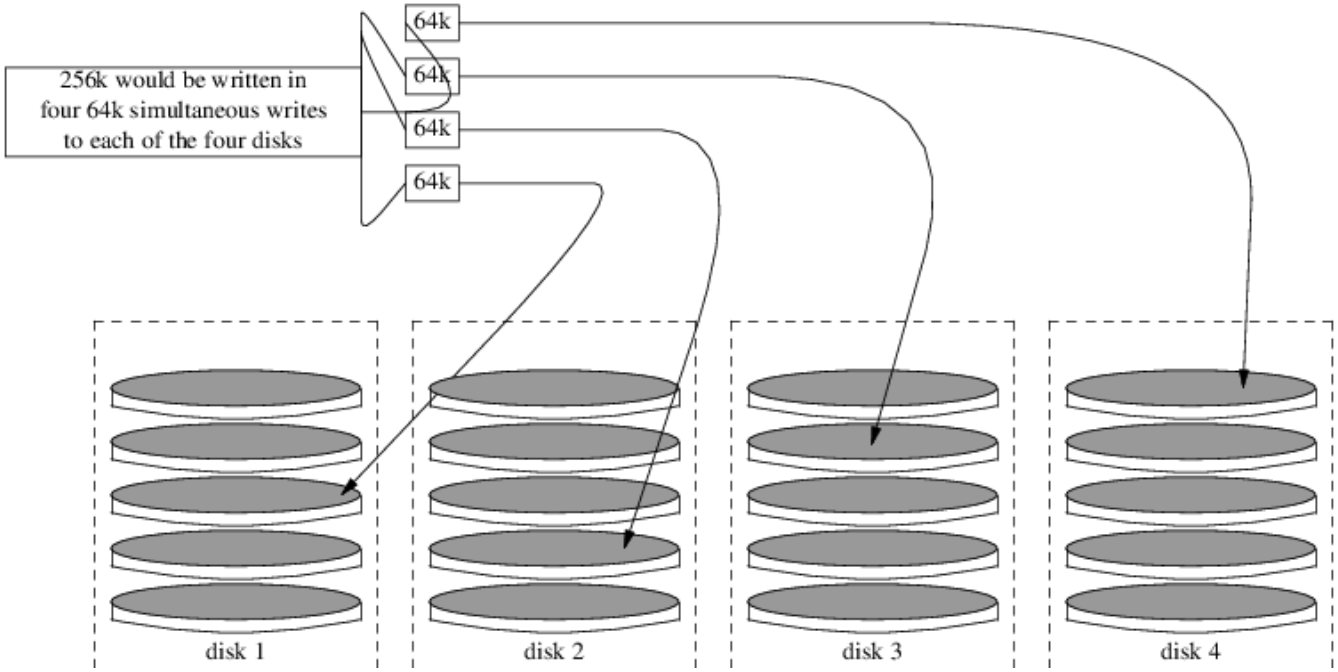
17.3. RAID0 - Создание дисковой последовательности (Striping)

Создание дисковой последовательности (Striping) - метод, применяемый, чтобы скомбинировать несколько физических дисков в один логический. Во многих случаях это делается с использованием аппаратных контроллеров. Дисковая подсистема GEOM

предоставляет программную поддержку RAID0, иногда называемую дисковой последовательностью (Stripe).

В RAID уровня 0 данные разбиваются на блоки, которые параллельно записываются на все диски массива. Вместо того, чтобы ждать записи 256k на один диск, RAID0 может параллельно записывать по 64k на каждый из четырёх дисков, обеспечивая более высокую производительность ввода/вывода. Производительность также может быть увеличена за счет использования большего числа дисков.

Все диски последовательности RAID0 должны быть одного размера, так как запись и считывание с дисков происходят параллельно.



Procedure: Создание дисковой последовательности из неформатированных ATA дисков

1. Загрузите модуль geom_stripe.ko:

```
# kldload geom_stripe
```

2. Убедитесь, что существует подходящая точка монтирования. Если вы планируете сделать логический диск корневым разделом, используйте временную точку монтирования, например /mnt:

```
# mkdir /mnt
```

3. Определите имена устройств, которые будут объединены в последовательность, и создайте новое устройство для последовательности. Например, чтобы создать дисковую последовательность из двух неиспользуемых и неразмеченных ATA дисков, например /dev/ad2 и /dev/ad3:

```
# gstripe label -v st0 /dev/ad2 /dev/ad3
Metadata value stored on /dev/ad2.
Metadata value stored on /dev/ad3.
Done.
```

4. Запишите стандартную метку, также известную как таблица разделов, в новый том, и установите стандартный загрузчик:

```
# bsdlabel -wB /dev/stripe/st0
```

5. Теперь в /dev/stripe кроме st0 появились ещё два устройства - st0a и st0c. Теперь создайте файловую систему на устройстве st0a, используя утилиту `newfs`:

```
# newfs -U /dev/stripe/st0a
```

На экране промелькнет множество цифр, и через несколько секунд процесс будет завершен. Логический диск создан и готов к монтированию.

Смонтируйте его вручную:

```
# mount /dev/stripe/st0a /mnt
```

Чтобы монтировать созданную дисковую последовательность автоматически во время загрузки, добавьте информацию о ней в файл /etc/fstab. Создайте постоянную точку монтирования и назовите её, к примеру, stripe:

```
# mkdir /stripe
# echo "/dev/stripe/st0a /stripe ufs rw 2 2" \
    >> /etc/fstab
```

Чтобы модуль geom_stripe.ko автоматически загружался во время инициализации системы, добавьте строку в /boot/loader.conf:

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

17.4. RAID1 - Зеркалирование (Mirroring)

Зеркалирование (Mirroring) - технология, применяемая как в корпоративной среде, так и на домашних компьютерах. Она позволяет создавать резервные копии "на лету". Зеркалирование, по сути, означает, что диск A является копией диска B. Или, возможно, диск C+D является копией диска A+B. Вне зависимости от конфигурации, основной аспект -

дублирование информации. Позже, эта информация может быть с легкостью восстановлена или сохранена как резервная копия без остановки системы, или даже физически помещена в хранилище данных.

Перед началом, убедитесь, что у вас есть два физических диска равной емкости. Далее в этом примере подразумевается, что это диски прямого доступа (direct access, da(4)) с интерфейсом SCSI.

17.4.1. Зеркалирование первичных дисков

В статье предполагается, что FreeBSD установлена на первый жесткий диск, определяемый системой как da0. Это устройство будет целевым для утилиты gmirror(8).

Перед построением зеркала включите дополнительную отладочную информацию и откройте доступ к устройству. Это достигается установкой следующего значения переменной sysctl(8) `kern.geom.debugflags`:

```
# sysctl kern.geom.debugflags=17
```

Теперь создайте зеркало. Начните процесс с сохранения метаданных на первом диске. В результате выполнения следующей команды будет создано устройство вида /dev/mirror/gm:

> Создание зеркала на диске, с которого произведена загрузка, может повлечь за собой потерю данных в том случае, если данными занят последний сектор диска. Риск повреждения данных меньше, если создание зеркала немедленно следует за свежей установкой FreeBSD.

```
# gmirror label -vb round-robin gm0 /dev/da0
```

Система должна выдать следующее сообщение:

```
Metadata value stored on /dev/da0.
Done.
```

Инициализируйте GEOM, это повлечет за собой загрузку модуля ядра /boot/kernel/geom_mirror.ko:

```
# gmirror load
```

> После успешного завершения команды будет создано устройство gm0 в каталоге /dev/mirror.

Включите автоматическую загрузку модуля geom_mirror.ko во время старта операционной системы:

```
# echo 'geom_mirror_load="YES"' >> /boot/loader.conf
```

Отредактируйте файл /etc/fstab, заменив в нём упоминания старого имени устройства da0 новым именем устройства зеркала gm0.

Если vi(1) - ваш любимый текстовый редактор, то эта задача решается просто:

```
# vi /etc/fstab
```

Сделайте резервную копию файла fstab, набрав в vi(1) `:w /etc/fstab.bak`. Затем замените все части строк, содержащие имя устройства da0, на имя gm0, набрав `:%s/da/mirror\/gm/g`.

Независимо от аппаратного интерфейса дисков (SCSI или ATA), устройство RAID будет именоваться всегда одинаково - gm. Содержимое файла fstab должно выглядеть подобно следующему:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/mirror/gm0s1b      none            swap    sw              0       0
/dev/mirror/gm0s1a      /               ufs     rw              1       1
/dev/mirror/gm0s1d      /usr            ufs     rw              0       0
/dev/mirror/gm0s1f      /home           ufs     rw              2       2
#/dev/mirror/gm0s2d     /store          ufs     rw              2       2
/dev/mirror/gm0s1e      /var            ufs     rw              2       2
/dev/acd0               /cdrom          cd9660  ro,noauto       0       0
```

Перезагрузите систему:

```
# shutdown -r now
```

С этого момента во время каждой загрузки система должна использовать устройство gm0 вместо устройства da0. Удостовериться в этом можно так: дождитесь загрузки системы, наберите команду `mount` и просмотрите её вывод:

```
# mount
Filesystem          1K-blocks    Used    Avail Capacity  Mounted on
/dev/mirror/gm0s1a    1012974  224604   707334    24%    /
devfs                       1       1        0   100%    /dev
/dev/mirror/gm0s1f   45970182   28596 42263972     0%    /home
/dev/mirror/gm0s1d    6090094 1348356  4254532    24%    /usr
/dev/mirror/gm0s1e    3045006 2241420   559986    80%    /var
devfs                       1       1        0   100%    /var/named/dev
```

Как и ожидалось, вывод выглядит корректно. И в заключение, чтобы начать синхронизацию данных, включите в зеркало диск da1 при помощи следующей команды:

```
# gmirror insert gm0 /dev/da1
```

Во время построения зеркала статус процесса построения может быть проверен следующей командой:

```
# gmirror status
```

Вывод вышеприведённой команды для построенного и синхронизированного зеркала выглядит подобно следующему:

```
      Name    Status  Components
mirror/gm0  COMPLETE  da0
                      da1
```

Если есть какие-либо неполадки или зеркало находится в процессе построения, в выводе команды будет обозначен статус `DEGRADED` вместо статуса `COMPLETE`.

17.4.2. Решение проблем

17.4.2.1. Система не загружается

Если система прекращает загрузку и выдает строку:

```
ffs_mountroot: can't find rootvp
Root mount failed: 6
mountroot>
```

Перезагрузите компьютер кнопкой питания или кнопкой "Reset". В загрузочном меню выберите опцию (6). Это приведет к тому, что система выдаст приглашение loader(8). Загрузите модуль ядра вручную:

```
OK? load geom_mirror
OK? boot
```

Если это сработало, модуль ядра по какой-либо причине не загрузился правильно. Проверьте корректность соответствующей записи в /boot/loader.conf. Если проблема осталась, добавьте строку:

```
options GEOM_MIRROR
```

в файл конфигурации ядра, пересоберите и переустановите ядро. Это должно устранить проблему.

17.4.3. Восстановление после дисковых сбоев

Примечательной особенностью зеркалирования является то, что если диск вышел из строя, то он, пожалуй, может быть заменён вообще без ущерба для данных.

Принимая во внимание предыдущую конфигурацию RAID1, предположим, что устройство da1 вышло из строя, и ему требуется замена. Перед заменой определите, какой именно диск вышел из строя, а потом выключите систему. Теперь дефектный диск может быть заменён новым, после чего необходимо снова загрузить систему. После загрузки системы для замещения диска в зеркале могут быть использованы следующие команды:

```
# gmirror forget gm0
```

```
# gmirror insert gm0 /dev/da1
```

Для наблюдения за статусом построения используйте команду `gmirror status`. Вывод этой команды достаточно прост и понятен.

17.5. Сетевые устройства GEOM Gate

GEOM включает в себя поддержку работы с удаленными устройствами по сети, например с дисками, CD-ROM и т.д. путем использования gate утилит. Это подобно работе с NFS.

Для начала необходимо создать файл экспорта. В этом файле указывается, кому разрешен доступ к экспортируемым ресурсам и какой уровень доступа предоставляется. Например для того, чтобы экспортировать четвертый слайс первого SCSI диска, достаточно следующей записи в файле /etc/gg.exports:

```
192.168.1.0/24 RW /dev/da0s4d
```

Это позволит всем компьютерам внутри частной сети получить доступ к разделу da0s4d.

Чтобы экспортировать устройство, убедитесь, что оно не смонтировано, и запустите сервер ggated(8):

```
# ggated
```

Теперь, чтобы смонтировать устройство на клиентском компьютере выполните следующие команды:

```
# ggatec create -o rw 192.168.1.1 /dev/da0s4d
```

```
ggate0
# mount /dev/ggate0 /mnt
```

С этого момента устройство доступно в точке монтирования /mnt.

Необходимо заметить, что попытка смонтировать устройство, уже смонтированное как сетевой или локальный диск, закончится неудачей.

Когда устройство больше не нужно, оно может быть размонтировано командой umount(8), как любое другое дисковое устройство.

17.6. Метки дисковых устройств

Во время загрузки системы, ядро FreeBSD создает файлы для обнаруженных устройств. Этот метод обнаружения устройств создает некоторые проблемы, например если новое дисковое устройство подключается через USB. Может получиться так, что этому диску будет присвоено имя устройства da0, а устройство с прежним именем da0 получит следующее имя, da1. Это приведет к проблемам монтирования файловых систем, записанных в /etc/fstab. На самом деле, это может даже помешать загрузке системы.

Одно из решений состоит в расположении SCSI устройств в таком порядке, чтобы новые устройства, добавляемые к SCSI контроллеру, занимали свободные номера устройств. Но что делать с USB устройствами, которые могут занять место основного SCSI диска? Это случается потому, что USB устройства обычно тестируются до SCSI контроллера. Решение может состоять в подключении этих устройств после загрузки системы. Другое решение - использование ATA диска и исключение SCSI устройств из /etc/fstab.

Есть и лучшее решение. С помощью утилиты `glabel`, администратор или пользователь могут пометить дисковые устройства и использовать эти метки в /etc/fstab. Поскольку `glabel` сохраняет метки в последнем секторе заданного устройства, они сохраняются и после перезагрузки. Используя эти метки вместо имени устройств, можно всегда смонтировать файловую систему независимо от назначенного имени устройства.

Очевидно, что метки должны быть постоянными. Утилита `glabel` может использоваться для создания как временных, так и постоянных меток. Только постоянные метки сохраняются после перезагрузок. Прочтите glabel(8) для получения более подробной информации о различии между метками.

17.6.1. Типы меток и примеры

Существует два типа меток, основной (generic) тип и метки файловой системы. Метки могут быть постоянными или временными. Постоянные метки создаются командой tunefs(8) или newfs(8). В дальнейшем они будут автоматически создаваться в подкаталоге каталога /dev, имя которого определяется в соответствии с типом файловой системы. Например, метки файловых систем UFS2 будут расположены в каталоге /dev/ufs. Постоянные метки также можно создать при помощи команды `glabel label`. Эти метки не зависят от типа файловой

системы, поэтому они будут перечисляться в каталоге /dev/label.

Временные метки не сохраняются после перезагрузки. Эти метки создаются в каталоге /dev/label, они хорошо подходят для экспериментов. Временную метку можно создать командой `glabel create`. За более детальной информацией обратитесь к странице справочника glabel(8).

Чтобы создать постоянную метку для файловой системы UFS2 не нарушая самих данных, выполните следующую команду:

```
# tunefs -L home /dev/da3
```

Если файловая система заполнена, это может привести к повреждению данных; в случае заполненной файловой системы надо или удалить ненужные файлы, или не добавлять метки.

Метка должна появиться в /dev/ufs и может быть добавлена в /etc/fstab:

```
/dev/ufs/home       /home           ufs     rw              2       2
```

Во время запуска `tunefs` файловая система не должна быть смонтирована.

Теперь файловую систему можно смонтировать как обычно:

```
# mount /home
```

Если модуль ядра geom_label.ko указан в /boot/loader.conf и загружается вместе с системой, или в ядре указана опция `GEOM_LABEL`, метку устройства можно изменять без какого-либо негативного для системы эффекта.

Файловая система может быть создана с меткой по умолчанию путем использования флага `-L` команды `newfs`. Обратитесь к странице справочника newfs(8) за более подробной информацией.

Для удаления метки можно использовать следующую команду:

```
# glabel destroy home
```

В следующем примере показано, как устанавливаются метки на разделы загрузочного диска.

Пример 33. Установка меток на разделы загрузочного диска

Установка и задействование постоянных меток на разделах загрузочного диска предоставит возможность операционной системе загружаться нормально в том случае,

если диск был переключен на другой контроллер, или даже переставлен на другую машину. В этом примере был задействован один диск ATA, определяемый системой как ad0. Также в примере подразумевается, что система использует типичную для FreeBSD схему разделения дискового пространства на слайсы и размещения на них файловых систем /, /var, /usr, /tmp и раздела подкачки.

Перезагрузите систему, дождитесь меню загрузчика. Нажатием клавиши `4` выберите однопользовательский режим. Далее, введите следующие команды:

```
# glabel label rootfs /dev/ad0s1a
GEOM_LABEL: Label for provider /dev/ad0s1a is label/rootfs
# glabel label var /dev/ad0s1d
GEOM_LABEL: Label for provider /dev/ad0s1d is label/var
# glabel label usr /dev/ad0s1f
GEOM_LABEL: Label for provider /dev/ad0s1f is label/usr
# glabel label tmp /dev/ad0s1e
GEOM_LABEL: Label for provider /dev/ad0s1e is label/tmp
# glabel label swap /dev/ad0s1b
GEOM_LABEL: Label for provider /dev/ad0s1b is label/swap
# exit
```

Система продолжит загрузку в многопользовательский режим. По завершении загрузки откройте файл /etc/fstab и замените в нём традиционные имена файлов устройств на соответствующие устройствам метки. Результат будет выглядеть подобно следующему:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/label/swap         none            swap    sw              0       0
/dev/label/rootfs       /               ufs     rw              1       1
/dev/label/tmp          /tmp            ufs     rw              2       2
/dev/label/usr          /usr            ufs     rw              2       2
/dev/label/var          /var            ufs     rw              2       2
```

Перезагрузите еще раз систему. Если всё прошло успешно, система загрузится как обычно, а вывод команды `mount` отобразит следующее:

```
# mount
/dev/label/rootfs on / (ufs, local)
devfs on /dev (devfs, local)
/dev/label/tmp on /tmp (ufs, local, soft-updates)
/dev/label/usr on /usr (ufs, local, soft-updates)
/dev/label/var on /var (ufs, local, soft-updates)
```

Начиная с FreeBSD 7.2, GEOM класс glabel(8) поддерживает новый тип меток для файловых систем UFS. Новый тип меток базируется на уникальных идентификаторах файловых систем, называемых `ufsid`. Во время загрузки системы они автоматически создаются и

помещаются в каталог /dev/ufsid. Перечисление меток должным образом в файле /etc/fstab делает возможным монтирование разделов по значениям `ufsid`. Чтобы получить перечень файловых систем и соответствующих им меток `ufsid`, выполните команду `glabel status`:

```
% glabel status
                   Name  Status  Components
ufsid/486b6fc38d330916     N/A  ad4s1d
ufsid/486b6fc16926168e     N/A  ad4s1f
```

В данном примере ad4s1d содержит файловую систему /var, а ad4s1f соответствует файловой системе /usr. Эти файловые системы можно также монтировать, указав значения их `ufsid` в файле /etc/fstab:

```
/dev/ufsid/486b6fc38d330916        /var        ufs        rw        2      2
/dev/ufsid/486b6fc16926168e        /usr        ufs        rw        2      2
```

Таким способом могут быть смонтированы любые разделы с метками `ufsid`, что исключает необходимость создания постоянных меток вручную и в то же время позволяет воспользоваться преимуществами монтирования по меткам.

17.7. Журналирование UFS средствами GEOM

С выходом FreeBSD 7.0 был реализован долгожданный механизм ведения журналов для файловых систем. Сама реализация этого механизма осуществляется средствами системы GEOM, а конфигурирование выполняется утилитой gjournal(8).

Что такое журналирование? Журналирование сохраняет протокол транзакций файловой системы, то есть: изменения, составляющие логически завершенную операцию записи, сперва вносятся в журнал, а модификация метаданных и данных самого файла выполняется позже. В дальнейшем журнал может быть задействован для повторного выполнения транзакций на файловой системе с целью предотвращения нарушения целостности файловой системы.

Журналирование - это еще одним механизм предотвращения утери данных и нарушения целостности файловой системы. В отличие от механизма Soft Updates, который отслеживает и периодически сохраняет обновления метаданных, и механизма снэпшотов, который создает образ файловой системы, сам журнал хранится в специально отведенном для этой задачи пространстве диска, и, в некоторых случаях, может содержаться целиком на отдельном диске.

В отличие от других реализаций журналирования файловых систем, метод `gjournal` работает на блочном уровне, он не встроен в файловую систему; это лишь надстройка над системой GEOM.

Чтобы включить поддержку `gjournal`, в файле конфигурации ядра FreeBSD должна присутствовать следующая опция (включено по умолчанию для FreeBSD 7.0 и более поздних версий систем):

```
options UFS_GJOURNAL
```

Журналируемым устройствам, монтируемым во время загрузки системы, также потребуется модуль ядра geom_journal.ko. Внесите следующую запись в файл /boot/loader.conf:

```
geom_journal_load="YES"
```

В качестве альтернативы, функции вышеупомянутого модуля можно встроить в специализированное ядро. Для этого добавьте следующую опцию в файл конфигурации ядра:

```
options GEOM_JOURNAL
```

Для создания журнала на новой файловой системе выполните следующие шаги (здесь и далее подразумевается, что da4 есть новый SCSI диск):

```
# gjournal load
# gjournal label /dev/da4
```

На этом этапе в каталоге /dev должны присутствовать файлы устройств /dev/da4 и /dev/da4.journal. Теперь необходимо создать файловую систему:

```
# newfs -O 2 -J /dev/da4.journal
```

Предыдущая команда создаст файловую систему UFS2 на журналируемом устройстве.

Смонтируйте устройство в требуемый каталог файловой системы:

```
# mount /dev/da4.journal /mnt
```

В случае наличия нескольких слайсов, журнал создается для каждого из них. Например, если есть два слайса, и они называются ad4s1 и ad4s2, то утилитой `gjournal` создаются файлы устройств ad4s1.journal и ad4s2.journal.

Для увеличения производительности может потребоваться хранение журнала на отдельном диске. В таких случаях необходимо указать имя поставщика журнала или устройства хранения после имени устройства, на котором планируется включение журналирования. Журналирование также может быть активировано утилитой `tunefs` на действующих файловых системах; однако, всегда создавайте резервную копию перед попытками изменить настройки файловой системы. В большинстве случаев, выполнение команды `gjournal` завершится ошибкой, если создание журнала невозможно, в то время как

некорректное использование команды `tunefs` не защитит против потери данных.

Также возможно журналирование загрузочного диска системы FreeBSD. За детальными инструкциями по этой возможности обратитесь к статье Настройка журналирования UFS для настольного компьютера.

Глава 18. Поддержка файловых систем

18.1. Краткий обзор

Файловые системы - неотъемлемая часть любой операционной системы. Они позволяют пользователям записывать и хранить файлы, получать доступ к данным, и, конечно-же, пользоваться жесткими дисками. У разных операционных систем есть одна общая черта - их основная файловая система (native filesystem). Для FreeBSD это Fast File System (или FFS), которая произошла от Unix™ File System (сокращенно UFS).

FreeBSD также поддерживает ряд других файловых систем, тем самым предоставляя возможность получать доступ к данным от других операционных систем локально, например: к данным, находящимся на подключенных USB устройствах хранения, флэш-накопителях и жестких дисках. В списке поддерживаемых есть файловые системы, разработанные для других операционных систем, например Linux® Extended File System (EXT) и Sun™ Z File System (ZFS).

FreeBSD имеет разные уровни поддержки для разных файловых систем. Для некоторых будет достаточно загрузки модуля ядра, другим может потребоваться установка набора утилит (toolset). Цель этого раздела - дать представления пользователям FreeBSD о возможностях использования других файловых систем на их операционных системах. Начнем с Sun™ Z file system.

После прочтения этого раздела вы будете знать:

- Разницу между основной и поддерживаемой файловой системой.
- Какие файловые системы поддерживаются FreeBSD.
- Как подключить, сконфигурировать, получить доступ и использовать поддерживаемые файловые системы.

Перед прочтением этого раздела вам необходимо:

- Понимать основы UNIX® и FreeBSD (Основы UNIX).
- Знать азы конфигурирования и компиляции ядра (Настройка ядра FreeBSD).
- Уметь устанавливать приложения сторонних разработчиков в FreeBSD (Установка приложений. порты и пакеты).
- Быть знакомым с именованием дисков и устройств хранения в FreeBSD (Устройства хранения).

18.2. Файловая система ZFS

Файловая система ZFS, разработанная компанией Sun™, основана на использовании метода пулов устройств хранения данных. Это значит, что емкость носителя занимается только тогда, когда она становится необходимой для сохранения данных. ZFS также была разработана с упором на максимальную целостность данных, поддерживая снимки (snapshot), множество копий и контрольные суммы данных. Новая модель репликации

данных, известная как RAID-Z, подобна RAID-5, но специально разработана для предотвращения повреждений данных при записи.

18.2.1. Настройка ZFS

Подсистема ZFS занимает значительную часть ресурсов системы. Чтобы получить от нее максимум эффективности в повседневном использовании, потребуется выполнить некоторые настройки. ZFS является экспериментальной функциональной возможностью в FreeBSD, но ситуация может измениться в ближайшем будущем; однако на данный момент рекомендуется выполнить следующие шаги.

18.2.1.1. Память

Общий размер ОЗУ должен быть как минимум равен одному гигабайту, хотя рекомендуется два гигабайта или более. Во всех нижеследующих примерах используется система с 1ГБ памяти совместно с другими специальными настройками.

Известно, что некоторые пользователи преуспели в использовании ZFS на системах, имеющих менее одного гигабайта памяти, но с таким ограниченным объемом ОЗУ и при серьезной загрузке машины очень вероятны паники FreeBSD из-за нехватки памяти.

18.2.1.2. Настройка ядра

Рекомендуется исключить из файла конфигурации ядра неиспользуемые драйвера и опции. Так как большинство драйверов устройств доступно в виде модулей, то они просто могут быть загружены с помощью соответствующих записей в файле /boot/loader.conf.

Пользователям архитектуры i386™ необходимо добавить следующую опцию в их файл конфигурации ядра, перестроить ядро и перезагрузиться:

```
options         KVA_PAGES=512
```

Эта опция расширит адресное пространство ядра, тем самым позволяя переменной `vm.kvm_size` быть установленной за текущий предел в 1 ГБ (2 ГБ для PAE). Чтобы найти наиболее подходящее значение для этой опции, разделите имеющийся объем ОЗУ, выраженный в мегабайтах, на 4. Приведенное выше значение `512` рекомендуется для систем с 2 ГБ оперативной памяти.

18.2.1.3. Параметры loader.conf

Адресное пространство kmem должно быть увеличено на всех FreeBSD архитектурах. На тестовой системе с одним гигабайтом физической памяти стабильная работа была получена со следующими параметрами, которые необходимо внести в файл /boot/loader.conf и перезагрузить систему.

```
vm.kmem_size="330M"
vm.kmem_size_max="330M"
vfs.zfs.arc_max="40M"
```

```
vfs.zfs.vdev.cache.size="5M"
```

За более детальными рекомендациями по тонкой настройке системы под ZFS, обратитесь к странице: http://wiki.freebsd.org/ZFSTuningGuide.

18.2.2. Использование ZFS

Существует стартовый механизм, позволяющий монтировать ZFS пулы во время инициализации системы. Чтобы его задействовать, выполните следующие команды:

```
# echo 'zfs_enable="YES"' >> /etc/rc.conf
# /etc/rc.d/zfs start
```

Здесь и далее в статье подразумевается, что в системе установлено три SCSI диска с именами устройств da0, da1 и da2. Использующим IDE диски необходимо подставить имена устройств ad вместо имен устройств SCSI.

18.2.2.1. Простой дисковый пул

Для создания простого пула ZFS без избыточности, задействовав при этом один жесткий диск, воспользуйтесь командой `zpool`:

```
# zpool create example /dev/da0
```

Чтобы увидеть новый пул, просмотрите вывод команды `df`:

```
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235230  1628718    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032846 48737598     2%    /usr
example      17547136       0 17547136     0%    /example
```

Этот вывод четко показывает, что пул `example` был не только создан, но также и *примонтирован*. Он также доступен, как и обычная файловая система, в нем можно создавать файлы, а пользователи могут просматривать его содержимое, например:

```
# cd /example
# ls
# touch testfile
# ls -al
total 4
drwxr-xr-x   2 root  wheel    3 Aug 29 23:15 .
drwxr-xr-x  21 root  wheel  512 Aug 29 23:12 ..
-rw-r--r--   1 root  wheel    0 Aug 29 23:15 testfile
```

Однако в этом примере простого пула не задействованы никакие функциональные возможности ZFS. Создайте файловую систему в этом пуле и активируйте сжатие данных на ней:

```
# zfs create example/compressed
# zfs set compression=gzip example/compressed
```

С этого момента для файловой системы ZFS `example/compressed` активировано сжатие данных. Попробуйте поместить на нее несколько больших файлов копируя их в /example/compressed.

А вот как можно отключить сжатие данных:

```
# zfs set compression=off example/compressed
```

Для того чтобы размонтировать файловую систему, выполните следующую команду и проверьте результат утилитой `df`:

```
# zfs umount example/compressed
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235232  1628716    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032864 48737580     2%    /usr
example      17547008       0 17547008     0%    /example
```

Снова смонтируйте файловую систему и проверьте результат при помощи `df`:

```
# zfs mount example/compressed
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
```

Пул и файловая система также отображается в выводе команды `mount`:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
example on /example (zfs, local)
example/data on /example/data (zfs, local)
example/compressed on /example/compressed (zfs, local)
```

Как вы уже убедились, файловые системы ZFS после создания могут использоваться как и обычные файловые системы; однако доступно множество других возможностей. В следующем примере мы создадим новую файловую систему `data`. На ней мы будем содержать важные данные, поэтому файловая система сконфигурирована хранить две копии каждого блока:

```
# zfs create example/data
# zfs set copies=2 example/data
```

Снова проверьте свободное и использованное место выполнив команду `df`:

```
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
example/data        17547008       0 17547008     0%    /example/data
```

Заметьте, что каждая файловая система в пуле имеет тот же объем свободного места. Мы использовали команду `df` на протяжении этих примеров, чтобы показать, что файловые системы занимают только необходимое им пространство, используя ресурс одного и того же пула. ZFS уходит от привычных понятий "том (volume)" и "раздел (partition)", позволяя файловым системам занимать один и тот же пул. Уничтожьте файловые системы, потом уничтожьте пул, так как в них уже нет нужды:

```
# zfs destroy example/compressed
# zfs destroy example/data
# zpool destroy example
```

Жесткие диски со временем выходят из строя, это неизбежно. Когда этот диск выйдет из строя, данные будут утеряны. Одним из способов избежать потери данных из-за вышедшего из строя жесткого диска является построение RAID массивов. ZFS поддерживает эту функциональную возможность в своем дизайне, и это описано в следующем разделе.

18.2.2.2. ZFS RAID-Z

Как уже было сказано выше, в этой статье подразумевается, что в нашей системе в распоряжении есть три SCSI диска: da0, da1 и da2 (или ad0 и далее в случае IDE дисков). Для того, чтобы создать RAID-Z пул, выполните следующую команду:

```
# zpool create storage raidz da0 da1 da2
```

Sun™ рекомендует использовать от трех до девяти жестких дисков в

> конфигурации RAID-Z. Если есть необходимость в использовании 10 или более дисков, подумайте над тем, чтобы разбить их на меньшие группы RAID-Z. Если у вас есть только два диска и вам всё-таки требуется избыточность, возможно лучшим вариантом будет создание ZFS зеркала. Смотрите страницу справочника zpool(8) для получения более подробных сведений.

По завершении команды должен создаться пул `storage`. Как и прежде, это может быть проверено при помощи команд mount(8) и df(1). Больше дисковых устройств может быть задействовано путем добавления их в конец списка параметров команды, приведенной выше. Создайте в пуле новую файловую систему, называемую `home`, в которой будут размещаться пользовательские файлы:

```
# zfs create storage/home
```

На данном этапе возможно активировать сжатие данных и организовать автоматическое создание копий пользовательских домашних каталогов и файлов. Это может быть достигнуто так же, как и ранее, при помощи следующих команд:

```
# zfs set copies=2 storage/home
# zfs set compression=gzip storage/home
```

Чтобы организовать в этой файловой системе хранение домашних каталогов пользователей, скопируйте сюда их содержимое и создайте соответствующие символические ссылки:

```
# cp -rp /home/* /storage/home
# rm -rf /home /usr/home
# ln -s /storage/home /home
# ln -s /storage/home /usr/home
```

С этого момента пользовательские данные сохраняются на новой файловой системе /storage/home. Для проверки создайте учетную запись нового пользователя и войдите ею в систему.

Попробуйте создать снимок (snapshot), к которому можно будет откатиться при необходимости:

```
# zfs snapshot storage/home@08-30-08
```

Заметьте, что снимок (snapshot) захватит реальную файловую систему, а не домашний каталог или файл. Символ `@` отделяет имя файловой системы или имя тома от имени снимка. Когда возникнет необходимость восстановить пользовательские домашние каталоги, выполните следующую команду:

```
# zfs rollback storage/home@08-30-08
```

Чтобы получить список имеющихся в наличии снимков, выполните команду `ls` в каталоге .zfs/snapshot. Например, чтобы увидеть сделанный ранее снимок, выполните следующую команду:

```
# ls /storage/home/.zfs/snapshot
```

Можно написать скрипт, выполняющий снимки пользовательских данных ежемесячно; однако, со временем, они могут занять значительную часть дискового пространства. Предыдущий снимок может быть удален используя следующую команду:

```
# zfs destroy storage/home@08-30-08
```

Нет причины после наших экспериментов далее держать в текущем состоянии /storage/home. Сделаем ее реальной файловой системой /home:

```
# zfs set mountpoint=/home storage/home
```

Выполнение команд `df` и `mount` покажет, что с этого момента операционная система воспринимает нашу файловую систему как обычную /home:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
storage on /storage (zfs, local)
storage/home on /home (zfs, local)
# df
Filesystem   1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a    2026030  235240  1628708    13%    /
devfs                1       1        0   100%    /dev
/dev/ad0s1d   54098308 1032826 48737618     2%    /usr
storage       26320512       0 26320512     0%    /storage
storage/home  26320512       0 26320512     0%    /home
```

На этом завершим конфигурацию RAID-Z. Чтобы во время ночных запусков periodic(8) получать информацию о статусе созданных файловых систем, выполните следующую команду:

```
# echo 'daily_status_zfs_enable="YES"' >> /etc/periodic.conf
```

18.2.2.3. Восстановление RAID-Z

Каждая система программных RAID массивов предоставляет возможность отображать информацию о своем состоянии. ZFS - не исключение. Статус устройств RAID-Z может быть просмотрен при помощи следующей команды:

```
# zpool status -x
```

Если пулы исправны и всё нормально, возвратится следующее сообщение:

```
all pools are healthy
```

А если есть какие-то неполадки, например диск выведен из массива, возвращенное состояние пула будет подобным следующему:

```
  pool: storage
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
    Sufficient replicas exist for the pool to continue functioning in a
    degraded state.
action: Online the device using 'zpool online' or replace the device with
    'zpool replace'.
 scrub: none requested
config:

    NAME        STATE     READ WRITE CKSUM
    storage     DEGRADED     0     0     0
      raidz1    DEGRADED     0     0     0
        da0     ONLINE       0     0     0
        da1     OFFLINE      0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Вывод показывает, что устройство было переведено в автономный режим администратором. Это верно для данного отдельного примера. Чтобы перевести диск в автономный режим, была выполнена команда:

```
# zpool offline storage da1
```

Теперь после останова системы возможно заменить da1. Когда система загрузится снова, выполните следующую команду чтобы заменить диск в массиве:

```
# zpool replace storage da1
```

С этого момента статус может быть проверен опять и на этот раз без флага `-x`:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: resilver completed with 0 errors on Sat Aug 30 19:44:11 2008
config:

    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

В выводе сообщается, что при перестроении массива ошибок обнаружено не было.

18.2.2.4. Проверка данных

Как уже было сказано ранее, ZFS использует `контрольные суммы` для проверки целостности сохраненных данных. Подсчет и сохранение контрольных сумм включается автоматически во время создания файловых систем и может быть отключен при помощи команды:

```
# zfs set checksum=off storage/home
```

Отключение подсчета контрольных сумм - не очень хорошая идея; особенно ввиду того, что они занимают мало места, а также при их использовании нет существенных расходов ресурсов системы. Пока подсчет включен, возможно выполнять проверки целостности данных ZFS, используя контрольные суммы. Этот процесс известен как "очистка (scrubbing)". Чтобы проверить целостность данных пула `storage`, выполните следующую команду:

```
# zpool scrub storage
```

Этот процесс может занять значительное время в зависимости от количества сохранённых данных. Очистка (scrubbing) порождает интенсивный ввод/вывод, поэтому только один экземпляр этой операции может выполняться в один момент времени. После завершения очистки (scrubbing) статус обновится, его можно просмотреть выполнив следующий запрос:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: scrub completed with 0 errors on Sat Aug 30 19:57:37 2008
config:
```

```
    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Время завершения отображается в простом виде в этом примере. Очистка помогает удостовериться в целостности данных на протяжении длительного времени.

В этом разделе была освещена лишь малая часть возможностей ZFS. За более подробной информацией обратитесь к страницам справочника zfs(8) и zpool(8).

Глава 19. Локализация - использование и настройка i18n/L10n

19.1. Краткий обзор

FreeBSD - это распределенный проект, пользователи и контрибьюторы которого находятся в самых разных частях света. Поэтому FreeBSD поддерживает локализацию на многие языки, что позволяет просматривать, вводить и обрабатывать данные на языках, отличных от английского. Можно выбрать среди большинства основных языков, в том числе: китайский, немецкий, японский, корейский, французский, русский и вьетнамский.

Термин интернационализация (`internationalization`) сокращают до i18n, по числу символов в английском слове. Сокращение L10n аналогично получается от слова `localization`. i18n/L10n методы, протоколы и приложения позволяют пользователям использовать языки по своему выбору.

В этой главе обсуждаются особенности интернационализации и локализации FreeBSD, включая следующие темы:

- Схема именования локалей.
- Установка региональных настроек для оболочки.
- Локализация консоли.
- Локализация Xorg.
- Поиск i18n-совместимых приложений.
- Информация по настройке для некоторых языков.

Перед чтением этой главы вам следует знать:

- Как установить дополнительные приложения сторонних разработчиков.

19.2. Использование локализации

Настройки локализации состоят из трёх компонентов: код языка, код страны и кодировка. Из этих частей формируются названия локалей:

```
КодЯзыка_КодСтраны.Кодировка
```

КодЯзыка и *КодСтраны* используются для определения страны и конкретного диалекта языка. Основные коды языка и страны содержит некоторые примеры пар *КодЯзыка_КодСтраны*:

Таблица 9. Основные коды языка и страны

КодЯзыка_КодСтраны	Описание
en_US	Английский, Соединенные Штаты
ru_RU	Русский, Россия
zh_TW	Традиционный китайский, Тайвань

Полный список локалей доступен по команде:

```
% locale -a | more
```

Чтобы определить текущую локаль:

```
% locale
```

Специфичные для языков наборы символов ISO8859-1, ISO8859-15, KOI8-R, CP437 описаны в multibyte(3). Актуальный список наборов символов находится на сайте IANA Registry.

Некоторые языки, такие как китайский или японский, не могут быть представлены с использованием символов ASCII, и для них требуется дополненная языковая кодировка с использованием расширенного или многобайтового представления символов. Такими кодировками являются EUC и Big5. Старые приложения могут ошибочно принимать символы в таких кодировках за управляющие, в то время как новые обычно их распознают. В зависимости от реализации, пользователю может потребоваться компиляция приложения с поддержкой расширенного или многобайтового представления символов или правильная его настройка.

Во FreeBSD используются Xorg-совместимые кодировки.

В продолжении этого раздела рассматриваются различные способы настройки локализации в системе FreeBSD. В следующем разделе выносятся соображения по поиску и компиляции приложений с поддержкой i18n.

19.2.1. Настройка локализации для оболочки

Для настроек локализации используется пользовательский ~/.login_conf или инициализационый файл пользовательской оболочки: ~/.profile, ~/.bashrc или ~/.cshrc.

Следует задать две переменные окружения:

- `LANG` задаёт локаль *

  `MM_CHARSET` задаёт набор символов MIME для приложений

В дополнение к настройкам пользовательской оболочки эти переменные также следует задать в конфигурации конкретного приложения и в конфигурации Xorg.

Существует два способа выполнить необходимые присвоения переменным: класс логина,

который является рекомендуемым, и файл инициализации. В следующих двух разделах будет показано, как использовать оба способа.

19.2.1.1. Настройка через классы логина

Первый способ является рекомендуемым, поскольку в нём необходимые для выбора локализации и набора символов MIME значения переменных окружения присваиваются для всех оболочек. Эту процедуру может выполнить пользователь для себя, а также это может сделать в виде настройки суперпользователь для всех пользователей системы.

В этом минимальном примере обе переменные задаются для кодировки Latin-1 в .login_conf домашнего каталога отдельного пользователя:

```
me:\
    :charset=ISO-8859-1:\
    :lang=de_DE.ISO8859-1:
```

Ниже дан ~/.login_conf, в котором переменные заданы для традиционного китайского в кодировке BIG-5. Здесь нужно больше переменных, потому что некоторые программы некорректно воспринимают переменные окружения локализации для Китая, Японии и Кореи:

```
#Пользователи, которые не хотят использовать денежные единицы
#и форматы времени Тайваня, могут вручную изменить каждую переменную
me:\
    :lang=zh_TW.Big5:\

:setenv=LC_ALL=zh_TW.Big5,LC_COLLATE=zh_TW.Big5,LC_CTYPE=zh_TW.Big5,LC_MESSAGES=zh_TW.
Big5,LC_MONETARY=zh_TW.Big5,LC_NUMERIC=zh_TW.Big5,LC_TIME=zh_TW.Big5:\
    :charset=big5:\
    :xmodifiers="@im=gcin": #Set gcin as the XIM Input Server
```

Как вариант, суперпользователь может настроить локализацию для всех пользователей системы. Следующие переменные в /etc/login.conf используются для установки локализации и набора символов MIME:

```
название_языка|подробное описание:\
    :charset=кодировка_MIME:\
    :lang=название_локализации:\
    :tc=default:
```

Таким образом, на предыдущем примере с Latin-1 это бы выглядело так:

```
german|German Users Accounts:\
    :charset=ISO-8859-1:\
    :lang=de_DE.ISO8859-1:\
```

```
    :tc=default:
```

За информацией по этим переменным обращайтесь к login.conf(5). Отметим, что там уже присутствует класс *russian*.

После каждого изменения /etc/login.conf не забывайте выполнить команду для обновления базы данных:

```
# cap_mkdb /etc/login.conf
```

19.2.1.1.1. Утилиты для смены класса логина

В дополнение к ручному редактированию /etc/login.conf имеется несколько утилит, которые позволяют задать локаль при создании новых пользователей:

Если для добавлении новых пользователей используется `vipw`, то чтобы задать локаль, укажите *язык*:

```
user:password:1111:11:язык:0:0:User Name:/home/user:/bin/sh
```

Если для добавления новых пользователей используется `adduser`, то язык по умолчанию можно предварительно выбрать для всех новых пользователей или указать его для отдельного пользователя.

Если все новые пользователи используют общий язык, задайте `defaultclass=язык` в /etc/adduser.conf.

Чтобы переопределить эту настройку при создании пользователя, введите требуемую локаль в запросе командной строки:

```
Enter login class: default []:
```

или укажите её в команде adduser(8):

```
# adduser -class язык
```

Если для добавления новых пользователей используется `pw`, укажите локаль так:

```
# pw useradd имя_пользователя -L язык
```

Изменить класс логина у существующего пользователя можно с помощью `chpass`, передав имя пользователя через параметр:

```
# chpass имя_пользователя
```

19.2.1.2. Файл инициализации оболочки

Второй способ не рекомендуется, поскольку для каждой используемой оболочки требуется ручная конфигурация, при этом в каждой оболочке используется собственный файл конфигурации и разный синтаксис. Например, чтобы задать немецкий язык в оболочке `sh`, эти строчки можно было бы добавить в ~/.profile для настройки оболочки отдельного пользователя. Также их можно было бы добавить в /etc/profile или /usr/shared/skel/dot.profile, чтобы применить ко всем пользователям:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Тем не менее, путь к файлу конфигурации и используемый синтаксис отличаются в оболочке `csh`. Следующие настройки можно одинаково успешно задать в ~/.csh.login, /etc/csh.login и /usr/shared/skel/dot.login:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Используемый в ~/.xinitrc синтаксис для настройки Xorg также зависит от оболочки. Первый пример для оболочки `sh`, и второй для `csh`:

```
LANG=de_DE.ISO8859-1; export LANG
```

```
setenv LANG de_DE.ISO8859-1
```

19.2.2. Настройка консоли

Для консоли имеется несколько локализованных шрифтов. Для их просмотра наберите `ls /usr/shared/syscons/fonts`. Чтобы настроить консольный шрифт, укажите в /etc/rc.conf*имя_шрифта* без расширения .fnt:

```
font8x16=имя_шрифта
font8x14=имя_шрифта
font8x8=имя_шрифта
```

Значения keymap и screenmap можно задать в /etc/rc.conf:

```
scrnmap=название_screenmap
keymap=название_keymap
```

```
keychange="последовательность fkey_number"
```

Чтобы просмотреть доступные таблицы screenmap, наберите `ls /usr/shared/syscons/scrnmaps`. Значение *screenmap* указывается без расширения .scm. Таблица screenmap с соответствующим шрифтом обычно используется в качестве обходного пути для расширения 8 бит до 9 бит в матрице символов шрифта адаптора VGA. Это будет приводить к вынесению букв за границы псевдографической области, если используется 8-битный шрифт.

Чтобы просмотреть доступные таблицы keymap, наберите `ls /usr/shared/syscons/keymaps`. Значение *keymap* указывается без расширения .kbd. Чтобы проверить keymap без перезагрузки системы, используйте kbdmap(1).

Запись `keychange` обычно нужна для сопоставления функциональных клавиш выбранному типу терминала, поскольку последовательности функциональных клавиш не могут быть определены в keymap.

После этого задайте правильный консольный тип терминала в /etc/ttys для всех виртуальных терминалов. Типы терминалов для набора символов содержит доступные типы терминалов:

Таблица 10. Типы терминалов для набора символов

Набор символов	Тип терминала
ISO8859-1 or ISO8859-15	`cons25l1`
ISO8859-2	`cons25l2`
ISO8859-7	`cons25l7`
KOI8-R	`cons25r`
KOI8-U	`cons25u`
CP437 (VGA default)	`cons25`
US-ASCII	`cons25w`

Для языков с расширенной или многобайтовой кодировкой установите консоль для данного языка из Коллекции Портов FreeBSD. Доступные консоли из Коллекции Портов содержит доступные порты. После установки смотрите pkg-message или страницы Справочника по конфигурации и использованию данного порта.

Таблица 11. Доступные консоли из Коллекции Портов

Язык	Расположение порта
Традиционный китайский (BIG-5)	chinese/big5con
Chinese/Japanese/Korean	chinese/cce
Chinese/Japanese/Korean	chinese/zhcon
Japanese	chinese/kon2
Japanese	japanese/kon2-14dot

Язык	Расположение порта
Japanese	japanese/kon2-16dot

Если moused включен в /etc/rc.conf, может потребоваться дополнительная настройка. По умолчанию драйвер syscons(4) выделяет для курсора мыши в таблице символов диапазон `0xd0-0xd3`. Если в языке этот диапазон используется, переместите диапазон курсора посредством добавления следующей строки в /etc/rc.conf:

```
mousechar_start=3
```

19.2.3. Настройка Xorg

X Window System описывает процедуру установки и настройки Xorg. Для настройки локализации Xorg в Коллекции Портов FreeBSD имеются дополнительные шрифты и методы ввода. Настройки i18n для отдельных приложений, такие как шрифты и меню, можно внести в ~/.Xresources, чтобы меню в графических приложениях отображались на выбранном языке.

Протокол X Input Method (XIM) - это стандарт Xorg для ввода неанглийских символов. Доступные метода ввода описывает приложения для методов ввода, которые содержатся в Коллекции Портов FreeBSD. Также доступны дополнительные приложения Fcitx и Uim.

Таблица 12. Доступные метода ввода

Язык	Метод ввода
китайский	chinese/gcin
китайский	chinese/ibus-chewing
китайский	chinese/ibus-pinyin
китайский	chinese/oxim
китайский	chinese/scim-fcitx
китайский	chinese/scim-pinyin
китайский	chinese/scim-tables
японский	japanese/ibus-anthy
японский	japanese/ibus-mozc
японский	japanese/ibus-skk
японский	japanese/im-ja
японский	japanese/kinput2
японский	japanese/scim-anthy
японский	japanese/scim-canna
японский	japanese/scim-honoka
японский	japanese/scim-honoka-plugin-romkan

Язык	Метод ввода
японский	japanese/scim-honoka-plugin-wnn
японский	japanese/scim-prime
японский	japanese/scim-skk
японский	japanese/scim-tables
японский	japanese/scim-tomoe
японский	japanese/scim-uim
японский	japanese/skkinput
японский	japanese/skkinput3
японский	japanese/uim-anthy
корейский	korean/ibus-hangul
корейский	korean/imhangul
корейский	korean/nabi
корейский	korean/scim-hangul
корейский	korean/scim-tables
вьетнамский	vietnamese/xvnkb
вьетнамский	vietnamese/x-unikey

19.3. Поиск приложений i18n

i18n приложения пишутся с применением набора i18n в библиотеках. Это позволяет разработчикам писать простые файлы и переводить отображаемые меню и надписи на любые языки.

В Коллекции портов FreeBSD содержится множество приложений со встроенной поддержкой символов с расширенным и многобайтовым представлением для нескольких языков. Чтобы упростить поиск таких приложений, в их названии содержится аббревиатура `i18n`. Тем не менее, они не всегда поддерживают нужный язык.

Некоторые приложения могут быть собраны с конкретной кодировкой. Обычно это делается через Makefile порта или передачей параметра configure. Для получения этой информации смотрите документацию i18n для соответствующего порта FreeBSD.

19.4. Настройка локализации для некоторых языков

В этом разделе приведены примеры локализации системы FreeBSD на русский язык. В завершение приводится дополнительная информация для локализации на другие языки.

19.4.1. Русский язык (кодировка KOI8-R)

В этом разделе приведены настройки, специфичные для локализации системы FreeBSD на русский язык. Для более полного описания каждой из настроек обращайтесь к разделу Использование локализации.

Чтобы задать эту локаль для программной оболочки, добавьте в ~/.login_conf каждого пользователя следующие строки:

```
me:My Account:\
    :charset=KOI8-R:\
    :lang=ru_RU.KOI8-R:
```

Чтобы настроить консоль, добавьте в /etc/rc.conf такие строки:

```
keymap="ru.utf-8"
scrnmap="utf-82cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
mousechar_start=3
```

Для каждой записи `ttyv` в /etc/ttys используйте `cons25r` в качестве типа терминала.

Чтобы настроить печать, требуется специальный выходной фильтр для перекодировки из KOI8-R в CP866, поскольку большинство принтеров для России поставляются с аппаратной кодовой страницей CP866. Для этой цели в состав FreeBSD включён фильтр по умолчанию /usr/libexec/lpr/ru/koi2alt. Для его использования добавьте в /etc/printcap такую запись:

```
lp|Russian local line printer:\
    :sh:of=/usr/libexec/lpr/ru/koi2alt:\
    :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Обратитесь к printcap(5) за более подробным разъяснением.

Чтобы настроить поддержку русских названий имён файлов при монтировании файловых систем MS-DOS®, включите в добавляемую в /etc/fstab запись `-L` с названием локали:

```
/dev/ad0s2      /dos/c  msdos   rw,-Lru_RU.KOI8-R 0 0
```

За дополнительной информацией обращайтесь к странице справочника mount_msdosfs(8).

Чтобы настроить русские шрифты в Xorg, установите пакет x11-fonts/xorg-fonts-cyrillic. Затем проверьте раздел `"Files"` в /etc/X11/xorg.conf. *Перед* всеми записями `FontPath` должна быть добавлена следующая строка:

```
FontPath   "/usr/local/lib/X11/fonts/cyrillic"
```

Дополнительные кириллические шрифты доступны в Коллекции Портов.

Для настройки ввода на русском языке добавьте следующие строки в /etc/xorg.conf, раздел `"Keyboard"`:

```
Option "XkbLayout"   "us,ru"
Option "XkbOptions"  "grp:toggle"
```

Убедитесь, что в этом файле закомментирован `XkbDisable`.

Для `grp:toggle` используйте `Right Alt`, для `grp:ctrl_shift_toggle` - `Ctrl` + `Shift`. Для `grp:caps_toggle` используйте `CapsLock`. Прежняя функция `CapsLock` всё ещё доступна в режиме LAT с использованием `Shift` + `CapsLock`. `grp:caps_toggle` по неустановленной причине не работает в Xorg.

Если на клавиатуре есть клавиши "Windows®" и некоторые неалфавитные клавиши работают неправильно, добавьте в /etc/xorg.conf следующую строку:

```
Option "XkbVariant" ",winkeys"
```

Ввод с клавиатуры XKB на русском может не работать с нелокализованными приложениями. Минимально локализованные приложения должны в начале программы вызывать функцию `XtSetLanguageProc (NULL, NULL, NULL);`.

За дальнейшими инструкциями по локализации приложений Xorg обращайтесь к странице http://koi8.pp.ru/xwin.html. Для получения более общей информации по KOI8-R смотрите http://koi8.pp.ru/.

19.4.2. Информация для других языков

В этом разделе приводится дополнительная информация по настройке других локалей.

Традиционный китайский для Тайваня

У проекта FreeBSD-Taiwan есть FreeBSD Chinese HOWTO.

Локализация на греческий язык

Исчерпывающая статья по поддержке греческого во FreeBSD есть в официальной греческой документации здесь.

Локализация на японский и корейский языки

Для японского обратитесь к http://www.jp.FreeBSD.org/, а для корейского к http://www.kr.FreeBSD.org/.

Неанглоязычная документация FreeBSD

Контрибьюторы FreeBSD перевели отдельные части документации FreeBSD на другие языки. Эти переводы доступны по ссылкам на сайте FreeBSD или из каталога /usr/shared/doc.

Глава 20. Обновление системы и смена версии FreeBSD

20.1. Краткий обзор

Между релизами над FreeBSD ведется постоянная работа. Некоторые отдают предпочтение официально выпущенным версиям, в то время как остальные предпочитают использовать последние разработки. Тем не менее, даже для официальных версий часто выходят обновления, связанные с безопасностью и другими критическими исправлениями. Независимо от используемой версии FreeBSD предоставляет все необходимые инструменты для поддержания системы в актуальном состоянии, а также позволяет легко перейти на другую версию. Эта глава описывает, как отслеживать систему в процессе её разработки, а также основные инструменты для поддержания системы FreeBSD в актуальном состоянии.

После чтения этой главы вы будете знать:

- Как поддерживать систему FreeBSD в актуальном состоянии при помощи freebsd-update, Subversion или CTM.
- Как узнать состояние установленной системы по отношению к известной нетронутой копии.
- Как поддерживать установленную документацию в актуальном состоянии при помощи Subversion или портов документации.
- Разницу между двумя ветвями разработки: FreeBSD-STABLE и FreeBSD-CURRENT.
- Как перестраивать и переустанавливать всю базовую систему.

Перед чтением этой главы вы должны:

- Правильно настроить сетевое подключение (Сложные вопросы работы в сети).
- Знать, как устанавливать дополнительное стороннее программное обеспечение (Установка приложений. порты и пакеты).

В этой главе для получения и обновления исходных текстов FreeBSD используется команда `svn`. Для этого нужно сперва установить порт или пакет devel/subversion.

20.2. Обновление FreeBSD

Своевременное применение обновлений безопасности и переход на более новую версию операционной системы - важные аспекты системного администрирования. FreeBSD включает в себя программу `freebsd-update`, которую можно использовать для решения обеих задач.

Эта программа используется для установки распространяемых в двоичном виде обновлений безопасности и исправлений для FreeBSD без необходимости ручной

компиляции и установки патчей или нового ядра. Двоичные обновления доступны для всех архитектур и версий, поддерживаемых группой безопасности. Перечень поддерживаемых версий и их ожидаемые даты окончания поддержки указаны на странице http://www.FreeBSD.org/security/.

Эта программа также используется для незначительных обновлений версии операционной системы, а также для перехода на другую ветвь выпуска релизов. Перед обновлением следует ознакомиться с объявлением о выпуске новой версии, так как там может содержаться важная информация, применимая к версии, на которую намечен переход. С соответствующими объявлениями можно ознакомиться по ссылке http://www.FreeBSD.org/releases/.

Если имеется задание `crontab`, запускающее freebsd-update(8), то перед сменой версии операционной системы его обязательно нужно выключить.

В этом разделе описывается конфигурационный файл `freebsd-update`, демонстрируется применение исправлений безопасности и обновление операционной системы со сменой младшей или старшей версии, а также обсуждаются некоторые соображения касаемо смены версии операционной системы.

20.2.1. Конфигурационный файл

Конфигурационный файл `freebsd-update` самодостаточен и работает по умолчанию. Некоторые пользователи могут пожелать отредактировать конфигурационный файл /etc/freebsd-update.conf для лучшего контроля над процессом обновления. В комментариях описываются доступные в этом файле параметры, но для следующих из них может потребоваться дополнительное разъяснение:

```
# Components of the base system which should be kept updated.
Components world kernel
```

Данный параметр определяет, какие части FreeBSD будут обновлены. По умолчанию обновляется вся базовая система (world) и ядро (kernel). Вместо этого можно указать отдельные компоненты, такие как `src/base` или `src/sys`. Тем не менее, лучшим вариантом будет оставить всё как есть, поскольку изменение этого перечня с целью добавления особых пунктов потребует от пользователя указания подряд всех пунктов. Со временем это может привести к негативным последствиям из-за возможной рассинхронизации между исходными текстами и двоичными файлами.

```
# Paths which start with anything matching an entry in an IgnorePaths
# statement will be ignored.
IgnorePaths /boot/kernel/linker.hints
```

Добавьте сюда пути к каталогам (например, /bin или /sbin), которые вы бы хотели оставить нетронутыми в процессе обновления. Этот параметр можно использовать для предотвращения перезаписывания локальных изменений программой `freebsd-update`.

```
# Paths which start with anything matching an entry in an UpdateIfUnmodified
# statement will only be updated if the contents of the file have not been
# modified by the user (unless changes are merged; see below).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile
```

Этот параметр позволяет обновлять конфигурационные файлы в указанных каталогах, только если они не содержат изменений. При наличии каких-либо изменений со стороны пользователя автоматическое обновление таких файлов отменяется. Есть другой параметр `KeepModifiedMetadata`, который предписывает команде `freebsd-update` сохранять изменения в процессе слияния.

```
# When upgrading to a new FreeBSD release, files which match MergeChanges
# will have any local changes merged into the version from the new release.
MergeChanges /etc/ /var/named/etc/ /boot/device.hints
```

Список каталогов с конфигурационными файлами, для которых `freebsd-update` попытается выполнить слияние. Процесс слияния файла представляет собой последовательность изменений в формате diff(1), похожую на mergemaster(8), но с меньшим количеством параметров. Результат слияния принимается, открывается редактор или `freebsd-update` прекращает работу. В случае сомнений сделайте резервную копию /etc и просто согласитесь со всеми изменениями. Для получения подробной информации по команде `mergemaster` смотрите Объединение файлов конфигурации.

```
# Directory in which to store downloaded updates and temporary
# files used by FreeBSD Update.
# WorkDir /var/db/freebsd-update
```

Этот каталог предназначен для размещения патчей и временных файлов. В случае, когда пользователь выполняет обновление со сменой версии, в этом месте нужно иметь по крайней мере гигабайт свободного дискового пространства.

```
# When upgrading between releases, should the list of Components be
# read strictly (StrictComponents yes) or merely as a list of components
# which *might* be installed of which FreeBSD Update should figure out
# which actually are installed and upgrade those (StrictComponents no)?
# StrictComponents no
```

Если выставлено значение `yes`, то `freebsd-update` будет исходить из того, что список `Components` является полным, и не будет пытаться выполнить изменения за пределами этого списка. В действительности `freebsd-update` попытается обновить все файлы, которые принадлежат списку `Components`.

20.2.2. Обновления безопасности

Процесс применения обновлений безопасности FreeBSD был упрощён, что позволяет

поддерживать систему в актуальном состоянии, используя `freebsd-update`. Для получения дополнительной информации по бюллетеням безопасности FreeBSD смотрите Сообщения безопасности FreeBSD.

Обновления безопасности можно загрузить и установить с использованием следующих команд. Первая команда определяет наличие незагруженных обновлений и показывает файлы, которые будут изменены в процессе обновления. Вторая команда выполняет обновление.

```
# freebsd-update fetch
# freebsd-update install
```

Если были установлены обновления ядра, то после этого нужно перезагрузить систему. Если обновление установилось для какого-либо работающего в системе двоичного файла, то следует перезапустить затронутые приложения, чтобы использовалась исправленная версия двоичного файла.

Можно настроить ежедневную автоматическую проверку наличия обновлений, добавив следующую запись в /etc/crontab:

```
@daily                                  root    freebsd-update cron
```

При наличии обновлений они будут автоматически загружены. Пользователю `root` будет отправлено письмо, так что эти обновления можно будет просмотреть и установить самостоятельно командой `freebsd-update install`.

На случай, если что-то пошло не так, в `freebsd-update` предусмотрен механизм возврата последнего набора изменений с использованием следующей команды:

```
# freebsd-update rollback
Uninstalling updates... done.
```

Если после завершения всех действий было изменено ядро или какой-либо из его модулей, система должна быть перезагружена, а все затронутые исполняемые файлы нужно перезапустить.

Команда `freebsd-update` позволяет автоматически обновлять только ядро GENERIC. Если используется ядро с собственной конфигурацией, его понадобится пересобрать и переустановить после того, как `freebsd-update` завершит установку обновлений. Тем не менее, `freebsd-update` обнаружит и обновит ядро GENERIC при наличии /boot/GENERIC, даже если оно не является текущим используемым ядром в системе.

Всегда храните копию ядра GENERIC в /boot/GENERIC. Оно пригодится при решении различных проблем, а также при выполнении обновления со сменой версии. Смотрите Собственная конфигурация ядра в FreeBSD 9.X и более поздних версиях для описания получения копии ядра GENERIC.

Если конфигурация в /etc/freebsd-update.conf не изменялась, `freebsd-update` вместе с остальными обновлениями установит обновлённые исходные тексты ядра. После этого можно обычным способом выполнить перестроение и переустановку нового ядра с собственной конфигурацией.

Обновления, получаемые с помощью `freebsd-update`, не всегда затрагивают ядро. Перестроение собственного ядра не является обязательным, если исходные тексты ядра не были изменены при выполнении `freebsd-update install`. Тем не менее, `freebsd-update` всегда обновляет /usr/src/sys/conf/newvers.sh. Текущий набор изменений, как указано в номере `-p` в выводе `uname -r`, получается из этого файла. Перестроение собственного ядра, даже если ничего больше не менялось, позволяет `uname` правильно сообщать текущий набор изменений в системе. Это в частности может помочь при сопровождении множества систем, поскольку позволяет быстро оценить наличие установленных обновлений в каждой из них.

20.2.3. Обновления со сменой старшей и младшей версий

Обновление с FreeBSD 9.0 на FreeBSD 9.1, называется обновлением со сменой младшего номера версии. Смена старшего номера версии происходит, когда FreeBSD переходит с одной значительной версии на другую, как, например, при обновлении с FreeBSD 9.X на FreeBSD 10.X. Оба типа обновлений можно произвести, указав `freebsd-update` версию, на которую нужно перейти.

Если в системе используется ядро с собственной конфигурацией, убедитесь перед началом обновления в наличии копии ядра GENERIC в /boot/GENERIC. Смотрите Собственная конфигурация ядра в FreeBSD 9.X и более поздних версиях для описания получения копии ядра GENERIC.

Следующая команда, будучи запущенной на FreeBSD 9.0, выполнит обновление до версии FreeBSD 9.1:

```
# freebsd-update -r 9.1-RELEASE upgrade
```

После своего запуска `freebsd-update` анализирует содержимое конфигурационного файла и собирает необходимую для проведения обновления информацию о текущей установленной системе. На экран будет выдан перечень компонентов, которые удалось и не удалось обнаружить установленными. Например:

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 9.0-RELEASE from update1.FreeBSD.org... done.
Fetching metadata index... done.
Inspecting system... done.

The following components of FreeBSD seem to be installed:
kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
src/gnu src/include src/krb5 src/lib src/libexec src/release src/rescue
src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
```

```
world/base world/info world/lib32 world/manpages

The following components of FreeBSD do not seem to be installed:
kernel/generic world/catpages world/dict world/doc world/games
world/proflibs

Does this look reasonable (y/n)? y
```

Следующим шагом `freebsd-update` попытается загрузить по сети файлы, необходимые для выполнения обновления. В некоторых случаях может потребоваться ответить на вопросы относительно того, что и как устанавливать.

Если используется ядро с собственной конфигурацией, то в этом случае появится предупреждение следующего вида:

```
WARNING: This system is running a "MYKERNEL" kernel, which is not a
kernel configuration distributed as part of FreeBSD 9.0-RELEASE.
This kernel will not be updated: you MUST update the kernel manually
before running "/usr/sbin/freebsd-update install"
```

На этом этапе предупреждение можно проигнорировать. На промежуточном этапе процесса обновления будет использовано обновлённое ядро GENERIC.

После того, как все изменения были загружены, они будут применены. Этот процесс может занять определённое время, в зависимости от производительности и текущей загруженности компьютера. Затем будет выполнено слияние конфигурационных файлов. Процесс слияния требует от пользователя определённого вмешательства, так как для файла можно выполнить слияние автоматически, а можно открыть текстовый редактор для слияния вручную. Результат успешного слияния будет показан на экране. Неудачное или пропущенное слияние вызовет преждевременное завершение программы. Можно подготовить резервную копию каталога /etc для таких важных файлов как master.passwd и group и выполнить их слияние вручную позднее.

На данном этапе система еще не модифицирована, и все изменения и слияния происходят в отдельном каталоге. Теперь, когда все изменения успешно применены, все конфигурационные файлы объединены и кажется, что процесс должен пройти плавно, изменения могут быть установлены на диск с помощью следующей команды:

```
# freebsd-update install
```

В первую очередь изменения будут применены к ядру и его модулям. При использовании ядра с собственной конфигурацией укажите для следующей загрузки обновлённое ядро /boot/GENERIC с помощью nextboot(8):

```
# nextboot -k GENERIC
```

Перед перезагрузкой с ядром GENERIC убедитесь, что оно содержит все необходимые драйвера для системы для корректной загрузки и подключения к сети, если машина обновляется удалённо. В частности, если в ядре содержится встроенная функциональность, которая обычно обеспечивается модулями ядра, загрузите эти драйвера с ядром GENERIC, временно указав их как модули в /boot/loader.conf. Рекомендуется отключить несущественные службы, а также любые локальные и сетевые диски до завершения процесса обновления.

Теперь компьютер должен быть перезагружен с новым ядром:

```
# shutdown -r now
```

После перезагрузки нужно повторно запустить команду `freebsd-update`. Команда прочитает, на каком этапе она находится, и перейдёт к удалению старых объектных файлов и совместно используемых библиотек.

```
# freebsd-update install
```

Количество этапов установки обновлений может быть два вместо трёх и зависит от того, были ли изменены номера версий каких-либо совместно используемых библиотек.

На этом процесс завершён. Если было выполнено обновление со сменой старшего номера версии, переустановите все порты и пакеты в соответствии с описанием, которое предоставляет Обновление пакетов после смены старшей версии системы.

20.2.3.1. Собственная конфигурация ядра в FreeBSD 9.X и более поздних версиях

Перед использованием `freebsd-update` убедитесь в наличии копии ядра GENERIC в /boot/GENERIC. Если ядро с собственной конфигурацией было собрано единожды, то в /boot/kernel.old будет находиться ядро `GENERIC`. Просто переименуйте этот каталог в /boot/kernel.

Если ядро с собственной конфигурацией было собрано более одного раза, получите копию ядра `GENERIC`, соответствующую текущей версии операционной системы. При наличии физического доступа копию ядра `GENERIC` можно установить с установочного носителя:

```
# mount /cdrom
# cd /cdrom/usr/freebsd-dist
# tar -C/ -xvf kernel.txz boot/kernel/kernel
```

Иначе, ядро GENERIC можно собрать и установить из исходных текстов:

```
# cd /usr/src
# make kernel __MAKE_CONF=/dev/null SRCCONF=/dev/null
```

Чтобы такое ядро было определено как ядро GENERIC программой freebsd-update, в файле конфигурации GENERIC должны отсутствовать изменения. Также предлагается, что ядро было собрано без использования каких-либо специальных параметров.

Загрузка с GENERIC не требуется, поскольку для freebsd-update достаточно существования /boot/GENERIC.

20.2.3.2. Обновление пакетов после смены старшей версии системы

После обновления системы со сменой младшей версии установленные приложения, в целом, продолжают работать без каких-либо проблем. Различные старшие версии используют различающиеся двоичные интерфейсы приложений (Application Binary Interface, ABI), из-за чего перестаёт работать большинство сторонних приложений. После обновления системы со сменой старшей версии все установленные пакеты и порты также нуждаются в обновлении. Пакеты можно обновить с использованием pkg upgrade. Для обновления установленных портов используется ports-mgmt/portmaster.

Принудительное обновление все установленных пакетов приведёт к их замене на последние версии из репозитория, даже если номер версии при этом не увеличивался. Это требуется из-за смены версии ABI при обновлении на другую старшую версию FreeBSD. Принудительное обновление можно выполнить так:

```
# pkg-static upgrade -f
```

Перестроение всех установленных приложений можно выполнить этой командой:

```
# portmaster -af
```

Эта команда будет отображать экран выбора конфигурации для каждого приложения, в котором доступны параметры конфигурации, с ожиданием пользовательского ввода. Чтобы не использовать такое поведение и всегда выбирать параметры по умолчанию, добавьте ключ -G в вышеприведённую команду.

После завершения процесса обновления программного обеспечения закончите процесс обновления последним запуском freebsd-update, для того чтобы убедиться, что ничто не было пропущено в процессе обновления:

```
# freebsd-update install
```

Если в качестве временной меры использовалось ядро GENERIC, то это подходящее время

для построения и установки нового ядра с собственной конфигурацией в соответствии с инструкциями в Настройка ядра FreeBSD.

Перезагрузите машину с новой версией FreeBSD. На этом процесс обновления завершён.

20.2.4. Сравнение состояния системы

С помощью команды `freebsd-update IDS` можно получить состояние установленной версии FreeBSD относительно известной доверенной копии. Эта команда проверяет текущую версию системных утилит, библиотек и конфигурационных файлов, и её можно использовать в качестве встроенной системы обнаружения вторжений (Intrusion Detection System, IDS).

> Эта команда не является заменой IDS, такой как security/snort. Поскольку `freebsd-update` сохраняет свои данные на диске, возможность подмены становится очевидной. И хотя эта возможность может быть уменьшена при использовании настройки `kern.securelevel`, а также используя для записи данных `freebsd-update` файловую систему, которая в остальное время смонтирована только на чтение, лучшим решением будет сравнить систему относительно эталона на физически защищенном носителе, таком как DVD или внешний USB диск с включённой защитой от записи.

Для того, чтобы начать сравнение, укажите файл для сохранения результатов:

```
# freebsd-update IDS >> outfile.ids
```

Запустится проверка системы, результат которой будет записан в указанный файл в виде списка файлов вместе с их контрольными суммами в формате SHA256 - для известных файлов из релиза и текущих в системе.

Строки в списке чрезмерно длинные, но зато такой формат вывода удобен для разбора. Так, для получения списка всех отличающихся от релиза файлов достаточно выполнить такую команду:

```
# cat outfile.ids | awk '{ print $1 }' | more
/etc/master.passwd
/etc/motd
/etc/passwd
/etc/pf.conf
```

Вывод специально обрезан, на самом деле файлов намного больше. Некоторые из них изменены в ходе нормальной работы: так, файл /etc/passwd был изменён после заведения пользователей в системе. Модули ядра могли измениться вследствие обновления через `freebsd-update`. Для исключения из проверки конкретных файлов и каталогов укажите их в качестве значения параметра `IDSIgnorePaths` в /etc/freebsd-update.conf.

20.3. Обновление документации

Документация является неотъемлемой частью операционной системы FreeBSD. И хотя актуальная версия документации FreeBSD всегда доступна на сайте FreeBSD (http://www.freebsd.org/doc/), может быть удобно иметь под рукой актуальную локальную копию сайта FreeBSD, руководств, FAQ и статей.

В этом разделе описывается, как использовать исходный текст или Коллекцию Портов FreeBSD для организации актуальной локальной копии документации FreeBSD.

За информацией о редактировании и отправке изменений для документации обращайтесь к FreeBSD Documentation Project Primer for New Contributors (FreeBSD Documentation Project Primer).

20.3.1. Обновление документации из исходного кода

Для перестроения документации FreeBSD из исходного текста требуется набор инструментов, который не является частью основной системы FreeBSD. Требуемые инструменты, включая svn, можно установить из пакета или порта textproc/docproj, разработанного в рамках проекта документации FreeBSD.

После установки используйте svn для получения копии исходных текстов документации:

```
# svn checkout https://svn.FreeBSD.org/doc/head /usr/doc
```

Первоначальная загрузка исходных текстов документации может занять некоторое время. Дайте ей завершиться.

Последующие обновления можно получить, выполнив:

```
# svn update /usr/doc
```

После того как в /usr/doc была загружена актуальная копия исходных текстов, всё готово для обновления установленной документации.

Полное обновление всех доступных языковых версий можно выполнить, набрав команду:

```
# cd /usr/doc
# make install clean
```

Для обновления только указанной языковой версии команду `make` можно запустить в соответствующем подкаталоге /usr/doc:

```
# cd /usr/doc/en_US.ISO8859-1
# make install clean
```

Альтернативный способ обновления документации заключается в запуске следующей команды из из /usr/doc или подкаталога с желаемой языковой версией:

```
# make update
```

Используемый при установке формат можно указать через `FORMATS`:

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

Для упрощения процесса частичного обновления документации и построения только нужных переводов имеется несколько параметров. Их можно задать как на общесистемном уровне, указав в /etc/make.conf, так и непосредственно в команде `make`.

Данные параметры включают:

`DOC_LANG`

Перечень языков и кодировок для построения и установки, например, `en_US.ISO8859-1` для англоязычной документации.

`FORMATS`

Единый формат или набор форматов для построения. На данный момент поддерживаются `html`, `html-split`, `txt`, `ps` и `pdf`.

`DOCDIR`

Путь для установки документации. По умолчанию /usr/shared/doc.

Для получения других переменных `make`, также работающих во FreeBSD в качестве общесистемных, обратитесь к make.conf(5).

20.3.2. Обновление документации из портов

В предыдущем разделе был представлен метод обновления документации FreeBSD из исходных текстов. В этом разделе описывается альтернативный метод с использованием Коллекции Портов, который позволяет:

- Установить предварительно собранный пакет документации без необходимости локального построения чего-либо или установки инструментария документации.
- Выполнить построение исходных текстов документации через инфраструктуру портов, что несколько упрощает этапы загрузки и построения.

Данный метод обновления документации FreeBSD предоставляется портами и пакетами документации, которые ежемесячно обновляет Группа Менеджеров Дерева Документации <doceng@FreeBSD.org>. Они перечислены в Коллекции Портов FreeBSD в категории docs (http://www.freshports.org/docs/).

Порты документации организованы следующим образом:

- Пакет или порт misc/freebsd-doc-en устанавливает всю англоязычную документацию.
- Метапакет или порт misc/freebsd-doc-all устанавливает всю документацию на всех доступных языках.
- Имеются пакеты и порты для каждого перевода, например, misc/freebsd-doc-hu для венгерской документации.

При использовании двоичных пакетов документация FreeBSD будет установлена во всех доступных форматах для данного языка. Например, следующая команда установит последнюю версию пакета венгерской документации:

```
# pkg install hu-freebsd-doc
```

Для пакетов используется другая схема наименования, которая отличается от названия соответствующего порта: `lang-freebsd-doc`, где *lang* соответствует сокращённому языковому коду, такому как `hu` для венгерского или `zh_cn` для упрощённого китайского.

Чтобы указать используемый формат документации, для этого вместо установки готового пакета нужно собрать порт самостоятельно. Ниже приводится пример построения и установки английской документации:

```
# cd /usr/ports/misc/freebsd-doc-en
# make install clean
```

В порте имеется меню конфигурации, в котором можно указать нужный формат. По умолчанию выбирается HTML с разделителями, такой как на http://www.FreeBSD.org, а также PDF.

Иначе, при построении порта документации можно указать параметры `make`, которые включают в себя:

`WITH_HTML`

Документ в формате HTML на одной странице. Сформированная документация сохраняется в файле article.html или book.html.

`WITH_PDF`

Сформированная документация сохраняется в файле article.pdf или book.pdf.

`DOCBASE`

Указывает место размещения документации. По умолчанию /usr/local/shared/doc/freebsd.

В примере ниже демонстрируется использование переменных для установки венгерской документации в PDF в указанный каталог:

```
# cd /usr/ports/misc/freebsd-doc-hu
```

```
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

Пакеты или порты документации обновляются согласно инструкциям в Установка приложений. порты и пакеты. Например, следующая команда выполняет обновление установленной документации на венгерском языке с помощью ports-mgmt/portmaster в режиме использования только готовых пакетов:

```
# portmaster -PP hu-freebsd-doc
```

20.4. Использование ветви разработки

Во FreeBSD имеется две ветки разработки: FreeBSD-CURRENT и FreeBSD-STABLE.

В этом разделе даётся объяснение для каждой из них и их предназначение, а также рассказывается, как синхронизировать систему с любой из этих веток.

20.4.1. Использование FreeBSD-CURRENT

FreeBSD-CURRENT является "передним краем" разработки FreeBSD и предназначена для пользователей с высокой технической грамотностью. Менее продвинутым пользователям, также желающим отслеживать ветку разработки, следует использовать FreeBSD-STABLE.

FreeBSD-CURRENT обозначает последнюю версию исходных текстов FreeBSD и включает в себя незавершённые работы, экспериментальные изменения и переходные механизмы, которые могут отсутствовать в следующем официальном релизе. Хотя многие разработчики FreeBSD выполняют компиляцию исходных текстов FreeBSD-CURRENT ежедневно, бывают периоды, когда исходные тексты могут не компилироваться. Обычно такие проблемы решаются сразу по мере возможности, но всё же выбор точки синхронизации исходных текстов является определяющим фактором, содержит ли FreeBSD-CURRENT новую функциональность или же мину замедленного действия.

FreeBSD-CURRENT предназначена для трёх основных групп:

1. Члены сообщества FreeBSD, активно работающие над некоторой частью дерева исходных текстов.
2. Члены сообщества FreeBSD, которые являются активными тестерами. Они тратят свое время на исправление проблем, вносят важные предложения по изменениям и общему развитию FreeBSD, присылают патчи.
3. Пользователи, которые хотят быть в курсе изменений, используют текущие исходные тексты для ознакомительных целей либо же иногда высказывают замечания или предоставляют собственный код.

FreeBSD-CURRENT *не* должна использоваться в качестве быстрого способа получить новые возможности, не дожидаясь выпуска следующей версии, поскольку предварительная версия не является полностью проверенной и скорее всего содержит ошибки. FreeBSD-CURRENT не является быстрым способом получения исправлений, поскольку любое

изменение является в равной мере источником исправления существующих ошибок и появления новых. FreeBSD-CURRENT не является "официально поддерживаемой" каким бы то ни было способом.

Чтобы отслеживать изменения во FreeBSD-CURRENT:

1. Подпишитесь на списки рассылки Список рассылки, посвящённый обсуждению FreeBSD-CURRENT и Список рассылки сообщений об изменениях в репозитории SVN для ветки head/-current дерева исходных текстов. Это *необходимо* для того, чтобы получать сообщения и важные бюллетени относительно текущего состояния FreeBSD-CURRENT.

   Список рассылки Список рассылки сообщений об изменениях в репозитории SVN для ветки head/-current дерева исходных текстов содержит записи из журнала коммитов по каждому изменению, а также сопутствующую информацию о возможных побочных эффектах.

   Чтобы подписаться на эти списки рассылки, перейдите по ссылке https://lists.freebsd.org, щёлкните на нужном списке и следуйте дальнейшим инструкциям. Для того чтобы отслеживать изменения всего дерева исходных текстов, а не только FreeBSD-CURRENT, подпишитесь на Список рассылки сообщений об изменениях в репозитории SVN для всего дерева исходных текстов (за исключением <quote>user</quote> и <quote>projects</quote>).

2. Загрузите исходные тексты FreeBSD-CURRENT. Обычно для этого используется svn, с помощью которого можно загрузить исходные тексты -CURRENT из ветки `head` с одного из зеркал Subversion, перечисленных в Сайты зеркала Subversion.

   Пользователи с очень медленным или ограниченным подключением могут рассматривать использование CTM, который описывается в Использование CTM, однако этот способ является менее надёжным по сравнению с рекомендуемым способом синхронизации исходных текстов посредством svn.

3. Вследствие больших размеров репозитория некоторые пользователи для ознакомления или изготовления патчей выбирают частичную загрузку. Тем не менее, для компиляции операционной системы из исходных текстов требуется загрузить FreeBSD-CURRENT *полностью*, а не только лишь выбранные части.

   Перед началом компиляции FreeBSD-CURRENT внимательно прочтите файл /usr/src/Makefile и следуйте инструкциям в Пересборка мира. Список рассылки, посвящённый обсуждению FreeBSD-CURRENT и /usr/src/UPDATING позволят быть в курсе прочих процедур, которые иногда бывают необходимы в процессе перехода к следующему релизу.

4. Будьте активным участником! Пользователям FreeBSD-CURRENT предлагается высказывать свои соображения по улучшению или исправлению ошибок. Предложения, к которым прилагается код, всегда приветствуются!

20.4.2. Использование FreeBSD-STABLE

FreeBSD-STABLE является веткой разработки, из которой выпускаются основные релизы.

Изменения в этой ветке происходят с меньшей скоростью и в предположении, что они сперва были проверены во FreeBSD-CURRENT. При этом она *остаётся* веткой разработки, и в любой момент времени исходные тексты FreeBSD-STABLE могут оказаться не готовы для обычного использования. Это просто другая ветка разработки, не предназначенная для конечных пользователей. Пользователям, у которых нет возможности заниматься тестированием, следует использовать самый последний выпуск FreeBSD.

Тем, кто заинтересован процессом разработки FreeBSD или желает поучаствовать, особенно поскольку от этого зависит следующий релиз FreeBSD, стоит отслеживать FreeBSD-STABLE.

Хотя ветка FreeBSD-STABLE должна всегда компилироваться и работать, это невозможно гарантировать. Поскольку гораздо больше людей работает с FreeBSD-STABLE, неудивительно, что в FreeBSD-STABLE иногда обнаруживаются ошибки и всплывают непредвиденные ситуации, которые не проявляли себя в FreeBSD-CURRENT. По этим причинам не рекомендуется слепо использовать FreeBSD-STABLE. Особенно важно *не* обновлять какие-либо сервера, находящиеся в эксплуатации, до FreeBSD-STABLE без тщательного тестирования кода в среде разработки.

Чтобы отслеживать изменения во FreeBSD-STABLE:

1. Подпишитесь на список рассылки Список рассылки, посвящённый обсуждению FreeBSD-STABLE;, чтобы быть в курсе о зависимостях процесса компиляции, которые могут появиться во FreeBSD-STABLE или любых других проблемах, требующих особого внимания. Также в этом списке рассылки разработчики делают объявления о спорных исправлениях или добавлениях, давая пользователям возможность высказать свое мнение о возможных тонких моментах.

   Подпишитесь на список рассылки svn, соответствующий используемой ветви. Например, при использовании 9-STABLE следует подписаться на Список рассылки сообщений об изменениях в репозитории SVN для ветки 9-stable дерева исходных текстов. Этот список рассылки содержит записи из журнала коммитов по каждому изменению, а также сопутствующую информацию о возможных побочных эффектах.

   Чтобы подписаться на эти списки рассылки, перейдите по ссылке https://lists.freebsd.org, щёлкните на нужном списке, и следуйте дальнейшим инструкциям. Для того чтобы отслеживать изменения всего дерева исходных текстов, подпишитесь на Список рассылки сообщений об изменениях в репозитории SVN для всего дерева исходных текстов (за исключением <quote>user</quote> и <quote>projects</quote>).

2. Чтобы установить новую систему FreeBSD-STABLE, установите самый последний релиз FreeBSD-STABLE, загрузив его с зеркалирующих сайтов FreeBSD или используйте ежемесячную стандартную сборку FreeBSD-STABLE. Обратитесь к www.freebsd.org/snapshots для получения дополнительной информации о снэпшотах.

   Чтобы скомпилировать новую или обновить существующую систему FreeBSD до FreeBSD-STABLE, используйте svn для загрузки исходных текстов нужной ветки. Имена веток вида `stable/9` перечислены на странице www.freebsd.org/releng. При отсутствии надёжного Интернет-соединения можно воспользоваться CTM (Использование CTM).

3. Перед началом компиляции или обновления до FreeBSD-STABLE внимательно прочтите

файл /usr/src/Makefile и следуйте инструкциям в Пересборка мира. Список рассылки, посвящённый обсуждению FreeBSD-STABLE; и /usr/src/UPDATING позволят быть в курсе прочих процедур, которые иногда бывают необходимы в процессе перехода к следующему релизу.

20.5. Синхронизация исходных текстов

Имеются различные способы синхронизации с исходными текстами FreeBSD. В этом разделе сравниваются основные из них, Subversion и CTM.

Хотя возможно частичное обновление дерева исходных текстов, единственной поддерживаемой процедурой обновления является обновление всего дерева и перекомпиляция всех программ, работающих в контексте пользователя, например тех, что находятся в каталогах /bin и /sbin, а также исходных текстов ядра. Обновление только части дерева исходных текстов, только ядра или только программ часто приводит к возникновению проблем от ошибок компиляции до аварийных остановов системы или потери данных.

Subversion для обновления исходных текстов использует модель *pull*. Пользователь или сценарий `cron` запускают программу `svn`, которая обновляет локальную версию исходных текстов. Subversion является предпочтительным способом обновления локального дерева исходных текстов, поскольку обновления являются актуальными с точностью до минуты и пользователь управляет временем их загрузки. Загрузку определённых файлов и каталогов легко ограничить, а запрашиваемые обновления формируются на лету на стороне сервера. О том, как актуализировать исходные тексты с использованием Subversion, описано в svn.

CTM не выполняет интерактивное сравнение имеющихся исходных текстов с находящимися в главном архиве, и не выполняет их загрузку. Вместо этого несколько раз в день на главной машине CTM запускается скрипт, находящий изменения в файлах с момента своего предыдущего запуска. Все обнаруженные изменения сжимаются, помечаются последовательным номером и кодируются для передачи по электронной почте в печатном формате ASCII. После получения эти "дельта-файлы CTM" могут быть переданы утилите `ctm.rmail`, которая осуществляет автоматическое декодирование, проверку и применение изменений к пользовательской копии исходных текстов. Этот процесс более эффективен по сравнению с используемым в Subversion и требует меньше ресурсов сервера, так как он выполнен по модели *push*, а не *pull*. Инструкции по использованию CTM для синхронизации исходных текстов даны в Использование CTM.

Если пользователь случайно уничтожил часть своего архива, Subversion обнаружит и перестроит повреждённую часть. CTM этого не делает, поэтому если пользователь удалил часть дерева исходных текстов и не имеет архивной копии, то нужно будет начать с самого начала (с последнего "базового дельта-файла"), перестроив всё с помощью CTM.

20.6. Пересборка мира

После того, как локальное дерево исходных текстов было синхронизировано с некоторой

версией FreeBSD (FreeBSD-STABLE или FreeBSD-CURRENT), его можно использовать для перестроения системы. Этот процесс известен как перестроение мира.

Перед перестроением мира убедитесь в выполнении следующих действий:

Procedure: *Перед* тем как приступать к построению мира

1. Сохраните резервную копию всех важных данных на другую систему или съёмный носитель, проверьте её целостность и держите под рукой загрузочный носитель. Невозможно переоценить важность создания резервной копии системы *до* начала перестроения системы. Хотя перестроение системы является простой задачей, неизбежно возникают ситуации, при которых ошибки в исходных текстах приводят к тому, что система перестаёт загружаться. Возможно, вам никогда не придётся этим воспользоваться, но, постучав по дереву, всегда лучше подстраховаться.
2. Проверьте последние сообщения в списке рассылки Список рассылки, посвящённый обсуждению FreeBSD-STABLE; или Список рассылки, посвящённый обсуждению FreeBSD-CURRENT (в зависимости от отслеживаемой ветки). Будьте в курсе любых известных проблем, и тех систем, которые они затрагивают. В случае возникновения подобной проблемы, дождитесь сообщения о том, что эта проблема решена. После этого повторите синхронизацию исходных текстов для получения необходимого исправления.
3. Прочтите /usr/src/UPDATING для получения информации о дополнительных шагах, необходимых для данной версии исходных текстов. В этом файле содержится важная информация о возможных проблемах и может быть указан порядок выполнения соответствующих команд. При большинстве обновлений требуются дополнительные шаги, например, переименование или удаление определённых файлов перед установкой нового мира. Эти шаги будут перечислены в конце файла, где в явном виде описывается текущая рекомендуемая последовательность действий при обновлении. Если содержимое UPDATING противоречит каким-либо шагам в этой главе, руководствуйтесь инструкциями в файле UPDATING, которые имеют больший приоритет.

Не используйте `make world`

В некоторой устаревшей документации рекомендуется использование `make world`. Эта команда пропускает некоторые важные шаги, поэтому использовать её следует лишь в том случае, если вы точно знаете, что делаете. Почти во всех случаях `make world` - это неправильный способ, вместо этого следует использовать описанную здесь процедуру.

20.6.1. Обзор процесса

Процесс построения мира подразумевает переход с более старой версии FreeBSD с использованием исходных текстов более новой версии, которые были получены согласно инструкциям в Синхронизация исходных текстов.

Во FreeBSD термин "world" обозначает ядро, исполняемые файлы основой системы,

библиотеки, файлы для программирования и встроенный компилятор. Имеет значение порядок, при котором эти компоненты собираются и устанавливаются.

Например, из-за ошибки в старом компиляторе невозможно было бы скомпилировать новое ядре. Поскольку новое ядро должно быть собрано новым компилятором, для этого в свою очередь необходимо собрать новый компилятор, но устанавливать его перед сборкой ядра необязательно.

Новый мир может зависеть от особенностей нового ядра, поэтому новое ядро должно быть установлено до установки нового мира. Старый мир может работать неправильно на новом ядре, поэтому новый мир должен быть установлен сразу после установки нового ядра.

Перед установкой нового мира могут потребоваться изменения в конфигурации, но некоторые из изменений могут не работать со старым миром. Следовательно, используются два разных этапа обновления конфигурации. В основной части процесса обновления выполняется только замена или добавление файлов. Существующие файлы при этом не удаляются. Поскольку это может повлечь проблемы, в /usr/src/UPDATING содержится информация о том, какие из файлов и на каком шаге нужно удалить вручную.

Исходя из этих соображений в следующей процедуре описана рекомендуемая последовательность обновления.

Хорошей практикой является запись в файл вывода команды `make`. Если что-то пошло не так, копию сообщения об ошибке можно отправить в один из списков рассылки FreeBSD.

Проще всего использовать для этого `script` с параметром, задающим имя файла для сохранения всего вывода. Не сохраняйте вывод в /tmp, так как этот каталог может быть очищен при следующей перезагрузке. Более подходящим местом является /var/tmp. Запустите команду непосредственно перед перестроением мира, а после завершения процесса наберите `exit`:

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
```

Procedure: Обзор процесса построения мира

Команды для построения мира должны запускаться в указанном здесь порядке. В этом разделе даётся краткое описание назначения каждой из команд.

1. Если процесс построения мира уже запускался ранее на этой системе, то в /usr/obj могла остаться копия предыдущей сборки. Удалите этот каталог для ускорения процесса построения нового мира и возможного сокращений работы по разрешению зависимостей.

   ```
   # chflags -R noschg /usr/obj/*
   ```

```
# rm -rf /usr/obj
```

2. Скомпилируйте новый компилятор и несколько сопутствующих инструментов и используйте их для компиляции остальной части мира. Результаты сохраняются в /usr/obj.

```
# cd /usr/src
# make buildworld
```

3. Для построения нового ядра используйте компилятор, расположенный в /usr/obj, чтобы защититься от ошибок несоответствия между компилятором и ядром. Это необходимо, так как определённые структуры данных могут поменяться, и при использовании различных версий ядра и исходных текстов перестанут работать `ps` и `top`.

```
# make buildkernel
```

4. Установите новое ядро и модули, чтобы их можно было использовать для загрузки. Если используется `kern.securelevel` со значением выше `1` *и* на файле ядра установлен `noschg` или подобный флаг, то для этого сперва придётся дополнительно перейти в однопользовательский режим. В противном случае эту команду можно без проблем запустить в многопользовательском режиме. Смотрите страницу Справочника init(8) для получения информации о `kern.securelevel`, а также chflags(1) для информации об использовании различных файловых флагов.

```
# make installkernel
```

5. Переведите систему в однопользовательский режим для минимизации проблем при обновлении уже работающих исполняемых файлов. Это также уменьшит вероятность возникновения проблем при работе старого мира на новом ядре.

```
# shutdown now
```

После перехода в однопользовательский режим, запустите эти команды, если в системе используется UFS:

```
# mount -u /
# mount -a -t ufs
# swapon -a
```

Если используется ZFS, запустите другие две команды. В данном примере zpool называется `zroot`:

```
# zfs set readonly=off zroot
# zfs mount -a
```

6. Дополнительно: Если желаемая картография клавиатуры отличается от используемой по умолчанию US English, её можно изменить с помощью kbdmap(1):

```
# kbdmap
```

7. Затем, если часы CMOS установлены на местное время (это так, если вывод date(1) не содержит правильное время и часовой пояс), выполните:

```
# adjkerntz -i
```

8. Пересборка мира не включает в себя добавление или обновление конфигурационных файлов в /etc, /var, /usr и некоторых других каталогах. Следующим шагом является выполнение первоначального обновления файлов конфигурации в /etc для подготовки к новому миру. Следующая команда ограничивается сравнением файлов, необходимых для успешного выполнения цели `installworld`. В частности, на этом шаге могут быть добавлены новые пользовательские группы, служебные учётные записи и сценарии автозапуска, которые были добавлены во FreeBSD со времени последнего обновления. Это необходимо для их использования при выполнении шага `installworld`. Смотрите Объединение файлов конфигурации для получения более подробных инструкций по этой команде:

```
# mergemaster -p
```

9. Установите новый мир и служебные исполняемые файлы, находящиеся в /usr/obj.

```
# cd /usr/src
# make installworld
```

10. Обновите остальные файлы конфигурации.

```
# mergemaster -iF
```

11. Удалите устаревшие файлы. Это важно, так как в противном случае они могут вызвать проблемы.

```
# make delete-old
```

12. Теперь нужна полная перезагрузка системы для того, чтобы загрузить новое ядро и

мир с использованием новых конфигурационных файлов.

```
# reboot
```

13. Убедитесь, что перед удалением старых версий библиотек все установленные порты были пересобраны согласно инструкциям в Обновление портов. По завершению удалите все старые библиотеки во избежание конфликтов с их новыми версиями. За подробным описанием этого шага обратитесь к Удаление устаревших файлов и библиотек.

```
# make delete-old-libs
```

Если для системы доступно окно обслуживания, обдумайте возможность компиляции системы в однопользовательском режиме вместо использования для этого многопользовательского режима с переводом в однопользовательский режим для установки. Переустановка системы затрагивает множество важных системных файлов, все стандартные системные исполняемые файлы, библиотеки и заголовочные файлы. Замена этих файлов на работающей системе (в частности, используемых в данный момент пользователями) может привести к неприятностям.

20.6.2. Файлы конфигурации

В процессе построения мира используется несколько файлов конфигурации.

Makefile, расположенный в /usr/src, описывает правила и порядок построения программ, составляющих FreeBSD.

В make.conf(5) описаны параметры, доступные для `make`, а также несколько общих примеров имеется в /usr/shared/examples/etc/make.conf. Добавляемые в /etc/make.conf параметры определяют поведение `make` при построении программ. Эти параметры действуют при каждом использовании `make`, включая компиляцию приложений из Коллекции Портов, компиляцию собственных программ на Си и построение операционной системы FreeBSD. Изменение некоторых настроек может иметь далекоидущие и порой неожиданные последствия. Прочтите комментарии в обоих местах и примите к сведению, что значения по умолчанию были выбраны как компромисс между производительностью и надёжностью.

Поведение при сборке операционной системы из исходных текстов задаётся в /etc/src.conf. В отличие от /etc/make.conf, содержимое /etc/src.conf влияет только на сборку самой операционной системы FreeBSD. Описание многих параметров, доступных в этом файле, имеется в src.conf(5). Будьте осторожны при выключении на первый взгляд ненужных модулей ядра или параметров сборки. Иногда между ними имеются неожиданные или неочевидные взаимозависимости.

20.6.3. Переменные и цели выполнения

Общий формат использования `make`:

```
# make -x -DVARIABLE target
```

В этом примере параметр `-x` передаётся `make`. Обратитесь к странице Справочника make(1) для получения примеров использования имеющихся параметров.

Чтобы передать переменную, укажите её имя с использованием `-D_VARIABLE_`. Поведение Makefile зависит от переменных. Они могут быть заданы в /etc/make.conf или указаны при использовании `make`. Например, эта переменная указывает, что библиотеки для профилирования собирать не нужно:

```
# make -DNO_PROFILE target
```

Это соответствует настройке в /etc/make.conf:

```
NO_PROFILE=    true     #    Обход построения библиотек для профилирования
```

target указывает программе `make` на то, что нужно сделать, а Makefile определяет доступные цели. Некоторые цели используются в процессе построения для разбиения его на этапы.

Разделение опций удобно по двум причинам. Во-первых, это позволяет выполнять сборку, не затрагивая компоненты рабочей системы. По этой причине можно спокойно запустить `buildworld` на машине, работающей в многопользовательском режиме. Но цель `installworld` всё же рекомендуется запускать в однопользовательском режиме.

Во-вторых, это позволяет использовать монтирование по NFS для обновления многих машин по сети согласно описанию в Отслеживание исходных текстов для нескольких машин.

Параметр `-j` приводит к запуску нескольких одновременно работающих процессов `make`. Поскольку процесс компиляции больше всего требователен к подсистеме ввода/вывода, а не к производительности процессора, это можно использовать и на машинах с одним процессором.

Используйте следующую команду на машине с одним CPU, чтобы иметь до 4 одновременно работающих процессов. Опубликованные в списке рассылки практические замеры показывают, что в среднем это даёт наибольший выигрыш в производительности.

```
# make -j4 buildworld
```

На многопроцессорной машине попробуйте подобрать значение между `6` и `10`, и посмотрите, как это отразится на скорости работы.

Если при выполнении команды `make buildworld` были заданы значения каких-либо переменных, то при выполнении `make installworld` нужно задать те же самые переменные. При этом `-j` *нельзя* использовать совместно с `installworld`.

Например, если выполнялась эта команда:

```
# make -DNO_PROFILE buildworld
```

то результат её выполнения должен устанавливаться командой:

```
# make -DNO_PROFILE installworld
```

В противном случае вторая команда попытается установить библиотеки для профилирования, которые не компилировались на этапе выполнения команды `make buildworld`.

20.6.4. Объединение файлов конфигурации

FreeBSD предоставляет утилиту mergemaster(8), которая является скриптом для оболочки Боурна и предназначена для определения разницы между конфигурационными файлами в каталоге /etc и конфигурационными файлами из дерева исходных текстов /usr/src/etc. Это является рекомендуемым способом синхронизации системных конфигурационных файлов с теми, что размещены в дереве исходных текстов.

Перед использованием `mergemaster` рекомендуется скопировать имеющийся каталог /etc в какое-нибудь безопасное место. `-R` задает выполнение рекурсивного копирования, а `-p` сохраняет даты и владельца файлов:

```
# cp -Rp /etc /etc.old
```

При запуске `mergemaster` строит временное корневое окружение, начиная с /, и заполняет его различными системными конфигурационными файлами. Затем эти файлы сравниваются с текущими установленными в системе. Файлы, которые имеют отличия, будут выданы в формате diff(1), где знак `+` означает добавленные или изменённые строки, а знак `-` означает строки, которые будут либо полностью удалены, либо заменены на новый файл. Обратитесь к страницам справочной системы по команде diff(1) для получения более полной информации о формате выдачи отличий в файлах.

Затем `mergemaster` выдаст каждый файл, в котором есть изменения, с вариантами действий: удалить новый файл, упоминаемый здесь как временный, установить временный файл в его неизменённом виде, объединить временный файл с установленным на данный момент, либо просмотреть результат ещё раз.

Выбор удаления временного файла укажет `mergemaster` оставить текущий файл без изменений и удалить его новую версию. Делать это не рекомендуется. Чтобы получить

помощь в любое время, наберите `?` в приглашении `mergemaster`. Если пользователь выбирает пропуск файла, запрос появится снова, после того как будут обработаны все остальные файлы.

Выбор установки немодифицированного временного файла приведёт к замене текущего файла новым. Для большинства немодифицированных файлов это является подходящим вариантом.

Выбор варианта с объединением файла приведёт к вызову текстового редактора, содержащего текст обоих файлов. Файлы можно объединить, просматривая оба файла на экране и выбирая те части из обоих, которые подходят для окончательного варианта. При сравнении файлов нажатие `l` выбирает содержимое слева, нажатие `r` выбирает содержимое справа. В окончательном варианте будет файл, состоящий из обеих частей, который и будет установлен. Этот вариант обычно используется для файлов, настройки в которых изменялись пользователем.

Выбор повторного просмотра результатов выдаст разницу между файлами.

После того как утилита `mergemaster` закончит работу с системными файлами, она выдаст запрос относительно других параметров. Она может запросить перестроение файла паролей и завершится запросом на удаление оставшихся временных файлов.

20.6.5. Удаление устаревших файлов и библиотек

В ходе жизненного цикла разработки FreeBSD файлы с их содержимым иногда становятся устаревшими. Это может быть вызвано тем, что функциональность реализуется в другом месте, сменился номер версии библиотеки или файл был целиком удалён из системы. Такие устаревшие файлы, библиотеки и каталоги следует удалять вместе с обновлением системы. Это не даст захламить систему старыми файлами, которые занимают место на диске и на архивных носителях. Кроме того, если в старой библиотеке имеется проблема безопасности или стабильности, такую систему следует обновить до более новой библиотеки, чтобы предотвратить крахи, вызванные работой старой версии. Файлы, каталоги и библиотеки, которые признаны устаревшими, перечислены в /usr/src/ObsoleteFiles.inc. Для удаления устаревших файлов в процессе обновления системы следует пользоваться следующими инструкциями.

После выполнения `make installworld` и последующего `mergemaster` проверьте наличие устаревших файлов и библиотек:

```
# cd /usr/src
# make check-old
```

Если были найдены какие-либо устаревшие файлы, их можно удалить с помощью следующей команды:

```
# make delete-old
```

Перед удалением каждого устаревшего файла запрашивается подтверждение. Используйте `BATCH_DELETE_OLD_FILES`, чтобы сократить этот процесс и позволить системе удалить эти файлы автоматически:

```
# make -DBATCH_DELETE_OLD_FILES delete-old
```

Аналогичного эффекта можно достичь, пропустив эти команды через `yes`:

```
# yes|make delete-old
```

> *Предупреждение*
>
> Удаление устаревших файлов приведёт к нарушению работы программ, которые всё ещё зависят от этих устаревших файлов. Это особенно верно для старых библиотек. В большинстве случаев программы, порты или библиотеки, использующие такую старую библиотеку, нужно перекомпилировать перед выполнением `make delete-old-libs`.

Программы для проверки наличия зависимостей от совместно используемых библиотек включают в себя sysutils/libchk и sysutils/bsdadminscripts.

Устаревшие совместно используемые библиотеки могут конфликтовать с более новыми библиотеками, что приводит к сообщениям следующего вида:

```
/usr/bin/ld: warning: libz.so.4, needed by /usr/local/lib/libtiff.so, may conflict
with libz.so.5
/usr/bin/ld: warning: librpcsvc.so.4, needed by /usr/local/lib/libXext.so, may
conflict with librpcsvc.so.5
```

Для решения этих проблем выясните, какой именно порт установил данную библиотеку:

```
# pkg which /usr/local/lib/libtiff.so
  /usr/local/lib/libtiff.so was installed by package tiff-3.9.4
# pkg which /usr/local/lib/libXext.so
  /usr/local/lib/libXext.so was installed by package libXext-1.1.1,1
```

Затем данный порт нужно удалить, пересобрать и переустановить. Для автоматизации этого процесса можно использовать ports-mgmt/portmaster. После того как все порты пересобраны и более не используют старые библиотеки, удалите эти старые библиотеки с помощью следующей команды:

```
# make delete-old-libs
```

Если что-то работает неправильно, можно с лёгкостью перестроить конкретную часть

системы. Например, если файл /etc/magic был случайно удалён в процессе обновления или переноса /etc, то команда `file` перестанет работать. В таком случае это можно исправить вот так:

```
# cd /usr/src/usr.bin/file
# make all install
```

20.6.6. Вопросы общего характера

Нужно ли полностью перестраивать систему при каждом изменении?

Это зависит от характера изменения. Например, если svn показывает, что с момента последнего запуска были изменены только следующие файлы:

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/shared/mk/bsd.port.mk
```

то перестраивать всю систему возможно незачем. Вместо этого можно перейти в соответствующие подкаталоги и выдать команду `make all install`. Однако если меняется что-то важное, например, src/lib/libc/stdlib, то вы должны перестроить всю систему.

Некоторые пользователи перестраивают систему каждые две недели, позволяя изменениям накопиться за это время. Другие перестраивают только те вещи, которые менялись, и внимательно отслеживают все зависимости. Всё это зависит от того, как часто пользователь хочет делать обновление и отслеживает ли он FreeBSD-STABLE или FreeBSD-CURRENT.

Почему прерывается компиляция с большим количеством ошибок по сигналу 11 (или с другим номером сигнала)?

Как правило, это говорит о проблемах с оборудованием. Построение системы является эффективным стресс-тестом для оборудования, в особенности памяти. Явным указателем на это является то, что при перезапуске make процедура построения прекращается в различные моменты времени.

Для исправления этой ошибки попробуйте заменить комплектующие машины, начиная с оперативной памяти, для определения сбоящей компоненты.

Можно ли удалить /usr/obj после окончания?

В этом каталоге содержатся все объектные файлы, которые создаются во время фазы компиляции. Обычно одним из первых шагов в процессе `make buildworld` является удаление этого каталога, чтобы начать заново. Сохранение /usr/obj после окончания имеет мало смысла, а его удаление освободит приблизительно 2 ГБ дискового пространства.

Могут ли быть продолжены прерванные процессы построения?

Это зависит от того, насколько далеко зашел процесс построения перед тем, как была обнаружена проблема. В общем случае процесс `make buildworld` строит новые копии необходимых инструментальных средств и системные библиотеки. Затем эти средства и библиотеки устанавливаются. Новые инструментальные средства и библиотеки затем используются для перестроения самих себя и повторно устанавливаются. Система в целом теперь перестраивается с новыми системными файлами.

На последней стадии выполнение этих команд является достаточно безопасным, поскольку они не отменяют работу предыдущего `make buildworld`:

```
# cd /usr/src
# make -DNO_CLEAN all
```

Если в выводе `make buildworld` появляется такое сообщение:

```
--------------------------------------------------------------
Building everything..
--------------------------------------------------------------
```

то делать так вероятно достаточно безопасно.

Если такое сообщение не выводится, всегда лучше подстраховаться и запустить сборку с самого начала.

Можно ли ускорить сборку мира?

Ускорить процесс сборки мира может несколько действий. Например, весь процесс можно выполнять в однопользовательском режиме. Однако, это не позволит пользователям иметь доступ к системе, пока этот процесс не завершится.

Тщательный подход к проектированию файловой системы или использование датасетов ZFS позволит почувствовать разницу. Задумайтесь о размещении /usr/src и /usr/obj на различных файловых системах. По возможности размещайте файловые системы на различных дисках и дисковых контроллерах. При монтировании /usr/src используйте параметр `noatime`, который отключает запись информации о времени доступа к файлу. Если /usr/src не расположен на собственной файловой системе, подумайте о перемонтировании /usr с `noatime`.

Файловая система, на которой располагается /usr/obj, может быть смонтирована (или перемонтирована) с параметром `async`. Это приведёт к тому, что операции записи на диск будут выполняться асинхронно. Другими словами, запись будет завершаться немедленно, но данные записываться на диск несколькими секундами позже. Это позволит объединять операции записи и приведёт к значительному приросту производительности.

Файловую систему с /usr/obj можно смонтировать с `async` для записи на диск в асинхронном режиме. В этом случае операции записи завершаются мгновенно, а сами

данные записываются на диск через несколько секунд. Это позволяет писать кластеризованно, что может дать значительный прирост производительности.

Имейте в виду, что эта опция делает вашу файловую систему менее устойчивой. С этой опцией имеется больше шансов, что при перезагрузке машины после неожиданного сбоя при пропадании напряжения файловая система окажется в невосстановимом состоянии.

Если каталог /usr/obj - это всё, что есть на этой файловой системе, то это не проблема. Если на той же самой файловой системе имеются какие-то важные данные, то проверьте давность ваших резервных копий перед включением этой опции.

Выключите генерацию профилирующего кода, установив "NO_PROFILE=true" в файле /etc/make.conf.

Передайте утилите make(1) параметр `-jn` для запуска параллельно нескольких процессов. Обычно это помогает вне зависимости от того, сколько процессоров установлено в машине.

Что делать, если что-то пошло не так?

Скрупулезно проверьте, чтобы в вашем окружении не было мешающих остатков от предыдущих построений:

```
# chflags -R noschg /usr/obj/usr
# rm -rf /usr/obj/usr
# cd /usr/src
# make cleandir
# make cleandir
```

Да, команду `make cleandir` действительно нужно выполнять дважды.

После этого повторите весь процесс снова, начиная с `make buildworld`.

Если у вас всё ещё есть проблемы, пришлите текст ошибки и вывод команды `uname -a` в Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD. Будьте готовы ответить на другие вопросы о конфигурации вашей системы!

20.7. Отслеживание исходных текстов для нескольких машин

Если нужно отслеживать одно и то же дерево исходных текстов на множестве машин, то загрузка кода и полное перестроение системы на каждой из них выглядит как ненужная трата ресурсов: дискового пространства, пропускной способности сети и процессорного времени. Решением является выделение одной машины, которая выполняет основной объём работы, в то время как остальные используют результаты работы посредством NFS. В этом разделе описывается именно этот метод. Для получения информации об

использовании NFS обращайтесь в Network File System (NFS).

Первым делом определите набор машин, на которых будет выполняться единый набор программ, который мы будем называть *набором для построения*. Каждая машина может иметь собственное уникальное ядро, но они будут работать с одними и теми же программами пользователя. Из этого набора выберите машину, которая будет являться *машиной построения*, на которой будут строиться ядро и всё окружение. В идеальном случае это быстрая машина с достаточно незагруженным CPU для выполнения команд `make buildworld` и `make buildkernel`.

Выберите *тестовую машину*, которая будет выполнять проверку обновлений программного обеспечения, прежде чем они пойдут в работу. Это *должна* быть машина, которая может находиться в нерабочем состоянии достаточно долго. Это также может быть машина построения, но не обязательно.

Всем машинам в этом наборе для построения нужно смонтировать /usr/obj и /usr/src по NFS с машины построения. В случае нескольких наборов для построения каталог /usr/src должен находиться на одной машине построения и монтироваться на остальных по NFS.

Удостоверьтесь, что /etc/make.conf и /etc/src.conf на всех машинах в заданном наборе для построения согласуются с машиной построения. Это означает, что машина построения должна строить все те части базовой системы, которые будут устанавливаться на каждой машине из набора для построения. Кроме того, у каждой машины построения должно быть задано имя ядра в переменной `KERNCONF` в /etc/make.conf, и машина построения должна перечислить их все в переменной `KERNCONF`, причём первым должно идти имя её собственного ядра. Машина построения должна хранить конфигурационные файлы ядра каждой машины в каталоге /usr/src/sys/arch/conf.

Постройте ядро и всё окружение на машине построения так, как это описано в Переменные и цели выполнения, но ничего не устанавливайте на самой машине. Вместо этого, установите собранное ядро на тестовой машине. Для этого смонтируйте /usr/src и /usr/obj по NFS. Затем выполните команду `shutdown now` для перехода в однопользовательский режим, для того чтобы установить новое ядро и всё окружение, после чего выполните команду `mergemaster` обычным образом. После этих действий перезагрузитесь для возврата к обычному режиму работы в многопользовательском режиме.

После того, как вы убедитесь в нормальной работе всего на тестовой машине, проведите эту процедуру для установки нового программного обеспечения на каждой из оставшихся машин в наборе для построения.

Такой же подход можно использовать и для дерева портов. Сперва нужно смонтировать /usr/ports по NFS на всех машинах в наборе для построения. Чтобы настроить /etc/make.conf для использования общего каталога с дистрибутивными файлами, задайте переменную `DISTDIR` так, чтобы она указывала на общедоступный каталог, доступный для записи тому пользователю, который отображается в пользователя `root` для точек монтирования NFS. Каждая машина должна задавать `WRKDIRPREFIX` так, чтобы она указывала на локальный каталог, если порты будут собираться локально. Если же пакеты будут распространяться, задайте на машине построения переменную `PACKAGES`, чтобы она указывала на каталог, соответствующий `DISTDIR`.

Часть IV: Сетевые коммуникации

FreeBSD это одна из наиболее широко используемых в высокопроизводительных сетевых серверах операционных систем. Главы этой части книги охватывают:

- Последовательные соединения
- PPP и PPP через Ethernet
- Электронную почту
- Запуск сетевых серверов
- Брандмауэры
- Другую сетевую тематику повышенной сложности

Эти главы предназначены для получения дополнительной информации. Нет необходимости читать их в определенной последовательности, или читать их все перед тем, как начать использовать FreeBSD в сети.

Глава 21. Последовательные соединения

21.1. Краткое описание

В UNIX® всегда была поддержка последовательных соединений. Фактически, самые первые UNIX® машины использовали последовательные линии для пользовательского ввода/вывода. Многое изменилось с тех пор, когда среднестатистический "терминал" состоял из 10-символов-в-секунду последовательного принтера и клавиатуры. Эта глава рассказывает о некоторых способах, которыми FreeBSD использует последовательные соединения.

Прочитав эту главу, вы узнаете:

- Как подсоединить терминалы к системе FreeBSD.
- Как использовать модем для дозвона на удаленные хосты.
- Как разрешить удаленным пользователям входить в вашу систему с помощью модема.
- Как загрузить систему с последовательной консоли.

Перед прочтением этой главы вам потребуется:

- Узнать как настраивать и устанавливать новое ядро (Настройка ядра FreeBSD).
- Понять, что такое права доступа и процессы UNIX® (Основы UNIX).
- Кроме этого вам потребуется техническое руководство на последовательное оборудование (модем или мультипортовую карту), которую вы хотите использовать с FreeBSD.

21.2. Введение

21.2.1. Терминология

bps

Бит в секунду (Bits per Second) - скорость передачи данных

DTE

Терминальное оборудование (Data Terminal Equipment) - например, ваш компьютер

DCE

Оборудование связи (Data Communications Equipment) - ваш модем

RS-232

Стандарт EIA для аппаратных последовательных соединений

При упоминании скорости передачи данных, в этой главе не используется термин "бод" ("baud"). Бод означает количество электрических импульсов, которые могут быть переданы за период времени, а "bps" это *корректный* термин для использования (он хотя бы не

создает столько проблем как предыдущий).

21.2.2. Кабели и порты

Для подсоединения модема или терминала к системе FreeBSD потребуется последовательный порт и подходящий кабель для последовательного устройства. Если вы уже знаете о аппаратном обеспечении и требуемых кабелях, можете пропустить этот раздел.

21.2.2.1. Кабели

Есть несколько различных видов последовательных кабелей. Два наиболее часто используемых в нашей ситуации типа это нуль-модемный и стандартный ("прямой") RS-232 кабель. Документация на оборудование должна описывать тип требуемого кабеля.

21.2.2.1.1. Нуль-модемные кабели

Нуль модемный кабель пропускает некоторые сигналы, такие как "Signal Ground", напрямую, а другие "заворачивает". Например, контакт "Transmitted Data" на одном конце соединяется с контактом "Received Data" на другом.

Вы можете сделать собственный кабель для использования с терминалами. Эта таблица показывает названия сигналов RS-232C и номера контактов на разъеме DB-25. Заметим, что стандарт описывает соединение контактов номер 1 как сигнал *Protective Ground*, но его часто не делают. Некоторым терминалам достаточно сигналов на контактах 2, 3 и 7; другим требуется большее число сигналов, как показано на примерах ниже:

Таблица 13. Нуль-модемный кабель DB-25 - DB-25

Сигнал	Контакт		Контакт	Сигнал
SG	7	соединен с	7	SG
TD	2	соединен с	3	RD
RD	3	соединен с	2	TD
RTS	4	соединен с	5	CTS
CTS	5	соединен с	4	RTS
DTR	20	соединен с	6	DSR
DTR	20	соединен с	8	DCD
DSR	6	соединен с	20	DTR
DCD	8	соединен с	20	DTR

Вот еще две распространенные в настоящее время схемы.

Таблица 14. Нуль-модемный кабель DB-9 - DB-9

Сигнал	Контакт		Контакт	Сигнал
RD	2	соединен с	3	TD

Сигнал	Контакт		Контакт	Сигнал
TD	3	соединен с	2	RD
DTR	4	соединен с	6	DSR
DTR	4	соединен с	1	DCD
SG	5	соединен с	5	SG
DSR	6	соединен с	4	DTR
DCD	1	соединен с	4	DTR
RTS	7	соединен с	8	CTS
CTS	8	соединен с	7	RTS

Таблица 15. Нуль-модемный кабель DB-9 - DB-25

Сигнал	Контакт		Контакт	Сигнал
RD	2	соединен с	2	TD
TD	3	соединен с	3	RD
DTR	4	соединен с	6	DSR
DTR	4	соединен с	8	DCD
SG	5	соединен с	7	SG
DSR	6	соединен с	20	DTR
DCD	1	соединен с	20	DTR
RTS	7	соединен с	5	CTS
CTS	8	соединен с	4	RTS

Для соединения одного контакта с одной стороны с двумя контактами на другой обычно пару контактов на одной стороне соединяют коротким проводом, а затем один из них - длинным с единственным контактом на дальней стороне.

Приведенные диаграммы описывают наиболее популярные схемы распайки. В других вариантах (описанных в книге *RS-232 Made Easy*) SG соединяется с SG, TD соединяется с RD, RTS и CTS соединяются с DCD, DTR соединяется с DSR, и наоборот.

21.2.2.1.2. Стандартные кабели RS-232C

Стандартный последовательный кабель пропускает все RS-232C сигналы напрямую. Так, "send data" на одном конце кабеля соединяется с контактом "send data" на другом конце. Этот тип кабеля предназначен для подсоединения модема, а также подходит для некоторых терминалов.

21.2.2.2. Порты

Последовательные порты это устройства, через которые данные передаются между

компьютером с FreeBSD и терминалом. Этот раздел описывает типы существующих портов и их адресацию в FreeBSD.

21.2.2.2.1. Типы портов

Существует несколько типов последовательных портов. Перед изготовлением кабеля, вам потребуется убедиться, что он подходит к портам терминала и системы FreeBSD.

Большинство терминалов используют порты DB25. Персональные компьютеры, включая PC под управлением FreeBSD, используют порты DB25 или DB9. Если у вас есть мультипортовая последовательная карта для PC, там могут быть RJ-12 или RJ-45 порты.

Обратитесь к сопровождающей документации на оборудование за информацией об используемых портах. Можно также определить тип используемых портов по их внешнему виду.

21.2.2.2.2. Имена портов

В FreeBSD доступ к каждому последовательному порту может быть получен через файл в каталоге /dev. Есть два различных типа файлов:

- Порты входящих соединений (dial-in) называются /dev/ttydN, где *N* это номер порта начиная с нуля. Обычно, порты входящих соединений используются для терминалов. Для корректной работы этим портам требуется, чтобы последовательный кабель передавал сигнал data carrier detect (DCD).
- Порты исходящих соединений (call-out) называются /dev/cuadN. Они обычно используются не для терминалов, а только для модемов. Вы можете использовать эти порты если последовательный кабель или терминал не поддерживает сигнал DCD.

> Call-out порты в FreeBSD 5.X и ранее именуются /dev/cuaaN.

Если вы соединили терминал с первым последовательным портом (COM1 в MS-DOS®), используйте /dev/ttyd0 для доступа к терминалу. Если терминал соединен со вторым последовательным портом (известным также как COM2), используйте /dev/ttyd1, и так далее.

21.2.3. Настройка ядра

FreeBSD с настройками по умолчанию поддерживает последовательные порты. В мире MS-DOS® они известны как COM1, COM2, COM3, и COM4. На данный момент в FreeBSD есть поддержка как "простых" мультипортовых карт с последовательными интерфейсами, таких как BocaBoard 1008 и 2016, так и более "умных" мультипортовых карт, например карт Digiboard и Stallion Technologies. Тем не менее, ядро по умолчанию определяет только стандартные COM порты.

Чтобы увидеть, как ядро определяет последовательные порты, просмотрите сообщения, выводимые во время загрузки ядра, или используйте команду `/sbin/dmesg` для вывода сообщений ядра еще раз. В частности, обратите внимание на сообщения, начинающиеся с символов `sio`.

Для просмотра только тех сообщений, которые содержат слово `sio`, используйте команду:

```
# /sbin/dmesg | grep 'sio'
```

Например, в системе с четырьмя последовательными портами, появятся такие специфичные для последовательных портов сообщения:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

Если ядро не распознает все последовательные порты, вам возможно потребуется настроить ядро FreeBSD, изменив файл /boot/device.hints. Вы можете также закомментировать или вовсе удалить строки, относящиеся к отсутствующим у вас устройствам.

Обратитесь к странице справочника sio(4) за дополнительной информацией о настройке последовательных портов и мультипортовых карт. Будьте осторожны при использовании настроек, которые работали в предыдущих версиях FreeBSD, поскольку флаги устройств и синтаксис изменились в новых версиях.

`port IO_COM1` это синоним для `port 0x3f8`, `IO_COM2` для `0x2f8`, `IO_COM3` для `0x3e8`, и `IO_COM4` для `0x2e8`. Это наиболее часто используемые для соответствующих последовательных портов адреса. Наиболее часто используемые прерывания 4, 3, 5, и 9. Имейте ввиду, что обычные последовательные порты *не могут* совместно использовать прерывания на ISA PC (на мультипортовых картах есть электроника, позволяющая всем чипам 16550A на плате совместно использовать одно или два IRQ).

21.2.4. Специальные файлы устройств

К большинству устройств ядра можно получить доступ через "специальные файлы устройств", расположенные в каталоге /dev. К устройствам sio можно получить доступ через /dev/ttydN (устройства входящих вызовов, dial-in) и /dev/cuadN (устройства исходящих вызовов, call-out). FreeBSD предоставляет также устройства инициализации (/dev/ttydN.init и /dev/cuadN.init в случае FreeBSD 6.X, /dev/ttyidN и /dev/cuaiaN для FreeBSD 5.X), устройства блокировки (/dev/ttydN.lock и /dev/cuadN.lock в случае FreeBSD 6.X, /dev/ttyldN и /dev/cualaN для FreeBSD 5.X). Первые используются для инициализации параметров порта при каждом его открытии (таких как `crtscts` для модемов, использующих сигналы `RTS/CTS` для управления потоком). Устройства блокировки используются для установки флага

блокировки на порт и предотвращения изменения определенных параметров пользователями или программами; обратитесь к страницам справочника termios(4), sio(4) и stty(1) соответственно за информацией о параметрах терминала, блокировании и инициализации устройств и настройке терминала.

21.2.5. Настройка последовательных портов

Устройство ttydN (или cuadN) это обычное устройство, которое потребуется открыть для приложений. Когда процесс открывает устройство применяются настройки ввода/вывода терминала по умолчанию. Вы можете посмотреть эти настройки с помощью команды

```
# stty -a -f /dev/ttyd1
```

Если вы измените настройки устройства, они будут действовать до его закрытия. После повторного открытия, оно вернется к настройкам по умолчанию. Для изменения настроек по умолчанию, вы можете открыть и изменить установки "начального состояния" устройства. Например, для включения по умолчанию режима `CLOCAL`, 8-битного соединения и контроля передачи `XON/XOFF` для ttyd5, выполните:

```
# stty -f /dev/ttyd5.init clocal cs8 ixon ixoff
```

Инициализация последовательных устройств контролируется файлом /etc/rc.d/serial. Этот файл определяет настройки последовательных устройств по умолчанию.

Для предотвращения изменения программами отдельных установок, настройте "состояние блокировки" устройства. Например, для установки значения скорости ttyd5 в 57600 bps, выполните:

```
# stty -f /dev/ttyd5.lock 57600
```

Теперь приложение, открывающее ttyd5 и пытающееся изменить скорость порта, получит скорость 57600 bps.

И конечно, сделайте запись начальных значений и состояния блокировки устройств доступной только учетной записи `root`.

21.3. Терминалы

Терминалы предоставляют удобный и дешевый способ доступа к системе FreeBSD, когда вы не сидите за консолью компьютера и не подключены к сети. Этот раздел описывает использование терминалов в FreeBSD.

21.3.1. Пользователи и типы терминалов

В первых системах UNIX® не было консолей. Вместо этого, пользователи входили и

запускали программы через терминалы, которые были подключены к последовательным портам компьютеров. Это очень похоже на использование модема и программного обеспечения терминала для дозвона до удаленной системы и выполнения только-текстовой работы.

Консоли современных PC поддерживают высококачественную графику, но возможность входа по последовательному порту на сегодняшний день все еще доступна почти в каждой UNIX® подобной операционной системе; FreeBSD не исключение. Используя терминал, подключенный к неиспользуемому последовательному порту, вы можете войти и запустить текстовую программу, которую обычно запускаете в текстовой консоли или в окне `xterm` системы X Window.

Для корпоративных пользователей, вы можете подсоединить множество терминалов к системе FreeBSD и поставить их на столы пользователей. Для домашнего пользователя, устаревший IBM PC или Macintosh® может быть подключен в качестве терминала к более мощному компьютеру под управлением FreeBSD. Вы можете превратить однопользовательский компьютер в мощную многопользовательскую систему.

В FreeBSD три вида терминалов:

- Простые (dumb) терминалы
- "PC, работающие в качестве терминалов"
- X терминалы

В оставшейся части раздела описывается каждый вид.

21.3.1.1. Простые терминалы

Простые терминалы это специализированное оборудование, позволяющее соединять компьютеры через последовательные линии. Они называются "простыми", поскольку их вычислительных возможностей хватает только для отображения, отправки и получения текста. Вы не сможете запустить на них никаких программ. Компьютер, к которому подсоединяется терминал, предоставляет все возможности для запуска текстовых редакторов, компиляторов, почтовых программ, игр и так далее.

Есть сотни видов простых терминалов, изготовленных различными производителями, включая DEC VT-100 и Wyse WY-75. Почти любой терминал может работать с FreeBSD. Некоторые high-end терминалы даже могут отображать графику, но только отдельные программные пакеты могут получить преимущество от этих расширенных возможностей.

Простые терминалы популярны в рабочей среде, где не требуется доступ к графическим приложениям, например тем, которые предоставляет система X Window.

21.3.1.2. PC, работающие в качестве терминалов

Если простые терминалы могут только отображать, отправлять и получать текст, возможностей абсолютно любого персонального компьютера хватит для работы в роли простого терминала. Все, что вам потребуется, это подходящий кабель и какая-нибудь программа *эмулятора терминала*.

Это популярная домашняя конфигурация. Например, когда ваша вторая половина занята работой на системной консоли FreeBSD, вы можете одновременно выполнять только-текстовую работу с менее мощного персонального компьютера, подключенного к системе FreeBSD.

21.3.1.3. X терминалы

X терминалы это наиболее сложный тип существующих терминалов. Вместо подключения к последовательному порту, они обычно подключаются к сети, например Ethernet. Вместо работы только с текстовыми приложениями, они могут отображать любое X приложение.

Мы представляем X терминалы только ради полноты описания. Тем не менее, эта глава *не* охватывает установку, настройку или использование X терминалов.

21.3.2. Настройка

Этот раздел описывает, что нужно сделать для настройки системы FreeBSD и включения входа в систему через терминал. Предполагается, что вы уже подключили терминал и настроили ядро для включения поддержки последовательного порта, к которому он подключен.

Обратитесь к главе Процесс загрузки FreeBSD за информацией о процессе `init`, отвечающем за контроль над всеми процессами и за инициализацию системы во время загрузки. Одна из задач, выполняемых `init` - чтение файла /etc/ttys и запуск процесса `getty` на доступных терминалах. Процесс `getty` отвечает за чтение имени пользователя и запуск программы `login`.

Таким образом, для настройки терминалов в системе FreeBSD необходимо выполнить следующие действия под `root`:

1. Добавить строку к /etc/ttys для файла из каталога /dev, представляющего последовательный порт, если этой строки еще нет.
2. Настроить запуск команды `/usr/libexec/getty` на этом порту и указать соответствующий тип *getty* в файле /etc/gettytab.
3. Указать тип терминала по умолчанию.
4. Переключить порт в состояние "on" ("включен")
5. Указать, должен ли порт быть "secure" ("безопасным")
6. Заставить `init` перечитать файл /etc/ttys.

Опционально, вы можете настроить свой тип *getty* для использования на шаге 2, добавив описание в файл /etc/gettytab. За описанием обратитесь к страницам справочника gettytab(5) и getty(8).

21.3.2.1. Добавление строки в /etc/ttys

В файле /etc/ttys находится список всех портов системы FreeBSD, на которые возможен вход.

Например, там находится первая виртуальная консоль ttyv0. Вы можете войти на консоль с помощью этой записи. Файл содержит записи и для других виртуальных консолей, последовательных портов, и псевдо-терминалов. Название файла последовательного порта из каталога /dev приводится без префикса /dev (например, устройство /dev/ttyv0 будет записано как ttyv0).

Установка FreeBSD по умолчанию включает файл /etc/ttys с поддержкой первых четырех последовательных портов: от ttyd0 до ttyd3. Если вы подключаете терминал к одному из этих портов, добавлять записи терминалов не потребуется.

Пример 34. Добавление записей терминалов в /etc/ttys

Предположим, вы хотите подключить два терминала к системе: Wyse-50 и старый 286 IBM PC с эмулятором терминала VT-100. Мы подключаем Wyse к второму последовательному порту и 286 к шестому последовательному порту (порт на мультипортовой карте). Соответствующие строки в /etc/ttys будут выглядеть так:

```
ttyd1  "/usr/libexec/getty std.38400"  wy50  on  insecure
ttyd5   "/usr/libexec/getty std.19200"  vt100  on  insecure
```

- Первое поле, как правило, указывает имя специального файла терминала, в соответствии с его именем в /dev.
- Второе поле - это команда, исполняемая для этого терминала, обычно getty(8). `getty` инициализирует и открывает линию, устанавливает ее скорость, приглашает пользователя к вводу имени пользователя, а затем выполняет программу login(1).Программа `getty` принимает один (опциональный) параметр в командной строке, тип *getty*. Тип *getty* определяет характеристики терминальной линии, такие как значение bps и четность. Программа `getty` считывает эти характеристики из файла /etc/gettytab.Файл /etc/gettytab содержит множество записей для терминалов, как для старых так и для новых. Почти во всех случаях запись, начинающаяся с текста `std`, предназначена для работы с аппаратными терминалами. Эти записи игнорируют четность. Запись `std` есть для каждого значения bps от 110 до 115200. Конечно, вы можете добавить собственные записи в этот файл. Страница справочника gettytab(5) содержит дополнительную информацию.При установке типа *getty* в файле /etc/ttys убедитесь в наличии соответствующей записи терминала.Например, Wyse-50 не использует четность и соединяется на 38400 bps. 286 PC не использует четность и соединяется на 19200 bps.
- Третье поле определяет тип терминала, обычно подключаемого к этой линии tty. Для портов входящих соединений обычно используется значение `unknown` или `dialup`, поскольку пользователь может подключить практически любой тип терминала или программу. Для аппаратных терминалов тип не меняется, поэтому вы можете поместить в это поле определенный тип терминала из базы данных termcap(5).Например, Wyse-50 использует реальный тип терминала, а 286 PC, работающий с Procomm, настроен на эмуляцию VT-100.
- Четвертое поле определяет должен ли порт быть включен. Размещение здесь `on` укажет процессу `init` запустить программу, указанную во втором поле, `getty`. Если

вы поместите `off` в это поле, команда `getty` не будет запущена и вход на этот порт станет невозможен.

- Последнее поле используется, чтобы указать, является ли порт безопасным. Пометка порта безопасным означает, что вы доверяете ему достаточно для того, чтобы разрешить учетной записи `root` (или любой учетной записи с UID 0) входить с этого порта. Небезопасные порты не разрешат вход `root`. На небезопасном порту пользователи должны войти с через непривилегированную учетную запись, а затем использовать su(1) или подобный механизм для получения привилегий суперпользователя.Настоятельно рекомендуется использовать "insecure" даже для терминалов, находящихся за закрытыми дверями. Довольно легко использовать `su` после входа, если вам потребуются привилегии суперпользователя.

21.3.2.2. Заставьте `init` перечитать /etc/ttys

После выполнения необходимых изменений в файле /etc/ttys, вам потребуется отправить сигнал SIGHUP (hangup) процессу `init`, чтобы заставить его перечитать его файл настройки. Например:

```
# kill -HUP 1
```

`init` это всегда первый из запущенных в в системе процессов, поэтому его PID всегда 1.

Если все установлено правильно, все кабели на месте и терминалы включены, процесс `getty` должен быть запущен на каждом терминале и вы увидите приглашение ко входу на каждом терминале.

21.3.3. Решение проблем с соединением

Даже при самом внимательном отношении к деталям, при настройке терминала все же могут возникнуть проблемы. В этом разделе приведен список симптомов и предлагается несколько решений.

21.3.3.1. Не появляется приглашение ко входу

Убедитесь, что терминал подключен и его питание включено. Убедитесь, что эмулятор терминала запущен на соответствующем порту.

Убедитесь, что кабель хорошо подключен и к терминалу и к компьютеру с FreeBSD. Убедитесь, что правильно выбран тип кабеля.

Убедитесь, что терминал и FreeBSD имеют одинаковые установки значения bps и четности. Если у вас видео терминал, убедитесь, что контраст и яркость включены. Если это принт-терминал, убедитесь, что бумага и чернила в порядке.

Убедитесь, что процесс `getty` запущен и обслуживает терминал. Например, для получения списка запущенных процессов `getty` с помощью `ps`, выполните:

```
# ps -axww|grep getty
```

Вы должны увидеть строку для соответствующего терминала. Например, если `getty` запущена на втором последовательном порту `ttyd1` и использует запись `std.38400` из файла /etc/gettytab, отобразится следующее:

```
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyd1
```

Если процесс `getty` не запущен, убедитесь, что вы включили порт в /etc/ttys. Не забудьте также запустить `kill -HUP 1` после изменения файла ttys.

Если процесс `getty` запущен, но на терминале по-прежнему не отображается приглашение ко входу, или если приглашение отображается, но войти невозможно, терминал или кабель, возможно, не поддерживают квитирование связи. Попробуйте изменить поле в /etc/ttys с `std.38400` на `3wire.38400`. Запись `3wire` похожа на `std`, но игнорирует квитирование связи. Вам может потребоваться уменьшить скорость соединения или включить программный контроль передачи при использовании `3wire` для предотвращения переполнений буфера.

21.3.3.2. Вместо приглашения ко входу на экране появляется "мусор"

Убедитесь, что терминал и FreeBSD имеют одинаковые установки значения bps и четности. Проверьте процесс `getty`, чтобы убедиться, что используется подходящий тип *getty*. Если это не так, отредактируйте /etc/ttys и запустите `kill -HUP 1`.

21.3.3.3. Символы появляются дважды, пароль отображается при вводе

Переключите терминал (или программу эмулятора терминала) с "half duplex" или "local echo" на "full duplex".

21.4. Входящие соединения по модему

Настройка системы FreeBSD для поддержки входящих соединений очень похожа на подсоединение терминалов за исключением того, что вы работаете с модемами вместо терминалов.

21.4.1. Внешние и внутренние модемы

Внешние модемы более удобны для дозвона, поскольку легко могут быть настроены с помощью параметров, сохраняемых в энергонезависимой памяти. На них обычно есть индикаторы, отображающие состояние основных RS-232 сигналов. Мигающие индикаторы впечатляют, но кроме того они также очень полезны для индикации правильной работы модема.

Внутренние модемы обычно не снабжаются энергонезависимой памятью, поэтому их настройка может ограничиваться установкой DIP переключателей. Если на внутреннем модеме есть индикаторы, их обычно сложно увидеть при закрытой крышке корпуса.

21.4.1.1. Модемы и кабели

Если вы используете внешний модем, несомненно потребуется подходящий кабель. Стандартный RS-232C кабель должен подойти, если подключены все обычные сигналы:

Таблица 16. Наименования сигналов

Сокращение	Наименование	Назначение
RD	Received Data	Принимаемые данные
TD	Transmitted Data	Передаваемые данные
DTR	Data Terminal Ready	Готовность терминала
DSR	Data Set Ready	Готовность данных
DCD	Data Carrier Detect	Наличие несущей
SG	Signal Ground	Сигнальная земля
RTS	Request to Send	Запрос на посылку
CTS	Clear to Send	Готовность к приему

FreeBSD требуются сигналы RTS и CTS для контроля передачи на скоростях выше 2400 bps, сигнал CD для определения, был ли ответ на сигнал или произошло отключение линии, и сигнал DTR для сброса модема после завершения сессии. Некоторые кабели не поддерживают все необходимые сигналы, поэтому, если вы столкнулись с проблемами, например, если сессия не завершается после отсоединения линии, причиной возможно являются проблемы с кабелем.

Как и другие UNIX® подобные операционные системы, FreeBSD использует аппаратные сигналы для определения того, был ли ответ на звонок или линия была отключена и требуется завершить работу модема и сбросить его в начальное состояние. FreeBSD избегает отправлять команды модему или просматривать отчеты о статусе от модема. Если вы знакомы с настройкой BBS, это может показаться неудобным.

21.4.2. Рекомендации по последовательным интерфейсам

FreeBSD поддерживает интерфейсы, основанные на NS8250, NS16450, NS16550, и NS16550A EIA RS-232C (CCITT V.24). Устройства 8250 и 16450 снабжены односимвольным буфером. Устройство 16550 снабжено 16-ти символьным буфером, который повышает производительность системы. (Ошибки в 16550 делают невозможным использование 16-символьного буфера, поэтому используйте 16550A если возможно). Поскольку устройства с односимвольным буфером предъявляют большие требования к операционной системе, чем с 16-ти символьным буфером, предпочтительны устройства на 16550A. Если в системе много активных последовательных портов или нагрузка велика, устройства на 16550A лучше подходят для поддержки соединений с малым количеством ошибок.

21.4.3. Краткий обзор

Как и с терминалами, `init` запускает процесс `getty` на каждом настроенном для входящих звонков последовательном порту. Например, если модем подключен к /dev/ttyd0, команда `ps`

`ax` может вывести следующее:

```
4850 ??  I      0:00.09 /usr/libexec/getty V19200 ttyd0
```

Когда пользователь дозванивается на подключенный модем, модем выдает сигнал CD (Carrier Detect). Ядро определяет, что несущая обнаружена и завершает открытие порта командой `getty`. `getty` отправляет приглашение `login:` на указанной скорости. `getty` ожидает в ответ набор символов, и, как правило, получает неправильный набор (обычно это происходит из-за того, что скорость соединения модема отличается от скорости `getty`). `getty` пробует подобрать скорость линии до тех пор, пока не получит правильный набор символов.

После того, как будет введено имя пользователя, `getty` выполняет /usr/bin/login, которая завершает вход, запрашивая пароль пользователя и запуская оболочку.

21.4.4. Файлы настройки

Есть три файла настройки системы в каталоге /etc, которые возможно потребуется отредактировать для включения удаленного доступа по модему в FreeBSD. Первый, /etc/gettytab, содержит информацию по настройке даемона /usr/libexec/getty. Второй, /etc/ttys, содержит информацию, указывающую /sbin/init на каких устройствах tty должны быть запущены процессы `getty`. Наконец, вы можете поместить команды инициализации портов в скрипт /etc/rc.d/serial.

В UNIX® есть две школы настройки модемов для входящих соединений. Одна предпочитает настраивать модемы и системы так, что не важно на какой скорости подсоединяется удаленный пользователь. Локальный интерфейс RS-232 компьютер-модем работает на жестко заданной скорости. Преимущество этой настройки в том, что удаленный пользователь всегда сразу видит приглашение ко входу. Обратная сторона в том, что система не знает, какова на самом деле скорость передачи данных, поэтому полноэкранные программы, такие как Emacs, не настраивают свои методы отображения на экране для работы с медленными соединениями.

Другая школа настраивает интерфейс RS-232 для работы с различной скоростью в зависимости от скорости подсоединения удаленного пользователя. Например, соединение модемов по протоколу V.32bis (14.4 Кбит/с) установит скорость порта RS-232 равной 19.2 Кбит/с, а соединение на скорости 2400 бит/с установит скорость RS-232 равной 2400 бит/с. Поскольку `getty` не понимает сообщений модема о скорости соединения, `getty` выдает приглашение `login:` на установленной по умолчанию скорости и считывает символы, полученные в ответе. Если пользователь видит "мусор" вместо приглашения ко входу, это означает, что нужно нажимать Enter до тех пор, пока не появится приглашение ко входу. Если скорости не совпадают, `getty` получает все, что вводит пользователь, в виде "мусора", пробует переключиться на другую скорость и выдает приглашение `login:` опять. Эта процедура может продолжаться до отвращения, но обычно требуется одно или два нажатия клавиши перед появлением нормально выглядящего приглашения. Очевидно, эта последовательность входа не так хороша, как метод с фиксированной скоростью, но при низкой скорости соединения работать с полноэкранными программами станет проще.

В этом разделе делается попытка дать сбалансированную информацию для настройки, но предпочтение будет отдано установке скорости соединения с модемом в соответствие скорости подключения.

21.4.4.1. /etc/gettytab

/etc/gettytab это файл в стиле termcap(5), содержащей информацию по настройке getty(8). Пожалуйста, обратитесь к странице справочника gettytab(5) за полной информацией о формате файла и за списком возможностей `getty`.

21.4.4.1.1. Настройка фиксированной скорости

Если вы зафиксировали скорость соединения модема на определенной скорости, редактировать файл /etc/gettytab скорее всего не потребуется.

21.4.4.1.2. Настройка изменяемой скорости

Вам потребуется сделать запись в /etc/gettytab для предоставления `getty` информации о скоростях, которые предполагается использовать для модема. Если у вас 2400 бит/с модем, возможно, подойдет существующая запись `D2400`.

```
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either way)
#
D2400|d2400|Fast-Dial-2400:\
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
        :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
        :nx=D2400:tc=300-baud:
```

Если у вас более скоростной модем, вам возможно потребуется добавить запись в /etc/gettytab; вот запись, которую вы можете использовать для 14.4 Кбит/с модема с максимальной скоростью интерфейса 19.2 Кбит/с:

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
        :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
        :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
        :nx=V9600:tc=std.19200:
```

Эта настройка включает 8-битные соединения без программного контроля четности.

В примере выше скорость порта будет переключаться в цикле начиная с 19.2 Кбит/с (для соединения по V.32bis), затем 9600 бит/с (для V.32), 2400 бит/с, 1200 бит/с, 300 бит/с, и обратно на 19.2 Кбит/с. Переключение скоростей в цикле реализовано с помощью `nx=` ("next table"). Каждая из линий использует `tc=` ("table continuation") для указания "стандартных" (std) настроек на каждой скорости.

Если у вас 28.8 Кбит/с модем и/или вы хотите получить преимущество от сжатия на скорости 14.4 Кбит/с, потребуются скорости выше, чем 19.2 Кбит/с. Вот пример записи из gettytab для начала соединения на скорости 57.6 Кбит/с:

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57.6 Kbps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
        :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
        :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
        :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
        :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
        :nx=VH9600:tc=std.57600:
```

Если у вас медленный CPU или сильно загруженная система без последовательных портов на базе 16550A, на скорости 57.6 Кбит/с могут возникнуть ошибки `sio` "silo".

21.4.4.2. /etc/ttys

Настройка файла /etc/ttys была описана в Добавление записей терминалов в /etc/ttys. Настройка модемов похожа, но потребуется передавать `getty` различные аргументы и указывать различные типы терминалов. Общий формат для фиксированной и переменной скорости такой:

```
ttyd0   "/usr/libexec/getty xxx"   dialup on
```

Первый пункт в строке выше это специальный файл устройства для этой записи - `ttyd0` означает, что `getty` будет запущена на /dev/ttyd0. Второй пункт, `"/usr/libexec/getty xxx"` (*xxx* будет замещено на запись из gettytab для начальной скорости), это процесс, который будет запущен на данном устройстве. Третий пункт, `dialup`, это тип терминала по умолчанию. Четвертый параметр, `on`, указывает `init`, что линия включена. Может быть пятый параметр, `secure`, но он должен использоваться только для терминалов, которые физически безопасны (таких как системная консоль).

Тип терминала по умолчанию (`dialup` в примере выше) может зависеть от личных

предпочтений. dialup это традиционный тип терминала по умолчанию на линиях для дозвона, который позволяет пользователям, зная что тип терминала dialup, автоматически настраивать свой тип терминала. Однако, автор находит более легким указание vt102 в качестве типа терминала по умолчанию, поскольку пользователи работают на своих удаленных системах с эмулятором терминала VT102.

После внесения изменений в /etc/ttys, вы можете отправить процессу init сигнал HUP перечитать файл. Используйте команду

```
# kill -HUP 1
```

для отправки сигнала. Если вы настраиваете систему в первый раз, то возможно захотите подождать, пока модем(ы) правильно настроятся и соединятся перед отправкой сигнала init.

21.4.4.2.1. Настройка фиксированной скорости

Для настройки соединения с фиксированной скоростью, в файле ttys должна быть запись с фиксированной скоростью для getty. Для модема, скорость порта которого фиксирована на значении 19.2 Кбит/с, строка в ttys может выглядеть так:

```
ttyd0   "/usr/libexec/getty std.19200"   dialup on
```

Если скорость модема фиксирована на другом значении, подставьте соответствующее значение в std.speed вместо std.19200. Убедитесь, что вы используете тип, описанный в /etc/gettytab.

21.4.4.2.2. Настройка переменной скорости

В настройке с переменной скоростью, запись в ttys должна обращаться к соответствующей "auto-baud" (sic) записи в /etc/gettytab. Например, если вы добавите предложенную выше запись для подключения модема с переменной скоростью, которая начинается с 19.2 Кбит/с (запись в gettytab начинается с V19200), запись в ttys может выглядеть так:

```
ttyd0   "/usr/libexec/getty V19200"   dialup on
```

21.4.4.3. /etc/rc.d/serial

Для высокоскоростных модемов, таких как V.32, V.32bis и V.34, требуется использование аппаратного контроля передачи (RTS/CTS). Вы можете добавить команды stty к файлу /etc/rc.d/serial для установки флага аппаратного контроля передачи в ядре FreeBSD для модемных портов.

Например, для установки флага termios crtscts на последовательном порту номер 1 (COM2) при инициализации устройств для входящей и исходящей связи, в /etc/rc.d/serialдолжны быть добавлены следующие строки:

```
# Serial port initial configuration
stty -f /dev/ttyd1.init crtscts
stty -f /dev/cuad1.init crtscts
```

21.4.5. Настройка модема

Если параметры вашего модема могут быть сохранены в энергонезависимой памяти, потребуется использовать терминальную программу (например, Telix под MS-DOS® или `tip` под FreeBSD) для установки параметров. Подсоединитесь к модему, используя ту же скорость соединения, которую использует `getty` в качестве начальной скорости, и настройте модем для соответствия следующим требованиям:

- CD включен после соединения
- DTR включен во время работы; сброс DTR отключает линию и переводит модем в начальное состояние
- CTS контроль переданных данных
- Контроль потока XON/XOFF отключен
- RTS контроль принятых данных
- "Тихий" режим (без кодов возврата)
- Эхо команд отключено

Прочтите документацию на модем для определения какие команды и/или DIP переключатели требуются чтобы установить эти настройки.

Например, для установки вышеуказанных параметров на внешнем 14,400 модеме U.S. Robotics® Sportster®, требуется отправить модему следующие команды:

```
ATZ
AT&C1&D2&H1&I0&R2&W
```

Вы, возможно, захотите настроить и другие параметры модема, такие как использование сжатия V.42bis и/или MNP5.

Внешний U.S. Robotics® Sportster® 14,400 модем также снабжен некоторыми DIP переключателями, которые требуется установить; для других модемов эти настройки могут быть использованы в качестве примера:

- Переключатель 1: вверх - нормальный DTR
- Переключатель 2: N/A (визуальные коды возврата/числовые коды возврата)
- Переключатель 3: вверх - подавление кодов возврата
- Переключатель 4: вниз - без эхо, offline команды
- Переключатель 5: вверх - авто ответ
- Переключатель 6: вверх - нормальный контроль несущей

- Переключатель 7: вверх - загрузить установки по умолчанию из NVRAM
- Переключатель 8: N/A (Smart/Dumb режимы)

Коды возврата должны быть отключены/подавлены для устранения проблем, которые могут возникнуть, если `getty` ошибочно выдаст приглашение `login:` модему в командном режиме и модем вернет (echo) эту команду или код возврата. Эта последовательность может привести к дополнительному и бессмысленному обмену командами между `getty` и модемом.

21.4.5.1. Настройка фиксированной скорости

Для настройки фиксированной скорости вам потребуется настроить модем с поддержкой постоянной скорости обмена данными модем-компьютер независимо от скорости соединения. На внешнем модеме U.S. Robotics® Sportster® 14,400 эти команды зафиксируют скорость передачи модем-компьютер на скорости, которая установлена при выполнении команды:

```
ATZ
AT&B1&W
```

21.4.5.2. Настройка переменной скорости

Для настройки переменной скорости вам потребуется настроить модем с поддержкой изменения скорости передачи данных через последовательный порт в соответствии через скоростью соединения. Следующие команды зафиксируют скорость передачи данных с коррекцией ошибок внешнего модема U.S. Robotics® Sportster® 14,400 на значении, которое установлено при выполнении команды, но сделают возможным изменение скорости последовательного порта для соединений без коррекции ошибок:

```
ATZ
AT&B2&W
```

21.4.5.3. Проверка настроек модема

Большинство высокоскоростных модемов предоставляют команды для просмотра текущих параметров модема в виде, отчасти приспособленном для чтения. Для внешних модемов U.S. Robotics® Sportster® 14,400 команда `ATI5` отображает установки, сохраненные в энергонезависимой памяти. Для просмотра действующих параметров модема (с учетом положения DIP переключателей), используйте команду `ATZ`, а затем `ATI4`.

Если ваш модем другого производителя, проверьте руководство к модему для аккуратной проверки параметров настройки модема.

21.4.6. Решение проблем

Вот несколько шагов, которые нужно выполнить для проверки настроек.

21.4.6.1. Проверьте систему FreeBSD

Подсоедините модем к системе FreeBSD, загрузите систему, и, если на модеме есть индикаторы, посмотрите, загорелся ли индикатор DTR при появлении приглашения `login:` на системной консоли - если он загорелся, это означает, что FreeBSD запустила процесс `getty` на соответствующем коммуникационном порту и модем ожидает входящего звонка.

Если индикатор DTR не загорелся, войдите на консоль системы FreeBSD и выполните команду `ps ax`, чтобы увидеть, пытается ли FreeBSD запустить процесс `getty` на соответствующем порту. Вы должны увидеть строки вроде этих среди показанных процессов:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyd0
115 ??  I      0:00.10 /usr/libexec/getty V19200 ttyd1
```

Если вы видите что-то другое, вроде этого:

```
114 d0  I      0:00.10 /usr/libexec/getty V19200 ttyd0
```

и модем все еще не принимает звонок, это означает, что `getty` завершила открытие коммуникационного порта. Это может означать проблему с кабелем или неправильную настройку модема, поскольку `getty` не должна открывать коммуникационный порт, пока модем не установит CD (обнаружение несущей).

Если вы не видите процессов `getty`, ожидающих открытия соответствующего порта ttydN, внимательно проверьте записи в /etc/ttys и попробуйте найти ошибки, если они есть. Проверьте также лог файл /var/log/messages, нет ли там сообщений от `init` или `getty`, имеющих отношение к проблеме. Если сообщения есть, проверьте еще раз файлы настройки /etc/ttys и /etc/gettytab, как и соответствующие специальные файлы устройств /dev/ttydN, чтобы обнаружить ошибки, отсутствующие записи или отсутствующие специальные файлы устройств.

21.4.6.2. Попробуйте позвонить на модем

Попробуйте дозвониться до системы; убедитесь, что используете 8 бит без четности и 1 стоп бит на удаленной системе. Если вы не получите приглашение сразу, или получите случайные данные, попробуйте нажимать Enter примерно раз в секунду. Если вы все еще не видите приглашения `login:` после нескольких попыток, попробуйте отправить команду `BREAK`. Если вы используете для дозвона высокоскоростной модем, попробуйте позвонить еще раз после фиксирования скорости интерфейса дозванивающегося модема (например, с помощью команды `AT&B1` для модема U.S. Robotics® Sportster®).

Если вы все еще не можете получить приглашение `login:`, проверьте /etc/gettytab еще раз и убедитесь, что

- Имя параметра `getty`, указанного в /etc/ttys, совпадает с именем параметра в /etc/gettytab
- Каждая запись `nx=` соответствует имени другой записи в gettytab

- Каждая запись `tc=` соответствует имени другой записи в gettytab

Если система FreeBSD не отвечает на звонок, убедитесь, что модем настроен для ответа на звонок при включении DTR. Если модем настроен правильно, проверьте, что DTR включается, взглянув на индикаторы модема (если они есть).

Если вы проверили все несколько раз и все еще не добились результата, сделайте перерыв и вернитесь к настройкам позже. Если опять ничего не получилось, возможно вам потребуется отправить письмо в Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD, описав модем и возникшую проблему, участники рассылки попробуют помочь вам.

21.5. Исходящие соединения по модему

Текст, приведенный ниже, это советы, позволяющие настроить ваш хост для доступа к другому компьютеру через модем. Они подходят для установления терминальной сессии с удаленным хостом.

Это подходит для входа на BBS.

Этот вид соединения может очень выручить, если требуется получить файл из интернет и есть проблемы с PPP. Если вам требуется зайти куда-то по FTP, а PPP не работает, используйте терминальную сессию для получения файла по FTP. Затем используйте zmodem для сброса его на свой компьютер.

21.5.1. Мой модем Stock Hayes не поддерживается, что я могу сделать?

На самом деле, страница руководства для `tip` устарела. Встроенная поддержка generic Hayes уже есть. Используйте `at=hayes` в файле /etc/remote.

Драйвер Hayes не умеет работать с некоторыми расширенными возможностями более новых модемов - сообщения вроде `BUSY`, `NO DIALTONE`, или `CONNECT 115200`. Вы должны отключить эти сообщения при использовании `tip` (с помощью `ATX0&W`).

Таймаут дозвона для `tip` составляет 60 секунд. Ваш модем должен использовать меньшее значение, или `tip` решит, что возникли проблемы со связью. Попробуйте `ATS7=45&W`.

Оригинальная `tip` не полностью поддерживает модемы Hayes. Решить это проблему можно отредактировав файл tipconf.h в каталоге /usr/src/usr.bin/tip/tip. Конечно, для этого вам потребуются исходные тексты.

Замените строку `#define HAYES 0` на `#define HAYES 1`. Затем выполните `make` и `make install`. После этого все должно работать отлично.

21.5.2. Как нужно выполнять команды AT?

Сделайте то, что называется "прямой" записью в файле /etc/remote. Например, если модем подключен к первому последовательному порту, /dev/cuad0, добавьте следующую строку:

```
cuad0:dv=/dev/cuad0:br#19200:pa=none
```

Используйте для br наибольшее значение bps, поддерживаемое модемом. Для подключения к модему выполните `tip cuad0`.

Или используйте `cu` под `root` так:

```
# cu -lline -sspeed
```

line это последовательный порт (например /dev/cuad0), а *speed* это скорость (например `57600`). После ввода команд AT наберите `~.` для выхода.

21.5.3. Знак `@` не работает для pn!

Знак `@` в телефонном номере указывает `tip` взять телефонный номер из /etc/phones. Но знак `@` это также специальный символ в таких файлах как /etc/remote. Экранируйте его с помощью обратной косой черты:

```
pn=\@
```

21.5.4. Как я могу позвонить по телефонному номеру из командной строки?

Поместите так называемую "generic" запись в файл /etc/remote. Например:

```
tip115200|Dial any phone number at 115200 bps:\
        :dv=/dev/cuad0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
        :dv=/dev/cuad0:br#57600:at=hayes:pa=none:du:
```

Затем вы можете сделать следующее:

```
# tip -115200 5551234
```

Если вы предпочитаете `cu` команде `tip`, используйте generic запись для `cu`:

```
cu115200|Use cu to dial any number at 115200bps:\
        :dv=/dev/cuad1:br#57600:at=hayes:pa=none:du:
```

и выполните:

```
# cu 5551234 -s 115200
```

21.5.5. Должен ли я вводить значение bps каждый раз?

Создайте запись `tip1200` или `cu1200`, но используйте то значение bps, которое записано в поле br. `tip` считает, что хорошее значение по умолчанию это 1200 bps, поэтому обращается к записи `tip1200`. Тем не менее, значение bps будет другим.

21.5.6. Я получаю доступ ко множеству хостов через терминальный сервер

Вместо ожидания соединения и ввода каждый раз `CONNECT <host>`, используйте возможность `tip cm`. Вот пример записи в /etc/remote:

```
pain|pain.deep13.com|Forrester's machine:\
        :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
        :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
        :dv=/dev/cuad2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

Она позволит вам вводить `tip pain` или `tip muffin` для соединения с хостами pain или muffin, и `tip deep13` для доступа к терминальному серверу.

21.5.7. Может ли tip соединяться более через одну линию для каждого сайта?

Эта проблема часто возникает в университете, где несколько модемных линий и несколько тысяч студентов, пытающихся их использовать.

Создайте запись для университета в /etc/remote и используйте `@` для `pn`:

```
big-university:\
        :pn=\@:tc=dialout
dialout:\
        :dv=/dev/cuad3:br#9600:at=courier:du:pa=none:
```

Затем, создайте список телефонов для университета в /etc/phones:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip` попробует связаться с каждым в указанном порядке, затем прекратит попытки. Если вы хотите продолжать соединяться, запустите `tip` в цикле.

21.5.8. Почему я должен дважды нажать `Ctrl` + `P` для отправки `Ctrl` + `P` один раз?

`Ctrl` + `P` это "управляющий" символ по умолчанию, используемый для указания `tip` того, что далее идут символьные данные. Вы можете сделать любой другой символ управляющим с помощью экранирования `~s`, которое означает "установить переменную".

Введите `~sforce=single-char`, завершив ввод новой строкой. *single-char* это любой одиночный символ. Если вы не введете *single-char*, управляющим символом станет nul, который можно получить,введя `Ctrl` + `2` или `Ctrl` + `Space`.Хорошее значение для *single-char* это `Shift` + `Ctrl` + `6`, которое используется только на некоторых терминальных серверах.

Вы можете использовать в качестве управляющего символа все, что захотите, поместив его в файл $HOME/.tiprc:

```
force=<single-char>
```

21.5.9. Почему все, что я ввожу, вдруг стало отображаться в верхнем регистре??

Вы нажали `Ctrl` + `A`, "повышающий символ" `tip`, который был специально введен для тех, у кого не работает клавиша caps-lock. Используйте `~s` как в примере выше для установки переменной `raisechar` в подходящее значение. Фактически, вы можете установить ее в то же значение, что и управляющий символ, если не собираетесь использовать ни один из них.

Вот пример .tiprc, отлично подходящий для пользователей Emacs, которым часто требуется вводить `Ctrl` + `2` и `Ctrl` + `A`:

```
force=^^
raisechar=^^
```

Символ ^^ это `Shift` + `Ctrl` + `6`.

21.5.10. Могу ли я передавать файлы с помощью `tip`?

Если вы соединяетесь с другой системой UNIX®, возможны передача и прием файлов с помощью команды `~p` (put) и `~t` (take). Эти команды запускают `cat` и `echo` в удаленной системе для приема и передачи файлов. Синтаксис следующий:

`~p` local-file [remote-file]

`~t` remote-file [local-file]

Коррекции ошибок нет, поэтому возможно лучше использовать другой протокол, например zmodem.

21.5.11. Как мне запустить zmodem с `tip`?

Для получения файла запустите отправляющую программу на удаленной стороне. Затем, наберите `~C rz` для начала локального приема файла.

Для отправки файлов запустите принимающую программу на удаленной стороне. Затем, наберите `~C sz файлы` для отправки их на удаленную систему.

21.6. Настройка последовательной консоли

21.6.1. Введение

FreeBSD может загружаться при использовании в качестве консоли текстового терминала на последовательном порту. Такая конфигурация может быть полезна в двух случаях: для системных администраторов, устанавливающих FreeBSD на компьютеры без подключенных клавиатуры или монитора, и для разработчиков, производящих отладку ядра или драйверов устройств.

Как описано в Процесс загрузки FreeBSD, процесс загрузки FreeBSD состоит из трех стадий. Первые две стадии реализованы в блоке загрузки, находящемся в начале слайса FreeBSD на загрузочном диске. На третей стадии загрузочный блок запускает загрузчик (/boot/loader).

Для настройки последовательной консоли вам потребуется настроить блок загрузки, загрузчик и ядро.

21.6.2. Настройка последовательной консоли, краткая версия

В этом разделе предполагается, что вы используете настройки по умолчанию и просто хотите увидеть краткий обзор настройки последовательной консоли.

1. Соедините кабелем последовательный порт COM1 и управляющий терминал.
2. Для того, чтобы сообщения в процессе загрузки выводились в последовательную консоль, выполните от имени суперпользователя команду

   ```
   # echo 'console="comconsole"' >> /boot/loader.conf
   ```

3. Отредактируйте /etc/ttys и измените `off` на `on` и `dialup` на `vt100` для записи `ttyd0`. В противном случае для входа с последовательной консоли не будет требоваться пароль, что может являться проблемой с точки зрения безопасности.
4. Перезагрузите систему и убедитесь, что последовательная консоль активировалась.

Если вам требуется иная конфигурация, обратитесь к более подробному описанию в разделе Настройка последовательной консоли.

21.6.3. Настройка последовательной консоли

1. Подготовьте кабель.

   Вам потребуется нуль-модемный или стандартный последовательный кабель и нуль-модемный адаптер. Обратитесь к Кабели и порты, где рассматриваются последовательные кабели.

2. Отключите клавиатуру.

   Большинство систем PC тестируют клавиатуру во время включения (POST) и выдают ошибку если клавиатура не обнаружена. Некоторые системы при отсутствии клавиатуры выдают звуковой сигнал и не загружаются пока клавиатура не будет подключена.

   Если компьютер сообщает об ошибке, но все же загружается, вам не потребуется делать что-то еще. (Некоторые компьютеры с Phoenix BIOS просто сообщают `Keyboard failed` и продолжают загрузку).

   Если компьютер не загружается без клавиатуры, вам потребуется настроить BIOS так, чтобы отсутствие клавиатуры игнорировалось (если это возможно). Обратитесь к руководству по материнской плате за деталями о том, как это сделать.

   > Установите параметр клавиатуры в настройках BIOS в значение "Not installed". При этом вы сможете продолжать использовать клавиатуру. Все, что делает этот параметр - указывает BIOS не тестировать клавиатуру во время загрузки, поэтому ее отсутствие не вызывает ошибки. Вы можете оставить клавиатуру подключенной, даже если с флагом "Not installed", и она все еще будет работать.

   > Если в к системе подключена PS/2® мышь, отключите ее, как и клавиатуру. Мышь PS/2® использует часть оборудования совместно с клавиатурой, поэтому если оставить ее подключенной, тестирование клавиатуры может ошибочно выдать наличие последней. Например, система Gateway 2000 Pentium 90 MHz ведет себя именно так. К тому же, это не проблема, поскольку мышь без клавиатуры как правило не нужна.

3. Подключите текстовый терминал к COM1 (sio0).

   Если у вас нет текстового терминала, используйте старый PC/XT с модемной программой, или последовательный порт на другом компьютере UNIX®. Если порта COM1 (sio0) нет, подключите его. На данный момент нет способа использовать другой порт вместо COM1 без перекомпиляции загрузочных блоков. Если вы уже используете COM1 для подключения другого устройства, временно удалите это устройство установите новый загрузочный блок и ядро как только FreeBSD заработает. (Предполагается, что COM1 будет доступен на файловом/вычислительном/терминальном сервере в любом случае; если вам

действительно требуется COM1 для чего-то другого (и вы не можете переключить это на COM2 (sio1)), возможно не стоит беспокоиться об этом сейчас.)

4. Убедитесь, что в файле настройки ядра установлены соответствующие флаги для COM1 (sio0).

   Подходящие флаги такие:

   `0x10`

   Включает поддержку консоли для этого устройства. Если установлен этот флаг, другие игнорируются. На данный момент поддержка консоли может быть включена не более чем на одном устройстве; предпочтительно на первом (в соответствии с порядком в конфигурационном файле) с установкой этого флага. Эта опция сама по себе не сделает последовательный порт консолью. Установите следующий флаг или используйте опцию `-h`, описанную ниже, вместе с этим флагом.

   `0x20`

   Включает поддержку консоли на устройстве (если нет другой консоли с более высоким приоритетом), независимо от наличия описываемой ниже опции `-h`. Флаг `0x20` должен использоваться вместе с флагом `0x10`.

   `0x40`

   Резервирует это устройство (совместно с флагом `0x10`) и делает устройство недоступным для обычной работы. Вы не должны использовать этот флаг для устройства последовательного порта, которое будет использоваться в качестве последовательной консоли. Используйте этот флаг только если устройство предназначено для удаленной отладки ядра. Обратитесь к Руководству для разработчиков за дополнительной информацией по удаленной отладке.

   Пример:

   ```
   device sio0 at isa? port IO_COM1 flags 0x10 irq 4
   ```

   Обратитесь к странице справочника sio(4) за подробностями.

   Если флаги не были установлены, вам потребуется запустить UserConfig (на другой консоли) или пересобрать ядро.

5. Создайте boot.config в корневом каталоге раздела `a` на загрузочном диске.

   Этот файл сообщит загрузочному блоку способ загрузки системы. Для активации последовательной консоли вам потребуется одна или несколько следующих опций - несколько опций могут быть указаны на одной строке:

   `-h`

   Переключает внутреннюю и последовательную консоль. Вы можете использовать ее для переключения устройств консоли. Например, при загрузке с

внутренней (видео) консоли, вы можете использовать `-h` для запуска загрузчика и ядра с использованием последовательного порта в качестве устройства консоли. При загрузке с последовательной консоли, вы можете использовать опцию `-h` для указания загрузчику и ядру использовать в качестве консоли видео дисплей.

`-D`

Переключает одно- и двухконсольную конфигурации. В одноконсольной конфигурации консоль может быть либо внутренней (видео дисплей), либо последовательным портом, в зависимости от состояния опции `-h`. В двухконсольной конфигурации и видео дисплей и последовательный порт станут консолями одновременно, независимо от состояния опции `-h`. Имейте ввиду, что конфигурация с двумя консолями работает только во время работы загрузочного блока. Как только управление переходит к загрузчику, остается только одна консоль, указанная опцией `-h`.

`-P`

Указывает загрузочному блоку протестировать клавиатуру. Если клавиатура не найдена, автоматически устанавливаются параметры `-D` и `-h`.

По причине ограничений на размер в существующей версии загрузочного блока, опция `-P` может протестировать только расширенные клавиатуры. Клавиатуры с менее чем 101 клавишами (и без клавиш F11 и F12) могут быть не обнаружены. Клавиатуры некоторых лэптопов могут быть не найдены из-за этого ограничения. Если это случилось, вы не сможете использовать опцию `-P`. К сожалению, не существует обходного пути решения этой проблемы.

Используйте или опцию `-P` для автоматического выбора консоли, или опцию `-h` для активации последовательной консоли.

Вы можете включить также другие опции, описанные в boot(8).

Опции, за исключением `-P`, будут переданы загрузчику (/boot/loader). Загрузчик определит будет ли консолью внутреннее видео устройство или последовательный порт, проверив только состояние опции `-h`. Это означает, что если вы включите в /boot.config опцию `-D`, но не `-h`, то сможете использовать консоль только во время работы загрузочного блока; загрузчик будет использовать внутреннее видео устройство в качестве консоли.

6. Загрузите компьютер.

Когда вы включите компьютер FreeBSD, загрузочный блок выведет содержимое /boot.config на консоль. Например:

```
/boot.config: -P
Keyboard: no
```

Вторая строка появится только если вы поместите -P в /boot.config и отражает наличие/отсутствие клавиатуры. Эти сообщения выводятся либо на последовательную, либо на внутреннюю консоль, или на обе, в зависимости от параметров в /boot.config.

Опции	Сообщения выводятся на
нет	внутренняя консоль
-h	последовательная консоль
-D	последовательная и внутренняя консоли
-Dh	последовательная и внутренняя консоли
-P, клавиатура присутствует	внутренняя консоль
-P, клавиатура отсутствует	последовательная консоль

После вывода вышеприведенных сообщений, происходит небольшая пауза перед тем, как запускается загрузчик и на консоли появляются следующие сообщения. В нормальной ситуации вам не потребуется прерывать загрузку в этот момент, но это можно сделать, чтобы убедиться, что все настроено правильно.

Нажмите на консоли любую клавишу кроме Enter для прерывания процесса загрузки. Загрузочный блок выдаст приглашение к дальнейшим действиям. Оно выглядит примерно так:

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Убедитесь, что сообщение выше появилось на последовательной, внутренней консоли или на обеих, в зависимости от опций в /boot.config. Если сообщение появилось там, где должно было появиться, нажмите Enter для продолжения процесса загрузки.

Если вам нужна последовательная консоль, но на терминале не видно приглашения, это означает проблемы с настройками. Введите -h и нажмите Enter/Return (если это возможно) для указания загрузочному блоку (а также загрузчику и ядру) выбрать последовательный порт в качестве консоли. Когда система загрузится, проверьте настройки еще раз и определите, что было сделано неправильно.

После запуска загрузчика и перехода в третью стадию процесса загрузки вы все еще можете переключиться между внутренней консолью и последовательной консолью путем установки соответствующих переменных окружения в загрузчике. Обращайтесь к разделу Изменение консоли из загрузчика.

21.6.4. Итоги

Здесь приведены краткие итоги по различным настройкам, рассмотренным в этом разделе и выбираемым в соответствии с ними консолям.

21.6.4.1. Вариант 1: вы устанавливаете для sio0 флаги 0x10

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Параметры в /boot.config	Консоль для загрузочного блока	Консоль для загрузчика	Консоль для ядра
нет	внутренняя	внутренняя	внутренняя
`-h`	последовательная	последовательная	последовательная
`-D`	последовательная и внутренняя	внутренняя	внутренняя
`-Dh`	последовательная и внутренняя	последовательная	последовательная
`-P`, клавиатура присутствует	внутренняя	внутренняя	внутренняя
`-P`, клавиатура отсутствует	последовательная и внутренняя	последовательная	последовательная

21.6.4.2. Вариант 2: вы устанавливаете для sio0 флаги 0x30

```
device sio0 at isa? port IO_COM1 flags 0x30 irq 4
```

Параметры в /boot.config	Консоль для загрузочного блока	Консоль для загрузчика	Консоль для ядра
нет	внутренняя	внутренняя	последовательная
`-h`	последовательная	последовательная	последовательная
`-D`	последовательная и внутренняя	внутренняя	последовательная
`-Dh`	последовательная и внутренняя	последовательная	последовательная
`-P`, клавиатура присутствует	внутренняя	внутренняя	последовательная
`-P`, клавиатура отсутствует	последовательная и внутренняя	последовательная	последовательная

21.6.5. Приемы работы с последовательной консолью

21.6.5.1. Установка более высокой скорости порта

По умолчанию, последовательный порт настроен так: 9600 бит/с, 8 бит, без четности, 1 стоп бит. Если вам необходимо изменить скорость, потребуется перекомпиляция как минимум загрузочных блоков. Добавьте следующую строку к /etc/make.conf и скомпилируйте новый загрузочный блок:

```
BOOT_COMCONSOLE_SPEED=19200
```

Обратитесь к Использование для консоли другого последовательного порта вместо sio0 за подробными инструкциями по сборке и установке новых загрузочных блоков.

Если последовательная консоль настраивается не путем установки параметра `-h`, или последовательная консоль, используемая ядром, отличается от той, что используется загрузочным блоком, потребуется добавить следующие опции к файлу настройки ядра и собрать новое ядро:

```
options CONSPEED=19200
```

21.6.5.2. Использование для консоли другого последовательного порта вместо sio0

Использование другого последовательного порта вместо sio0 для консоли потребует кое-какой перекомпиляции. Если вы по каким-либо причинам хотите использовать другой последовательный порт, перекомпилируйте загрузочный блок, загрузчик и ядро согласно приведенной ниже инструкции.

1. Получите исходные тексты ядра (глава Обновление системы и смена версии FreeBSD)
2. Отредактируйте /etc/make.conf и установите `BOOT_COMCONSOLE_PORT` в соответствии с адресом порта, который вы хотите использовать (0x3F8, 0x2F8, 0x3E8 или 0x2E8). Могут быть использованы только устройства от sio0 до sio3 (от COM1 до COM4); мультипортовые последовательные карты не будут работать. Установка прерываний не требуется.
3. Создайте файл настройки ядра и добавьте соответствующие флаги для порта, который планируется использовать. Например, если вы хотите использовать для консоли sio1 (COM2):

   ```
   device sio1 at isa? port IO_COM2 flags 0x10 irq 3
   ```

   или

```
device sio1 at isa? port IO_COM2 flags 0x30 irq 3
```

Флаги для других последовательных устройств не устанавливайте.

4. Соберите и установите загрузочный блок и загрузчик:

```
# cd /sys/boot
# make clean
# make
# make install
```

5. Соберите и установите ядро.
6. Запишите загрузочный блок на загрузочный диск с помощью bsdlabel(8) и загрузитесь с новым ядром.

21.6.5.3. Вход в отладчик DDB с последовательной линии

Если вы хотите войти в отладчик ядра с последовательной консоли (полезно для удаленной диагностики, но опасно если вы введете неправильный BREAK на последовательном порту!), потребуется собрать ядро со следующими параметрами:

```
options BREAK_TO_DEBUGGER
options DDB
```

21.6.5.4. Получение приглашения на последовательной консоли

Хотя это не обязательно, вам может потребоваться приглашение *login* по последовательной линии, в дополнение к уже доступным загрузочным сообщениям и отладочной сессии ядра. Здесь описано как сделать это.

Откройте файл /etc/ttys с помощью редактора и найдите строки:

```
ttyd0 "/usr/libexec/getty std.9600" unknown off secure
ttyd1 "/usr/libexec/getty std.9600" unknown off secure
ttyd2 "/usr/libexec/getty std.9600" unknown off secure
ttyd3 "/usr/libexec/getty std.9600" unknown off secure
```

Строки от `ttyd0` до `ttyd3` соответствуют портам от COM1 до COM4. Измените `off` на `on` для требуемого порта. Если вы изменили скорость последовательного порта, может потребоваться изменить `std.9600` для соответствия текущим настройкам, например `std.19200`.

Возможно, вы захотите заменить тип терминала `unknown` на тип реально используемого терминала.

После редактирования файла потребуется выполнить `kill -HUP 1` для включения новых настроек.

21.6.6. Изменение консоли из загрузчика

Предыдущий раздел описывает настройку последовательной консоли изменением параметров загрузочного блока. Этот раздел показывает, как указать консоль, вводя команды и переменные окружения для загрузчика. Поскольку загрузчик загружается после загрузочного блока, на третьей стадии загрузочного процесса, настройки загрузчика превалируют над настройками загрузочного блока.

21.6.6.1. Настройка последовательной консоли

Вы можете прямо указать загрузчику и ядру использовать последовательную консоль, записав одну строку в /boot/loader.rc:

```
set console="comconsole"
```

Это сработает независимо от настроек загрузочного блока, рассмотренных в предыдущем разделе.

Поместите эту строку в самое начало /boot/loader.rc, чтобы увидеть на последовательной консоли все загрузочные сообщения.

Вы можете также указать внутреннюю консоль:

```
set console="vidconsole"
```

Если вы не установите переменную загрузчика `console`, загрузчик, а затем и ядро будут использовать ту консоль, которая установлена параметром `-h` для загрузочного блока.

В версиях 3.2 или выше, вы можете указать консоль в /boot/loader.conf.local или /boot/loader.conf вместо /boot/loader.rc. С этим методом /boot/loader.rc должен выглядеть примерно так:

```
include /boot/loader.4th
start
```

Затем, создайте /boot/loader.conf.local и поместите туда следующую строку.

```
console=comconsole
```

или

```
console=vidconsole
```

Обращайтесь к loader.conf(5) за дополнительной информацией.

На данный момент у загрузчика нет параметра, эквивалентного параметру `-P` загрузочного блока и нет способа автоматического выбора внутренней и последовательной консоли в зависимости от наличия клавиатуры.

21.6.6.2. Использование для консоли отличного от sio0 последовательного порта

Вам потребуется перекомпилировать загрузчик для использования отличного от sio0 последовательного порта в качестве консоли. Следуйте процедуре, описанной в разделе Использование для консоли другого последовательного порта вместо sio0.

21.6.7. Предостережения

Идея в том, чтобы настроить выделенный сервер, который не требует графического оборудования или подсоединенной клавиатуры. К сожалению, хотя многие системы способны загрузиться без клавиатуры, есть совсем немного систем, способных загрузиться без графического адаптера. Компьютеры с AMI BIOS могут быть настроены для загрузки без графического адаптера простой установкой параметра настройки CMOS "graphics adapter" в значение "Not installed".

Однако, многие компьютеры не поддерживают этот параметр и не смогут загрузиться без графического оборудования. Для этих компьютеров вам потребуется оставить подключенной любую графическую карту (даже если это просто старая моно карта), хотя монитор и не подключен.

Глава 22. PPP и SLIP

22.1. Краткий обзор

В FreeBSD существует множество способов соединения одного компьютера с другим. Для установления соединения с отдельной сетью или интернет через обычный модем, или для открытия доступа к собственному компьютеру необходимо использование PPP или SLIP. В этой главе детально описана настройка таких модемных сервисов.

После прочтения этой главы вы будете знать:

- Как настроить PPP уровня пользователя (user PPP).
- Как настроить PPP уровня ядра (kernel PPP).
- Как настроить PPPoE (PPP over Ethernet).
- Как настроить PPPoA (PPP over ATM).
- Как настроить SLIP клиента и сервер.

Перед прочтением этой главы вам потребуется:

- Ознакомиться с основными сетевыми технологиями.
- Понимать основы и назначение модемного соединения и PPP и/или SLIP.

Вы возможно захотите узнать, в чем главное различие между PPP уровня пользователя и ядра. Ответ прост: PPP уровня пользователя обрабатывает входящие и исходящие данные в пространстве пользователя, а не в ядре. В терминах копирования данных между ядром и пространством пользователя это дорогостоящий путь, который однако позволяет значительно расширить возможности реализации PPP. PPP уровня пользователя для связи с внешним миром использует устройство tun, а PPP уровня ядра - устройство ppp.

В дальнейшем в этой главе PPP уровня пользователя будет обозначаться просто как ppp, пока не потребуется различать его и любое другое программное обеспечение PPP, такое как pppd. Если не указано иначе, все команды, приведенные в этой главе, должны выполняться под `root`.

22.2. PPP уровня пользователя

22.2.1. Настройка PPP уровня пользователя

22.2.1.1. Предположения

В этом документе предполагается, что у вас есть следующее:

- Учетная запись у провайдера интернет (Internet Service Provider, ISP), к которому вы подсоединяетесь, используя PPP.
- Модем или другое подключенное к системе и правильно настроенное устройство,

позволяющее подключиться к провайдеру.

- Номер модемного пула провайдера.
- Имя пользователя (логин) и пароль (обычная UNIX® пара логин/пароль, или PAP/CHAP пара логин/пароль).
- IP адреса одного или нескольких серверов имен. Обычно провайдер дает для этих целей два IP адреса. Если нет ни одного, вы можете использовать команду `enable dns` в ppp.conf и ppp настроит список серверов имен. Эта возможность зависит от наличия поддержки согласования DNS в реализации PPP провайдера.

Следующая информация может поставляться провайдером, но не является совершенно необходимой:

- IP адрес шлюза провайдера. Шлюз это компьютер, к которому вы подключитесь и который будет настроен в качестве *маршрута по умолчанию* (*default route*). Если у вас нет этой информации, она может быть получена от PPP сервера после подключения.

  Программой ppp этот IP адрес обозначается как `HISADDR`.

- Сетевая маска, которую вы должны использовать. Если провайдер не предоставил ее значение, вы можете использовать `255.255.255.255`.
- Если провайдер предоставил статический IP и имя хоста, используйте их. Иначе позвольте удаленной стороне назначить свободный IP адрес.

Если у вас нет всей необходимой информации, свяжитесь с провайдером.

В этом разделе строки файлов настройки из многих примеров пронумерованы. Эти номера приведены только для обсуждения настроек, они не должны помещаться в действующую настройку. Правильные отступы с табуляцией и пробелами также важны.

22.2.1.2. Создание файлов устройств PPP

В обычной ситуации, большинству пользователей нужно только одно устройство tun (/dev/tun0). Ссылки на tun0 ниже могут быть заменены на tunN, где *N* это любой номер устройства, соответствующий вашей системе.

Для систем FreeBSD без devfs(5) (FreeBSD 4.X более ранние), необходимо проверить устройство tun0 (это не требуется, если включена devfs(5), поскольку файлы устройств будут создаваться автоматически).

Простейший способ убедиться, что устройство tun0 настроено правильно, это пересоздать устройство. Для пересоздания устройства выполните следующее:

```
# cd /dev
# sh MAKEDEV tun0
```

Если вам необходимы 16 туннельных устройств, потребуется их создать. Это можно сделать,

выполнив следующие команды:

```
# cd /dev
# sh MAKEDEV tun15
```

22.2.1.3. Автоматическая настройка PPP

И `ppp` и `pppd` (реализация PPP уровня ядра) используют файлы настройки, расположенные в каталоге /etc/ppp. Примеры для ppp уровня пользователя можно найти в /usr/shared/examples/ppp/.

Настройка `ppp` требует редактирования нескольких файлов, в зависимости от ваших потребностей. То, что вы поместите в эти файлы, зависит в некоторой степени от того, предоставит ли провайдер статический IP адрес (т.е. вы получите один определенный IP адрес и будете использовать его постоянно) или динамический (т.е. ваш IP адрес будет изменяться при каждом подключении к провайдеру).

22.2.1.3.1. PPP и статические IP адреса

Вам потребуется отредактировать файл настройки /etc/ppp/ppp.conf. Он похож на приведенный ниже пример.

> Строки, оканчивающиеся на `:`, вводятся без отступа в начале строки, остальные строки должны быть введены с отступом, как показано в примере.

```
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION (built COMPILATIONDATE)
4       set device /dev/cuaa0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7                 \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8       set timeout 180
9       enable dns
10
11    provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set login "TIMEOUT 10 \"\" \"\" gin:--gin: \\U word: \\P col: ppp"
16      set timeout 300
17      set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18      add default HISADDR
```

Строка 1

Начинает настройку по умолчанию (default). Команды этой настройки выполняются

автоматически при запуске ppp.

Строка 2

Включает параметры протоколирования. Когда настройка работает удовлетворительно, эта строка должна быть сокращена до следующей формы

```
set log phase tun
```

для предотвращения появления слишком больших лог файлов.

Строка 3

Указывает PPP как идентифицировать себя на удаленной стороне. PPP идентифицирует себя на удаленной стороне если возникают проблемы согласования и установки соединения, предоставляя информацию, по которой администратор на удаленной стороне может воспользоваться для решения таких проблем.

Строка 4

Указывает устройство, к которому подключен модем. COM1 это /dev/cuaa0, а COM2 это /dev/cuaa1.

Строка 5

Устанавливает желаемую скорость подключения к модему. Если 115200 не работает (хотя должна работать для любого относительно нового модема) попробуйте 38400.

Строки 6 и 7

Строка дозвона. PPP уровня пользователя применяет expect-send синтаксис, похожий на синтаксис chat(8). Обратитесь к странице справочника за информацией о возможностях этого языка.

Обратите внимание, что эта команда продолжается на следующей строке для улучшения читаемости. Любая команда в ppp.conf может быть продолжена на следующей строке, если последний символ предыдущей строки “\”.

Строка 8

Устанавливает предельное время ожидания для соединения. Значение 180 секунд используется по умолчанию, так что строка с этим значением чисто косметическая.

Строка 9

Указывает PPP запросить у удаленной стороны сервера имен. Если вы работаете с локальным сервером имен, эта строка должна быть закомментирована или удалена.

Строка 10

Пустая строка для улучшения читаемости. Пустые строки игнорируются PPP.

Строка 11

Определяет настройки для провайдера, называемого "provider". Это имя может быть изменено на имя вашего провайдера, чтобы в дальнейшем вы могли использовать `load`

`provider` для начала соединения.

Строка 12

Определяет телефонный номер для этого провайдера. Несколько телефонных номеров могут быть указаны с помощью двоеточия (`:`) или символа канала (`|`) в качестве разделителя. Различия между двумя разделителями описаны в ppp(8). Главным образом они заключаются в том, что если вы хотите перебирать номера, используйте двоеточие. Если вы хотите дозваниваться по первому номеру в первую очередь, и использовать другие номера только если дозвон по первому завершится неудачно, используйте символ канала. Всегда заключайте список номеров в кавычки, как показано в примере.

Вы должны включить телефонный номер в кавычки (`"`) если в нем используются пробелы. Отсутствие кавычек может вызвать простую, но трудно обнаруживаемую ошибку.

Строки 13 и 14

Задают имя пользователя и пароль. При использовании приглашения login в стиле UNIX®, эти значения используются командой `set login` через переменные \U и \P. При соединении с использованием PAP или CHAP, эти значения используются во время аутентификации.

Строка 15

Если вы используете PAP или CHAP, приглашение на вход не появится, и эта строка должна быть закомментирована или удалена. Обратитесь к странице аутентификация PAP и CHAP за дальнейшей информацией.

Строка для входа записана в том же chat-подобном синтаксисе, что и строка для дозвона. В этом примере строка работает для сервиса, сессия входа которого выглядит примерно так:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

Вам потребуется изменить эту строку для использования с другим сервисом. При первом составлении скрипта убедитесь, что вы включили "chat" протоколирование, чтобы убедиться, что соединение происходит как ожидалось.

Строка 16

Установка максимального времени ожидания по умолчанию для соединения. В данном случае соединение будет разорвано автоматически после 300 секунд неактивности. Если вы не хотите, чтобы соединение разрывалось, установите эту переменную в нуль, или используйте параметр командной строки `-ddial`.

Строка 17

Устанавливает адрес интерфейса. Строка *x.x.x.x* должна быть заменена на IP адрес, который выделил вам провайдер. Строка *y.y.y.y* должна быть заменена на IP адрес шлюза

провайдера (компьютер, к которому вы подключаетесь). Если провайдер не сообщил адрес шлюза, используйте 10.0.0.2/0. Если вам требуется использовать "вычисленные" адреса, убедитесь, что создана запись в /etc/ppp/ppp.linkup в соответствии с инструкциями для PPP и динамических IP адресов. Если эта строка опущена, ppp не может быть запущен в режиме -auto.

Строка 18

Добавляет маршрут по умолчанию к шлюзу провайдера. Специальное слово HISADDR заменяется адресом шлюза, указанном в строке 17. Важно, чтобы эта строка появилась после строки 17, до нее переменная HISADDR еще не инициализирована.

Если вы не будете запускать ppp с параметром -auto, эта строка должна быть перемещена в файл ppp.linkup.

Нет необходимости добавлять запись в ppp.linkup, если у вас статический IP адрес и ppp работает в режиме -auto, поскольку таблица маршрутизации настроена правильно еще до подключения. Однако, вы возможно захотите создать запись для запуска программ после соединения. Эта ситуация описана далее в примере по sendmail.

Примерные файлы настройки находятся в каталоге /usr/shared/examples/ppp/.

22.2.1.3.2. PPP и динамические IP адреса

Если провайдер не выделил статический IP адрес, ppp может быть настроен для определения локального и удаленного адреса. Это делается путем "вычисления" IP адреса и настройки его программой ppp с использованием IP Configuration Protocol (IPCP) после установления соединения. Файл настройки ppp.conf тот же, что и в примере PPP и статические IP адреса, со следующим изменением:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255
```

Как и раньше, не включайте номер строки, он используется только для ссылки на строку в этом примере. Требуется отступ хотя бы в один пробел.

Строка 17

Номер после символа / это число бит в адресе, которые будут запрошены ppp. Вы можете использовать более подходящие вам IP адреса, но пример выше всегда будет работать.

Если вы не используете режим -auto, потребуется создать запись в /etc/ppp/ppp.linkup. Этот файл используется после установки соединения. На этот момент ppp уже настроит адреса интерфейсов и станет возможным добавление записей в таблицу маршрутизации:

```
1     provider:
2      add default HISADDR
```

Строка 1

При установке соединения, ppp ищет запись в ppp.linkup по следующим правилам:

сначала в соответствии с меткой, используемой в ppp.conf. Если это не сработает, ведется поиск записи для IP адреса шлюза. Это метка в IP записывается в виде IP адреса. Если запись все еще не найдена, используется запись `MYADDR`.

Строка 2

Эта строка сообщает `ppp` добавить маршрут по умолчанию, указывающий на `HISADDR`. `HISADDR` будет заменен на IP адрес шлюза, определенного IPCP.

Детальный пример находится в записи `pmdemand` файлов /usr/shared/examples/ppp/ppp.conf.sample и /usr/shared/examples/ppp/ppp.linkup.sample.

22.2.1.3.3. Прием входящих звонков

При настройке ppp для приема входящих звонков на компьютере, подключенном к локальной сети, вам необходимо решить, перенаправлять ли пакеты в локальную сеть. Если вы будете делать это, выделите удаленной стороне IP адрес из диапазона адресов локальной сети, и используйте команду `enable proxy` в файле /etc/ppp/ppp.conf. Вам потребуется также убедиться, что в файле /etc/rc.conf присутствует строка:

```
gateway_enable="YES"
```

22.2.1.3.4. Какой getty?

Раздел Настройка FreeBSD для входящих соединений дает хорошее описание включения входящих соединений с использованием getty(8).

Альтернатива `getty` это mgetty, более интеллектуальная версия `getty`, разработанная специально для приема входящих звонков..

Преимущество использование `mgetty` в том, что она активно *общается* с модемами, то есть если порт будет выключен в /etc/ttys, модем не будет отвечать на звонок.

Последние версии `mgetty` (от 0.99beta и выше) поддерживают также автоматическое определение потоков PPP, позволяя клиентам безскриптовое подключение к серверу.

Обратитесь к разделу Mgetty и AutoPPP за дальнейшей информацией по `mgetty`.

22.2.1.3.5. Права PPP

Программа `ppp` обычно запускается с правами пользователя `root`. Если вы хотите разрешить запуск `ppp` в режиме сервера с правами обычного пользователя путем запуска `ppp` как описано ниже, этого пользователя необходимо добавить в группу `network` в файле /etc/group.

Вам также потребуется дать ему доступ к одному или более разделов конфигурации, используя команду `allow`:

```
allow users fred mary
```

Если эта команда используется в разделе `default`, она дает пользователю полный доступ.

22.2.1.3.6. Оболочки PPP для пользователей с динамическими IP

Создайте файл, называющийся /etc/ppp/ppp-shell и содержащий следующее:

```
#!/bin/sh
IDENT=`echo $0 | sed -e 's/^.*-\(.*\)$/\1/'`
CALLEDAS="$IDENT"
TTY=`tty`

if [ x$IDENT = xdialup ]; then
        IDENT=`basename $TTY`
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

Этот скрипт должен быть исполняемым. Теперь создайте на этот скрипт символическую ссылку с именем ppp-dialup с помощью следующей команды:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

Используйте этот скрипт в качестве *оболочки* для удаленных пользователей. Ниже приведен пример записи в /etc/password для удаленных пользователей PPP с именем пользователя `pchilds` (не забывайте использовать `vipw` для редактирования файла паролей).

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Создайте каталог /home/ppp, который доступен для чтения и содержит следующие файлы нулевой длины:

```
-r--r--r--   1 root     wheel      0 May 27 02:23 .hushlogin
-r--r--r--   1 root     wheel           0 May 27 02:22 .rhosts
```

Это предотвратит отображение /etc/motd.

22.2.1.3.7. Оболочки PPP для пользователей со статическими IP

Создайте файл ppp-shell как в примере выше, и для каждой учетной записи со статически назначаемым IP создайте символическую ссылку на ppp-shell.

Например, если у вас три пользователя удаленного доступа, `fred`, `sam` и `mary`, которые подключаются к вашей сети класса C, выполните следующее:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
```

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

Оболочка каждого из этих пользователей удаленного доступа должна быть символической ссылкой, созданной выше (например, оболочка пользователя `mary` должна быть /etc/ppp/ppp-mary).

22.2.1.3.8. Настройка ppp.conf для пользователей с динамическими IP

Файл /etc/ppp/ppp.conf должен содержать примерно такие строки:

```
default:
  set debug phase lcp chat
  set timeout 0

ttyd0:
  set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
  enable proxy

ttyd1:
  set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
  enable proxy
```

Необходимо соблюдать отступы.

Раздел `default:` загружается для каждого соединения. Для каждой строки, включенной в /etc/ttys, создайте запись, подобную `ttyd0:` выше. Каждая строка должна содержать уникальные IP адреса из вашего пула IP адресов, выделенных пользователям с динамическими IP.

22.2.1.3.9. Настройка ppp.conf для пользователей со статическими IP

Помимо содержимого, описанного в примере файла /usr/shared/examples/ppp/ppp.conf выше, вам потребуется добавить раздел для каждого из пользователей со статическими IP. Мы продолжим использовать имена `fred`, `sam` и `mary` в качестве примера.

```
fred:
  set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
  set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
  set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

Если требуется, файл /etc/ppp/ppp.linkup должен также содержать информацию о маршрутизации для каждого пользователя со статическим IP. В строке ниже через

клиентское соединение добавляется маршрут к сети класса C`203.14.101.0`.

```
fred:
  add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
  add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
  add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

22.2.1.3.10. `mgetty` и AutoPPP

Настройка и компиляция `mgetty` с параметром `AUTO_PPP` позволяет `mgetty` определять LCP фазу PPP соединений и автоматически порождать оболочку ppp. Однако, поскольку стандартный метод логин/пароль не используется, необходима аутентификация пользователей через PAP или CHAP.

В этом разделе предполагается, что пользователь успешно настроил, скомпилировал и установил версию `mgetty` с параметром `AUTO_PPP` (v0.99beta или более поздняя).

Убедитесь, что в файле /usr/local/etc/mgetty+sendfax/login.config имеется следующая строка:

```
/AutoPPP/ -     -             /etc/ppp/ppp-pap-dialup
```

Это укажет `mgetty` запускать скрипт ppp-pap-dialup для обнаруженных соединений PPP.

Создайте файл /etc/ppp/ppp-pap-dialup, содержащий следующее (этот файл должен быть выполняемым):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Для каждой линии, включенной в /etc/ttys, создайте соответствующую запись в /etc/ppp/ppp.conf. Она будет отлично сочетаться с тем, что было создано выше.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Для каждого пользователя, входящего по этому методу, в файле /etc/ppp/ppp.secret должна присутствовать запись с логином/паролем, или, в качестве альтернативы, для аутентификации пользователей по PAP через /etc/password, необходимо использовать следующий параметр.

```
enable passwdauth
```

Если вы хотите присвоить некоторым пользователям статический IP, задайте его в качестве третьего аргумента в /etc/ppp/ppp.secret. Обратитесь к к /usr/shared/examples/ppp/ppp.secret.sample за примерами.

22.2.1.3.11. MS расширения

Возможна настройка PPP для выдачи адресов DNS и NetBIOS по запросу.

Для включения этих расширений с PPP версии 1.x, необходимо добавить к соответствующему разделу /etc/ppp/ppp.confследующие строки.

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Для PPP версии 2 и выше:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Клиентам будут выдаваться адреса первичного и вторичного серверов имен, и адрес хоста NetBIOS.

Если в версии 2 и выше строка `set dns` будет опущена, PPP использует значения из /etc/resolv.conf.

22.2.1.3.12. Аутентификация PAP и CHAP

Некоторые провайдеры настраивают систему так, что аутентификация производится с использованием либо PAP либо CHAP. В этом случае, приглашение `login:` при соединении не выдается и соединение PPP начинается сразу.

PAP менее безопасен, чем CHAP, но безопасность в данном случае не страдает, поскольку хотя пароли и передаются открытым текстом, они передаются только по модемной линии. У кракеров не так много возможностей для кражи паролей.

В примерах выше PPP и статические IP адреса или PPP и динамические IP адреса должны быть сделаны следующие изменения:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

Строка 13

Эта строка указывает имя пользователя PAP/CHAP. Вам потребуется заменить *MyUserName* на правильное значение.

Строка 14

Эта строка указывает пароль PAP/CHAP. Вам потребуется заменить *MyPassword* на правильное значение. Вы можете также добавить дополнительную строку, такую как:

```
16      accept PAP
```

или

```
16      accept CHAP
```

для явного указания протокола, но и PAP и CHAP поддерживаются по умолчанию.

Строка 15

Ваш провайдер обычно не потребует входа на сервер при использовании PAP или CHAP. Следовательно, вы должны отключить строку "set login".

22.2.1.3.13. Изменение настроек `ppp` "на лету"

Возможно изменение настроек `ppp` программы во время ее работы в фоновом режиме, но только если открыт соответствующий диагностический порт. Для его открытия добавьте в настройку следующую строку:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

С этой директивой ppp будет прослушивать заданный UNIX® сокет, запрашивая у клиентов пароль перед выдачей доступа. Символы`%d` заменяются на номер используемого устройства tun.

Как только сокет открыт, в скриптах, выполняющих настройку работающей программы, можно использовать pppctl(8).

22.2.1.4. Использование NAT

PPP имеет возможность использовать встроенный NAT без преобразования пакетов в ядре. Эта возможность может быть включена следующей строкой в /etc/ppp/ppp.conf:

```
nat enable yes
```

Альтернативно, PPP NAT может быть включен параметром командной строки `-nat`. Существует также переменная /etc/rc.conf`ppp_nat`, которая включена по умолчанию.

Если вы используете эту возможность, вы также можете найти полезными параметры /etc/ppp/ppp.conf для включения пересылки входящих соединений:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

или отключить все входящие соединения

```
nat deny_incoming yes
```

22.2.1.5. Завершающая настройка системы

Теперь, когда программа `ppp` настроена, осталось выполнить еще несколько действий прежде, чем все это заработает. Они выполняются путем редактирования файла /etc/rc.conf.

Просматривая этот файл, убедитесь, что добавлена строка `hostname=`, например:

```
hostname="foo.example.com"
```

Если провайдер предоставил вам статический IP адрес и имя, возможно лучше всего использовать это имя в качестве имени хоста.

Проверьте переменную `network_interfaces`. Если вы хотите настроить систему для дозвона по требованию, убедитесь, что устройство tun0 добавлено в список, иначе удалите его.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```

Переменная `ifconfig_tun0` должна быть пуста, необходимо также создать файл /etc/start_if.tun0. В этом файле должна находиться строка:

```
ppp -auto mysystem
```

Этот скрипт выполняется во время настройки сети, запуская даемона ppp в автоматическом режиме. Если эта машина является шлюзом для локальной сети, вы можете также использовать переключатель `-alias`. Обратитесь к странице справочника за дальнейшей информацией.

Убедитесь, что программа маршрутизации отключена переменной в файле /etc/rc.conf:

```
router_enable="NO"
```

Важно, чтобы даемон `routed` не был запущен, поскольку он может удалить запись маршрута

по умолчанию, создаваемую ppp.

Возможно, стоит обратить внимание на переменную sendmail_flags и убедиться, что она не включает параметр -q, иначе sendmail попытается сразу же обратиться к сети, и компьютер может начать дозвон. Вы можете попробовать:

```
sendmail_flags="-bd"
```

Обратная сторона этого решения в том, что необходим запуск очереди sendmail после поднятия соединения ppp:

```
# /usr/sbin/sendmail -q
```

Вы можете использовать команду !bg в файле ppp.linkup для автоматического выполнения этой задачи:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

Если вам это не нужно, возможна настройка "dfilter" для блокирования SMTP трафика. Обратитесь файлам примеров за дальнейшей информацией.

Все, что осталось, это перегрузить компьютер. После перезагрузки вы можете либо выполнить:

```
# ppp
```

и затем набрать dial provider для запуска сессии PPP, либо, если вы хотите, чтобы программа ppp начинала соединение автоматически при появлении исходящего трафика (и файл start_if.tun0 не создан), выполните:

```
# ppp -auto provider
```

22.2.1.6. Итоги

Для первоначальной настройки ppp необходимо пройти следующие шаги:

Сторона клиента:

1. Убедитесь, что устройство tun встроено в ядро.
2. Убедитесь, что устройства tunN находятся в каталоге /dev.

3. Создайте запись в /etc/ppp/ppp.conf. Пример pmdemand должен подойти для большинства провайдеров.
4. Если у вас динамический IP адрес, создайте запись в /etc/ppp/ppp.linkup.
5. Обновите файл /etc/rc.conf.
6. Создайте скрипт start_if.tun0, если необходим дозвон по требованию.

Сторона сервера:

1. Убедитесь, что устройство tun встроено в ядро.
2. Убедитесь, что устройства tunN находятся в каталоге /dev.
3. Создайте запись в /etc/passwd (используя программу vipw(8)).
4. Создайте профиль в домашнем каталоге пользователя, запускающий `ppp -direct direct-server` или подобную команду.
5. Создайте запись в /etc/ppp/ppp.conf. Пример direct-server должен подойти.
6. Создайте запись в /etc/ppp/ppp.linkup.
7. Обновите файл /etc/rc.conf.

22.3. PPP уровня ядра

22.3.1. Настройка PPP уровня ядра

Перед началом настройки PPP на вашем компьютере, убедитесь, что `pppd` находится в /usr/sbin и каталог /etc/ppp существует.

`pppd` может работать в двух режимах:

1. В качестве "клиента" - когда вы хотите подключить компьютер к внешнему миру через последовательное соединение PPP или модемную линию.
2. В качестве "сервера" - когда компьютер подключен к сети и используется для подключения других компьютеров через PPP.

В обеих случаях вам потребуется настроить файл параметров (/etc/ppp/options или ~/.ppprc если на вашем компьютере более одного пользователя работают с PPP).

Вам потребуется также программа для модемных/последовательных линий (предпочтительно comms/kermit) для дозвона и установки соединения с удаленным хостом.

22.3.2. Использование `pppd` в качестве клиента

Для подключения к линии PPP терминального сервера Cisco может использоваться следующий файл /etc/ppp/options.

```
crtscts  # enable hardware flow control
modem           # modem control line
noipdefault     # remote PPP server must supply your IP address
                # if the remote host does not send your IP during IPCP
                # negotiation, remove this option
passive         # wait for LCP packets
domain ppp.foo.com      # put your domain name here

:<remote_ip>    # put the IP of remote PPP host here
                # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to <local_ip>:<remote_ip>

defaultroute    # put this if you want that PPP server will be your
                # default router
```

Для подключения:

1. Дозвонитесь на удаленный хост, используя kermit (или любую другую модемную программу), и введите ваше имя пользователя и пароль (или те данные, которые требуются для установления PPP соединения с удаленным хостом).
2. Выйдите из kermit (без обрыва соединения).
3. Введите следующее:

   ```
   # /usr/src/usr.sbin/pppd.new/pppd /dev/tty01 19200
   ```

   Убедитесь, что выбраны подходящая скорость и правильное имя устройства.

Теперь ваш компьютер подключен по PPP. Если соединение не состоялось, вы можете добавить параметр `debug` к файлу /etc/ppp/options, и отследить проблему по сообщениям на консоли.

Следующий скрипт /etc/ppp/pppup проведет все 3 стадии в автоматическом режиме:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
```

```
        kill -9 ${pid}
fi

ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

/etc/ppp/kermit.dial это скрипт kermit, который дозванивается до удаленного хоста и проходит необходимую авторизацию (пример такого скрипта находится в конце этого раздела).

Используйте следующий скрипт /etc/ppp/pppdown для отключения от PPP линии:

```
#!/bin/sh
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill -TERM ${pid}
fi

ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

Проверьте, запущен ли еще pppd, выполнив /usr/etc/ppp/ppptest, который выглядит примерно так:

```
#!/bin/sh
pid=`ps ax| grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
        echo 'pppd running: PID=' ${pid-NONE}
else
        echo 'No pppd running.'
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

Для обрыва соединения, выполните /etc/ppp/kermit.hup, который должен содержать:

```
set line /dev/tty01 ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
echo \13
exit
```

Существует альтернативный метод, использующий `chat` вместо `kermit`:

Для установления соединения `pppd` достаточно двух файлов.

/etc/ppp/options:

```
/dev/cuaa1 115200

crtscts     # enable hardware flow control
modem       # modem control line
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault # remote PPP serve must supply your IP address
            # if the remote host doesn't send your IP during
                # IPCP negotiation, remove this option
passive         # wait for LCP packets
domain <your.domain>  # put your domain name here

:       # put the IP of remote PPP host here
            # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to <local_ip>:<remote_ip>

defaultroute    # put this if you want that PPP server will be
            # your default router
```

/etc/ppp/login.chat.script:

Все это может быть расположено на одной строке.

```
ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDT<phone.number>
  CONNECT "" TIMEOUT 10 ogin:-\\r-ogin: <login-id>
  TIMEOUT 5 sword: <password>
```

Как только эти файлы будут созданы и отредактированы, необходимо только запустить pppd, вот так:

```
# pppd
```

22.3.3. Использование pppd в качестве сервера

/etc/ppp/options должен содержать примерно следующее:

```
crtscts                          # Hardware flow control
netmask 255.255.255.0            # netmask (not required)
192.114.208.20:192.114.208.165   # IP's of local and remote hosts
                                 # local ip must be different from one
                                 # you assigned to the Ethernet (or other)
                                 # interface on your machine.
                                 # remote IP is IP address that will be
                                 # assigned to the remote machine
domain ppp.foo.com               # your domain
passive                          # wait for LCP
modem                            # modem line
```

Следующий скрипт /etc/ppp/pppserv укажет pppd работать в качестве сервера:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete
```

```
# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200
```

Используйте этот скрипт /etc/ppp/pppservdown для остановки сервера:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans
```

Следующий kermit скрипт (/etc/ppp/kermit.ans) включит/отключит режим ответа модема на входящие звонки. Он должен выглядеть примерно так:

```
set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
inp 5 OK
echo \13
out ATS0=1\13   ; change this to out ATS0=0\13 if you want to disable
```

```
                ; autoanswer mode
inp 5 OK
echo \13
exit
```

Скрипт, называющийся /etc/ppp/kermit.dial, используется для дозвона и аутентификации на удаленном хосте. Вам потребуется приспособить его под собственные нужды. Поместите ваш логин и пароль в этот скрипт; вам также потребуется изменить операторы input в зависимости от ответов от модема и удаленного хоста.

```
;
; put the com line attached to the modem here:
;
set line /dev/tty01
;
; put the modem speed here:
;
set speed 19200
set file type binary            ; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto                ; Then SET CARRIER if necessary,
set dial display on             ; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
set input case ignore
def \%x 0                       ; login prompt counter
goto slhup

:slcmd                          ; put the modem in command mode
echo Put the modem in command mode.
clear                           ; Clear unread characters from input buffer
pause 1
output +++                      ; hayes escape sequence
input 1 OK\13\10                ; wait for OK
if success goto slhup
output \13
pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd              ; if modem doesn't answer OK, try again
```

```
:slhup                          ; hang up the phone
clear                           ; Clear unread characters from input buffer
pause 1
echo Hanging up the phone.
output ath0\13                  ; hayes command for on hook
input 2 OK\13\10
if fail goto slcmd              ; if no OK answer, put modem in command mode

:sldial                         ; dial the number
pause 1
echo Dialing.
output atdt9,550311\13\10               ; put phone number here
assign \%x 0                    ; zero the time counter

:look
clear                           ; Clear unread characters from input buffer
increment \%x                   ; Count the seconds
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 60 goto look
else goto slhup

:sllogin                        ; login
assign \%x 0                    ; zero the time counter
pause 1
echo Looking for login prompt.

:slloop
increment \%x                   ; Count the seconds
clear                           ; Clear unread characters from input buffer
output \13
;
; put your expected login prompt here:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 10 goto slloop         ; try 10 times to get a login prompt
else goto slhup                 ; hang up and start again if 10 failures
```

```
:sluid
;
; put your userid here:
;
output ppp-login\13
input 1 {Password: }
;
; put your password here:
;
output ppp-password\13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo \7No dialtone.  Check the telephone line!\7
exit 1

; local variables:
; mode: csh
; comment-start: "; "
; comment-start-skip: "; "
; end:
```

22.4. Решение проблем с соединениями PPP

Этот раздел охватывает несколько вопросов, которые могут возникнуть при использовании PPP через модемные соединения. Например, предположим, что вам потребовалось узнать, какое именно приглашение отображает система, до которой вы дозваниваетесь. Некоторые провайдеры выдают приглашение `ssword`, другие `password`; если `ppp` скрипт не обрабатывает такие приглашения, попытка входа завершится неудачно. Наиболее общий способ отладки соединений `ppp` это подключение вручную. Ниже дана пошаговая информация по подключению вручную.

22.4.1. Проверьте файлы устройств

Если вы пересобирали ядро, проверьте устройство sio. Если вы не перенастраивали ядро, нет причин для беспокойства. Просто проверьте вывод `dmesg` для модемного устройства следующей командой:

```
# dmesg | grep sio
```

Вы должны получить информацию о устройствах sio. Это COM порты, которые нам необходимы. Если ваш модем работает как стандартный последовательный порт, вы увидите его на sio1, или COM2. Если это так, вам не требуется пересобирать ядро, необходимо лишь создать последовательное устройство. Вы можете сделать это, зайдя в /dev и запустив скрипт MAKEDEV. Создайте последовательные устройства:

```
# sh MAKEDEV cuaa0 cuaa1 cuaa2 cuaa3
```

Если модем находится на sio1, или COM2 в DOS, модемным устройством будет /dev/cuaa1.

22.4.2. Подключение вручную

Подключение к интернет с контролированием `ppp` вручную это быстрый, легкий и действенный способ отладки соединения или получения информации о обслуживании провайдером клиентского соединения `ppp`. Запустите PPP из командной строки. В примерах в качестве имени хоста мы будем использовать *example*. Запустите `ppp`, введя команду `ppp`:

```
# ppp
```

Теперь программа `ppp` запущена.

```
ppp ON example> set device /dev/cuaa1
```

Мы указали модемное устройство, в данном случае cuaa1.

```
ppp ON example> set speed 115200
```

Мы установили скорость подключения к модему, в данном случае 115,200 кбит/с.

```
ppp ON example> enable dns
```

Сообщает `ppp` настроить разрешение имен, добавив строки в /etc/resolv.conf. Если `ppp` не может определить имя хоста, его можно настроить позже вручную.

```
ppp ON example> term
```

Переключение в "терминальный" режим для контроля модема вручную.

```
deflink: Entering terminal mode on /dev/cuaa1
type '~h' for help
```

```
at
OK
atdt123456789
```

Использование команды `at` для инициализации модема, а затем `atdt` и номера провайдера для начала дозвона.

```
CONNECT
```

Подтверждение соединения, если есть проблемы с соединением, не вызванные проблемами в оборудовании, здесь мы попытаемся решить их.

```
ISP Login:myusername
```

Здесь вам предлагается ввести имя пользователя в приглашение, выводимое сервером провайдера.

```
ISP Pass:mypassword
```

Здесь предлагается ввести пароль, предоставленный провайдером. Как и при входе в FreeBSD, пароль не отображается.

```
Shell or PPP:ppp
```

В зависимости от вашего провайдера, это приглашение может не появиться. Здесь задается вопрос, хотите ли вы использовать оболочку на компьютере провайдера или запустить `ppp`. В этом примере мы выбрали `ppp`, поскольку хотим соединиться с интернет.

```
Ppp ON example>
```

Обратите внимание, что в этом примере первая буква `p` появилась в верхнем регистре. Это означает, что мы успешно подключились к провайдеру.

```
PPp ON example>
```

Мы успешно аутентифицировались у провайдера и ожидаем присвоения IP адреса.

```
PPP ON example>
```

Мы завершили согласование IP адресов, соединение успешно установлено.

```
PPP ON example>add default HISADDR
```

Это маршрут по умолчанию, его необходимо настроить, чтобы сделать возможной связь с внешним миром, поскольку на этот момент установлена связь только с удаленной стороной. Если маршрут не устанавливается из-за уже существующего маршрута, добавьте символ `!` перед `add`. Альтернативно вы можете сделать настройку перед установкой соединения.

Если все прошло удачно, на данный момент должно работать соединение с интернет, которое можно поместить в фоновый режим клавишами `CTRL` + `z`. Переход строки `PPP` в приглашении в нижний регистр (`ppp`) означает, что соединение было разорвано. Таким образом, символы P в верхнем регистре означают наличие соединения, а в нижнем - потерю соединения.

22.4.2.1. Отладка

Если у вас выделенная линия и нет необходимости устанавливать соединение, выключите контроль передачи данных CTS/RTS командой `set ctsrts off`. Это обычно необходимо при подключении к поддерживающим PPP терминальным серверам, когда PPP прерывается при попытке записать данные в ваше соединение, и ожидает сигнала CTS, или Clear To Send, который не появляется. Если вы используете этот параметр, используйте также параметр `set accmap`, который может быть необходим для устранения зависимости от оборудования путем пропускания определенных символов через соединение, обычно XON/XOFF. Обратитесь к странице справочника ppp(8) за более подробной информацией по этому параметру и его использованию.

Если у вас старый модем, может потребоваться использование `set parity even`. Проверка четности по умолчанию отключена, но она используется для для устранения ошибок (с серьезным увеличением объема трафика) на старых модемах и у некоторых провайдеров.

PPP может не вернуться в командный режим, что обычно вызвано ошибкой согласования, когда провайдер ждет от вас начала процесса согласования. В этот момент использование команды `~p` заставит ppp начать отправку настроечной информации.

Если вы не получаете приглашения на вход, скорее всего вам требуется использование аутентификации PAP или CHAP вместо UNIX® стиля как в примерах выше. Для использования PAP или CHAP просто добавьте следующие параметры PPP перед переходом в терминальный режим:

```
ppp ON example> set authname myusername
```

Необходимо заменить *myusername* на имя пользователя, выданное провайдером.

```
ppp ON example> set authkey mypassword
```

Где *mypassword* должен быть заменен на пароль, выданный провайдером.

Если вы подключаетесь, не можете определить ни одно доменное имя, попробуйте использовать ping(8) с каким-либо IP адресом и просмотрите выводимую информацию. Если 100 процентов (100%) пакетов теряются, скорее всего не назначен маршрут по умолчанию. Дважды проверьте, что параметр `add default HISADDR` установлен во время соединения. Если вы можете подключиться к удаленному IP адресу, возможно, что адрес сервера имен не был добавлен в /etc/resolv.conf. Этот файл должен выглядеть примерно так:

```
domain example.com
```

```
nameserver x.x.x.x
nameserver y.y.y.y
```

Где *x.x.x.x* и *y.y.y.y* должны быть заменены на IP адреса DNS серверов провайдера. Эта информация может предоставляться провайдером, вы можете выяснить это в службе поддержки.

Вы можете также настроить syslog(3) для протоколирования PPP соединения. Просто добавьте:

```
!ppp
*.*     /var/log/ppp.log
```

в файл /etc/syslog.conf. В большинстве случаев эти строки уже присутствуют.

22.5. Использование PPP через Ethernet (PPPoE)

Этот раздел описывает настройку PPP через Ethernet (PPP over Ethernet, PPPoE).

22.5.1. Настройка ядра

Для PPPoE более не требуется настройка ядра. Если необходимая поддержка netgraph не встроена в ядро, она будет динамически загружена ppp.

22.5.2. Настройка ppp.conf

Вот пример работающего ppp.conf:

```
default:
  set log Phase tun command # you can add more detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
  set device PPPoE:xl1 # replace xl1 with your Ethernet device
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set dial
  set login
  add default HISADDR
```

22.5.3. Запуск ppp

Под пользователем `root` вы можете запустить:

```
# ppp -ddial name_of_service_provider
```

22.5.4. Запуск ppp при загрузке

Добавьте к /etc/rc.conf следующее:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES"   # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

22.5.5. Использование тега сервиса PPPoE

Иногда для установки соединения необходимо использовать тег сервиса. Теги сервисов используются для различения PPPoE серверов, подключенных к одной сети.

В документации, предоставляемой провайдером, должна находиться необходимая информация о теге сервиса. Если вы не можете ее обнаружить, обратитесь в службу поддержки.

В крайнем случае, вы можете попробовать метод, предложенный в программе Roaring Penguin PPPoE, которая находится в коллекции портов. Учтите, что этот метод может сделать ваш модем неработоспособным, так что дважды подумайте перед тем, как воспользоваться им. Просто установите программу, поставляемую с модемом вашим провайдером. Затем, войдите в меню **System** программы. Имя вашего профиля должно находиться здесь. Обычно это *ISP*.

Имя профиля (тег сервиса) будет использоваться в записи настройки PPPoE в файле ppp.conf (часть команды `set device`, относящаяся к провайдеру). Обратитесь к странице справочника ppp(8) за более подробной информацией. Эта команда должна выглядеть примерно так:

```
set device PPPoE:xl1:ISP
```

Не забудьте изменить *xl1* на имя устройства вашей Ethernet карты.

Не забудьте изменить *ISP* на профиль, определенный вами ранее.

Дополнительная информация:

- Cheaper Broadband with FreeBSD on DSL, опубликовал Renaud Waldura.
- Nutzung von T-DSL und T-Online mit FreeBSD by Udo Erdelhoff (на немецком).

22.5.6. PPPoE с 3Com® HomeConnect™ ADSL Modem Dual Link

Этот модем не поддерживает RFC 2516 (*Метод соединения PPP через Ethernet (PPPoE)*, написанный L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, и R. Wheeler). Вместо этого, для фреймов Ethernet используются различные коды типов пакетов. Сообщите 3Com если считаете, что они должны соблюдать спецификации PPPoE.

Для включения поддержки этого нестандартного устройства, в FreeBSD необходимо

установить переменную sysctl. Это может быть сделано автоматически, поместите в /etc/sysctl.conf следующую переменную:

```
net.graph.nonstandard_pppoe=1
```

или, для непосредственного включения, выполните команду `sysctl net.graph.nonstandard_pppoe=1`.

К сожалению, поскольку эта настройка влияет на всю систему, невозможно одновременно взаимодействовать с нормальным PPPoE клиентом или сервером и 3Com® HomeConnect™ ADSL Modem.

22.6. Использование PPP через ATM (PPPoA)

Далее описано как настроить PPP через ATM (PPP over ATM, PPPoA). PPPoA популярен у европейских провайдеров DSL.

22.6.1. Использование PPPoA с Alcatel SpeedTouch™ USB

Поддержка PPPoA для этого устройства в FreeBSD поставляется в виде порта, поскольку встроенное программное обеспечение поставляется под лицензионным соглашением Alcatel и не может свободно распространяться с основной системой FreeBSD.

Для установки этой программы, просто используйте коллекцию портов. Установите порт net/pppoa и следуйте инструкциям.

Как и многие устройства USB, Alcatel SpeedTouch™ USB должен загрузить встроенное программное обеспечение с компьютера. Возможна автоматизация этого процесса в FreeBSD, чтобы эта передача происходила при подключении устройства к порту USB. Для включения этой автоматической передачи необходимо добавить в /etc/usbd.conf нижеприведенную информацию. Этот файл необходимо отредактировать под пользователем `root`.

```
device "Alcatel SpeedTouch USB"
    devname "ugen[0-9]+"
    vendor 0x06b9
    product 0x4061
    attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"
```

Для включения даемона USB, usbd, поместите в /etc/rc.conf следующую строку:

```
usbd_enable="YES"
```

Возможна также настройка ppp для дозвона при запуске. Для этого добавьте в /etc/rc.conf нижеприведенные строки. Опять же, для этого вам необходимо войти под пользователем `root`.

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_profile="adsl"
```

Необходимо также использовать пример файла ppp.conf, поставляемый с портом net/pppoa.

22.6.2. Использование mpd

Вы можете использовать mpd для подключения к различным сервисам, в частности к сервисам PPTP. Вы можете найти mpd в коллекции портов, net/mpd. Многие ADSL модемы требуют, чтобы PPTP туннель был создан между модемом и компьютером, один из таких модемов это Alcatel SpeedTouch™ Home.

Сначала установите порт, затем настройте mpd под собственные нужды и настройки провайдера. Порт помещает набор примеров настройки в каталог PREFIX/etc/mpd/. *PREFIX* означает каталог, в который устанавливаются порты, по умолчанию это /usr/local/. Полное руководство по настройке mpd доступно в HTML формате после установки порта. Оно находится в PREFIX/shared/doc/mpd/. Ниже находится пример настройки mpd для соединения с ADSL сервисом. Настройка разделена на два файла, первый это mpd.conf:

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname username ①
    set bundle password password ②
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

    open
```

① Имя пользователя, используемое для аутентификации у провайдера.

② Пароль, используемый для аутентификации у провайдера.

Файл mpd.links содержит информацию о соединении или соединениях, которые вы

планируете установить. Пример mpd.links, соответствующий приведенному выше примеру:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ①
    set pptp peer 10.0.0.138 ②
```

① IP адрес компьютера FreeBSD, с которого вы будете использовать mpd.

② IP адрес модема ADSL. Для Alcatel SpeedTouch™ Home этот адрес по умолчанию `10.0.0.138`.

Инициализация соединения возможно простым выполнением следующей команды под `root`:

```
# mpd -b adsl
```

Вы можете просмотреть статус соединения с помощью следующей команды:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
     inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Использование mpd это рекомендуемый способ подключения к сервису ADSL из FreeBSD.

22.6.3. Использование pptpclient

Возможно также использование FreeBSD для подключения к сервисам PPPoA с помощью net/pptpclient.

Для подключения к сервису DSL с использованием net/pptpclient, установите порт или пакет и отредактируйте /etc/ppp/ppp.conf. Вам потребуется работать под `root`. Пример настройки ppp.conf дан ниже. За дальнейшей информацией по параметрам ppp.conf обратитесь к странице справочника ppp, ppp(8).

```
adsl:
 set log phase chat lcp ipcp ccp tun command
 set timeout 0
 enable dns
 set authname username ①
 set authkey password ②
 set ifaddr 0 0
 add default HISADDR
```

① Имя пользователя вашей учетной записи у провайдера DSL.

② Пароль для вашей учетной записи.

Поскольку вам необходимо поместить пароль в незашифрованном виде в файл ppp.conf, убедитесь что никто другой не сможет прочесть содержимое этого файла. Следующая последовательность команд сделает этот файл доступным для чтения только пользователю `root`. Обратитесь к страницам справочника chmod(1) и chown(8) за дальнейшей информацией.

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Следующая команда откроет туннель для PPP сессии к вашему DSL маршрутизатору. Модемы Ethernet DSL поставляются с настроенным IP адресом локальной сети, к которому вы подключаетесь. У Alcatel SpeedTouch™ Home этот адрес `10.0.0.138`. В документации на ваш маршрутизатор должно быть указано, какой адрес используется. Для открытия туннеля и начала PPP сессии выполните:

```
# pptp address adsl
```

Чтобы вернуться в приглашение командной строки после выполнения этой команды, добавьте символ "&" в конец строки.

Устройство виртуального туннеля tun будет создано для взаимодействия между процессами pptp и ppp. Как только вы вернетесь в приглашение, или процесс pptp выполнит соединение, вы можете проверить туннель примерно такой командой:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffff00
        Opened by PID 918
```

Если вы не сможете соединиться, проверьте настройку маршрутизатора, которая обычно доступна через telnet или через веб браузер. Если вы все еще не можете подключиться, проверьте вывод команды `pptp` и содержимое лог файла ppp, /var/log/ppp.log.

22.7. Использование SLIP

22.7.1. Настройка SLIP клиента

Ниже дан один из способов настройки FreeBSD для подключения к SLIP сети со статическим адресом. Для динамического подключения (адрес изменяется при каждом дозвоне) возможно потребуется более сложная настройка.

Сначала определите, к какому последовательному порту подключен модем. Многие создают символическую ссылку, такую как /dev/modem, на настоящий файл устройства, /dev/cuaaN.

Это позволяет абстрагироваться от имени файла устройства, например если вы переносите модем на другой порт. Довольно сложно править множество файлов в /etc и .kermrc во всей системе!

/dev/cuaa0 это COM1, cuaa1 это COM2, и т.д.

Убедитесь, что в вашем файле настройки ядра присутствует строка:

```
pseudo-device   sl      1
```

В FreeBSD 5.X, используйте вместо этой строки следующую:

```
device   sl
```

Эта строка включена в ядро GENERIC, так что если вы ее не удаляли, проблем быть не должно.

22.7.1.1. То, что необходимо сделать только один раз

1. Добавьте ваш компьютер, шлюз и сервера имен в файл /etc/hosts. Вот пример такого файла:

   ```
   127.0.0.1          localhost loghost
   136.152.64.181          water.CS.Example.EDU water.CS water
   136.152.64.1            inr-3.CS.Example.EDU inr-3 slip-gateway
   128.32.136.9            ns1.Example.EDU ns1
   128.32.136.12           ns2.Example.EDU ns2
   ```

2. Убедитесь, что в файле /etc/host.conf`hosts` находится перед `bind` (для FreeBSD версий до 5.0). Начиная с FreeBSD 5.0, система использует файл /etc/nsswitch.conf, убедитесь, что параметр `files` находится перед `dns` в строке `hosts` этого файла. Без этого параметра могут происходить странные вещи.

3. Отредактируйте файл /etc/rc.conf.

   a. Установите имя хоста, настроив переменную hostname:

      ```
      hostname="myname.my.domain"
      ```

      Здесь необходимо использовать полное доменное имя вашего компьютера в интернет.

   b. Добавьте sl0 к списку сетевых интерфейсов, изменив переменную:

```
network_interfaces="lo0"
```

на:

```
network_interfaces="lo0 sl0"
```

c. Измените параметры sl0, добавив строку:

```
ifconfig_sl0="inet ${hostname} slip-gateway netmask 0xffffff00 up"
```

d. Назначьте маршрутизатор по умолчанию, изменив строку:

```
defaultrouter="NO"
```

на:

```
defaultrouter="slip-gateway"
```

4. Создайте файл /etc/resolv.conf, содержащий:

```
domain CS.Example.EDU
nameserver 128.32.136.9
nameserver 128.32.136.12
```

Как вы видите, здесь указаны адреса серверов имен. Конечно, реальные имена доменов и адреса для вас будут другими.

5. Перегрузите компьютер и убедитесь, что его имя хоста настроено правильно.

22.7.1.2. Создание SLIP соединения

1. Дозвонитесь на удаленный сервер, введите `slip` в приглашение, имя своего компьютера и пароль. Все, что требуется ввести в вашем случае. Если вы используете kermit, попробуйте такой скрипт:

```
# kermit setup
set modem hayes
set line /dev/modem
set speed 115200
set parity none
set flow rts/cts
```

```
set terminal bytesize 8
set file type binary
# The next macro will dial up and login
define slip dial 643-9600, input 10 =>, if failure stop, -
output slip\x0d, input 10 Username:, if failure stop, -
output silvia\x0d, input 10 Password:, if failure stop, -
output ***\x0d, echo \x0aCONNECTED\x0a
```

Конечно, вам потребуется заменить имя хоста и пароль на ваши собственные. После этого, для подключения просто введите `slip` из приглашения kermit.

Хранение пароля в любом месте файловой системы в незашифрованном виде это обычно плохая идея. Вы делаете это на свой риск.

2. Выйдите из kermit (вы можете приостановить его, нажав `Ctrl` + `z`) и введите под `root`:

```
# slattach -h -c -s 115200 /dev/modem
```

Если вы сможете выполнить `ping` для хостов по другую сторону маршрутизатора, вы подключились! Если это не работает, попробуйте параметр `slattach -a` вместо `-c`.

22.7.1.3. Как прервать соединение:

Сделайте следующее:

```
# kill -INT `cat /var/run/slattach.modem.pid`
```

для остановки `slattach`. Помните, что вы должны работать под `root` для выполнения этой команды. Затем вернитесь в kermit (запустив `fg`, если он приостановлен) и выйдите из него (`q`).

Страница справочника `slattach` сообщает, что для отключения интерфейса необходимо использовать `ifconfig sl0 down`, но это похоже не играет никакой роли. (`ifconfig sl0` сообщает о том же.)

Иногда модем может не сбросить соединение (это бывает довольно часто). В этом случае просто запустите kermit и выйдите из него еще раз. При второй попытке соединение обычно разрывается.

22.7.1.4. Решение проблем

Вот наиболее часто встречающиеся ситуации:

- Не используются параметры `slattach`-c` или `-a` (это может быть не фатально, но иногда вызывает проблемы.)

- Используется `s10` вместо `sl0` (с некоторыми шрифтами сложно увидеть разницу).
- Попробуйте использовать `ifconfig sl0` для просмотра статуса интерфейса. Например, вы можете получить такую информацию:

```
# ifconfig sl0
sl0: flags=10<POINTOPOINT>
        inet 136.152.64.181 --> 136.152.64.1 netmask ffffff00
```

- Если вы получите сообщение `no route to host` от команды ping, возможно это проблема с таблицей маршрутизации. Используйте команду `netstat -r` для отображения существующих маршрутов:

```
# netstat -r
Routing tables
Destination      Gateway            Flags     Refs     Use  IfaceMTU    Rtt
Netmasks:

(root node)
(root node)

Route Tree for Protocol Family inet:
(root node) =>
default          inr-3.Example.EDU  UG          8   224515  sl0 -      -
localhost.Exampl localhost.Example. UH          5    42127  lo0 -       0.438
inr-3.Example.ED water.CS.Example.E UH          1        0  sl0 -      -
water.CS.Example localhost.Example. UGH        34 47641234  lo0 -       0.438
(root node)
```

Предыдущий пример получен на относительно загруженной системе. Числа в вашей системе будут сильно зависеть от загрузки сети.

22.7.2. Настройка SLIP сервера

Этот документ предоставляет решение для настройки SLIP сервера в системе FreeBSD, что обычно означает настройку системы для автоматического запуска соединений при удаленном входе SLIP клиентов.

22.7.2.1. Предварительные требования

Информация в этом разделе чисто техническая, поэтому требуются некоторые предварительные знания. Предполагается, что вы знакомы с сетевым протоколом TCP/IP, и в частности, с адресацией сетей и хостов, сетевыми масками, делением на подсети, маршрутизацией и протоколами маршрутизации, такими как RIP. Настройка SLIP сервисов на сервере удаленного доступа требует знания этих концепций, и если вы не знакомы с ними, прочтите или книгу *TCP/IP Network Administration* от Craig Hunt, опубликованную O'Reilly & Associates, Inc. (ISBN Number 0-937175-82-X), или книги Douglas Comer по протоколу TCP/IP.

В дальнейшем предполагается, что вы уже настроили ваш модем (модемы) и настроили соответствующие системные файлы для разрешения входа через них. Если вы еще не подготовили систему соответствующим образом, обратитесь к руководству по настройке сервисов удаленного входа; просмотрите список руководств на http://www.FreeBSD.org/ru/docs/. Вы можете также обратиться к странице справочника sio(4) за информацией о драйвере последовательного порта и к страницам gettytab(5), getty(8) и init(8) за информацией по настройке системы для удаленного входа в систему через модемы, и возможно stty(1) за информацией о настройке параметров последовательных портов (таких как `clocal` для подключаемых непосредственно последовательных интерфейсов).

22.7.2.2. Краткий обзор

В типичной конфигурации FreeBSD работает в качестве SLIP сервера так: пользователь SLIP дозванивается на FreeBSD SLIP сервер и входит в систему со специальным SLIP логином, использующим /usr/sbin/sliplogin в качестве оболочки. Программа `sliplogin` просматривает файл /etc/sliphome/slip.hosts на предмет строки, соответствующей специальному пользователю, и если находит совпадение, подключает последовательную линию к доступному SLIP интерфейсу, а затем запускает shell скрипт /etc/sliphome/slip.login для настройки SLIP интерфейса.

22.7.2.2.1. Пример входа на SLIP сервер

Например, идентификатор пользователя на SLIP сервере `Shelmerg`. Соответствующая запись в /etc/master.passwd будет выглядеть примерно так:

```
Shelmerg:password:1964:89::0:0:Guy Helmer -
SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

Когда `Shelmerg` входит в систему, `sliplogin` ищет строку в /etc/sliphome/slip.hosts, в которой находится соответствующий идентификатор пользователя; например, строка может быть такой:

```
Shelmerg    dc-slip sl-helmer       0xfffffc00        autocomp
```

После обнаружения этой строки `sliplogin` подключает последовательную линию к следующему доступному SLIP интерфейсу, а затем выполняет /etc/sliphome/slip.login примерно так:

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xfffffc00 autocomp
```

Если все проходит нормально, /etc/sliphome/slip.login вызовет `ifconfig` для SLIP интерфейса, к которому подключилась программа `sliplogin` (slip интерфейс 0 в примере выше, первый параметр в списке, задаваемом slip.login) для установки локального IP адреса ((`dc-slip`), удаленного IP адреса (`sl-helmer`), сетевой маски для SLIP интерфейса (`0xfffffc00`), и любых дополнительных флагов (`autocomp`). Если что-то идет не так, `sliplogin` обычно протоколирует

соответствующие сообщения в через уровень `daemon` syslog; эти сообщения как правило попадают в /var/log/messages (обратитесь к страницам справочника syslogd(8) и syslog.conf(5), а также проверьте файл /etc/syslog.conf, чтобы выяснить, что протоколирует `syslogd` и куда помещается информация).

Достаточно примеров - давайте начнем настройку системы.

22.7.2.3. Настройка ядра

Стандартное ядро FreeBSD обычно поставляется с двумя SLIP интерфейсами ((sl0 и sl1); вы можете использовать команду `netstat -i`, чтобы выяснить, определены ли эти интерфейсы в вашем ядре.

Пример вывода `netstat -i`:

```
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
ed0   1500  <Link>0.0.c0.2c.5f.4a         291311     0   174209     0   133
ed0   1500  138.247.224 ivory             291311     0   174209     0   133
lo0   65535 <Link>                           79     0       79     0     0
lo0   65535 loop        localhost            79     0       79     0     0
sl0*  296   <Link>                            0     0        0     0     0
sl1*  296   <Link>                            0     0        0     0     0
```

Наличие в выводе `netstat -i` интерфейсов sl0 и sl1 означает, что SLIP интерфейсы встроены в ядро (символ * показывает неактивность интерфейсов).

Ядро FreeBSD по умолчанию не пересылает пакеты между интерфейсами (компьютер FreeBSD не работает как маршрутизатор), вследствие требований RFC (см. RFCs 1009 [Requirements for Internet Gateways], 1122 [Requirements for Internet Hosts - Communication Layers], и возможно 1127 [A Perspective on the Host Requirements RFCs]). Если вы хотите, чтобы FreeBSD SLIP работал в качестве маршрутизатора, отредактируйте файл /etc/rc.conf и присвойте переменной `gateway_enable` значение `YES`.

Для вступления изменений в силу потребуется перезагрузка.

В файле настройки стандартного ядра (/sys/i386/conf/GENERIC) находится строка:

```
pseudo-device sl 2
```

Она определяет число доступных устройств SLIP в ядре; Число в конце строки определяет максимально возможное количество одновременных SLIP соединений.

Обратитесь к Настройка ядра FreeBSD за информацией по настройке ядра FreeBSD.

22.7.2.4. Настройка sliplogin

Как упоминалось ранее, в каталоге /etc/sliphome находятся три файла, являющиеся частью настройки для /usr/sbin/sliplogin (для `sliplogin` существует страница справочника,

sliplogin(8)): slip.hosts, определяющий список пользователей SLIP и связанные с ними IP адреса; slip.login, который обычно всего лишь настраивает SLIP интерфейс; slip.logout, который восстанавливает состояние системы до запуска slip.login после завершения последовательного соединения.

22.7.2.4.1. Настройка slip.hosts

/etc/sliphome/slip.hosts содержит строки, в которых находится как минимум четыре параметра, разделенных пробелами:

- ID пользователя SLIP
- Локальный адрес (локальный для SLIP сервера) SLIP соединения
- Удаленный адрес SLIP соединения
- Сетевая маска

Локальные и удаленные адреса могут быть именами хостов (разрешаемыми в IP адреса через файл /etc/hosts или через службу доменных имен, в зависимости от настроек в файле /etc/nsswitch.conf для FreeBSD 5.X, или /etc/host.conf для FreeBSD 4.X), а сетевая маска может быть именем, разрешаемым через файл /etc/networks. В системе, используемой в качестве примера, файл /etc/sliphome/slip.hosts выглядит так:

```
#
# login local-addr      remote-addr     mask            opt1    opt2
#                                               (normal,compress,noicmp)
#
Shelmerg  dc-slip       sl-helmerg      0xfffffc00      autocomp
```

В конце строки находятся один или более параметров.

- `normal` - нет сжатия заголовков
- `compress` - сжимать заголовки
- `autocomp` - сжимать заголовки, если удаленная сторона это позволяет
- `noicmp` - запретить ICMP пакеты (любые "ping" пакеты будут отброшены и не станут помехой для другого трафика)

Выбор локального и удаленного адреса для SLIP соединений зависит от того, используете ли вы выделенную TCP/IP сеть, или используете на SLIP сервере "ARP прокси". (это не "настоящий" ARP прокси, но данная терминология используется в этом разделе). Если вы не уверены, какой метод выбрать, или как присвоить IP адреса, обратитесь к книгам по TCP/IP, упомянутым выше (Предварительные требования).

Если вы собираетесь использовать отдельную подсеть для SLIP клиентов, потребуется выделить адреса за пределом адресов вашей сети и присвоить каждому SLIP клиенту IP адрес из данной подсети. Затем вам возможно потребуется настроить статический маршрут в используемую для SLIP подсеть через SLIP сервер на ближайшем IP маршрутизаторе.

Иначе, если вы будете использовать метод "proxy ARP", потребуется присвоить SLIP

клиентам IP адреса, не входящие в Ethernet подсеть сервера SLIP, а также настроить скрипты /etc/sliphome/slip.login и /etc/sliphome/slip.logout, чтобы использовать arp(8) для управления записями ARP прокси в таблице ARP сервера SLIP.

22.7.2.4.2. Настройка slip.login

Типичный файл /etc/sliphome/slip.login выглядит примерно так:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

Этот файл slip.login всего лишь запускает `ifconfig` для соответствующего SLIP интерфейса с заданными локальным и удаленным адресом и сетевой маской.

Если вы решили использовать метод "ARP прокси" (вместо использования отдельной подсети для SLIP клиентов), ваш файл /etc/sliphome/slip.login должен выглядеть примерно так:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Answer ARP requests for the SLIP client with our Ethernet addr
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

Дополнительная строка в этом slip.login, `arp -s $5 00:11:22:33:44:55 pub`, создает ARP запись в ARP таблице SLIP сервера. При соединении другого узла в Ethernet с IP адресом SLIP клиента, SLIP сервер выдает ответ с собственным Ethernet MAC адресом.

При использовании примера выше убедитесь, что заменили Ethernet MAC адрес (`00:11:22:33:44:55`) на MAC адрес Ethernet карты вашей системы, или ваш "ARP прокси" точно не будет работать! Вы можете определить Ethernet MAC адрес SLIP сервера, просмотрев

вывод команды `netstat -i` выше; информация об адресе находится второй строке:

```
ed0   1500  <Link>0.2.c1.28.5f.4a         191923    0   129457     0   116
```

Это означает, что в данной системе Ethernet MAC адрес `00:02:c1:28:5f:4a` - точки в MAC адресе, выдаваемые `netstat -i`, должны быть заменены на двоеточия, необходимо также добавить нуль в начало каждого односимвольного шестнадцатеричного номера для преобразования этого адреса в форму, пригодную для arp(8); обратитесь к странице справочника arp(8) за полной информацией по использованию.

При создании /etc/sliphome/slip.login и /etc/sliphome/slip.logout, должен быть установлен бит "выполнения" (`chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout`), или `sliplogin` не сможет их выполнить.

22.7.2.4.3. Настройка slip.logout

/etc/sliphome/slip.logout не является совершенно необходимым (если только вы не реализуете "ARP прокси"), но если вы решили создать его, воспользуйтесь следующим примером:

```
#!/bin/sh -
#
#       slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

Если вы используете "ARP прокси", потребуется удаление записи ARP для SLIP клиента через /etc/sliphome/slip.logout:

```
#!/bin/sh -
#
#       @(#)slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Quit answering ARP requests for the SLIP client
```

```
/usr/sbin/arp -d $5
```

Команда `arp -d $5` удаляет запись ARP, добавленную slip.login при входе SLIP клиента.

Повторяем: убедитесь, что на файл /etc/sliphome/slip.logout установлен бит выполнения (`chmod 755 /etc/sliphome/slip.logout`).

22.7.2.5. Соглашения о маршрутизации

Если вы не используете "ARP прокси" метод для маршрутизации пакетов между SLIP клиентами и остальной сетью (и возможно интернет), вам возможно потребуется статический маршрут (маршруты) до ближайшего шлюза (шлюзов) для маршрутизации подсети SLIP клиентов через SLIP сервер.

22.7.2.5.1. Статические маршруты

Добавление статических маршрутов может стать для кого-то проблемой (это даже невозможно, если у вас нет соответствующих прав). Если в вашей организации сеть с несколькими маршрутизаторами, некоторые маршрутизаторы, например Cisco и Proteon, требуют не только настройки статического маршрута в подсеть SLIP, но и указания, о каких статических маршрутах сообщать другим маршрутизаторам, так что для наладки работоспособности статической маршрутизации может потребоваться некоторое исследование и отладка.

22.7.2.5.2. Запуск GateD®

GateD® это закрытое программно обеспечение, более недоступное в исходных текстах (дополнительная информация находится на вебсайте GateD®). Этот раздел существует лишь в целях обратной совместимости для тех, кто все еще использует старую версию.

Альтернатива головной боли со статическими маршрутами это установка GateD® на FreeBSD SLIP сервере и настройка его для использования соответствующих протоколов маршрутизации (RIP/OSPF/BGP/EGP) для сообщения другим маршрутизаторам о вашей SLIP подсети. Вам потребуется создать /etc/gated.conf для настройки gated. Ниже дан пример:

```
#
# gated configuration file for dc.dsu.edu; for gated version 3.5alpha5
# Only broadcast RIP information for xxx.xxx.yy out the ed Ethernet interface
#
#
# tracing options
#
traceoptions "/var/tmp/gated.output" replace size 100k files 2 general ;

rip yes {
  interface sl noripout noripin ;
  interface ed ripin ripout version 1 ;
  traceoptions route ;
```

```
} ;

#
# Turn on a bunch of tracing info for the interface to the kernel:
kernel {
  traceoptions remnants request routes info interface ;
} ;

#
# Propagate the route to xxx.xxx.yy out the Ethernet interface via RIP
#

export proto rip interface ed {
  proto direct {
      xxx.xxx.yy mask 255.255.252.0 metric 1; # SLIP connections
  } ;
} ;

#
# Accept routes from RIP via ed Ethernet interfaces

import proto rip interface ed {
  all ;
} ;
```

В примере выше используется широковещательная рассылка информации о маршрутизации для подсети SLIP *xxx.xxx.yy* протоколом RIP на сеть Ethernet; если вы используете другой драйвер Ethernet вместо ed, потребуется соответственно изменить запись для ed. В этом примере отладочная информация переправляется в /var/tmp/gated.output; вы можете выключить отладку, если GateD® работает. Вам потребуется заменить *xxx.xxx.yy* в сетевом адресе на вашу подсеть SLIP (убедитесь, что изменение сетевой маски в `proto direct` работает нормально).

Как только вы установили и настроили GateD®, потребуется сообщить стартовым скриптам FreeBSD запускать его вместо routed. Простейший способ сделать это - установить переменные `router` и `router_flags` в /etc/rc.conf. Обратитесь к странице справочника GateD® за информацией о параметрах командной строки.

Глава 23. Электронная почта

23.1. Краткий обзор

"Электронная почта" называемая также email, является на сегодняшний день одним из самых популярных средств связи. Эта глава описывает основы работы с почтовым сервером в FreeBSD, а также введение в процесс отправки и получения почты в FreeBSD; однако, это не полноценный справочник и фактически в главу не вошло много важной информации. Более подробно эта тема рассмотрена во множестве прекрасных книг, список которых приведен в Библиография.

После прочтения этой главы вы узнаете:

- Какие программные компоненты задействованы в отправке и получении электронной почты.
- Какие основные файлы настройки sendmail имеются в FreeBSD.
- Разницу между удаленными и локальными почтовыми ящиками.
- Как запретить спамерам использовать ваш почтовый сервер для пересылки почты.
- Как установить и настроить альтернативный агент передачи почты (Mail Transfer Agent, MTA), заменив им sendmail.
- Как разрешить наиболее часто встречающиеся проблемы с почтовым сервером.
- Как настроить систему только для отправки почты.
- Как использовать почту с коммутируемым подключением к сети.
- Как настроить SMTP аутентификацию для дополнительной защиты.
- Как установить и настроить почтовый агент пользователя (Mail User Agent, MUA), например mutt, для отправки и получения почты.
- Как загрузить почту с удаленного POP или IMAP сервера.
- Как автоматически применять фильтры и правила к входящей почте.

Перед прочтением этой главы вам потребуется:

- Правильно настроить сетевое подключение (Сложные вопросы работы в сети).
- Правильно настроить DNS для почтового сервера (Сетевые серверы).
- Знать как устанавливать дополнительное программное обеспечение сторонних разработчиков (Установка приложений. порты и пакеты).

23.2. Использование электронной почты

В работе почтовой системы задействованы пять основных частей: пользовательский почтовый клиент (Mail User Agent, MUA), почтовый сервис (даемон) (Mail Transfer Agent, MTA), сервер DNS, удаленный или локальный почтовый ящик, и конечно сам почтовый сервер.

23.2.1. Пользовательский почтовый клиент

Обычно, это программа типа mutt, alpine, elm, mail, а также программы с графическим интерфейсом, такие, как balsa или xfmail, или интегрированные приложения (например, какой-либо WWW браузер типа Netscape). Все эти программы общаются с локальным почтовым сервером, вызывая какой-либо даемон, или напрямую по протоколу TCP.

23.2.2. Почтовый даемон

FreeBSD по умолчанию поставляется с sendmail, но помимо того поддерживает множество других даемонов почтового сервера, вот лишь некоторые из них:

- exim;
- postfix;
- qmail.

Почтовый даемон выполняет только две функции: он отвечает за прием входящей почты и отправку исходящей. Он *не* отвечает за выдачу почты по протоколам POP или IMAP, и не обеспечивает подключения к локальным почтовым ящикам mbox или Maildir. Для этих целей вам может потребоваться дополнительный даемон.

Старые версии sendmail содержат некоторые серьезные ошибки безопасности, которые могут привести к получению атакующим локального и/или удаленного доступа к вашему компьютеру. Убедитесь, что вы работаете с современной версией, свободной от таких ошибок. Или установите альтернативный MTA из Коллекции Портов FreeBSD.

23.2.3. Email и DNS

Служба имен доменов (Domain Name System, DNS) и соответствующий ей даемон `named` играют важную роль в доставке почты. Для доставки почты с вашего сайта другому, даемон почтового сервера обратится к DNS для определения удаленного хоста, отвечающего за доставку почты по назначению. Тот же процесс происходит при доставке почты с удаленного хоста на ваш почтовый сервер.

DNS отвечает за сопоставления имен хостов IP адресам, как и за хранение информации, предназначенной для доставки почты, известной как MX записи. Запись MX (Mail eXchanger) определяет хост или хосты, которые будут получать почту для определенного домена. Если для вашего имени хоста или домена нет записи MX, почта будет доставлена непосредственно на ваш хост, IP адрес которого определен в записи A.

Вы можете просмотреть MX записи для любого домена с помощью команды host(1), как показано в примере ниже:

```
% host -t mx FreeBSD.org
FreeBSD.org mail is handled (pri=10) by mx1.FreeBSD.org
```

23.2.4. Получение почты

Получение почты для вашего домена выполняет почтовый сервер. Он сохраняет отправленную в ваш домен почту в формате либо mbox (это метод по умолчанию), либо Maildir, в зависимости от настроек. После сохранения почты ее можно либо прочитать локально, используя такие приложения как mail(1), mutt, или удаленно, по таким протоколам как POP или IMAP. Это означает, что для локального чтения почты вам не потребуется устанавливать сервер POP или IMAP.

23.2.4.1. Доступ к удаленным почтовым ящикам по протоколам POP и IMAP

Для удаленного доступа к почтовым ящикам вам потребуется доступ к POP или IMAP серверу. Хотя удаленный доступ обеспечивают оба протокола POP и IMAP, последний предоставляет множество дополнительных возможностей, вот некоторые из них:

- IMAP может как хранить сообщения на удаленном сервере, так и забирать их.
- IMAP поддерживает одновременные обновления.
- IMAP может быть очень полезен для низкоскоростных соединений, поскольку позволяет пользователям получить структуру сообщений без их загрузки; он также может использоваться для выполнения таких задач как поиск на сервере, для минимизации объема передаваемых между клиентом и сервером данных.

Для установки POP или IMAP сервера необходимо выполнить следующие действия:

1. Выберите IMAP или POP сервер, который подходит вам наилучшим образом. Следующие POP и IMAP серверы хорошо известны и могут быть приведены в качестве примера:
   - qpopper;
   - teapop;
   - imap-uw;
   - courier-imap;
   - dovecot;
2. Установите POP или IMAP даемон, выбранный из Коллекции Портов.
3. Если потребуется, настройте /etc/inetd.conf для запуска POP или IMAP сервера.

Необходимо отметить, что и POP и IMAP серверы передают информацию, включая имя пользователя и пароль, в незашифрованном виде. Это означает, что если вы хотите защитить передачу информации по этим протоколам, потребуется использовать туннелирование сессий через ssh(1) или при помощи SSL. Туннелирование соединений описано в Туннелирование SSH, а SSL - в OpenSSL.

23.2.4.2. Доступ к локальным почтовым ящикам

Доступ к почтовым ящикам может быть осуществлен непосредственно путем использования MUA на сервере, где эти ящики расположены. Это можно сделать используя приложения вроде mutt или mail(1).

23.2.5. Почтовый хост

Почтовый хост это сервер, который отвечает за отправку и получение почты для вашего компьютера, и возможно, для всей вашей сети.

23.3. Настройка sendmail

В FreeBSD по умолчанию программой передачи почты (Mail Transfer Agent, MTA) является sendmail(8). Работа sendmail заключается в приеме почты от почтовых программ пользователей (Mail User Agents, MUA) и отправке ее на соответствующий адрес, в соответствии с имеющимися настройками. sendmail может также принимать входящие соединения по сети и доставлять почту в локальные почтовые ящики или перенаправлять их другой программе.

sendmail использует следующие файлы настройки:

Имя файла	Назначение
/etc/mail/access	Файл базы данных доступа sendmail
/etc/mail/aliases	Синонимы почтовых ящиков
/etc/mail/local-host-names	Список хостов, для которых sendmail принимает почту
/etc/mail/mailer.conf	Настройки почтовой программы
/etc/mail/mailertable	Таблица доставки почтовой программы
/etc/mail/sendmail.cf	Основной файл настройки sendmail
/etc/mail/virtusertable	Таблицы виртуальных пользователей и доменов

23.3.1. /etc/mail/access

База данных доступа определяет список хостов или IP адресов, имеющих доступ к локальному почтовому серверу, а также тип предоставляемого доступа. Хосты могут быть перечислены как `OK`, `REJECT`, `RELAY` или просто переданы процедуре обработки ошибок sendmail с заданным сообщением об ошибке. Хостам, перечисленным с параметром по умолчанию `OK`, разрешено отправлять почты на этот хост, если адрес назначения почты принадлежит локальной машине. Все почтовые соединения от хостов, перечисленных с параметром `REJECT`, отбрасываются. Для хостов, перечисленных с параметром `RELAY`, разрешена передача через этот сервер почты с любым адресом назначения.

Пример 35. Настройка базы данных доступа sendmail

```
cyberspammer.com        550 We do not accept mail from spammers
FREE.STEALTH.MAILER@        550 We do not accept mail from spammers
another.source.of.spam          REJECT
okay.cyberspammer.com           OK
128.32                          RELAY
```

В этом примере приведены пять записей. К отправителям, чей адрес соответствует записи в левой части таблицы, применяется правило записанное в правой части таблицы. В первых двух примерах код ошибки будет передан процедуре обработке ошибок sendmail. В этом случае на удаленном хосте будет получено соответствующее сообщение. В следующем примере почта отбрасывается почта от определенного хоста, `another.source.of.spam`. В четвертом примере разрешается прием почты от хоста `okay.cyberspammer.com`, имя которого более точно совпадает с этой записью, чем с `cyberspammer.com` в примере выше. При более точном совпадении правила перезаписываются. В последнем примере разрешается пересылка почты от хостов с IP адресами, начинающимися с `128.32`. Эти хосты смогут отправлять почту через этот почтовый сервер для других почтовых серверов.

После изменения этого файла для обновления базы данных вам потребуется запустить `make` в каталоге /etc/mail/.

23.3.2. /etc/mail/aliases

База данных синонимов содержит список виртуальных почтовых ящиков, принадлежащих другим пользователям, файлам, программам, или другим синонимам. Вот несколько примеров, которые могут быть использованы для /etc/mail/aliases:

Пример 36. Mail Aliases

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

Формат файла прост; имя почтового ящика слева от двоеточия сопоставляется назначению(ям) справа. В первом примере производится сопоставление почтового ящика `root` почтовому ящику `localuser`, для которого затем опять будет произведен поиск в базе данных синонимов. Если совпадений не обнаружится, сообщение будет доставлено локальному пользователю `localuser`. В следующем примере приведен список рассылки. Почта на адрес `ftp-bugs` рассылается на три локальных почтовых ящика: `joe`, `eric` и `paul`. Обратите внимание, что удалённый почтовый ящик может быть задан в виде user@example.com. В следующем примере показана запись почты в файл, в данном случае /dev/null. И в последнем примере показано отправление почты программе, в данном случае почтовое сообщение переправляется через канал UNIX® на стандартный вход

/usr/local/bin/procmail.

После обновления этого файла вам потребуется запустить `make` в каталоге /etc/mail/ для обновления базы данных.

23.3.3. /etc/mail/local-host-names

В этом файле находится список имен хостов, принимаемых программой sendmail(8) в качестве локальных. Поместите в этот файл любые домены или хосты, для которых sendmail должен принимать почту. Например, если этот почтовый сервер должен принимать почту для домена `example.com` и хоста `mail.example.com`, его файл local-host-names может выглядеть примерно так:

```
example.com
mail.example.com
```

После обновления этого файла необходимо перезапустить sendmail(8), чтобы он смог перечитать изменения.

23.3.4. /etc/mail/sendmail.cf

Основной файл настройки sendmail, sendmail.cf управляет общим поведением sendmail, включая все, от перезаписи почтовых адресов до отправки удаленным серверам сообщений об отказе от пересылки почты. Конечно, файл настройки с таким многообразием возможностей очень сложен и подробное его описание выходит за рамки данного раздела. К счастью, для стандартных почтовых серверов изменять этот файл придется не часто.

Основной файл настройки sendmail может быть собран из макроса m4(1), определяющего возможности и поведение sendmail. Подробнее этот процесс описан в файле /usr/src/contrib/sendmail/cf/README.

Для применения изменений после правки файла необходимо перезапустить sendmail.

23.3.5. /etc/mail/virtusertable

Файл virtusertable сопоставляет виртуальные почтовые домены и почтовые ящики реальным почтовым ящикам. Эти почтовые ящики могут быть локальными, удаленными, синонимами, определенными в /etc/mail/aliases, или файлами.

Пример 37. Пример таблицы виртуального домена

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

В примере выше мы видим сопоставление адресов для домена `example.com`. Почта

обрабатывается по первому совпадению с записью в этом файле. Первая запись сопоставляет адрес root@example.com локальному почтовому ящику `root`. Вторая запись сопоставляет postmaster@example.com локальному почтовому ящику `postmaster` на хосте `noc.example.net`. Наконец, до этого момента адрес в домене `example.com` не совпал ни с одним из предыдущих, будет применено последнее сопоставление, в которому соответствует всякое другое почтовое сообщение, отправленное на любой адрес в `example.com`. Это сообщение будет доставлено в локальный почтовый ящик `joe`.

23.4. Установка другой почтовой программы

Как уже упоминалось, FreeBSD поставляется с MTA (Mail Transfer Agent) sendmail. Следовательно, по умолчанию именно эта программа отвечает за вашу исходящую и входящую почту.

Однако, по различным причинам некоторые системные администраторы заменяют системный MTA. Эти причины варьируются от простого желания попробовать другой MTA до потребности в определенных возможностях пакета, основанного на другой почтовой программе. К счастью, вне зависимости от причины, в FreeBSD такая замена выполняется просто.

23.4.1. Установка нового MTA

Вам предоставлен широкий выбор MTA. Начните с поиска в Коллекции Портов FreeBSD, где их немало. Конечно, вы можете использовать любой MTA по желанию, взятый откуда угодно, если только сможете запустить его под FreeBSD.

Начните с установки нового MTA. После установки у вас будет возможность решить, действительно ли он подходит вашем нуждам, а также настроить новое программное обеспечение перед тем, как заменить им sendmail. При установке новой программы убедитесь, что она не пытается перезаписать системные файлы, такие как /usr/bin/sendmail. Иначе ваша новая почтовая программа фактически начнет работать до того, как вы ее настроите.

Обратитесь к документации на выбранный MTA за информацией по его настройке.

23.4.2. Отключение sendmail

> Если вы отключите сервис исходящей почты sendmail, необходимо заменить его альтернативной системой доставки почты. Если вы не сделаете этого, системные программы, такие как periodic(8), не смогут отправлять сообщения по электронной почте как обычно. Многие программы в вашей системе могут требовать наличия функционирующей sendmail-совместимой системы. Если приложения будут продолжать использовать программу sendmail для отправки почты после того, как вы её отключили, почта может попасть в неактивную очередь sendmail и никогда не будет доставлена.

Для полного отключения sendmail, включая сервис исходящей почты, используйте

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

в /etc/rc.conf.

Если вы хотите отключить только сервис входящей почты sendmail, установите

```
sendmail_enable="NO"
```

в /etc/rc.conf. Дополнительная информация о параметрах запуска sendmail доступна на странице справочника rc.sendmail(8).

23.4.3. Запуск нового MTA при загрузке

Новый MTA можно запускать автоматически при загрузке системы добавив соответствующую строку в /etc/rc.conf. Ниже приведен пример для postfix:

```
# echo 'postfix_enable=<<YES>>' >> /etc/rc.conf
```

С этого момента MTA будет запускаться автоматически во время загрузки системы.

23.4.4. Замещение sendmail как почтовой программы по умолчанию

Программа sendmail настолько распространена в качестве стандартной программы для систем UNIX®, что многие программы считают, что она уже установлена и настроена. По этой причине многие альтернативные MTA предоставляют собственные совместимые реализации интерфейса командной строки sendmail; это облегчает их использование в качестве "прозрачной" замены sendmail.

Поэтому если вы используете альтернативную почтовую программу, потребуется убедиться, что когда программное обеспечение пытается выполнить стандартные исполняемые файлы sendmail, такие как /usr/bin/sendmail, на самом деле выполняются программы вновь установленной почтовой системы. К счастью, FreeBSD предоставляет систему, называемую mailwrapper(8), которая выполняет эту работу за вас.

Когда установлен sendmail, файл /etc/mail/mailer.conf выглядит примерно так:

```
sendmail    /usr/libexec/sendmail/sendmail
send-mail   /usr/libexec/sendmail/sendmail
mailq       /usr/libexec/sendmail/sendmail
newaliases  /usr/libexec/sendmail/sendmail
hoststat    /usr/libexec/sendmail/sendmail
purgestat   /usr/libexec/sendmail/sendmail
```

Это означает, что когда выполняется какая-то из этих стандартных программ (например сам sendmail), система на самом деле вызывает копию mailwrapper, называемую sendmail, которая обращается к mailer.conf и выполняет вместо этого /usr/libexec/sendmail/sendmail. Такая схема делает простой замену программ, которые на самом деле выполняются, когда вызываются стандартные функции sendmail.

Поэтому если вы хотите выполнять /usr/local/supermailer/bin/sendmail-compat вместо sendmail, отредактируйте /etc/mail/mailer.conf так:

```
sendmail    /usr/local/supermailer/bin/sendmail-compat
send-mail   /usr/local/supermailer/bin/sendmail-compat
mailq       /usr/local/supermailer/bin/mailq-compat
newaliases  /usr/local/supermailer/bin/newaliases-compat
hoststat    /usr/local/supermailer/bin/hoststat-compat
purgestat   /usr/local/supermailer/bin/purgestat-compat
```

23.4.5. Запуск новой почтовой программы

Как только вы все настроили, потребуется или уничтожить процесс sendmail, который уже не нужен и запустить новую почтовую программу, или просто перегрузить систему. Перезагрузка также даст вам возможность проверить, правильно ли настроена система для автоматического запуска MTA при загрузке.

23.5. Поиск и устранение неисправностей

23.5.1. Почему я должен использовать FQDN для хостов вне моей подсети?

Вы, видимо, обнаружили, что хост, к которому вы обратились, оказался на самом деле в другом домене; например, если вы находитесь в домене `foo.bar.edu` и хотите обратиться к хосту `mumble` в домене `bar.edu`, то должны указать его полное доменное имя, `mumble.bar.edu`, а не просто `mumble`.

Традиционно, программа разрешения имен BSD BIND позволяла это делать. Однако, текущая версия BIND, поставляемая с FreeBSD, больше не добавляет имена доменов, отличающихся от того, в котором вы находитесь, для не полностью указанных имен хостов. То есть, имя `mumble` будет опознан как `mumble.foo.bar.edu` или будет искаться в корневом домене.

Это отличается от предыдущего поведения, при котором поиск продолжался в доменах `mumble.bar.edu` и `mumble.edu`. Если вам интересны причины объявления такого поведения плохой практикой и даже ошибкой в безопасности, обратитесь к RFC 1535.

Хорошим решением будет поместить строку

```
search foo.bar.edu bar.edu
```

вместо ранее используемой:

```
domain foo.bar.edu
```

в файл /etc/resolv.conf. Однако удостоверьтесь, что порядок поиска не нарушает "границ полномочий между локальным и внешним администрированием", в терминологии RFC 1535.

23.5.2. sendmail выдает ошибку mail loops back to myself

В FAQ по sendmail дан следующий ответ:

```
Я получаю такие сообщения об ошибке:

553 MX list for domain.net points back to relay.domain.net
554 <user@domain.net>... Local configuration error

Как можно решить эту проблему?

Согласно записям MX, почта для домена domain.net перенаправляется на хост
relay.domain.net, однако последний не распознается как
domain.net.  Добавьте domain.net в файл
/etc/mail/local-host-names
[известный как /etc/sendmail.cw до версии 8.10]
(если вы используете
FETURE(use_cw_file)) или добавьте Cw domain.net в файл
/etc/mail/sendmail.cf.
```

FAQ по sendmail можно найти на http://www.sendmail.org/faq/ и рекомендуется прочесть его при желании произвести некоторые "усовершенствования" настроек почтовой системы.

23.5.3. Как организовать работу почтового сервера при коммутируемом соединении с Интернет?

Вы хотите подключить к интернет компьютер с FreeBSD, работающий в локальной сети. Компьютер с FreeBSD будет почтовым шлюзом для локальной сети. PPP соединение не выделенное.

Существует как минимум два пути, чтобы сделать это. Один способ это использование UUCP.

Другой способ это использование постоянно работающего интернет сервера для обеспечения вторичного MX сервиса вашего домена. Например, домен вашей компании `example.com`, и провайдер интернет настроил `example.net` для обеспечения вторичного MX сервиса:

```
example.com.          MX        10      example.com.
```

```
                        MX        20      example.net.
```

Только один хост должен быть указан в качестве последнего получателя (добавьте запись `Cw example.com` в файл /etc/mail/sendmail.cf на машине `example.com`).

Когда программа `sendmail` (со стороны отправителя) "захочет" доставить почту, она попытается соединиться с вашим хостом (`example.com`) через модемное подключение. Скорее всего, ей это не удастся (вы, вероятнее всего, не будете подключены к интернет). Программа sendmail автоматически перейдет ко вторичному MX серверу, т.е. вашему провайдеру (`example.net`). Вторичный MX сервер будет периодически пытаться соединиться с вашим хостом и доставить почту на основной сервер MX (`example.com`).

Вы можете воспользоваться следующим сценарием, чтобы забирать почту каждый раз, когда вы входите в систему:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Если же вы хотите написать отдельный пользовательский скрипт, лучше воспользоваться командой `sendmail -qRexample.com` вместо вышеприведенного сценария, так как в этом случае вся почта в очереди для хоста `example.com` будет обработана немедленно.

Рассмотрим эту ситуацию подробнее:

Вот пример сообщения из freebsd-isp.

```
> Мы предоставляем вторичный MX для наших клиентов.  Вы соединяетесь
> с нашим сервером несколько раз в день, чтобы забрать почту для вашего
> первичного (главного) MX (мы не соединяемся с ним каждый раз, когда
> приходит новая почта для его доменов).  Далее, sendmail отправляет
> почту, находящуюся в очереди каждые 30 минут, и клиент должен быть
> подключен к Интернет в течении 30 минут, чтобы удостовериться, что
> вся почта ушла на основной MX-сервер.
>
> Может быть, есть какая-либо команда, которая заставит sendmail
> немедленно отправить все почту, находящуюся в очереди?  Естественно,
> пользователи не обладают какими-либо повышенными привилегиями на
> нашем сервере.

В разделе privacy flags файла
sendmail.cf, определяется опция
Opgoaway,restrictqrun

Уберите restrictqrun, чтобы разрешить рядовым
пользователям инициировать работу с очередью.  Вам также может понадобиться
изменить порядок MX-серверов.  Так, если вы предоставляете первый (основной)
```

```
MX-сервер для ваши пользователей, мы указываем:

# If we are the best MX for a host, try directly instead of generating
# local config error.
OwTrue

Таким образом, удаленный хост будет доставлять почту непосредственно к вам,
не пытаясь установить соединение с клиентом.  Затем уже вы, в свою очередь,
отсылаете ее клиенту.  Удостоверьтесь, что в DNS есть записи про
customer.com и hostname.customer.com.  Просто
добавьте запись A в DNS для customer.com.
```

23.5.4. Почему я продолжаю получать ошибки Relaying Denied при отправки почты через другие хосты?

В установке FreeBSD по умолчанию, sendmail настроен для отправки почты только от хоста, на котором он работает. Например, если доступен POP сервер, пользователи смогут проверять почту из школы, с работы или других удаленных точек, но не смогут отправлять письма. Обычно, через некоторое время после попытки будет отправлено письмо от MAILER-DAEMON с сообщением об ошибке `5.7 Relaying Denied`.

Есть несколько путей разрешения этой ситуации. Самый прямой путь это использование адреса вашего провайдера в файле relay-domains, расположенном в /etc/mail/relay-domains. Быстрый способ сделать это:

```
# echo "your.isp.example.com" > /etc/mail/relay-domains
```

После создания или редактирования этого файла вы должны перезапустить sendmail. Это отлично работает, если вы администратор сервера и не хотите отправлять почту локально, или хотите воспользоваться почтовым клиентом/системой на другом компьютере или даже через другого провайдера. Это также очень полезно, если у вас настроены одна или две почтовые записи. Если необходимо добавить несколько адресов, вы можете просто открыть этот файл в текстовом редакторе и добавить домены, по одному на строку:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

Теперь будет отправляться любая почта, посылаемая через вашу систему любым хостом из этого списка (предоставляемого пользователем, имеющим учетную запись в вашей системе). Это отличный способ разрешить пользователям отправлять почту через вашу систему удаленно, одновременно он блокирует отправку спама.

23.6. Расширенное руководство

В следующем разделе рассматриваются более сложные темы, такие как настройка почты и включение почтовой системы для всего домена.

23.6.1. Базовая конфигурация

Изначально, вы можете отправлять почту "во внешний мир" если правильно составлен файл /etc/resolv.conf или запущен свой сервер имен. Если вы хотите, чтобы почта, предназначенная для хоста в вашем домене, доставлялась MTA (например, sendmail) на вашем хосте FreeBSD, есть два пути:

- Запустите свой собственный сервер DNS, тем самым организовав собственный домен, например, `FreeBSD.org`
- Получайте почту для вашего хоста непосредственно. Это работает при доставке почты непосредственно на DNS имя вашей машины. Например, `example.FreeBSD.org`.

Независимо от выбранного из предложенных выше вариантов, для доставки почты непосредственно на ваш хост у него должен быть постоянный IP адрес (а не динамический, как у большинства PPP соединений). Если вы находитесь за брандмауэром, то последний должен пропускать SMTP-пакеты. Если вы хотите, чтобы почта приходила непосредственно на ваш хост, необходимо убедиться в одном из двух:

- Убедитесь, что запись (с наименьшим номером) MX в DNS соответствует IP адресу вашего хоста.
- Убедитесь, что в DNS для вашего хоста вообще отсутствует MX-запись.

Выполнение любого из перечисленных условий обеспечит доставку почты для вашего хоста.

Попробуйте это:

```
# hostname
example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

Если вы это видите, то можно без проблем посылать почту на fyourlogin@example.FreeBSD.org (предполагается, что sendmail на `example.FreeBSD.org` работает правильно).

Однако, если вы видите это:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

то вся почта, посланная на `example.FreeBSD.org` будет собираться на `hub` (для того же пользователя), вместо того, чтобы быть отосланной непосредственно на ваш хост.

Эта информация обрабатывается вашим DNS сервером. Соответствующая запись DNS, указывающая, через какой хост будет проходить ваша почта, называется MX (*M*ail e *X*changer). Если для хоста отсутствует такая запись, почта будет приходить прямо на этот хост.

Допустим, что запись MX для хоста `freefall.FreeBSD.org` в какой-то момент выглядела так:

```
freefall        MX  30  mail.crl.net
freefall        MX  40  agora.rdrop.com
freefall        MX  10  freefall.FreeBSD.org
freefall        MX  20  who.cdrom.com
```

Вы видите, что для хоста `freefall` существуют несколько MX-записей. Запись с наименьшим номером соответствует хосту, получающему почту непосредственно, если он доступен; если он недоступен по каким-то причинам, другие сервера (иногда называемые ("резервными MX") временно получают почту, и хранят ее пока не станут доступны хосты с меньшими номерами, в конечном итоге отправляя почту на эти хосты.

Чтобы альтернативные MX-хосты использовались наиболее эффективно, они должны быть независимо подключены к Интернет. Ваш провайдер (или дружественный сайт) скорее всего без проблем сможет оказать подобные услуги.

23.6.2. Почта для вашего домена

Для настройки "почтового хоста" (почтовый сервер) вам потребуется, чтобы почта, направляемая различным рабочим станциям, пересылалась этому хосту. Обычно вам необходима доставка всей почты для любого хоста вашего домена (в данном случае `*.FreeBSD.org`) на почтовый сервер, чтобы пользователи могли получать свою почту на с этого сервера.

Чтобы облегчить себе (и другим) жизнь, создайте на обеих машинах учетные записи с одинаковыми именами пользователей, например, с помощью команды adduser(8).

Сервер, который вы будете использовать в качестве почтового, должен быть объявлен таковым для каждой машины в домене. Вот фрагмент примерной конфигурации:

```
example.FreeBSD.org A   204.216.27.XX       ; Рабочая станция
            MX  10 hub.FreeBSD.org  ; Почтовый шлюз
```

Таким образом, вся корреспонденция, адресованная рабочей станции, будет обрабатываться вашим почтовым сервером, независимо от того, что указано в A-записи.

Все это можно реализовать только в том случае, если вы используете сервер DNS. Если вы по каким-либо причинам не имеете возможности установить свой собственный сервер имен, необходимо договориться с провайдером или теми, кто поддерживает ваш DNS.

Если вы хотите поддерживать несколько виртуальных почтовых серверов, может пригодиться следующая информация. Допустим, что ваш клиент зарезервировал домен, например, `customer1.org`, и вам требуется, чтобы почта, предназначенная для `customer1.org` приходила на ваш хост, например, `mail.myhost.com`. В таком случае, DNS должен выглядеть так:

```
customer1.org       MX  10  mail.myhost.com
```

Заметьте, что если вам требуется только получать почту для домена, соответствующая A-запись *не* нужна.

Помните, что если вы попытаетесь каким-либо образом обратиться к хосту `customer1.org`, у вас вряд ли что-либо получится, если нет A-записи для этого хоста.

Последнее, что вы должны сделать - это сказать программе sendmail, для каких доменов и/или хостов она должна принимать почту. Это можно сделать несколькими способами:

- Добавьте названия этих хостов в файл /etc/mail/local-host-names, если вы используете `FEATURE(use_cw_file)`. Если у вас sendmail версии ниже 8.10, необходимо отредактировать файл /etc/sendmail.cw.
- Добавьте строку `Cwyour.host.com` в файл /etc/sendmail.cf или /etc/mail/sendmail.cf (если у вас sendmail версии 8.10 или более поздней).

23.7. Настройка почты только для отправки

Существует множество случаев, когда может потребоваться только отправка почты через почтовый сервер. Вот отдельные примеры:

- У вас настольный компьютер, но вы хотите использовать такие программы как send-pr(1). Для пересылки почты вам потребуется использовать почтовый сервер провайдера.
- Ваш компьютер является сервером, где почта не хранится локально, необходима только переправка всей почты через внешний почтовый сервер.

Практически любой MTA способен работать и в этих условиях. К сожалению, может быть очень сложно правильно настроить полноценный MTA для работы только с исходящей почтой. Такие программы, как sendmail и postfix слишком избыточны для этих целей.

К тому же, если вы используете обычные средства доступа в интернет, условий для запуска "почтового сервера" может быть недостаточно.

Простейшим способом удовлетворить имеющиеся потребности может быть установка порта mail/ssmtp. Выполните под `root` следующие команды:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

После установки потребуется настроить mail/ssmtp с помощью файла из четырех строк, расположенного в /usr/local/etc/ssmtp/ssmtp.conf:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Убедитесь, что используете существующий почтовый адрес для `root`. Введите сервер вашего провайдера для пересылки исходящей почты вместо `mail.example.com` (некоторые провайдеры называют его "сервером исходящей почты" или "SMTP сервером").

Убедитесь, что вы выключили sendmail, включая сервис исходящей почты. За подробностями обращайтесь к Отключение sendmail.

У пакета mail/ssmtp имеются и другие параметры. Обратитесь к файлу с примером настройки в /usr/local/etc/ssmtp или к странице справочника ssmtp за примерами и дополнительной информацией.

Установка ssmtp таким способом позволит правильно работать любым программам на вашем компьютере, которым требуется отправка почты, но не нарушит политику вашего провайдера и не позволит вашему компьютеру быть использованным спамерами.

23.8. Использование почты с коммутируемым соединением

Если у вас есть статический IP, настройки по умолчанию менять не потребуется. Установите имя хоста в соответствии с присвоенным именем интернет и sendmail будет делать свою работу.

Если у вас динамический IP адрес и используется коммутируемое PPP соединение с интернет, у вас возможно уже есть почтовый ящик на сервере провайдера. Предположим, что домен провайдера называется `example.net`, и что ваше имя пользователя `user`, ваш компьютер называется `bsd.home`, и провайдер сообщил вам, что возможно использование `relay.example.net` в качестве сервера для пересылки почты.

Для получения почты из почтового ящика необходима установка соответствующей программы. Хорошим выбором является утилита fetchmail, она поддерживает множество различных протоколов. Эта программа доступна в виде пакета или из Коллекции Портов (mail/fetchmail). Обычно провайдер предоставляет доступ по протоколу POP. Если вы работаете с пользовательским PPP, то можете автоматически забирать почту после установления соединения с интернет с помощью следующей записи в /etc/ppp/ppp.linkup:

```
MYADDR:
!bg su user -c fetchmail
```

Если вы используете sendmail (как показано ниже) для доставки почты к не-локальным учетным записям, вам возможно потребуется обработка почтовой очереди sendmail сразу после установки соединения с интернет. Для выполнения этой работы поместите в /etc/ppp/ppp.linkup следующую команду сразу после `fetchmail`:

```
!bg su user -c "sendmail -q"
```

Предполагается, что учетная запись для `user` существует на `bsd.home`. В домашнем каталоге `user` на `bsd.home`, создайте файл .fetchmailrc:

```
poll example.net protocol pop3 fetchall pass MySecret
```

Этот файл не должен быть доступен на чтение никому, кроме `user`, поскольку в нем находится пароль `MySecret`.

Для отправки почты с правильным заголовком `from:`, вам потребуется сообщить sendmail использовать user@example.net вместо user@bsd.home. Вы можете также указать sendmail отправлять почту через `relay.example.net`, для более быстрой пересылки почты.

Должен подойти следующий файл .mc:

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(`example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.example.net')
Dmbsd.home
define(`confDOMAIN_NAME',`bsd.home')dnl
define(`confDELIVERY_MODE',`deferred')dnl
```

Обратитесь к предыдущему разделу за информацией о том, как преобразовать этот файл .mc в файл sendmail.cf. Не забудьте также перезапустить sendmail после обновления sendmail.cf.

23.9. SMTP аутентификация

Наличие SMTP аутентификации на почтовом сервере дает множество преимуществ. SMTP аутентификация может добавить дополнительный уровень безопасности к sendmail, и

позволяет мобильным пользователям, подключающимся к разным хостам, возможность использовать тот же почтовый сервер без необходимости перенастройки почтового клиента при каждом подключении.

1. Установите security/cyrus-sasl2 из портов. Вы можете найти этот порт в security/cyrus-sasl2. В пакете security/cyrus-sasl2 есть множество параметров компиляции. Для используемого здесь метода SMTP аутентификации убедитесь, что параметр `LOGIN` не отключен.
2. После установки security/cyrus-sasl2, отредактируйте /usr/local/lib/sasl2/Sendmail.conf (или создайте его если он не существует) и добавьте следующую строку:

   ```
   pwcheck_method: saslauthd
   ```

3. Затем установите security/cyrus-sasl2-saslauthd и добавьте в /etc/rc.conf следующую строку:

   ```
   saslauthd_enable="YES"
   ```

   а затем запустите saslauthd:

   ```
   # service saslauthd start
   ```

   Этот даемон является посредником для аутентификации sendmail через базу данных passwd FreeBSD. Это позволяет избежать проблем, связанных с созданием нового набора имен пользователей и паролей для каждого пользователя, которому необходима SMTP аутентификация, пароль для входа в систему и для отправки почты будет одним и тем же.

4. Теперь отредактируйте /etc/make.conf и добавьте следующие строки:

   ```
   SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
   SENDMAIL_LDFLAGS=-L/usr/local/lib
   SENDMAIL_LDADD=-lsasl2
   ```

   Эти параметры необходимы sendmail для подключения cyrus-sasl2 во время компиляции. Убедитесь, что cyrus-sasl2 был установлен до перекомпиляции sendmail.

5. Перекомпилируйте sendmail, выполнив следующие команды:

   ```
   # cd /usr/src/lib/libsmutil
   # make cleandir && make obj && make
   # cd /usr/src/lib/libsm
   # make cleandir && make obj && make
   ```

```
# cd /usr/src/usr.sbin/sendmail
# make cleandir && make obj && make && make install
```

Компиляция sendmail должна пройти без проблем, если /usr/src не был сильно изменен и доступны необходимые разделяемые библиотеки.

6. После компилирования и переустановки sendmail, отредактируйте файл /etc/mail/freebsd.mc (или тот файл, который используется в качестве .mc; многие администраторы используют в качестве имени этого файла вывод hostname(1) для обеспечения уникальности). Добавьте к нему следующие строки:

```
dnl set SASL options
TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
```

Эти параметры настраивают различные методы, доступные sendmail для аутентификации пользователей. Если вы хотите использовать вместо pwcheck другой метод, обратитесь к прилагаемой документации.

7. Наконец, запустите make(1) в каталоге /etc/mail. Из файла .mc будет создан файл .cf, называющийся freebsd.cf (или с тем именем, которое было использовано для файла .mc). Затем используйте команду `make install restart`, которая скопирует файл в sendmail.cf, и правильно перезапустит sendmail. Дополнительная информация об этом процессе находится в /etc/mail/Makefile.

Если все шаги пройдены успешно, введите информацию для аутентификации в настройки почтового клиента и отправьте тестовое сообщение. Для определения причин возможных ошибок установите параметр sendmail `LogLevel` в 13 и просмотрите /var/log/maillog.

За дальнейшей информацией обратитесь к странице sendmail, посвященной SMTP аутентификации.

23.10. Почтовые программы пользователей

Почтовая программа пользователя (Mail User Agent, MUA) это приложение, используемое для отправки и получения почты. Кроме того, поскольку почта "эволюционирует" и становится более сложной, MUA совершенствуют свои функции по обработке почты, становятся более удобны в использовании. FreeBSD поддерживает множество различных пользовательских почтовых программ, каждая из которых может быть легко установлена из Коллекции Портов FreeBSD. Пользователи могут выбирать между графическими почтовыми клиентами, такими как evolution или balsa, консольными клиентами, такими как mutt, alpine или `mail`, или Web-интерфейсами, используемыми в некоторых больших организациях.

23.10.1. mail

В FreeBSD в качестве MUA по умолчанию используется mail(1). Это консольный MUA, предоставляющий все основные функции, необходимые для отправки и получения текстовых сообщений, хотя его возможности по работе с вложениями ограничены и он может работать только с локальными почтовыми ящиками.

Хотя `mail` не поддерживает работу с серверами POP или IMAP, эти почтовые ящики могут быть загружены в локальный файл mbox с помощью fetchmail, который будет обсуждаться далее в этой главе (Использование fetchmail).

Для отправки и получения почты выполните `mail`:

```
% mail
```

Содержимое почтового ящика в каталоге /var/mail будет автоматически прочитано утилитой `mail`. Если почтовый ящик пуст, утилита завершит работу с сообщением о том, что почта не была обнаружена. После чтения почтового ящика запустится интерфейс программы и будет отображен список сообщений. Сообщения нумеруются автоматически и будут выглядеть как в этом примере:

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost        Mon Mar  8 14:05  14/510   "test"
 N  2 root@localhost        Mon Mar  8 14:05  14/509   "user account"
 N  3 root@localhost        Mon Mar  8 14:05  14/509   "sample"
```

Теперь сообщения могут быть прочитаны с помощью команды `t`, завершаемой номером сообщения, которое должно быть отображено. В этом примере мы прочтем первое сообщение:

```
& t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)

This is a test message, please reply if you receive it.
```

Как видно в примере выше, клавиша `t` выводит сообщение со всеми заголовками. Для повторного вывода списка сообщений необходимо использовать клавишу `h`.

Если требуется ответить на сообщение, используйте для ответа `mail`, нажав клавишу `R` или `r`.

Клавиша `R` используется в `mail` для ответа только отправителю, а `r` для ответа и отправителю, и другим получателям сообщения. Вы можете также завершить эти команды номером письма, на которое хотите составить ответ. После этого необходимо ввести ответ, конец сообщения должен быть завершен символом `.` на новой строке. Пример можно увидеть ниже:

```
& R 1
To: root@localhost
Subject: Re: test

Thank you, I did get your email.
.
EOT
```

Для отправки нового сообщения используйте клавишу `m` и введите адрес получателя. Несколько получателей могут быть указаны через запятую. Введите тему сообщения и его содержимое. Конец сообщения отмечается помещением символа `.` на новой строке.

```
& mail root@localhost
Subject: I mastered mail

Now I can send and receive email using mail ... :)
.
EOT
```

В утилите `mail` для вызова справки в любой момент может быть использована команда `?`, для получения помощи по `mail` необходимо также обратиться к странице справочника mail(1).

> Как упоминалось выше, команда mail(1) не была первоначально предназначена для работы с вложениями, и поэтому их поддержка довольно слабая. Современные MUA, такие как mutt, работают с вложениями гораздо более уверенно. Но если вы все же предпочитаете использовать `mail`, установите порт converters/mpack.

23.10.2. mutt

mutt это небольшая но очень мощная почтовая программа с отличными возможностями, в числе которых:

- Возможность сортировки сообщений по дискуссиям;
- Поддержка PGP для подписи и шифрования сообщений;
- Поддержка MIME;
- Поддержка Maildir;
- Широкие возможности настройки.

Все эти возможности делают mutt одним из самых лучших почтовых клиентов. Обратитесь к http://www.mutt.org за дополнительной информацией по mutt.

Стабильная версия mutt может быть установлена из порта mail/mutt. После установки порта, mutt может быть запущен следующей командой:

```
% mutt
```

mutt автоматически прочтет содержимое пользовательского почтового ящика в каталоге /var/mail и отобразит почту, если она имеется в наличии. Если почты в ящике пользователя нет, mutt будет ожидать команд от пользователя. В примере ниже показан mutt со списком сообщений:

```
q:Quit  d:Del  u:Undel  s:Save  m:Mail  r:Reply  g:Group  ?:Help
   1 N   Mar 09 Super-User      (   1) test
   2 N   Mar 09 Super-User      (   1) user account
   3 N   Mar 09 Super-User      (   1) sample

-*-Mutt: /var/mail/marcs [Msgs:3 New:3 1.6K]---(date/date)--------------(all)---
```

Для чтения почты выберите сообщение с помощью клавиш навигации и нажмите `Enter`. Пример mutt, отображающего сообщение, показан ниже:

```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.  d:Del  r:Reply  j:Next ?:Help
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Tue,  9 Mar 2004 10:28:36 +0200 (SAST)
From: Super-User <root@localhost>

This is a test message, please reply if you receieve it.

-N  - 1/1: Super-User              test                                  -- (all)
```

Как и команда mail(1), mutt позволяет пользователям отвечать как только отправителю, так и всем получателям. Для ответа только отправителю почты, используйте клавишу `r`. Для группового ответа и отправителю сообщения и всем получателям используйте клавишу `g`.

mutt использует vi(1) в качестве редактора для создания писем и ответа на них. Редактор можно заменить путем создания или редактирования собственного .muttrc в своем домашнем каталоге и установки переменной `editor`, или установкой переменной окружения `EDITOR`. Обратитесь к http://www.mutt.org/ за более подробной информацией о настройке mutt.

Для создания нового почтового сообщения нажмите `m`. После введения темы mutt запустит vi(1) для создания письма. Как только письмо будет завершено, сохраните его и закройте `vi`, mutt продолжит работу, отобразив окно с сообщением, которое должно быть отправлено. Для отправки сообщения нажмите `y`. Пример окна с сообщением показан ниже:

```
y:Send  q:Abort  t:To  c:CC  s:Subj  a:Attach file  d:Descrip  ?:Help
    From: Marc Silver <marcs@localhost>
      To: Super-User <root@localhost>
      Cc:
     Bcc:
 Subject: Re: test
Reply-To:
     Fcc:
Security: Clear

-- Attachments
- I     1 /tmp/mutt-bsd-c0hobscQ            [text/plain, 7bit, us-ascii, 1.1K]

-- Mutt: Compose  [Approx. msg size: 1.1K   Atts: 1]----------------------------
```

mutt также содержит исчерпывающий справочник, к которому можно обратиться из большинства меню, нажав клавишу `?`. Верхняя строка также показывает клавиатурные сокращения, которые могут быть использованы.

23.10.3. alpine

alpine предназначен для начинающих пользователей, но включает некоторые дополнительные возможности.

> В программе alpine ранее были обнаружены некоторые уязвимости, позволяющие удаленному взломщику выполнять произвольный код с правами пользователя локальной системы путем отправки специально подготовленного письма. Все эти *известные* проблемы были исправлены, но код alpine написан в очень небезопасном стиле и офицеры безопасности FreeBSD считают, что возможно наличие других не обнаруженных уязвимостей. Имейте это ввиду при установке alpine.

Текущая версия alpine может быть установлена из порта mail/alpine. Как только порт установлен, alpine можно запустить командой:

```
% alpine
```

При первом запуске alpine отображает страницу приветствия с кратким введением, а также просьбу команды разработчиков alpine отправить анонимное почтовое сообщение, позволяющее им определить количество пользователей, работающих с их почтовым клиентом. Для отправки анонимного сообщения нажмите `Enter`, или `E` для выхода из приветствия без отправки анонимного сообщения. Пример приветствия показан ниже:

```
  PINE 4.58   GREETING TEXT                                        No Messages

                    <<<This message will appear only once>>>

           Welcome to Pine ... a Program for Internet News and Email

We hope you will explore Pine's many capabilities. From the Main Menu,
select Setup/Config to see many of the options available to you. Also
note that all screens have context-sensitive help text available.

SPECIAL REQUEST: This software is made available world-wide as a public
service of the University of Washington in Seattle. In order to justify
continuing development, it is helpful to have an idea of how many people
are using Pine. Are you willing to be counted as a Pine user? Pressing
Return will send an anonymous (meaning, your real email address will not
be revealed) message to the Pine development team at the University of
Washington for purposes of tallying.

              Pine is a trademark of the University of Washington.

                              [ALL of greeting text]
? Help      E Exit this greeting                  - PrevPage % Print
          Ret [Be Counted!]                     Spc NextPage
```

Затем отображается главное меню, перемещение по которому осуществляется с помощью клавиш навигации. В главном меню находятся ссылки для составления новых писем, просмотра почтовых каталогов, и даже управления адресной книгой. Ниже главного меню показаны клавиатурные сокращения, выполняющие соответствующие задачи.

По умолчанию alpine открывает каталог inbox. Для просмотра списка сообщений нажмите `I`, или выберите MESSAGE INDEX, как показано ниже:

```
  PINE 4.58   MAIN MENU                          Folder: INBOX  3 Messages

         ?     HELP              -  Get help using Pine

         C     COMPOSE MESSAGE   -  Compose and send a message

         I     MESSAGE INDEX     -  View messages in current folder

         L     FOLDER LIST       -  Select a folder to view

         A     ADDRESS BOOK      -  Update address book

         S     SETUP             -  Configure Pine Options

         Q     QUIT              -  Leave the Pine program

   Copyright 1989-2003.  PINE is a trademark of the University of Washington.

? Help                    P PrevCmd                R RelNotes
O OTHER CMDS > [Index]    N NextCmd                K KBLock
```

В списке показаны сообщения в текущем каталоге, они могут быть просмотрены с помощью клавиш навигации. Подсвеченные сообщения можно прочесть нажав `Enter`.

```
  PINE 4.58   MESSAGE INDEX                    Folder: INBOX  Message 1 of 3 ANS

  A   1 Mar  9 Super-User          (471) test
  A   2 Mar  9 Super-User          (479) user account
  A   3 Mar  9 Super-User          (473) sample

? Help        < FldrList   P PrevMsg       - PrevPage D Delete     R Reply
O OTHER CMDS  > [ViewMsg]  N NextMsg     Spc NextPage U Undelete   F Forward
```

На снимке экрана ниже показан пример письма, отображаемого alpine. Внизу экрана даны клавиатурные сокращения. Например, `r` используется для указания MUA ответить на отображаемое в данный момент сообщение.

```
  PINE 4.58   MESSAGE TEXT                     Folder: INBOX  Message 1 of 3 ALL ANS

Date: Tue,  9 Mar 2004 10:28:36 +0200 (SAST)
From: Super-User <root@localhost>
To: marcs@localhost
Subject: test

This is a test message, please reply if you receieve it.

                                  [ALL of message]
? Help        < MsgIndex   P PrevMsg       - PrevPage D Delete     R Reply
O OTHER CMDS  > ViewAttch  N NextMsg     Spc NextPage U Undelete   F Forward
```

Ответ на письмо в alpine осуществляется с помощью редактора pico, который

устанавливается по умолчанию вместе с alpine. pico упрощает навигацию в сообщении гораздо проще для новых пользователей, чем vi(1) или mail(1). Как только ответ будет готов, сообщение можно отправить нажав `Ctrl` + `X`. alpine запросит подтверждение.

```
  PINE 4.58   COMPOSE MESSAGE REPLY                    Folder: INBOX  3 Messages

To      : Super-User <root@localhost>
Cc      :
Attchmnt:
Subject : Re: test
----- Message Text -----

I did recieve your message...

^G Get Help  ^X Send      ^R Read File ^Y Prev Pg   ^K Cut Text  ^O Postpone
^C Cancel    ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

Программа alpine может быть настроена через пункт SETUP главного меню. Обратитесь к странице http://www.washington.edu/alpine/ за дальнейшей информацией.

23.11. Использование fetchmail

fetchmail это полноценный IMAP и POP клиент, позволяющий пользователям автоматически загружать почту с удаленных серверов IMAP и POP в локальные почтовые ящики; так доступ к почтовым ящикам упрощается. fetchmail может быть установлен из порта mail/fetchmail и предоставляет различные возможности, в том числе:

- Поддержка протоколов POP3, APOP, KPOP, IMAP, ETRN и ODMR.
- Возможность пересылки почты через SMTP, что позволяет использовать функции фильтрации, перенаправления и синонимов.
- Может быт запущен в режиме даемона для периодической проверки поступающих сообщений.
- Может забирать почту с нескольких почтовых ящиков и рассылать ее различным локальным пользователям в зависимости от настроек.

Описание всех возможностей fetchmail выходит за пределы этой главы, за дополнительной информацией обратитесь к документации по fetchmail. Утилита fetchmail требует наличия файла настройки .fetchmailrc. Этот файл включает информацию о сервере, а также информацию для аутентификации. Поскольку этот файл содержит важную информацию, правильно будет сделать его доступным для чтения только владельцем с помощью

следующей команды:

```
% chmod 600 .fetchmailrc
```

В следующем примере файл .fetchmailrc предназначен для загрузки одного почтового ящика по протоколу POP. Этот файл указывает fetchmail соединиться с `example.com` с именем пользователя `joesoap` и паролем `XXX`. В примере подразумевается, что пользователь `joesoap` существует также и в локальной системе.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

В следующем примере производится подключение к нескольким POP и IMAP серверам, при необходимости почта перенаправляется другим локальным пользователям:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

Утилита fetchmail может работать в режиме даемона с флагом `-d`, заданным с интервалом (в секундах), через который fetchmail должен опрашивать серверы, перечисленные в .fetchmailrc. В следующем примере fetchmail будет забирать почту каждые 600 секунд:

```
% fetchmail -d 600
```

Дополнительную информацию о fetchmail можно найти на сайте http://fetchmail.berlios.de/.

23.12. Использование procmail

Утилита procmail это невероятно мощное приложение, используемое для фильтрации входящей почты. Она позволяет пользователям определять "правила", которые могут быть сопоставлены входящим письмам для выполнения определенных действий или для перенаправления почты в альтернативные почтовые ящики и/или на почтовые адреса. procmail может быть установлен с помощью порта mail/procmail. После установки он может быть непосредственно интегрирован в большинство MTA; сверьтесь с документацией на ваш MTA. Другой способ интеграции procmail - добавление в файл .forward, находящийся в домашнем каталоге пользователя, следующей строки:

```
"|exec /usr/local/bin/procmail || exit 75"
```

В этом разделе будут показаны основы настройки правил procmail, а также краткое описание их действия. Эти и другие правила должны быть помещены в файл .procmailrc,

который должен находиться в домашнем каталоге пользователя.

Большую часть этих правил также можно найти на странице справочника procmailex(5).

Перенаправление всей почты от user@example.com на внешний адрес goodmail@example2.com:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Перенаправление всей почты объемом меньше 1000 байт на внешний адрес goodmail@example2.com:

```
:0
* < 1000
! goodmail@example2.com
```

Перенаправление всей почты, отправляемой на alternate@example.com, в почтовый ящик alternate:

```
:0
* ^TOalternate@example.com
alternate
```

Перенаправление всей почты с "Spam" в /dev/null:

```
:0
^Subject:.*Spam
/dev/null
```

Полезный пример, обрабатывающий входящую почту со списков рассылки `FreeBSD.org` и помещающий каждый список в отдельный почтовый ящик.

```
:0
* ^Sender:.owner-freebsd-\/[^@]+@FreeBSD.ORG
{
    LISTNAME=${MATCH}
    :0
    * LISTNAME??^\/[^@]+
    FreeBSD-${MATCH}
}
```

Глава 24. Сетевые серверы

24.1. Краткий обзор

Эта глава посвящена некоторым наиболее часто используемым сетевым службам систем UNIX®. Мы опишем, как установить, настроить, протестировать и поддерживать многие различные типы сетевых сервисов. Для облегчения вашей работы в главу включены примеры конфигурационных файлов.

После чтения этой главы вы будете знать:

- Как управлять даемоном inetd.
- Как настроить сетевую файловую систему.
- Как настроить сетевой сервер информации для совместного использования учётных записей пользователей.
- Как настроить автоматическое конфигурирование сетевых параметров при помощи DHCP.
- Как настроить сервер имён.
- Как настроить Apache HTTP сервер.
- Как настроить файловый и сервер печати для Windows® клиентов с использованием Samba.
- Как синхронизировать дату и время, а также настроить сервер времени с протоколом NTP.
- Как настроить стандартный демон протоколирования, `syslogd`, принимать сообщения от удалённых хостов.

Перед чтением этой главы вы должны:

- Понимать основы работы скриптов /etc/rc.
- Свободно владеть основными сетевыми терминами.
- Знать как устанавливать дополнительные программы сторонних разработчиков (Установка приложений. порты и пакеты).

24.2. "Супер-сервер"inetd

24.2.1. Обзор

inetd(8) иногда называют также "супер-сервером Интернет", потому что он управляет соединениями к многим сервисам. Когда inetd принимает соединение, он определяет, для какой программы предназначено соединение, запускает соответствующий процесс и предоставляет ему сокет, ссылка на который передается процессу в качестве стандартных устройств ввода, вывода и сообщения об ошибках. Для не слишком нагруженных серверов запуск через inetd может уменьшить общую нагрузку на систему по сравнению с запуском

каждого даемона индивидуально в выделенном режиме.

В первую очередь inetd используется для вызова других даемонов, но несколько простых протоколов, таких, как chargen, auth и daytime, обслуживаются непосредственно.

Этот раздел посвящен основам настройки inetd посредством его параметров командной строки и его конфигурационного файла, /etc/inetd.conf.

24.2.2. Настройки

inetd инициализируется посредством системы rc(8). Параметр `inetd_enable` по умолчанию установлен в `NO`, однако может быть включен утилитой sysinstall в процессе установки. Указание

```
inetd_enable="YES"
```

или

```
inetd_enable="NO"
```

в файле /etc/rc.conf разрешит или запретит запуск inetd во время загрузки. Команда

```
/etc/rc.d/inetd rcvar
```

покажет текущие установки переменных, относящихся к inetd.

Кроме того, через `inetd_flags` даемону inetd могут быть переданы различные параметры командной строки.

24.2.3. Параметры командной строки

Как и большинство даемонов, для inetd существует большое количество разнообразных опций, изменяющих его поведение. Полный из список таков:

`inetd [-d] [-l] [-w] [-W] [-c maximum] [-C rate] [-a address | hostname] [-p filename] [-R rate] [-s maximum] [configuration file]`

Опции могут передаваться inetd при помощи переменной `inetd_flags` файла /etc/rc.conf. По умолчанию переменная `inetd_flags` установлена в `-wW -C 60`, то есть включает обработку TCP wrapping и запрещает обращаться с одного IP-адреса к сервису более чем 60 раз в минуту.

Несмотря на то, что ниже по тексту мы упоминаем опции ограничения частоты обращения к службам (rate-limiting), в большинстве случаев начинающим пользователям нет необходимости менять эти параметры. Эти опции могут стать полезными в том случае, если вы обнаружите, что ваша система принимает чрезмерное количество соединений. Полный список опций можно найти на странице справочника inetd(8).

-c maximum

Определение максимального числа одновременных запусков каждой службы; по умолчание не ограничено. Может быть переопределено индивидуально для каждой службы при помощи параметра `max-child`.

-C rate

Определение по умолчанию максимального количества раз, которое служба может быть вызвана с одного IP-адреса в минуту; по умолчанию не ограничено. Может быть переопределено для каждой службы параметром `max-connections-per-ip-per-minute`.

-R rate

Определяет максимальное количество раз, которое служба может быть вызвана в минуту; по умолчанию 256. Частота, равная 0, не ограничивает число вызовов.

-s maximum

Задает максимальное количество процессов, одновременно обслуживающих один сервис для одного IP-адреса; по умолчанию не ограничено. Может переопределяться для каждой службы параметром `max-child-per-ip`.

24.2.4. inetd.conf

Настройка inetd производится через файл /etc/inetd.conf.

Если в файле /etc/inetd.conf делались изменения, то inetd можно заставить считать его конфигурационный файл повторно посредством команды

Пример 38. Перезагрузка конфигурационного файла inetd

```
# /etc/rc.d/inetd reload
```

В каждой строке конфигурационного файла описывается отдельный даемон. Комментариям в файле предшествует знак "#". Строки в файле /etc/inetd.conf имеют такой формат:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]
user[:group][/login-class]
server-program
server-program-arguments
```

Пример записи для даемона ftpd(8), использующего IPv4:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

service-name

Это имя сервиса, предоставляемого конкретным даемоном. Оно должно соответствовать сервису, указанному в файле /etc/services. Здесь определяется, какой порт должен обслуживать inetd. При создании нового сервиса он должен помещаться сначала в файл /etc/services.

socket-type

`stream`, `dgram`, `raw` либо `seqpacket`. `stream` должен использоваться для ориентированных на соединение даемонов TCP, когда как `dgram` используется для даемонов, использующих транспортный протокол UDP.

protocol

Одно из следующих:

Протокол	Описание
tcp, tcp4	TCP IPv4
udp, udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	TCP как для IPv4, так и для v6
udp46	UDP как для IPv4, так и для v6

{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]

`wait|nowait` определяет, может ли даемон, вызванный из inetd, работать с собственным сокетом, или нет. Сокеты типа `dgram` должны использовать параметр `wait`, когда как даемоны с потоковыми сокетами, которые обычно многопоточны, должны использовать `nowait`. `wait` обычно передает много сокетов одному даемону, когда как `nowait` порождает даемон для каждого нового сокета.

Максимальное число порожденных даемонов, которых может создать inetd, может быть задано параметром `max-child`. Если нужно ограничение в десять экземпляров некоторого даемона, то после параметра `nowait` нужно задать `/10`. При задании `/0` ограничения на количество экземпляров снимаются.

Кроме `max-child`, могут быть задействованы два других параметра, ограничивающих максимальное число соединений от одного источника. `max-connections-per-ip-per-minute` ограничивает количество соединений от одного IP-адреса в течение минуты, так что значение, равное десяти, будет ограничивать любой заданный IP-адрес на выполнение десяти попыток подключения к некоторому сервису в минуту. Параметр `max-child-per-ip` ограничивает количество дочерних процессов, которые могут быть одновременно задействованы на обслуживание одного IP-адреса. Эти опции полезны для предотвращения намеренного или ненамеренного расходования ресурсов и атак типа Denial of Service (DoS) на машину.

В этом поле одно из значений `wait` или `nowait` обязательны. `max-child`, `max-connections-per-ip-per-minute` и `max-child-per-ip` опциональны.

Многопоточный даемон типа stream без ограничений `max-child`, `max-connections-per-ip-per-minute` или `max-child-per-ip` будет определен просто как `nowait`.

Тот же самый даемон с ограничением в максимум десять даемонов будет определен так: `nowait/10`.

Та же конфигурация с ограничением в двадцать соединений на IP-адрес в минуту и общим ограничением в максимум десять порожденных даемонов выглядит так: `nowait/10/20`.

Эти параметры, используемые все со значениями по умолчанию даемоном fingerd(8), имеют такой вид:

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd fingerd -s
```

Наконец, пример, описывающий ограничение на 100 даемонов в целом, при этом не более чем по 5 на один IP-адрес, будет выглядеть так: `nowait/100/0/5`.

user

Это имя пользователя, под которым должен работать соответствующий даемон. Чаще всего даемоны работают как пользователь `root`. Для обеспечения безопасности некоторые серверы запускаются как пользователь `daemon` или как пользователь с минимальными правами `nobody`.

server-program

Полный маршрут к даемону, который будет выполняться при установлении соединения. Если даемон является сервисом, предоставляемым самим inetd, то нужно задать ключевое слово `internal`.

server-program-arguments

Этот параметр работает вместе с параметром `server-program`, задавая параметры, начиная с `argv[0]`, передаваемые даемону при запуске. Если в командной строке задано `mydaemon -d`, то `mydaemon -d` будет являться значением для `server-program-arguments`. И снова, если даемон является внутренней службой, то здесь нужно использовать `internal`.

24.2.5. Безопасность

В зависимости от выбранных при установке параметров, многие из служб inetd могут оказаться по умолчанию включенными. Если нет особой нужды в некотором даемоне, подумайте, не стоит ли его выключить? Поместите знак "#" перед ненужным даемоном в /etc/inetd.conf и пошлите сигнал для inetd. Некоторые даемоны, такие, как fingerd, вообще нежелательны, потому что они дают информацию, которая может оказаться полезной атакующему.

Некоторые даемоны не заботятся о безопасности и имеют большие тайм-ауты для соединений или вообще их не имеют. Это позволяет атакующему неспешно устанавливать соединения к конкретному даемону, истощая имеющиеся ресурсы. Может оказаться полезным задать для некоторых даемонов ограничения `max-connections-per-ip-per-minute`,

`max-child` и `max-child-per-ip`, особенно если вы обнаружите слишком большое число соединений.

По умолчанию механизм TCP wrapping включен. Обратитесь к справочной странице по hosts_access(5) для получения более подробной информации о задании ограничений TCP для различных даемонов, запускаемых посредством inetd.

24.2.6. Разное

daytime, time, echo, discard, chargen и auth все являются услугами, предоставляемыми самим inetd.

Сервис auth предоставляет идентификационные сетевые услуги и поддается настройке; прочие сервисы ненастраиваемы.

Обратитесь к справочной странице по inetd(8) для получения более подробной информации.

24.3. Network File System (NFS)

Кроме поддержки многих прочих типов файловых систем, во FreeBSD встроена поддержка сетевой файловой системы (Network File System), известной как NFS. NFS позволяет системе использовать каталоги и файлы совместно с другими машинами, посредством сети. Посредством NFS пользователи и программы могут получать доступ к файлам на удалённых системах точно так же, как если бы это были файлы на собственных дисках.

Вот некоторые из наиболее заметных преимуществ, которые даёт использование NFS:

- Отдельно взятые рабочие станции используют меньше собственного дискового пространства, так как совместно используемые данные могут храниться на одной отдельной машине и быть доступными для других машин в сети.
- Пользователям не нужно иметь домашние каталоги, отдельные для каждой машины в вашей сети. Домашние каталоги могут располагаться на сервере NFS и их можно сделать доступными отовсюду в сети.
- Устройства хранения информации, такие, как дискеты, приводы CD-ROM и устройства Zip®, могут использоваться другими машинами в сети. Это может привести к уменьшению переносимых устройств хранения информации в сети.

24.3.1. Как работает NFS

NFS строится по крайней мере из двух основных частей: сервера и одного или большего количества клиентов. Клиент обращается к данным, находящимся на сервере, в режиме удалённого доступа. Для того, чтобы это нормально функционировало, нужно настроить и запустить несколько процессов.

На сервере работают следующие даемоны:

Даемон	Описание
nfsd	Даемон NFS, обслуживающий запросы от клиентов NFS.
mountd	Даемон монтирования NFS, который выполняет запросы, передаваемые ему от nfsd(8).
rpcbind	Этот даемон позволяет клиентам NFS определить порт, используемый сервером NFS.

Клиент может запустить также даемон, называемый nfsiod. nfsiod обслуживает запросы, поступающие от сервера от сервера NFS. Он необязателен, увеличивает производительность, однако для нормальной и правильной работы не требуется. Для получения дополнительной информации обратитесь к разделу справочной системы о nfsiod(8).

24.3.2. Настройка NFS

Настройка NFS является достаточно незамысловатым процессом. Все процессы, которые должны быть запущены, могут быть запущены во время загрузки посредством нескольких модификаций в вашем файле /etc/rc.conf.

Проверьте, что на NFS-сервере в файле /etc/rc.conf имеются такие строки:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
nfs_server_flags="-u -t -n 4"
mountd_flags="-r"
```

mountd запускается автоматически, если включена функция сервера NFS.

На клиенте убедитесь, что в файле /etc/rc.conf присутствует такой параметр:

```
nfs_client_enable="YES"
```

Файл /etc/exports определяет, какие файловые системы на вашем сервере NFS будут экспортироваться (иногда их называют "совместно используемыми"). Каждая строка в /etc/exports задаёт файловую систему, которая будет экспортироваться и какие машины будут иметь к ней доступ. Кроме машин, имеющих доступ, могут задаваться другие параметры, влияющие на характеристики доступа. Имеется полный набор параметров, которые можно использовать, но здесь пойдёт речь лишь о некоторых из них. Описания остальных параметров можно найти на страницах справочной системы по exports(5).

Вот несколько примерных строк из файла /etc/exports:

В следующих примерах даётся общая идея того, как экспортировать файловые системы,

хотя конкретные параметры могут отличаться в зависимости от ваших условий и конфигурации сети. К примеру, чтобы экспортировать каталог /cdrom для трёх машин, находящихся в том же самом домене, что и сервер (поэтому отсутствует доменное имя для каждой машины) или для которых имеются записи в файле /etc/hosts. Флаг `-ro` указывает на использование экспортируемой файловой системы в режиме только чтения. С этим флагом удалённая система не сможет никоим образом изменить экспортируемую файловую систему.

```
/cdrom -ro host1 host2 host3
```

В следующей строке экспортируется файловая система /home, которая становится доступной трем хостам, указанным по их IP-адресам. Это полезно, если у вас есть собственная сеть без настроенного сервера DNS. Как вариант, файл /etc/hosts может содержать внутренние имена хостов; пожалуйста, обратитесь к справочную систему по hosts(5) для получения дополнительной информации. Флаг `-alldirs` позволяет рассматривать подкаталоги в качестве точек монтирования. Другими словами, это не монтирование подкаталогов, но разрешение клиентам монтировать только каталоги, которые им требуются или нужны.

```
/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

В строке, приведённой ниже, файловая система /a экспортируется таким образом, что она доступна двум клиентам из других доменов. Параметр `-maproot=root` позволяет пользователю `root` удалённой системы осуществлять запись на экспортируемую файловую систему как пользователь `root`. Если параметр `-maproot=root` не задан, то даже если пользователь имеет права доступа `root` на удалённой системе, он не сможет модифицировать файлы на экспортированной файловой системе.

```
/a  -maproot=root  host.example.com box.example.org
```

Для того, чтобы клиент смог обратиться к экспортированной файловой системе, он должен иметь права сделать это. Проверьте, что клиент указан в вашем файле /etc/exports.

В файле /etc/exports каждая строка содержит информацию об экспортировании для отдельной файловой системы для отдельно взятого хоста. Удалённый хост может быть задан только один раз для каждой файловой системы, и может иметь только одну запись, используемую по умолчанию, для каждой локальной файловой системы. К примеру, предположим, что /usr является отдельной файловой системой. Следующий /etc/exports будет некорректен:

```
# Invalid when /usr is one file system
/usr/src   client
/usr/ports client
```

Одна файловая система, /usr, имеет две строки, задающие экспортирование для одного и того же хоста, `client`. Правильный формат в этом случае таков:

```
/usr/src /usr/ports  client
```

Свойства отдельной файловой системы, экспортируемой некоторому хосту, должны задаваться в одной строке. Строки без указания клиента воспринимаются как отдельный хост. Это ограничивает то, как вы можете экспортировать файловые системы, но для большинства это не проблема.

Ниже приведён пример правильного списка экспортирования, где /usr и /exports являются локальными файловыми системами:

```
# Экспортируем src и ports для client01 и client02, но
# только client01 имеет права пользователя root на них
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports        client02
# Клиентские машины имеют пользователя root и могут монтировать всё в
# каталоге /exports.  Кто угодно может монтировать /exports/obj в режиме чтения
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

Даемон mountd должен быть проинформирован об изменении файла /etc/exports, чтобы изменения вступили в силу. Это может быть достигнуто посылкой сигнала HUP процессу `mountd`:

```
# kill -HUP `cat /var/run/mountd.pid`
```

или вызовом скрипта `mountd` подсистемы rc(8) с соответствующим параметром:

```
# /etc/rc.d/mountd onereload
```

За подробной информацией о работе скриптов rc.d обращайтесь к Использование rc во FreeBSD 5.X и последующих версиях.

Как вариант, при перезагрузке FreeBSD всё настроится правильно. Хотя выполнять перезагрузку вовсе не обязательно. Выполнение следующих команд пользователем `root` запустит всё, что нужно.

На сервере NFS:

```
# rpcbind
# nfsd -u -t -n 4
# mountd -r
```

На клиенте NFS:

```
# nfsiod -n 4
```

Теперь всё должно быть готово к реальному монтированию удалённой файловой системы. В приводимых примерах сервер будет носить имя `server`, а клиент будет носить имя `client`. Если вы только хотите временно смонтировать удалённую файловую систему, или всего лишь протестировать ваши настройки, то просто запустите команды, подобные приводимым здесь, работая как пользователь `root` на клиентской машине:

```
# mount server:/home /mnt
```

По этой команде файловая система /home на сервере будет смонтирована в каталог /mnt на клиенте. Если всё настроено правильно, вы сможете войти в каталог /mnt на клиенте и увидеть файлы, находящиеся на сервере.

Если вы хотите автоматически монтировать удалённую файловую систему при каждой загрузке компьютера, добавьте файловую систему в /etc/fstab. Вот пример:

```
server:/home      /mnt    nfs     rw      0   0
```

На страницах справочной системы по fstab(5) перечислены все доступные параметры.

24.3.3. Блокировка файлов

Некоторым приложениям (например, mutt) для корректной работы необходима возможность блокировки файлов (file locking). При работе по NFS блокировка файлов может осуществляться при помощи демона rpc.lockd. Чтобы его активировать, добавьте следующие записи в файл /etc/rc.conf как на клиенте, так и на сервере NFS (предполагается, что и клиент, и сервер уже сконфигурированы):

```
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Запустите демоны, выполнив следующие команды:

```
# /etc/rc.d/lockd start
# /etc/rc.d/statd start
```

Если нет необходимости в настоящей блокировке файлов между сервером NFS и клиентами, то клиент NFS может быть настроен так, чтобы выполнять блокировки файлов локально, для чего необходимо передать опцию `-L` команде mount_nfs(8). За подробностями обратитесь к странице справочника mount_nfs(8).

24.3.4. Практическое использование

У NFS есть много вариантов практического применения. Ниже приводится несколько наиболее широко распространённых способов её использования:

- Настройка несколько машин для совместного использования CDROM или других носителей. Это более дешёвый и зачастую более удобный способ установки программного обеспечения на несколько машин.
- В больших сетях может оказаться более удобным настроить центральный сервер NFS, на котором размещаются все домашние каталоги пользователей. Эти домашние каталоги могут затем экспортироваться в сеть так, что пользователи всегда будут иметь один и тот же домашний каталог вне зависимости от того, на какой рабочей станции они работают.
- Несколько машин могут иметь общий каталог /usr/ports/distfiles. Таким образом, когда вам нужно будет установить порт на несколько машин, вы сможете быстро получить доступ к исходным текстам без их загрузки на каждой машине.

24.3.5. Автоматическое монтирование с amd

amd(8) (даемон автоматического монтирования) автоматически монтирует удалённую файловую систему, как только происходит обращение к файлу или каталогу в этой файловой системе. Кроме того, файловые системы, которые были неактивны некоторое время, будут автоматически размонтированы даемоном amd. Использование amd является простой альтернативой статическому монтированию, так как в последнем случае обычно всё должно быть описано в файле /etc/fstab.

amd работает, сам выступая как сервер NFS для каталогов /host и /net. Когда происходит обращение к файлу в одном из этих каталогов, amd ищет соответствующий удаленный ресурс для монтирования и автоматически его монтирует. /net используется для монтирования экспортируемой файловой системы по адресу IP, когда как каталог /host используется для монтирования ресурса по удаленному имени хоста.

Обращение к файлу в каталоге /host/foobar/usr укажет amd на выполнение попытки монтирования ресурса /usr, который находится на хосте `foobar`.

Пример 39. Монтирование ресурса при помощи amd

Вы можете посмотреть доступные для монтирования ресурсы отдалённого хоста командой `showmount`. К примеру, чтобы посмотреть ресурсы хоста с именем `foobar`, вы можете использовать:

```
% showmount -e foobar
Exports list on foobar:
/usr                    10.10.10.0
/a                10.10.10.0
% cd /host/foobar/usr
```

Как видно из примера, `showmount` показывает /usr как экспортируемый ресурс. При переходе в каталог /host/foobar/usr даемон amd пытается разрешить имя хоста `foobar` и автоматически смонтировать требуемый ресурс.

amd может быть запущен из скриптов начальной загрузки, если поместить такую строку в файл /etc/rc.conf:

```
amd_enable="YES"
```

Кроме того, даемону amd могут быть переданы настроечные флаги через параметр `amd_flags`. По умолчанию `amd_flags` настроен следующим образом:

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map"
```

Файл /etc/amd.map задает опции, используемые по умолчанию при монтировании экспортируемых ресурсов. В файле /etc/amd.conf заданы настройки некоторых более сложных возможностей amd.

Обратитесь к справочным страницам по amd(8) и amd.conf(8) для получения более полной информации.

24.3.6. Проблемы взаимодействия с другими системами

Некоторые сетевые адаптеры для систем PC с шиной ISA имеют ограничения, которые могут привести к серьезным проблемам в сети, в частности, с NFS. Эти проблемы не специфичны для FreeBSD, однако эту систему они затрагивают.

Проблема, которая возникает практически всегда при работе по сети систем PC (FreeBSD) с высокопроизводительными рабочими станциями, выпущенными такими производителями, как Silicon Graphics, Inc. и Sun Microsystems, Inc. Монтирование по протоколу NFS будет работать нормально, и некоторые операции также будут выполняться успешно, но неожиданно сервер окажется недоступным для клиент, хотя запросы к и от других систем будут продолжаться обрабатываться. Такое встречается с клиентскими системами, не зависимо от того, является ли клиент машиной с FreeBSD или рабочей станцией. Во многих системах при возникновении этой проблемы нет способа корректно завершить работу клиента. Единственным выходом зачастую является холодная перезагрузка клиента, потому что ситуация с NFS не может быть разрешена.

Хотя "правильным" решением является установка более производительного и скоростного сетевого адаптера на систему FreeBSD, имеется простое решение, приводящее к удовлетворительным результатам. Если система FreeBSD является *сервером*, укажите параметр `-w=1024` на клиенте при монтировании. Если система FreeBSD является *клиентом*, то смонтируйте файловую систему NFS с параметром `-r=1024`. Эти параметры могут быть заданы в четвертом поле записи в файле fstab клиента при автоматическом монтировании, или при помощи параметра `-o` в команде mount(8) при монтировании вручную.

Нужно отметить, что имеется также другая проблема, ошибочно принимаемая за

приведенную выше, когда серверы и клиенты NFS находятся в разных сетях. Если это тот самый случай, *проверьте*, что ваши маршрутизаторы пропускают нужную информацию UDP, в противном случае вы ничего не получите, что бы вы ни предпринимали.

В следующих примерах `fastws` является именем хоста (интерфейса) высокопроизводительной рабочей станции, а `freebox` является именем хоста (интерфейса) системы FreeBSD со слабым сетевым адаптером. Кроме того, /sharedfs будет являться экспортируемой через NFS файловой системой (обратитесь к страницам справочной системы по команде exports(5)), а /project будет точкой монтирования экспортируемой файловой системы на клиенте. В любом случае, отметьте, что для вашего приложения могут понадобиться дополнительные параметры, такие, как `hard`, `soft` или `bg`.

Пример системы FreeBSD (`freebox`) как клиента в файле /etc/fstab на машине `freebox`:

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

Команда, выдаваемая вручную на машине `freebox`:

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

Пример системы FreeBSD в качестве сервера в файле /etc/fstab на машине `fastws`:

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

Команда, выдаваемая вручную на машине `fastws`:

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

Практически все 16-разрядные сетевые адаптеры позволят работать без указанных выше ограничений на размер блоков при чтении и записи.

Для тех, кто интересуется, ниже описывается, что же происходит в при появлении этой ошибки, и объясняется, почему ее невозможно устранить. Как правило, NFS работает с "блоками" размером 8 килобайт (хотя отдельные фрагменты могут иметь меньшие размеры). Так, пакет Ethernet имеет максимальный размер около 1500 байт, то "блок" NFS разбивается на несколько пакетов Ethernet, хотя на более высоком уровне это все тот же единый блок, который должен быть принят, собран и *подтвержден* как один блок. Высокопроизводительные рабочие станции могут посылать пакеты, которые соответствуют одному блоку NFS, сразу друг за другом, насколько это позволяет делать стандарт. На слабых, низкопроизводительных адаптерах пакеты, пришедшие позже, накладываются поверх ранее пришедших пакетов того же самого блока до того, как они могут быть переданы хосту и блок как единое целое не может быть собран или подтвержден. В результате рабочая станция входит в ситуацию тайм-аута и пытается повторить передачу, но уже с полным блоком в 8 КБ, и процесс будет повторяться снова, до бесконечности.

Задав размер блока меньше размера пакета Ethernet, мы достигаем того, что любой полностью полученный пакет Ethernet может быть подтвержден индивидуально, и избежим тупиковую ситуацию.

Наложение пакетов может все еще проявляться, когда высокопроизводительные рабочие станции сбрасывают данные на PC-систему, однако повторение этой ситуации не обязательно с более скоростными адаптерами с "блоками" NFS. Когда происходит наложение, затронутые блоки будут переданы снова, и скорее всего, они будут получены, собраны и подтверждены.

24.4. Network Information System (NIS/YP)

24.4.1. Что это такое?

NIS, что является сокращением от Network Information Services (Сетевые Информационные Службы), которые были разработаны компанией Sun Microsystems для централизованного администрирования систем UNIX® (изначально SunOS™). В настоящее время эти службы практически стали промышленным стандартом; все основные UNIX®-подобные системы (Solaris™, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD и так далее) поддерживают NIS.

NIS первоначально назывались Yellow Pages (или yp), но из-за проблем с торговым знаком Sun изменила это название. Старое название (и yp) всё ещё часто употребляется.

Это система клиент/сервер на основе вызовов RPC, которая позволяет группе машин в одном домене NIS совместно использовать общий набор конфигурационных файлов. Системный администратор может настроить клиентскую систему NIS только с минимальной настроечной информацией, а затем добавлять, удалять и модифицировать настроечную информацию из одного места.

Это похоже на систему доменов Windows NT®; хотя их внутренние реализации не так уж и похожи, основные функции сравнимы.

24.4.2. Термины/программы, о которых вы должны знать

Существует несколько терминов и некоторое количество пользовательских программ, которые будут нужны, когда вы будете пытаться сделать NIS во FreeBSD, и в случае создания сервера, и в случае работы в качестве клиента NIS:

Термин	Описание
Имя домена NIS	Главный сервер NIS и все его клиенты (включая вторичные серверы), имеют доменное имя NIS. Как и в случае с именем домена Windows NT®, имя домена NIS не имеет ничего общего с DNS.

Термин	Описание
rpcbind	Для обеспечения работы RPC (Remote Procedure Call, Удалённого Вызова Процедур, сетевого протокола, используемого NIS), должен быть запущен даемон rpcbind. Если даемон rpcbind не запущен, невозможно будет запустить сервер NIS, или работать как NIS-клиент.
ypbind	"Связывает" NIS-клиента с его NIS-сервером. Он определяет имя NIS-домена системы, и при помощи RPC подключается к серверу. ypbind является основой клиент-серверного взаимодействия в среде NIS; если на клиентской машине программа ypbind перестанет работать, то эта машина не сможет получить доступ к серверу NIS.
ypserv	Программа `ypserv`, которая должна запускаться только на серверах NIS: это и есть сервер NIS. Если ypserv(8) перестанет работать, то сервер не сможет отвечать на запросы NIS (к счастью, на этот случай предусмотрен вторичный сервер). Есть несколько реализаций NIS (к FreeBSD это не относится), в которых не производится попыток подключиться к другому серверу, если ранее используемый сервер перестал работать. Зачастую единственным средством, помогающим в этой ситуации, является перезапуск серверного процесса (или сервера полностью) или процесса ypbind на клиентской машине.
rpc.yppasswdd	Программа `rpc.yppasswdd`, другой процесс, который запускается только на главных NIS-серверах: это даемон, позволяющий клиентам NIS изменять свои пароли NIS. Если этот даемон не запущен, то пользователи должны будут входить на основной сервер NIS и там менять свои пароли.

24.4.3. Как это работает?

В системе NIS существует три типа хостов: основные (master) серверы, вторичные (slave) серверы и клиентские машины. Серверы выполняют роль централизованного хранилища информации о конфигурации хостов. Основные серверы хранят оригиналы этой информации, когда как вторичные серверы хранят ее копию для обеспечения

избыточности. Клиенты связываются с серверами, чтобы предоставить им эту информацию.

Информация во многих файлах может совместно использоваться следующим образом. Файлы master.passwd, group и hosts используются совместно через NIS. Когда процессу, работающему на клиентской машине, требуется информация, как правило, находящаяся в этих файлах локально, то он делает запрос к серверу NIS, с которым связан.

24.4.3.1. Типы машин

- *Основной сервер NIS*. Такой сервер, по аналогии с первичным контроллером домена Windows NT®, хранит файлы, используемые всеми клиентами NIS. Файлы passwd, group и различные другие файлы, используемые клиентами NIS, находятся на основном сервере.

> Возможно использование одной машины в качестве сервера для более чем одного домена NIS. Однако, в этом введении такая ситуация не рассматривается, и предполагается менее масштабное использование NIS.

- *Вторичные серверы NIS*. Похожие на вторичные контроллеры доменов Windows NT®, вторичные серверы NIS содержат копии оригинальных файлов данных NIS. Вторичные серверы NIS обеспечивают избыточность, что нужно в критичных приложениях. Они также помогают распределять нагрузку на основной сервер: клиенты NIS всегда подключаются к тому серверу NIS, который ответил первым, в том числе и к вторичным серверам.
- *Клиенты NIS*. Клиенты NIS, как и большинство рабочих станций Windows NT®, аутентифицируются на сервере NIS (или на контроллере домена Windows NT® для рабочих станций Windows NT®) во время входа в систему.

24.4.4. Использование NIS/YP

В этом разделе приводится пример настройки NIS.

24.4.4.1. Планирование

Давайте предположим, что вы являетесь администратором в маленькой университетской лаборатории. В настоящий момент в этой лаборатории с 15 машинами отсутствует единая точка администрирования; на каждой машине имеются собственные файлы /etc/passwd и /etc/master.passwd. Эти файлы синхронизируются друг с другом только вручную; сейчас, когда вы добавляете пользователя в лаборатории, вы должны выполнить команду `adduser` на всех 15 машинах. Понятно, что такое положение вещей нужно исправлять, так что вы решили перевести сеть на использование NIS, используя две машины в качестве серверов.

Итак, конфигурация лаборатории сейчас выглядит примерно так:

Имя машины	IP-адрес	Роль машины
`ellington`	`10.0.0.2`	Основной сервер NIS

Имя машины	IP-адрес	Роль машины
`coltrane`	`10.0.0.3`	Вторичный сервер NIS
`basie`	`10.0.0.4`	Факультетская рабочая станция
`bird`	`10.0.0.5`	Клиентская машина
`cli[1-11]`	`10.0.0.[6-17]`	Другие клиентские машины

Если вы определяете схему NIS первый раз, ее нужно хорошо обдумать. Вне зависимости от размеров вашей сети, есть несколько ключевых моментов, которые требуют принятия решений.

24.4.4.1.1. Выбор имени домена NIS

Это имя не должно быть "именем домена", которое вы использовали. Более точно это имя называется "именем домена NIS". Когда клиент рассылает запросы на получение информации, он включает в них имя домена NIS, частью которого является. Таким способом многие сервера в сети могут указать, какой сервер на какой запрос должен отвечать. Думайте о домене NIS как об имени группы хостов, которые каким-то образом связаны.

Некоторые организации в качестве имени домена NIS используют свой домен Интернет. Это не рекомендуется, так как может вызвать проблемы в процессе решения сетевых проблем. Имя домена NIS должно быть уникальным в пределах вашей сети и хорошо, если оно будет описывать группу машин, которые представляет. Например, художественный отдел в компании Acme Inc. может находиться в домене NIS с именем "acme-art". В нашем примере положим, что мы выбрали имя `test-domain`.

Несмотря на это, некоторые операционные системы (в частности, SunOS™) используют свое имя домена NIS в качестве имени домена Интернет. Если одна или более машин в вашей сети имеют такие ограничения, вы *обязаны* использовать имя домена Интернет в качестве имени домена NIS.

24.4.4.1.2. Требования к серверу

Есть несколько вещей, которые нужно иметь в виду при выборе машины для использования в качестве сервера NIS. Одной из обескураживающей вещью, касающейся NIS, является уровень зависимости клиентов от серверов. Если клиент не может подключиться к серверу своего домена NIS, зачастую машину просто становится нельзя использовать. Отсутствие информации о пользователях и группах приводит к временной остановке работы большинства систем. Зная это, вы должны выбрать машину, которая не должна подвергаться частым перезагрузкам и не используется для разработки. Сервер NIS в идеале должен быть отдельно стоящей машиной, единственным целью в жизни которой является быть сервером NIS. Если вы работаете в сети, которая не так уж сильно загружена, то можно поместить сервер NIS на машине, на которой запущены и другие сервисы, просто имейте в виду, что если сервер NIS становится недоступным, то это негативно отражается на *всех* клиентах NIS.

24.4.4.2. Серверы NIS

Оригинальные копии всей информации NIS хранится на единственной машине, которая называется главным сервером NIS. Базы данных, которые используются для хранения информации, называются картами NIS. Во FreeBSD эти карты хранятся в /var/yp/[domainname], где [domainname] является именем обслуживаемого домена NIS. Один сервер NIS может поддерживать одновременно несколько доменов, так что есть возможность иметь несколько таких каталогов, по одному на каждый обслуживаемый домен. Каждый домен будет иметь свой собственный независимый от других набор карт.

Основной и вторичный серверы обслуживают все запросы NIS с помощью даемона `ypserv`. `ypserv` отвечает за получение входящих запросов от клиентов NIS, распознавание запрашиваемого домена и отображение имени в путь к соответствующему файлы базы данных, а также передаче информации из базы данных обратно клиенту.

24.4.4.2.1. Настройка основного сервера NIS

Настройка основного сервера NIS может оказаться сравнительно простой, в зависимости от ваших потребностей. В поставку FreeBSD сразу включена поддержка NIS. Все, что вам нужно, это добавить следующие строки в файл /etc/rc.conf, а FreeBSD сделает за вас всё остальное..

```
nisdomainname="test-domain"
```

1. В этой строке задается имя домена NIS, которое будет `test-domain`, еще до настройки сети (например, после перезагрузки).

   ```
   nis_server_enable="YES"
   ```

2. Здесь указывается FreeBSD на запуск процессов серверов NIS, когда дело доходит до сетевых настроек.

   ```
   nis_yppasswdd_enable="YES"
   ```

3. Здесь указывается на запуск даемона `rpc.yppasswdd`, который, как это отмечено выше, позволит пользователям менять свой пароль NIS с клиентской машины.

> В зависимости от ваших настроек NIS, вам могут понадобиться дополнительные строки. Обратитесь к "разделу о серверах NIS, которые являются и клиентами NIS" ниже для получения подробной информации.

После добавления вышеприведенных строк, запустите команду `/etc/netstart`, работая как администратор. По ней произойдет настройка всего, при этом будут использоваться значения, заданные в файле /etc/rc.conf. И наконец, перед инициализацией карт NIS, запустите вручную демон ypserv:

```
# /etc/rc.d/ypserv start
```

24.4.4.2.2. Инициализация карт NIS

Карты NIS являются файлами баз данных, которые хранятся в каталоге /var/yp. Они генерируются из конфигурационных файлов, находящихся в каталоге /etc основного сервера NIS, за одним исключением: файл /etc/master.passwd. На это есть весомая причина, вам не нужно распространять пароли пользователя `root` и других административных пользователей на все серверы в домене NIS. По этой причине, прежде чем инициализировать карты NIS, вы должны сделать вот что:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

Вы должны удалить все записи, касающиеся системных пользователей (`bin`, `tty`, `kmem`, `games` и так далее), а также записи, которые вы не хотите распространять клиентам NIS (например, `root` и другие пользователи с UID, равным 0 (администраторы)).

> Проверьте, чтобы файл /var/yp/master.passwd был недоступен для записи ни для группы, ни для остальных пользователей (режим доступа 600)! Воспользуйтесь командой `chmod`, если это нужно.

Когда с этим будет покончено, самое время инициализировать карты NIS! В поставку FreeBSD включен скрипт с именем `ypinit`, который делает это (обратитесь к его справочной странице за дополнительной информацией). Отметьте, что этот скрипт имеется в большинстве операционных систем UNIX®, но не во всех. В системе Digital Unix/Compaq Tru64 UNIX он называется `ypsetup`. Так как мы генерируем карты для главного сервера NIS, то при вызове программы `ypinit` мы передаем ей параметр `-m`. Для генерации карт NIS в предположении, что вы уже сделали шаги, описанные выше, выполните следующее:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
```

```
coltrane
Is this correct?  [y/n: y] y

[..вывод при генерации карт..]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

Программа ypinit должна была создать файл /var/yp/Makefile из /var/yp/Makefile.dist. При создании этого файла предполагается, что вы работаете в окружении с единственным сервером NIS и только с машинами FreeBSD. Так как в домене test-domain имеется также и вторичный сервер, то вы должны отредактировать файл /var/yp/Makefile:

```
ellington# vi /var/yp/Makefile
```

Вы должны закомментировать строку, в которой указано

```
NOPUSH = "True"
```

(она уже не раскомментирована).

24.4.4.2.3. Настройка вторичного сервера NIS

Настройка вторичного сервера NIS осуществляется ещё проще, чем настройка главного сервера. Войдите на вторичный сервер и отредактируйте файл /etc/rc.conf точно также, как вы делали это ранее. Единственным отличием является то, что при запуске программы ypinit мы теперь должны использовать опцию -s. Применение опции -s требует также указание имени главного сервера NIS, так что наша команда должна выглядеть так:

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n

Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
```

```
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
Don't forget to update map ypservers on ellington.
```

Теперь у вас должен быть каталог с именем /var/yp/test-domain. Копии карт главного сервера NIS должны быть в этом каталоге. Вы должны удостовериться, что этот каталог обновляется. Следующие строки в /etc/crontab вашего вторичного сервера должны это делать:

```
20  *   *   *   *   root   /usr/libexec/ypxfr passwd.byname
21  *   *   *   *   root   /usr/libexec/ypxfr passwd.byuid
```

Эти две строки заставляют вторичный сервер синхронизировать свои карты с картами главного сервера. Эти строки не являются обязательными, так как главный сервер

автоматически пытается передать вторичным серверам все изменения в своих картах NIS. Однако, учитывая важность информации о паролях для клиентов, зависящих от вторичного сервера, рекомендуется выполнять частые обновления карт с паролями. Это особенно важно в загруженных сетях, в которых обновления карт могут не всегда завершаться успешно.

А теперь точно также запустите команду `/etc/netstart` на вторичном сервере, по которой снова выполнится запуск сервера NIS.

24.4.4.3. Клиенты NIS

Клиент NIS выполняет так называемую привязку к конкретному серверу NIS при помощи даемона `ypbind`. `ypbind` определяет домен, используемый в системе по умолчанию (тот, который устанавливается по команде `domainname`), и начинает широковещательную рассылку запросов RPC в локальной сети. В этих запросах указано имя домена, к серверу которого `ypbind` пытается осуществить привязку. Если сервер, который был настроен для обслуживания запрашиваемого домена, получит широковещательный запрос, он ответит `ypbind`, который, в свою очередь запомнит адрес сервера. Если имеется несколько серверов (например, главный и несколько вторичных), то `ypbind` будет использовать адрес первого ответившего. С этого момента клиентская система будет направлять все свои запросы NIS на этот сервер. Время от времени `ypbind` будет "пинать" сервер для проверки его работоспособности. Если на один из тестовых пакетов не удастся получить ответа за разумное время, то `ypbind` пометит этот домен как домен, с которым связка разорвана, и снова начнет процесс посылки широковещательных запросов в надежде найти другой сервер.

24.4.4.3.1. Настройка клиента NIS

Настройка машины с FreeBSD в качестве клиента NIS достаточно проста.

1. Отредактируйте файл /etc/rc.conf, добавив туда следующие строки для того, чтобы задать имя домена NIS и запустить `ypbind` во время запуска сетевых служб:

```
nisdomainname="test-domain"
nis_client_enable="YES"
```

2. Для импортирования всех возможных учётных записей от сервера NIS, удалите все записи пользователей из вашего файла /etc/master.passwd и воспользуйтесь командой `vipw` для добавления следующей строки в конец файла:

```
+::::::::::
```

> Эта строчка даст всем пользователям с корректной учетной записью в картах учетных баз пользователей доступ к этой системе. Есть множество способов настроить ваш клиент NIS, изменив эту строку. Посмотрите ниже текст, касающийся сетевых групп, чтобы получить

более подробную информацию. Дополнительная информация для изучения находится в книге издательства O'Reilly под названием `Managing NFS and NIS`.

Вы должны оставить хотя бы одну локальную запись (то есть не импортировать ее через NIS) в вашем /etc/master.passwd и эта запись должна быть также членом группы `wheel`. Если с NIS что-то случится, эта запись может использоваться для удаленного входа в систему, перехода в режим администратора и исправления неисправностей.

3. Для импортирования всех возможных записей о группах с сервера NIS, добавьте в ваш файл /etc/group такую строчку:

```
+:*::
```

Для немедленного запуска клиента NIS выполните следующую команду с правами пользователя `root`:

```
# /etc/netstart
# /etc/rc.d/ypbind start
```

После завершения выполнения этих шагов у вас должно получиться запустить команду `ypcat passwd` и увидеть карту учетных записей сервера NIS.

24.4.5. Безопасность NIS

В общем-то любой пользователь, зная имя вашего домена, может выполнить запрос RPC к ypserv(8) и получить содержимое ваших карт NIS. Для предотвращения такого неавторизованного обмена ypserv(8) поддерживает так называемую систему "securenets", которая может использоваться для ограничения доступа к некоторой группе хостов. При запуске ypserv(8) будет пытаться загрузить информацию, касающуюся securenets, из файла /var/yp/securenets.

Имя каталога зависит от параметра, указанного вместе с опцией `-p`. Этот файл содержит записи, состоящие из указания сети и сетевой маски, разделенных пробелом. Строчки, начинающиеся со знака "#", считаются комментариями. Примерный файл securenets может иметь примерно такой вид:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
```

```
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0      255.255.240.0
```

Если ypserv(8) получает запрос от адреса, который соответствует одному из этих правил, он будет отрабатывать запрос обычным образом. Если же адрес не подпадает ни под одно правило, запрос будет проигнорирован и в журнал будет записано предупреждающее сообщение. Если файл /var/yp/securenets не существует, `ypserv` будет обслуживать соединения от любого хоста.

Программа `ypserv` также поддерживает пакет программ TCP Wrapper от Wietse Venema. Это позволяет администратору для ограничения доступа вместо /var/yp/securenets использовать конфигурационные файлы TCP Wrapper.

Хотя оба этих метода управления доступом обеспечивают некоторую безопасность, они, как основанные на проверке привилегированного порта, оба подвержены атакам типа "IP spoofing". Весь сетевой трафик, связанный с работой NIS, должен блокироваться вашим брандмауэром.

Серверы, использующие файл /var/yp/securenets, могут быть не в состоянии обслуживать старых клиентов NIS с древней реализацией протокола TCP/IP. Некоторые из этих реализаций при рассылке широковещательных запросов устанавливают все биты машинной части адреса в ноль и/или не в состоянии определить маску подсети при вычислении адреса широковещательной рассылки. Хотя некоторые из этих проблем могут быть решены изменением конфигурации клиента, другие могут привести к отказу от использования /var/yp/securenets.

Использование /var/yp/securenets на сервере с такой архаичной реализацией TCP/IP является весьма плохой идеей, и приведёт к потере работоспособности NIS в большой части вашей сети.

Использование пакета TCP Wrapper увеличит время отклика вашего сервера NIS. Дополнительной задержки может оказаться достаточно для возникновения тайм-аутов в клиентских программах, особенно в загруженных сетях или с медленными серверами NIS. Если одна или более ваших клиентских систем страдают от таких проблем, вы должны преобразовать такие клиентские системы во вторичные серверы NIS и сделать принудительную их привязку к самим себе.

24.4.6. Запрет входа некоторых пользователей

В нашей лаборатории есть машина `basie`, о которой предполагается, что она является исключительно факультетской рабочей станцией. Мы не хотим исключать эту машину из домена NIS, однако файл passwd на главном сервере NIS содержит учетные записи как для работников факультета, так и студентов. Что мы можем сделать?

Есть способ ограничить вход некоторых пользователей на этой машине, даже если они

присутствуют в базе данных NIS. Чтобы это сделать, вам достаточно добавить `-username` в конец файла /etc/master.passwd на клиентской машине, где *username* является именем пользователя, которому вы хотите запретить вход. Рекомендуется сделать это с помощью утилиты `vipw`, так как `vipw` проверит ваши изменения в /etc/master.passwd, а также автоматически перестроит базу данных паролей по окончании редактирования. Например, если мы хотим запретить пользователю `bill` осуществлять вход на машине `basie`, то мы сделаем следующее:

```
basie# vipw
[add -bill to the end, exit]
vipw: rebuilding the database...
vipw: done

basie# cat /etc/master.passwd

root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5::0:0:System &:/:/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/shared/man:/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/sbin/nologin
+::::::::
-bill

basie#
```

24.4.7. Использование сетевых групп

Способ, описанный в предыдущем разделе, работает достаточно хорошо, если вам нужны особые правила для очень малой группы пользователей или машин. В более крупных сетях вы *забудете* о запрете входа определенных пользователей на важные машины или даже будете настраивать каждую машину по отдельности, теряя таким образом главное преимущество использования NIS: *централизованное* администрирование.

Ответом разработчиков NIS на эту проблему являются *сетевые группы*. Их назначение и смысл можно сравнить с обычными группами, используемыми в файловых системах UNIX®. Главное отличие заключается в отсутствии числового идентификатора и возможности задать сетевую группу включением как пользователей, так и других сетевых групп.

Сетевые группы были разработаны для работы с большими, сложными сетями с сотнями пользователей и машин. С одной стороны, хорошо, если вам приходится с такой ситуацией. С другой стороны, эта сложность делает невозможным описание сетевых групп с помощью простых примеров. Пример, используемый в дальнейшем, демонстрирует эту проблему.

Давайте предположим, что успешное внедрение системы NIS в вашей лаборатории заинтересовало ваше руководство. Вашим следующим заданием стало расширение домена NIS для включения в него некоторых других машин студенческого городка. В двух таблицах перечислены имена новых машин и пользователей, а также их краткое описание.

Имена пользователей	Описание
`alpha`, `beta`	Обычные служащие IT-департамента
`charlie`, `delta`	Практиканты IT-департамента
`echo`, `foxtrott`, `golf`, ...	Обычные сотрудники
`able`, `baker`, ...	Проходящие интернатуру

Имена машин	Описание
`war`, `death`, `famine`, `pollution`	Ваши самые важные серверы. Только служащим IT позволяется входить на эти машины.
`pride`, `greed`, `envy`, `wrath`, `lust`, `sloth`	Менее важные серверы. Все сотрудники департамента IT могут входить на эти машины.
`one`, `two`, `three`, `four`, ...	Обычные рабочие станции. Только *реально нанятым* служащим позволяется использовать эти машины.
`trashcan`	Очень старая машина без каких-либо критичных данных. Даже проходящим интернатуру разрешено ее использовать.

Если вы попытаетесь реализовать эти требования, ограничивая каждого пользователя по отдельности, то вам придется добавить на каждой машине в файл passwd по одной строчке `-user` для каждого пользователя, которому запрещено входить на эту систему. Если вы забудете даже одну строчку, у вас могут начаться проблемы. Гораздо проще делать это правильно во время начальной установки, однако вы постепенно *будете забывать* добавлять строчки для новых пользователей во время повседневной работы. В конце концов, Мерфи был оптимистом.

Использование в этой ситуации сетевых групп дает несколько преимуществ. Нет необходимости описывать по отдельности каждого пользователя; вы ставите в соответствие пользователю одну или несколько сетевых групп и разрешаете или запрещаете вход всем членам сетевой группы. Если вы добавляете новую машину, вам достаточно определить ограничения на вход для сетевых групп. Если добавляется новый пользователь, вам достаточно добавить его к одной или большему числу сетевых групп. Эти изменения независимы друг от друга: нет больше комбинаций "для каждого пользователя и

машины". Если настройка вашей системы NIS тщательно спланирована, то для разрешения или запрещения доступа к машинам вам нужно будет модифицировать единственный конфигурационный файл.

Первым шагом является инициализация карты NIS по имени netgroup. Программа ypinit(8) во FreeBSD по умолчанию этой карты не создаёт, хотя реализация NIS будет её поддерживает, как только она будет создана. Чтобы создать пустую карту, просто наберите

```
ellington# vi /var/yp/netgroup
```

и начните добавлять содержимое. Например, нам нужно по крайней мере четыре сетевых группы: сотрудники IT, практиканты IT, обычные сотрудники и интернатура.

```
IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain) (,foxtrott,test-domain) \
    (,golf,test-domain)
INTERNS (,able,test-domain) (,baker,test-domain)
```

`IT_EMP`, `IT_APP` и так далее являются именами сетевых групп. Несколько слов в скобках служат для добавления пользователей в группу. Три поля внутри группы обозначают следующее:

1. Имя хоста или хостов, к которым применимы последующие записи. Если имя хоста не указано, то запись применяется ко всем хостам. Если же указывается имя хоста, то вы получите мир темноты, ужаса и страшной путаницы.
2. Имя учетной записи, которая принадлежит этой сетевой группе.
3. Домен NIS для учетной записи. Вы можете импортировать в вашу сетевую группу учетные записи из других доменов NIS, если вы один из тех несчастных, имеющих более одного домена NIS.

Каждое из этих полей может содержать шаблоны, подробности даны в странице справочника по netgroup(5).

Не нужно использовать имена сетевых групп длиннее 8 символов, особенно если в вашем домене NIS имеются машины, работающие под управлением других операционных систем. Имена чувствительны к регистру; использование заглавных букв для имен сетевых групп облегчает распознавание пользователей, имен машин и сетевых групп.

Некоторые клиенты NIS (отличные от FreeBSD) не могут работать с сетевыми группами, включающими большое количество записей. Например, в некоторых старых версиях SunOS™ возникают проблемы, если сетевая группа содержит более 15 *записей*. Вы можете обойти это ограничение, создав несколько подгрупп с 15 или меньшим количеством пользователей и настоящую сетевую группу, состоящую из подгрупп:

```
BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) [...]
BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
BIGGRP3  (,joe31,domain)  (,joe32,domain)
BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
```

Вы можете повторить этот процесс, если вам нужно иметь более 225 пользователей в одной сетевой группе.

Активация и распространение вашей карты NIS проста:

```
ellington# cd /var/yp
ellington# make
```

Это приведет к созданию трех карт NIS netgroup, netgroup.byhost и netgroup.byuser. Воспользуйтесь утилитой ypcat(1) для проверки доступности ваших новых карт NIS:

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

Вывод первой команды должен соответствовать содержимому файла /var/yp/netgroup. Вторая команда не выведет ничего, если вы не зададите сетевые группы, специфичные для хоста. Третья команда может использоваться пользователем для получения списка сетевых групп.

Настройка клиента достаточно проста. Чтобы настроить сервер `war`, вам достаточно запустить vipw(8) и заменить строку

```
+:::::::::
```

на

```
+@IT_EMP:::::::::
```

Теперь только данные, касающиеся пользователей, определенных в сетевой группе `IT_EMP`, импортируются в базу паролей машины `war` и только этим пользователям будет разрешен вход.

К сожалению, это ограничение также касается и функции `~` командного процессора и всех подпрограмм, выполняющих преобразование между именами пользователей и их числовыми ID. Другими словами, команда `cd ~user` работать не будет, команда `ls -l` будет выдавать числовые идентификаторы вместо имён пользователей, а `find . -user joe -print` работать откажется, выдавая сообщение `No such user`. Чтобы это исправить, вам нужно будет выполнить импорт всех записей о пользователях *без разрешения на вход на ваши*

серверы.

Это можно сделать, добавив еще одну строку в файл /etc/master.passwd. Эта строка должна содержать:

`+:::::::::/sbin/nologin`, что означает "Произвести импортирование всех записей с заменой командного процессора на /sbin/nologin в импортируемых записях". Вы можете заменить любое поле в строке с паролем, указав значение по умолчанию в вашем /etc/master.passwd.

Проверьте, что строка `:::::::::/sbin/nologin` помещена после `@IT_EMP:::::::::`. В противном случае все пользовательские записи, импортированные из NIS, будут иметь /sbin/nologin в качестве оболочки.

После этого изменения при появлении нового сотрудника IT вам будет достаточно изменять только одну карту NIS. Вы можете применить подобный метод для менее важных серверов, заменяя старую строку `+:::::::::` в их файлах /etc/master.passwd на нечто, подобное следующему:

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+:::::::::/sbin/nologin
```

Соответствующие строки для обычных рабочих станций могут иметь такой вид:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+:::::::::/sbin/nologin
```

И все было прекрасно до того момента, когда через несколько недель изменилась политика: Департамент IT начал нанимать интернатуру. Интернатуре в IT позволили использовать обычные рабочие станции и менее важные серверы; практикантам позволили входить на главные серверы. Вы создали новую сетевую группу `IT_INTERN`, добавили в нее новую интернатуру и начали изменять настройки на всех и каждой машине... Как говорит старая мудрость: "Ошибки в централизованном планировании приводят к глобальному хаосу".

Возможность в NIS создавать сетевые группы из других сетевых групп может использоваться для предотвращения подобных ситуаций. Одним из вариантов является создание сетевых групп на основе ролей. Например, вы можете создать сетевую группу с именем `BIGSRV` для задания ограничений на вход на важные серверы, другую сетевую группу с именем `SMALLSRV` для менее важных серверов и третью сетевую группу под названием `USERBOX` для обычных рабочих станций. Каждая из этих сетевых групп содержит сетевые группы, которым позволено входить на эти машины. Новые записи для вашей карты NIS сетевой группы должны выглядеть таким образом:

```
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
```

```
USERBOX   IT_EMP  ITINTERN USERS
```

Этот метод задания ограничений на вход работает весьма хорошо, если вы можете выделить группы машин с одинаковыми ограничениями. К сожалению, такая ситуация может быть исключением, но не правилом. В большинстве случаев вам нужна возможность определять ограничения на вход индивидуально для каждой машины.

Задание сетевых групп в зависимости от машин является другой возможностью, которой можно воспользоваться при изменении политики, описанной выше. При таком развитии событий файл /etc/master.passwd на каждой машине содержит две строки, начинающиеся с "+". Первая из них добавляет сетевую группу с учётными записями, которым разрешено входить на эту машину, а вторая добавляет все оставшиеся учетные записи с /sbin/nologin в качестве командного процессора. Хорошей идеей является использование "ИМЕНИ МАШИНЫ" заглавными буквами для имени сетевой группы. Другими словами, строки должны иметь такой вид:

```
+@BOXNAME:::::::::
+::::::::::/sbin/nologin
```

Как только вы завершите эту работу для всех ваших машин, вам не нужно будет снова модифицировать локальные версии /etc/master.passwd. Все будущие изменения могут быть выполнены изменением карты NIS. Вот пример возможной карты сетевой группы для этого случая с некоторыми полезными дополнениями:

```
# Сначала определяем группы пользователей
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)
#
# Теперь задаем несколько групп на основе ролей
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# И группы для специальных задач
# Открыть пользователям echo и golf доступ к антивирусной машине
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)
#
# Сетевые группы, специфичные для машин
# Наши главные серверы
WAR   BIGSRV
FAMINE    BIGSRV
# Пользователю india необходим доступ к этому серверу
```

```
POLLUTION  BIGSRV  (,india,test-domain)
#
# Этот очень важен и ему требуются большие ограничения доступа
DEATH     IT_EMP
#
# Антивирусная машина, упомянутая выше
ONE   SECURITY
#
# Ограничить машину единственным пользователем
TWO   (,hotel,test-domain)
# [...далее следуют другие группы]
```

Если вы используете какие-либо базы данных для управления учетными записями ваших пользователей, вы должны смочь создать первую часть карты с помощью инструментов построения отчетов вашей базы данных. В таком случае новые пользователи автоматически получат доступ к машинам.

И последнее замечание: Не всегда бывает разумно использовать сетевые группы на основе машин. Если в студенческих лабораториях вы используете несколько десятков или даже сотен одинаковых машин, то вам нужно использовать сетевые группы на основе ролей, а не основе машин, для того, чтобы размеры карты NIS оставались в разумных пределах.

24.4.8. Важные замечания

Есть некоторые действия, которые нужно будет выполнять по-другому, если вы работаете с NIS.

- Каждый раз, когда вы собираетесь добавить пользователя в лаборатории, вы должны добавить его *только* на главном сервере NIS и *обязательно перестроить карты NIS*. Если вы забудете сделать это, то новый пользователь не сможет нигде войти, кроме как на главном сервере NIS. Например, если в лаборатории нам нужно добавить нового пользователя `jsmith`, мы делаем вот что:

  ```
  # pw useradd jsmith
  # cd /var/yp
  # make test-domain
  ```

  Вместо `pw useradd jsmith` вы можете также запустить команду `adduser jsmith`.

- *Не помещайте административные учетные записи в карты NIS*. Вам не нужно распространять административных пользователей и их пароли на машины, которые не должны иметь доступ к таким учётным записям.
- *Сделайте главный и вторичные серверы NIS безопасными и минимизируйте их время простоя*. Если кто-то либо взломает, либо просто отключит эти машины, то люди без права входа в лабораторию с легкостью получат доступ.

  Это основное уязвимое место в любой централизованно администрируемой системе. Если вы не защищаете ваши серверы NIS, вы будете иметь дело с толпой разозлённых

пользователей!

24.4.9. Совместимость с NIS v1

ypserv из поставки FreeBSD имеет встроенную поддержку для обслуживания клиентов NIS v1. Реализация NIS во FreeBSD использует только протокол NIS v2, хотя другие реализации имеют поддержку протокола v1 для совместимости со старыми системами. Даемоны ypbind, поставляемые с такими системами, будут пытаться осуществить привязку к серверу NIS v1, даже если это им не нужно (и они будут постоянно рассылать широковещательные запросы в поиске такого сервера даже после получения ответа от сервера v2). Отметьте, что хотя имеется поддержка обычных клиентских вызовов, эта версия ypserv не отрабатывает запросы на передачу карт v1; следовательно, она не может использоваться в качестве главного или вторичного серверов вместе с другими серверами NIS, поддерживающими только протокол v1. К счастью, скорее всего, в настоящий момент такие серверы практически не используются.

24.4.10. Серверы NIS, которые также являются клиентами NIS

Особое внимание следует уделить использованию ypserv в домене со многими серверами, когда серверные машины являются также клиентами NIS. Неплохо бы заставить серверы осуществить привязку к самим себе, запретив рассылку запросов на привязку и возможно, перекрестную привязку друг к другу. Если один сервер выйдет из строя, а другие будут зависеть от него, то в результате могут возникнуть странные ситуации. Постепенно все клиенты попадут в тайм-аут и попытаются привязаться к другим серверам, но полученная задержка может быть значительной, а странности останутся, так как серверы снова могут привязаться друг к другу.

Вы можете заставить хост выполнить привязку к конкретному серверу, запустив команду `ypbind` с флагом `-S`. Если вы не хотите делать это вручную каждый раз при перезагрузке вашего сервера NIS, то можете добавить в файл /etc/rc.conf такие строки:

```
nis_client_enable="YES"   # run client stuff as well
nis_client_flags="-S NIS domain,server"
```

Дополнительную информацию можно найти на странице справки по ypbind(8).

24.4.11. Форматы паролей

Одним из общих вопросов, которые возникают в начале работы с NIS, является вопрос совместимости форматов паролей. Если ваш сервер NIS использует пароли, зашифрованные алгоритмом DES, то он будет поддерживать только тех клиентов, что также используют DES. К примеру, если в вашей сети имеются клиенты NIS, использующие Solaris™, то вам, скорее всего, необходимо использовать пароли с шифрованием по алгоритму DES.

Чтобы понять, какой формат используют ваши серверы и клиенты, загляните в файл /etc/login.conf. Если хост настроен на использование паролей, зашифрованных по алгоритму DES, то класс `default` будет содержать запись вроде следующей:

```
default:\
    :passwd_format=des:\
    :copyright=/etc/COPYRIGHT:\
    [Последующие строки опущены]
```

Другими возможными значениями для `passwd_format` являются `blf` и `md5` (для паролей, шифруемых по стандартам Blowfish и MD5 соответственно).

Если вы внесли изменения в файл /etc/login.conf, то вам также нужно перестроить базу данных параметров входа в систему, что достигается запуском следующей команды пользователем `root`:

```
# cap_mkdb /etc/login.conf
```

Формат паролей, которые уже находятся в файле /etc/master.passwd, не будет изменён до тех пор, пока пользователь не сменит свой пароль *после* перестроения базы данных параметров входа в систему.

После этого, чтобы удостовериться в том, что пароли зашифрованы в том формате, который выбран вами, нужно проверить, что строка `crypt_default` в /etc/auth.conf указывает предпочтение выбранного вами формата паролей. Для этого поместите выбранный формат первым в списке. Например, при использовании DES-шифрования паролей строка будет выглядеть так:

```
crypt_default   =   des blf md5
```

Выполнив вышеперечисленные шаги на каждом из серверов и клиентов NIS, работающих на FreeBSD, вы можете обеспечить их согласованность относительно используемого в вашей сети формата паролей. Если у вас возникли проблемы с аутентификацией клиента NIS, начать её решать определённо стоит отсюда. Запомните: если вы хотите использовать сервер NIS в гетерогенной сети, вам, наверное, нужно будет использовать DES на всех системах в силу того, что это минимальный общий стандарт.

24.5. Автоматическая настройка сети (DHCP)

24.5.1. Что такое DHCP?

DHCP, или Dynamic Host Configuration Protocol (Протокол Динамической Конфигурации Хостов), описывает порядок, по которому система может подключиться к сети и получить необходимую информацию для работы в ней. Во FreeBSD используется `dhclient`, импортированный из OpenBSD 3.7. Вся информация здесь, относительно `dhclient` относится либо к ISC, либо к DHCP клиентам. DHCP сервер включён в ISC дистрибутив.

24.5.2. Что описывается в этом разделе

В этом разделе описываются, как компоненты клиентской части ISC или OpenBSD DHCP клиента, так и компоненты ISC DHCP системы со стороны сервера. Программа, работающая на клиентской стороне, `dhclient`, интегрирована в поставку FreeBSD, а серверная часть доступна в виде порта net/isc-dhcp42-server. Кроме ссылок ниже, много полезной информации находится на страницах справочной системы, описывающих dhclient(8), dhcp-options(5) и dhclient.conf(5).

24.5.3. Как это работает

Когда на клиентской машине выполняется программа `dhclient`, являющаяся клиентом DHCP, она начинает широковещательную рассылку запросов на получение настроечной информации. По умолчанию эти запросы делаются на 68 порт UDP. Сервер отвечает на UDP 67, выдавая клиенту адрес IP и другую необходимую информацию, такую, как сетевую маску, маршрутизатор и серверы DNS. Вся эта информация даётся в форме "аренды" DHCP и верна только определенное время (что настраивается администратором сервера DHCP). При таком подходе устаревшие адреса IP тех клиентов, которые больше не подключены к сети, могут автоматически использоваться повторно.

Клиенты DHCP могут получить от сервера очень много информации. Подробный список находится в странице Справочника dhcp-options(5).

24.5.4. Интеграция с FreeBSD

DHCP клиент от OpenBSD, `dhclient`, полностью интегрирован во FreeBSD. Поддержка клиента DHCP есть как в программе установки, так и в самой системе, что исключает необходимость в знании подробностей конфигурации сети в любой сети, имеющей сервер DHCP.

DHCP поддерживается утилитой sysinstall. При настройке сетевого интерфейса из программы sysinstall второй вопрос, который вам задается: "Do you want to try DHCP configuration of the interface?" ("Хотите ли вы попробовать настроить этот интерфейс через DHCP?"). Утвердительный ответ приведёт к запуску программы `dhclient`, и при удачном его выполнении к автоматическому заданию информации для настройки интерфейса.

Есть две вещи, которые вы должны сделать для того, чтобы ваша система использовала DHCP при загрузке:

- Убедитесь, что устройство bpf включено в компиляцию вашего ядра. Чтобы это сделать, добавьте строчку `device bpf` в конфигурационный файл ядра и перестройте ядро. Более подробная информация о построении ядер имеется в Настройка ядра FreeBSD.

  Устройство bpf уже является частью ядра GENERIC, которое поставляется вместе с FreeBSD, так что, если вы не используете другое ядро, то вам и не нужно его делать для того, чтобы работал DHCP.

  > Те, кто беспокоится о безопасности, должны иметь в виду, что устройство bpf является также тем самым устройством, которое позволяет работать программам-снифферам пакетов (хотя для этого они должны быть

> запущены пользователем `root`). Наличие устройства bpf*необходимо* для использования DHCP, но если вы чересчур беспокоитесь о безопасности, то вам нельзя добавлять устройство bpf в ядро только для того, чтобы в неопределённом будущем использовать DHCP.

- По умолчанию, конфигурирование FreeBSD по протоколу DHCP выполняется фоновым процессом, или *асинхронно*. Остальные стартовые скрипты продолжают работу не ожидая завершения процесса конфигурирования, тем самым ускоряя загрузку системы.

  Фоновое конфигурирование не создает проблем в случае, если сервер DHCP быстро отвечает на запросы, и процесс конфигурирования происходит быстро. Однако, в некоторых случаях настройка по DHCP может длиться значительное время. При этом запуск сетевых сервисов может потерпеть неудачу, если будет произведен ранее завершения конфигурирования по DHCP. Запуск DHCP в *синхронном* режиме предотвращает проблему, откладывая выполнение остальных стартовых скриптов до момента завершения конфигурирования по DHCP.

  Для осуществления фонового конфигурирования по DHCP (асинхронный режим), используйте значение “DHCP” в /etc/rc.conf:

  ```
  ifconfig_fxp0="DHCP"
  ```

  Для откладывания запуска стартовых скриптов до завершения конфигурирования по DHCP (синхронный режим), укажите значение “SYNCDHCP”:

  ```
  ifconfig_fxp0="SYNCDHCP"
  ```

  > Замените используемое в этих примерах имя *fxp0* на имя интерфейса, который необходимо сконфигурировать динамически, как это описано в Настройка карт сетевых интерфейсов.

  Если `dhclient` в вашей системе находится в другом месте или если вы хотите задать дополнительные параметры для `dhclient`, то также укажите следующее (изменив так, как вам нужно):

  ```
  dhclient_program="/sbin/dhclient"
  dhclient_flags=""
  ```

Сервер DHCP, dhcpd, включён как часть порта net/isc-dhcp42-server в коллекцию портов. Этот порт содержит DHCP-сервер от ISC и документацию.

24.5.5. Файлы

- /etc/dhclient.conf

  `dhclient` требует наличия конфигурационного файла, /etc/dhclient.conf. Как правило, файл

содержит только комментарии, а настройки по умолчанию достаточно хороши. Этот настроечный файл описан на страницах справочной системы по dhclient.conf(5).

- /sbin/dhclient

  `dhclient` скомпонован статически и находится в каталоге /sbin. На страница Справочника dhclient(8) дается более подробная информация о `dhclient`.

- /sbin/dhclient-script

  `dhclient-script` является специфичным для FreeBSD скриптом настройки клиента DHCP. Он описан в dhclient-script(8), но для нормального функционирования никаких модификаций со стороны пользователя не требуется.

- /var/db/dhclient.leases

  В этом файле клиент DHCP хранит базу данных выданных к использованию адресов в виде журнала. На странице dhclient.leases(5) дается гораздо более подробное описание.

24.5.6. Дополнительная литература

Полное описание протокола DHCP дается в RFC 2131. Кроме того, дополнительная информация есть на сервере http://www.dhcp.org/.

24.5.7. Установка и настройка сервера DHCP

24.5.7.1. Чему посвящён этот раздел

Этот раздел даёт информацию о том, как настроить систему FreeBSD для работы в качестве сервера DHCP на основе реализации пакета DHCP от ISC (Internet Systems Consortium).

Серверная часть пакета не поставляется как часть FreeBSD, так что вам потребуется установить порт net/isc-dhcp42-server для получения этого сервиса. Обратитесь к Установка приложений. порты и пакеты для получения более полной информации об использовании коллекции портов.

24.5.7.2. Установка сервера DHCP

Для того, чтобы настроить систему FreeBSD на работу в качестве сервера DHCP, вам необходимо обеспечить присутствие устройства bpf(4), вкомпилированного в ядро. Для этого добавьте строку `device bpf` в файл конфигурации вашего ядра. Для получения более полной информации о построении ядер, обратитесь к Настройка ядра FreeBSD.

Устройство bpf уже входит в состав ядра GENERIC, поставляемого с FreeBSD, так что вам не нужно создавать собственное ядро для обеспечения работы DHCP.

Те, кто обращает особое внимание на вопросы безопасности, должны заметить, что bpf является тем устройством, что позволяет нормально работать снифферам пакетов (хотя таким программам требуются привилегированный доступ). Наличие устройства bpf*обязательно* для

> использования DHCP, но если вы очень обеспокоены безопасностью, наверное, вам не нужно включать bpf в ваше ядро только потому, что в отдалённом будущем вы собираетесь использовать DHCP.

Следующим действием, которое вам нужно выполнить, является редактирование примерного dhcpd.conf, который устанавливается в составе порта net/isc-dhcp42-server. По умолчанию это файл /usr/local/etc/dhcpd.conf.sample, и вы должны скопировать его в файл /usr/local/etc/dhcpd.conf перед тем, как его редактировать.

24.5.7.3. Настройка сервера DHCP

dhcpd.conf состоит из деклараций относительно подсетей и хостов, и проще всего описывается на примере:

```
option domain-name "example.com";①
option domain-name-servers 192.168.4.100;②
option subnet-mask 255.255.255.0;③

default-lease-time 3600;④
max-lease-time 86400;⑤
ddns-update-style none;⑥

subnet 192.168.4.0 netmask 255.255.255.0 {
  range 192.168.4.129 192.168.4.254;⑦
  option routers 192.168.4.1;⑧
}

host mailhost {
  hardware ethernet 02:03:04:05:06:07;⑨
  fixed-address mailhost.example.com;⑩
}
```

① Этот параметр задаёт домен, который будет выдаваться клиентам в качестве домена, используемого по умолчанию при поиске. Обратитесь к страницам справочной системы по resolv.conf(5) для получения дополнительной информации о том, что это значит.

② Этот параметр задаёт список разделённых запятыми серверов DNS, которые должен использовать клиент.

③ Маска сети, которая будет выдаваться клиентам.

④ Клиент может запросить определённое время, которое будет действовать выданная информация. В противном случае сервер выдаст настройки с этим сроком (в секундах).

⑤ Это максимальное время, на которое сервер будет выдавать конфигурацию. Если клиент запросит больший срок, он будет подтверждён, но будет действовать только `max-lease-time` секунд.

⑥ Этот параметр задаёт, будет ли сервер DHCP пытаться обновить DNS при выдаче или освобождении конфигурационной информации. В реализации ISC этот параметр является *обязательным*.

⑦ Это определение того, какие IP-адреса должны использоваться в качестве резерва для выдачи клиентам. IP-адреса между и включая границы, будут выдаваться клиентам.

⑧ Объявление маршрутизатора, используемого по умолчанию, который будет выдаваться клиентам.

⑨ Аппаратный MAC-адрес хоста (чтобы сервер DHCP мог распознать хост, когда тот делает запрос).

⑩ Определение того, что хосту всегда будет выдаваться один и тот же IP-адрес. Заметьте, что указание здесь имени хоста корректно, так как сервер DHCP будет разрешать имя хоста самостоятельно до того, как выдать конфигурационную информацию.

Когда вы закончите составлять свой dhcpd.conf, нужно разрешить запуск сервера DHCP в файле /etc/rc.conf, добавив в него строки

```
dhcpd_enable="YES"
dhcpd_ifaces="dc0"
```

Замените `dc0` именем интерфейса (или именами интерфейсов, разделяя их пробелами), на котором(ых) сервер DHCP должен принимать запросы от клиентов.

Затем вы можете стартовать сервер DHCP при помощи команды

```
# /usr/local/etc/rc.d/isc-dhcpd start
```

Если в будущем вам понадобится сделать изменения в настройке вашего сервера, то важно заметить, что посылка сигнала `SIGHUP` приложению dhcpd_не приведёт_ к перезагрузке настроек, как это бывает для большинства демонов. Вам нужно послать сигнал `SIGTERM` для остановки процесса, а затем перезапустить его при помощи вышеприведённой команды.

24.5.7.4. Файлы

- /usr/local/sbin/dhcpd

  dhcpd скомпонован статически и расположен в каталоге /usr/local/sbin. Страницы справочной системы dhcpd(8), устанавливаемые портом, содержат более полную информацию о dhcpd.

- /usr/local/etc/dhcpd.conf

  dhcpd требует наличия конфигурационного файла, /usr/local/etc/dhcpd.conf, до того, как он будет запущен и начнёт предоставлять сервис клиентам. Необходимо, чтобы этот файл содержал все данные, которая будет выдаваться обслуживаемым клиентам, а также информацию о работе сервера. Этот конфигурационный файл описывается на страницах справочной системы dhcpd.conf(5), которые устанавливаются портом.

- /var/db/dhcpd.leases

  Сервер DHCP ведёт базу данных выданной информации в этом файле, который

записывается в виде протокола. Страницы справочной системы dhcpd.leases(5), устанавливаемые портом, дают гораздо более подробное описание.

- /usr/local/sbin/dhcrelay

  dhcrelay используется в сложных ситуациях, когда сервер DHCP пересылает запросы от клиента другому серверу DHCP в отдельной сети. Если вам нужна такая функциональность, то установите порт net/isc-dhcp42-relay. На страницах справочной системы dhcrelay(8), которые устанавливаются портом, даётся более полное описание.

24.6. Domain Name System (DNS)

24.6.1. Обзор

По умолчанию во FreeBSD используется одна из версий программы BIND (Berkeley Internet Name Domain), являющейся самой распространенной реализацией протокола DNS. DNS - это протокол, при помощи которого имена преобразуются в IP-адреса и наоборот. Например, в ответ на запрос о `www.FreeBSD.org` будет получен IP-адрес веб-сервера Проекта FreeBSD, а запрос о `ftp.FreeBSD.org` возвратит IP-адрес соответствующей машины с FTP-сервером. Точно также происходит и обратный процесс. Запрос, содержащий IP-адрес машины, возвратит имя хоста. Для выполнения запросов к DNS вовсе не обязательно иметь в системе работающий сервер имён.

FreeBSD в настоящее время поставляется с сервером DNSBIND9, предоставляющим расширенные настройки безопасности, новую схему расположения файлов конфигурации и автоматические настройки для chroot(8).

В сети Интернет DNS управляется через достаточно сложную систему авторизированных корневых серверов имён, серверов доменов первого уровня (Top Level Domain, TLD) и других менее крупных серверов имён, которые содержат и кэшируют информацию о конкретных доменах.

На данный момент пакет BIND поддерживается Internet Systems Consortium https://www.isc.org/.

24.6.2. Используемая терминология

Для понимания этого документа нужно понимать значения некоторых терминов, связанных с работой DNS.

Термин	Определение
Прямой запрос к DNS (forward DNS)	Преобразование имён хостов в адреса IP
Ориджин (origin)	Обозначает домен, покрываемый конкретным файлом зоны
named, bind	Общеупотребительные названия для обозначения пакета BIND, обеспечивающего работу сервера имён во FreeBSD.

Термин	Определение
Резолвер	Системный процесс, посредством которого машина обращается к серверу имён для получения информации о зоне
Обратный DNS (reverse DNS)	Преобразование адресов IP в имена хостов
Корневая зона	Начало иерархии зон Интернет. Все зоны находятся под корневой зоной, подобно тому, как все файлы располагаются ниже корневого каталога.
Зона	Отдельный домен, поддомен или часть DNS, управляемая одним сервером.

Примеры зон:

- `.` - так обычно обозначается в документации корневая зона.
- `org.` - домен верхнего уровня (TLD) в корневой зоне.
- `example.org.` является зоной в домене верхнего уровня (TLD) `org.`.
- `1.168.192.in-addr.arpa` является зоной, в которую включены все IP-адреса, формирующие пространство адресов `192.168.1.*`.

Как можно видеть, уточняющая часть имени хоста появляется слева. Например, `example.org.` более точен, чем `org.`, также, как `org.` более точен, чем корневая зона. Расположение каждой части имени хоста сильно похоже на файловую систему: каталог /dev расположен в корневой файловой системе, и так далее.

24.6.3. Причины, по которым вам может понадобиться сервер имён

Сервера имён обычно используются в двух видах: авторитетный сервер имён и кэширующий сервер имён, также называемый распознавателем (resolver).

Авторитетный сервер имён нужен, когда:

- нужно предоставлять информацию о DNS остальному миру, отвечая на запросы авторизированно.
- зарегистрирован домен, такой, как `example.org` и в этом домене требуется поставить имена машин в соответствие с их адресами IP.
- блоку адресов IP требуется обратные записи DNS (IP в имена хостов).
- резервный (slave) сервер имён должен отвечать на запросы.

Кэширующий сервер имён нужен, когда:

- локальный сервер DNS может кэшировать информацию и отвечать на запросы быстрее, чем это происходит при прямом опросе внешнего сервера имён.

Например, когда кто-нибудь запрашивает информацию о `www.FreeBSD.org`, то обычно

резолвер обращается к серверу имён вашего провайдера, посылает запрос и ожидает ответа. С локальным кэширующим сервером DNS запрос во внешний мир будет делаться всего один раз. Последующие запросы не будут посылаться за пределы локальной сети, потому что информация уже имеется в кэше.

24.6.4. Как это работает

Во FreeBSD даемон BIND называется named.

Файл	Описание
named(8)	Даемон BIND
rndc(8)	Программа управления даемоном сервера имён
/etc/namedb	Каталог, в котором располагается вся информация о зонах BIND
/etc/namedb/named.conf	Конфигурационный файл для даемона

Файлы зон обычно располагаются в каталоге /etc/namedb и содержат информацию о зоне DNS, за которую отвечает сервер имён.

В зависимости от способа конфигурации зоны на сервере файлы зон могут располагаться в подкаталогах master, slave или dynamic иерархии /etc/namedb. Эти файлы содержат DNS информацию, которую и будет сообщать в ответ на запросы сервер имен.

24.6.5. Запуск BIND

Так как сервер имён BIND устанавливается по умолчанию, его настройка сравнительно проста.

Стандартная конфигурация named запускает простой кэширующий сервер в ограниченной среде chroot(8), который прослушивает запросы на интерфейсе обратной связи (loopback) с адресом (127.0.0.1). Для одноразового запуска даемона в этой конфигурации используйте команду

```
# /etc/rc.d/named onestart
```

Чтобы даемон named запускался во время загрузки, поместите в /etc/rc.conf следующую строку:

```
named_enable="YES"
```

Разумеется, существует множество различных конфигураций /etc/namedb/named.conf, лежащих за рамками данного документа. Разнообразные опции запуска named во FreeBSD описаны в переменных `named__*_` файла /etc/defaults/rc.conf и странице справочника rc.conf(5). Кроме того, полезной может оказаться Использование rc во FreeBSD 5.X и

последующих версиях.

24.6.6. Конфигурационные файлы

Файлы конфигурации даемона named расположены в каталоге /etc/namedb и, за исключением случая, когда вам требуется просто резолвер, требуют модификации.

24.6.6.1. /etc/namedb/named.conf

```
// $FreeBSD$
//
// If you are going to set up an authoritative server, make sure you
// understand the hairy details of how DNS works.  Even with
// simple mistakes, you can break connectivity for affected parties,
// or cause huge amounts of useless Internet traffic.

options {
    // All file and path names are relative to the chroot directory,
    // if any, and should be fully qualified.
    directory   "/etc/namedb/working";
    pid-file    "/var/run/named/pid";
    dump-file   "/var/dump/named_dump.db";
    statistics-file "/var/stats/named.stats";

// If named is being used only as a local resolver, this is a safe default.
// For named to be accessible to the network, comment this option, specify
// the proper IP address, or delete this option.
    listen-on   { 127.0.0.1; };

// If you have IPv6 enabled on this system, uncomment this option for
// use as a local resolver.  To give access to the network, specify
// an IPv6 address, or the keyword "any".
//  listen-on-v6    { ::1; };

// These zones are already covered by the empty zones listed below.
// If you remove the related empty zones below, comment these lines out.
    disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
    disable-empty-zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";
    disable-empty-zone
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";

// If you've got a DNS server around at your upstream provider, enter
// its IP address here, and enable the line below.  This will make you
// benefit from its cache, thus reduce overall DNS traffic in the Internet.
/*
    forwarders {
        127.0.0.1;
    };
*/
```

```
// If the 'forwarders' clause is not empty the default is to 'forward first'
// which will fall back to sending a query from your local server if the name
// servers in 'forwarders' do not have the answer.  Alternatively you can
// force your name server to never initiate queries of its own by enabling the
// following line:
//  forward only;

// If you wish to have forwarding configured automatically based on
// the entries in /etc/resolv.conf, uncomment the following line and
// set named_auto_forward=yes in /etc/rc.conf.  You can also enable
// named_auto_forward_only (the effect of which is described above).
//  include "/etc/namedb/auto_forward.conf";
```

Как и говорится в комментариях, если вы хотите получить эффект от использования кэша провайдера, то можно включить раздел `forwarders`. В обычном случае сервер имён будет рекурсивно опрашивать определённые серверы имён Интернет до тех пор, пока не получит ответ на свой запрос. При включении этого раздела он будет автоматически опрашивать сервер имён вашего провайдера (или тот, который здесь указан), используя преимущества его кэша. наличия нужной информации. Если соответствующий сервер имён провайдера работает быстро и имеет хороший канал связи, то в результате такой настройки вы можете получить хороший результат.

`127.0.0.1` здесь работать *не будет*. Измените его на IP-адрес сервера имён провайдера.

```
/*
       Modern versions of BIND use a random UDP port for each outgoing
       query by default in order to dramatically reduce the possibility
       of cache poisoning.  All users are strongly encouraged to utilize
       this feature, and to configure their firewalls to accommodate it.

       AS A LAST RESORT in order to get around a restrictive firewall
       policy you can try enabling the option below.  Use of this option
       will significantly reduce your ability to withstand cache poisoning
       attacks, and should be avoided if at all possible.

       Replace NNNNN in the example with a number between 49160 and 65530.
    */
    // query-source address * port NNNNN;
};

// If you enable a local name server, don't forget to enter 127.0.0.1
// first in your /etc/resolv.conf so this server will be queried.
// Also, make sure to enable it in /etc/rc.conf.

// The traditional root hints mechanism. Use this, OR the slave zones below.
zone "." { type hint; file "/etc/namedb/named.root"; };
```

```
/*  Slaving the following zones from the root name servers has some
    significant advantages:
    1. Faster local resolution for your users
    2. No spurious traffic will be sent from your network to the roots
    3. Greater resilience to any potential root server failure/DDoS

    On the other hand, this method requires more monitoring than the
    hints file to be sure that an unexpected failure mode has not
    incapacitated your server.  Name servers that are serving a lot
    of clients will benefit more from this approach than individual
    hosts.  Use with caution.

    To use this mechanism, uncomment the entries below, and comment
    the hint zone above.

    As documented at http://dns.icann.org/services/axfr/ these zones:
    "." (the root), ARPA, IN-ADDR.ARPA, IP6.ARPA, and ROOT-SERVERS.NET
    are available for AXFR from these servers on IPv4 and IPv6:
    xfr.lax.dns.icann.org, xfr.cjr.dns.icann.org
*/
/*
zone "." {
    type slave;
    file "/etc/namedb/slave/root.slave";
    masters {
        192.5.5.241;    // F.ROOT-SERVERS.NET.
    };
    notify no;
};

zone "arpa" {
    type slave;
    file "/etc/namedb/slave/arpa.slave";
    masters {
        192.5.5.241;    // F.ROOT-SERVERS.NET.
    };
    notify no;
};
*/

/*  Serving the following zones locally will prevent any queries
    for these zones leaving your network and going to the root
    name servers.  This has two significant advantages:
    1. Faster local resolution for your users
    2. No spurious traffic will be sent from your network to the roots
*/
// RFCs 1912 and 5735 (and BCP 32 for localhost)
zone "localhost"    { type master; file "/etc/namedb/master/localhost-forward.db"; };
zone "127.in-addr.arpa" { type master; file "/etc/namedb/master/localhost-reverse.db";
};
zone "255.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
```

```
// RFC 1912-style zone for IPv6 localhost address
zone "0.ip6.arpa"   { type master; file "/etc/namedb/master/localhost-reverse.db"; };

// "This" Network (RFCs 1912 and 5735)
zone "0.in-addr.arpa"   { type master; file "/etc/namedb/master/empty.db"; };

// Private Use Networks (RFCs 1918 and 5735)
zone "10.in-addr.arpa"     { type master; file "/etc/namedb/master/empty.db"; };
zone "16.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "17.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "18.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "19.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "20.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "21.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "22.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "23.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "24.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "25.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "26.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "27.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "28.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "29.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "30.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "31.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "168.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// Link-local/APIPA (RFCs 3927 and 5735)
zone "254.169.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IETF protocol assignments (RFCs 5735 and 5736)
zone "0.0.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// TEST-NET-[1-3] for Documentation (RFCs 5735 and 5737)
zone "2.0.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "100.51.198.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "113.0.203.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Range for Documentation (RFC 3849)
zone "8.b.d.0.1.0.0.2.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// Domain Names for Documentation and Testing (BCP 32)
zone "test" { type master; file "/etc/namedb/master/empty.db"; };
zone "example" { type master; file "/etc/namedb/master/empty.db"; };
zone "invalid" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.com" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.net" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.org" { type master; file "/etc/namedb/master/empty.db"; };

// Router Benchmark Testing (RFCs 2544 and 5735)
zone "18.198.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
```

```
zone "19.198.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IANA Reserved - Old Class E Space (RFC 5735)
zone "240.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "241.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "242.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "243.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "244.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "245.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "246.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "247.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "248.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "249.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "250.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "251.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "252.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "253.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "254.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Unassigned Addresses (RFC 4291)
zone "1.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "3.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "4.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "5.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "6.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "7.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "8.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "9.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "a.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "b.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "c.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "d.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "e.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "0.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "1.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "2.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "3.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "4.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "5.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "6.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "7.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "8.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "9.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "a.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "b.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "0.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "1.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "2.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "3.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "4.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "5.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
```

```
zone "6.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "7.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 ULA (RFC 4193)
zone "c.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "d.f.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Link Local (RFC 4291)
zone "8.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "9.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "a.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "b.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Deprecated Site-Local Addresses (RFC 3879)
zone "c.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "d.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "e.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };
zone "f.e.f.ip6.arpa"   { type master; file "/etc/namedb/master/empty.db"; };

// IP6.INT is Deprecated (RFC 4159)
zone "ip6.int"      { type master; file "/etc/namedb/master/empty.db"; };

// NB: Do not use the IP addresses below, they are faked, and only
// serve demonstration/documentation purposes!
//
// Example slave zone config entries.  It can be convenient to become
// a slave at least for the zone your own domain is in.  Ask
// your network administrator for the IP address of the responsible
// master name server.
//
// Do not forget to include the reverse lookup zone!
// This is named after the first bytes of the IP address, in reverse
// order, with ".IN-ADDR.ARPA" appended, or ".IP6.ARPA" for IPv6.
//
// Before starting to set up a master zone, make sure you fully
// understand how DNS and BIND work.  There are sometimes
// non-obvious pitfalls.  Setting up a slave zone is usually simpler.
//
// NB: Don't blindly enable the examples below. :-)  Use actual names
// and addresses instead.

/* An example dynamic zone
key "exampleorgkey" {
    algorithm hmac-md5;
    secret "sf87HJqjkqh8ac87a02lla==";
};
zone "example.org" {
    type master;
    allow-update {
        key "exampleorgkey";
    };
```

```
    file "dynamic/example.org";
};
*/

/* Example of a slave reverse zone
zone "1.168.192.in-addr.arpa" {
    type slave;
    file "/etc/namedb/slave/1.168.192.in-addr.arpa";
    masters {
        192.168.1.1;
    };
};
*/
```

Это примеры описаний прямой и обратной зон из файла named.conf для вторичных серверов.

Для каждого новой зоны, которую будет обслуживать сервер имён, в файл named.conf должна быть добавлена запись.

К примеру, самая простая запись для домена `example.org` может выглядеть вот так:

```
zone "example.org" {
    type master;
    file "master/example.org";
};
```

Зона является первичной, что отражается в поле `type`, и информация о зоне хранится в файле /etc/namedb/master/example.org, что указывается в поле `file`.

```
zone "example.org" {
    type slave;
    file "slave/example.org";
};
```

В случае вторичной зоны информация о ней передается с основного сервера имён для заданной зоны и сохраняется в указанном файле. Если и когда основной сервер имён выходит и строя или недосягаем, то скачанная информация о зоне будет находиться на вторичных серверах, и они смогут обслуживать эту зону.

24.6.6.2. Файлы зон

Пример файла зоны `example.org` для основного сервера (располагающийся в файле /etc/namedb/master/example.org) имеет такой вид:

```
$TTL 3600        ; 1 hour default TTL
example.org.    IN      SOA      ns1.example.org. admin.example.org. (
```

```
                                2006051501      ; Serial
                                10800           ; Refresh
                                3600            ; Retry
                                604800          ; Expire
                                300             ; Negative Response TTL
                        )

; DNS Servers
                IN      NS      ns1.example.org.
                IN      NS      ns2.example.org.

; MX Records
                IN      MX 10   mx.example.org.
                IN      MX 20   mail.example.org.

                IN      A       192.168.1.1

; Machine Names
localhost       IN      A       127.0.0.1
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5

; Aliases
www             IN      CNAME   example.org.
```

Заметьте, что все имена хостов, оканчивающиеся на ".", задают полное имя, тогда как все имена без символа "." на конце считаются заданными относительно origin. Например, `ns1` преобразуется в `ns1.example.org.`

Файл зоны имеет следующий формат:

```
recordname      IN recordtype  value
```

Наиболее часто используемые записи DNS:

SOA

начало зоны ответственности

NS

авторитативный сервер имен

A

адрес хоста

CNAME

каноническое имя для алиаса

MX

обмен почтой

PTR

указатель на доменное имя (используется в обратных зонах DNS)

```
example.org. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serial
                        10800           ; Refresh after 3 hours
                        3600            ; Retry after 1 hour
                        604800          ; Expire after 1 week
                        300 )           ; Negative Response TTL
```

`example.org.`

имя домена, а также ориджин для этого файла зоны.

`ns1.example.org.`

основной/авторитативный сервер имён для этой зоны.

`admin.example.org.`

человек, отвечающий за эту зону, адрес электронной почты с символом "@" замененным на точку. (admin@example.org становится `admin.example.org`)

`2006051501`

последовательный номер файла. При каждом изменении файла зоны это число должно увеличиваться. В настоящее время для нумерации многие администраторы предпочитают формат `ггггммддвв`. `2006051501` будет означать, что файл последний раз изменялся 15.05.2006, а последнее число `01` означает, что это была первая модификация файла за день. Последовательный номер важен, так как он служит для того, чтобы вторичные серверы узнавали об обновлении зоны.

```
        IN      NS      ns1.example.org.
```

Это NS-запись. Такие записи должны иметься для всех серверов имён, которые будут отвечать за зону.

```
localhost       IN      A       127.0.0.1
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5
```

Записи типа A служат для обозначения имён машин. Как это видно выше, имя `ns1.example.org` будет преобразовано в `192.168.1.2`.

```
        IN      A       192.168.1.1
```

Эта строка присваивает IP адрес `192.168.1.1` текущему ориджину, в данном случае домену `example.org`.

```
www      IN CNAME   @
```

Записи с каноническими именами обычно используются для присвоения машинам псевдонимов. В этом примере `www` является псевдонимом для "главной" машины, имя которой по воле случая совпало с именем домена `example.org` (`192.168.1.1`). Записи типа CNAME нельзя использовать совместно с другими типами записей для одного и того же имени хоста (recordname).

```
            IN MX   10      mail.example.org.
```

MX-запись указывает, какие почтовые серверы отвечают за обработку входящей электронной почты для зоны. `mail.example.org` является именем почтового сервера, а 10 обозначает приоритет этого почтового сервера.

Можно иметь несколько почтовых серверов с приоритетами, например, 10, 20 и так далее. Почтовый сервер, пытающийся доставить почту для `example.org`, сначала попробует связаться с машиной, имеющий MX-запись с самым большим приоритетом (наименьшим числовым значением в поле MX), затем с приоритетом поменьше и так далее, до тех пор, пока почта не будет отправлена.

Для файлов зон in-addr.arpa (обратные записи DNS) используется тот же самый формат, отличающийся только использованием записей PTR вместо A или CNAME.

```
$TTL 3600

1.168.192.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serial
                        10800           ; Refresh
                        3600            ; Retry
                        604800          ; Expire
                        300 )           ; Negative Response TTL

        IN      NS      ns1.example.org.
        IN      NS      ns2.example.org.

1       IN      PTR     example.org.
2       IN      PTR     ns1.example.org.
3       IN      PTR     ns2.example.org.
4       IN      PTR     mx.example.org.
5       IN      PTR     mail.example.org.
```

В этом файле дается полное соответствие имён хостов IP-адресам в нашем описанном ранее вымышленном домене.

Следует отметить, что все имена в правой части PTR-записи должны быть полными доменными именами (то есть, заканчиваться точкой ".").

24.6.7. Кэширующий сервер имён

Кэширующий сервер имён - это сервер имен, чья главная задача - разрешение рекурсивных запросов. Он просто выполняет запросы от своего имени и сохраняет результаты для последующего использования.

24.6.8. * DNSSEC

Этот раздел не переведен.

24.6.9. Безопасность

Хотя BIND является самой распространенной реализацией DNS, всегда стоит вопрос об обеспечении безопасности. Время от времени обнаруживаются возможные и реальные бреши в безопасности.

FreeBSD автоматически запускает named в ограниченном окружении (chroot(8)); помимо этого, есть еще несколько механизмов, помогающих защититься от возможных атак на сервис DNS.

Весьма полезно прочесть сообщения безопасности CERT и подписаться на Список рассылки FreeBSD, посвящённый срочным сообщениям, связанным с безопасностью для того, чтобы быть в курсе текущих проблем с обеспечением безопасности Internet и FreeBSD.

Если возникает проблема, то наличие последних исходных текстов и свежесобранного named может способствовать её решению.

24.6.10. Дополнительная литература

Справочная информация по BIND/named: rndc(8), named(8), named.conf(8), nsupdate(8), dnssec-signzone(8), dnssec-keygen(8)

- Официальная страница ISC BIND
- Официальный форум ISC BIND
- Книга издательства O'Reilly DNS and BIND 5th Edition
- Root DNSSEC
- DNSSEC Trust Anchor Publication for the Root Zone
- RFC1034 - Domain Names - Concepts and Facilities
- RFC1035 - Domain Names - Implementation and Specification
- RFC4033 - DNS Security Introduction and Requirements

- RFC4034 - Resource Records for the DNS Security Extensions
- RFC4035 - Protocol Modifications for the DNS Security Extensions
- RFC4641 - DNSSEC Operational Practices
- RFC5011 - Automated Updates of DNS Security (DNSSEC Trust Anchors

24.7. Apache HTTP сервер

24.7.1. Обзор

FreeBSD используется в качестве платформы для многих из самых нагруженных серверов в мире. Большинство серверов в интернет используют Apache HTTP сервер. Пакеты Apache должны быть включены в поставку FreeBSD. Если вы не установили их во вместе с системой, воспользуйтесь портами www/apache13 или www/apache22.

Как только Apache был успешно установлен, его необходимо настроить.

> В этом разделе рассказывается о версии 1.3.X Apache HTTP сервера, поскольку эта версия наиболее широко используется в FreeBSD. Apache 2.X содержит много новых технологий, но здесь они не обсуждаются. За дополнительной информацией о Apache 2.X, обращайтесь к http://httpd.apache.org/.

24.7.2. Настройка

В FreeBSD основной файл настройки Apache HTTP сервера устанавливается в /usr/local/etc/apache/httpd.conf. Это обычный текстовый UNIX® файл настройки с строками комментариев, начинающимися с символа `#`. Исчерпывающее описание всех возможных параметров настройки находится за пределом рассмотрения этой книги, поэтому здесь будут описаны только наиболее часто модифицируемые директивы.

`ServerRoot "/usr/local"`

Указывает верхний каталог установки Apache по умолчанию. Бинарные файлы находятся в bin и sbin, подкаталоги расположены относительно корневого каталога сервера, файлы настройки находятся в etc/apache.

`ServerAdmin you@your.address`

Адрес, на который должны будут отправляться сообщения о проблемах с сервером. Этот адрес выводится на некоторые генерируемые сервером страницы, например с сообщениями об ошибках.

`ServerName www.example.com`

`ServerName` позволяет вам устанавливать имя хоста, которое отправляется обратно клиентам, если оно отличается от того, с которым настроен хост (например, использование `www` вместо реального имени хоста).

`DocumentRoot "/usr/local/www/data"`

`DocumentRoot`: Каталог, внутри которого будут храниться документы. По умолчанию, все запросы обрабатываются внутри этого каталога, но символические ссылки и синонимы могут использоваться для указания на другие каталоги.

Хорошей идеей будет сделать резервные копии настроек Apache перед внесением изменений. Как только вы будете удовлетворены первоначальной настройкой, можно запускать Apache.

24.7.3. Запуск Apache

Apache не запускается из inetd, как это делают многие другие сетевые серверы. Он настроен для автономного запуска, чтобы обеспечивать большую производительность при обработке HTTP запросов от браузеров клиентов. Для упрощения запуска, остановки и перезапуска сервера существует shell скрипт. Для запуска Apache в первый раз просто выполните:

```
# /usr/local/sbin/apachectl start
```

Вы можете остановить сервер в любой момент, выполнив:

```
# /usr/local/sbin/apachectl stop
```

После внесения любых изменений в файл настроек, вам потребуется перезапустить сервер:

```
# /usr/local/sbin/apachectl restart
```

Для перезапуска Apache без прерывания имеющихся соединений, выполните:

```
# /usr/local/sbin/apachectl graceful
```

Дополнительная информация находится на странице справочного руководства apachectl(8).

Для запуска Apache при старте системы, добавьте в /etc/rc.conf следующую строку:

```
apache_enable="YES"
```

или для Apache 2.2:

```
apache22_enable="YES"
```

Если вы хотите передать программе Apache`httpd` дополнительные параметры командной при загрузке системы, они могут быть помещены в дополнительную строку rc.conf:

```
apache_flags=""
```

Теперь, когда веб сервер запущен, вы можете просмотреть свой веб сайт, задав в строке браузера адрес `http://localhost/`. По умолчанию отображается веб страница /usr/local/www/data/index.html.

24.7.4. Виртуальный хостинг

Apache поддерживает два различных типа виртуального хостинга (Virtual Hosting). Первый метод основан на именах (Name-based Virtual Hosting). Он использует полученные от клиента заголовки HTTP/1.1 для определения имени хоста. Это позволяет многим различным доменам использовать один и тот же IP адрес.

Для настройки Apache на использование этого типа хостинга добавьте в httpd.conf запись подобную следующей:

```
NameVirtualHost *
```

Если веб сервер назывался `www.domain.tld` и вы хотите настроить виртуальный домен для `www.someotherdomain.tld`, необходимо добавить в httpd.conf следующие записи:

```
<VirtualHost *>
ServerName www.domain.tld
DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
ServerName www.someotherdomain.tld
DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

Замените адреса и пути к документам на те, что вы будете использовать.

За дополнительной информацией по настройке виртуальных хостов обращайтесь к официальной документации Apache: http://httpd.apache.org/docs/vhosts/.

24.7.5. Модули Apache

Существуют множество различных модулей Apache, которые добавляют функциональность к основному серверу. Коллекция портов FreeBSD предоставляет простой способ установки Apache с некоторыми наиболее популярными дополнительными модулями.

24.7.5.1. mod_ssl

Модуль mod_ssl использует библиотеку OpenSSL для сильной криптографии через протоколы Secure Sockets Layer (SSL v2/v3) и Transport Layer Security (TLS v1). Этот модуль

содержит все необходимое для запроса подписанного сертификата из центра сертификации для защищенного веб сервера на FreeBSD.

Если вы еще не установили Apache, версия Apache 1.3.X с mod_ssl может быть установлена через порт www/apache13-modssl. Поддержка SSL также доступна для Apache 2.X через порт www/apache22, где она включена по умолчанию.

24.7.5.2. Apache и скриптовые языки

Для большинства скриптовых языков созданы модули Apache. На базе таких модулей возможно создание других модулей Apache, написанных полностью на скриптовом языке. Они также часто используются как встроенные в сервер интерпретаторы, что исключает накладные расходы на запуск внешнего интерпретатора и сокращает время построения динамических страниц.

24.7.6. Построение динамических сайтов

В последнее десятилетие все большее число компаний обращает внимание на Интернет как площадку для ведения и расширения бизнеса. Среди прочего, этот процесс подчеркивает потребность в интерактивном содержимом сайтов. Некоторые компании, такие как Microsoft®, представляют свои закрытые решения; сообщество разработчиков открытых программ отвечает на вызов. Среди современных решений для предоставления динамического контента следует отметить Django, Ruby on Rails, mod_perl и mod_php.

24.7.6.1. Django

Django - это распространяемая под лицензией BSD инфраструктура, позволяющая разработчикам быстро создавать элегантные, высокопроизводительные веб-приложения. Она предоставляет в распоряжение разработчика объектно-реляционное отображение (object-relational mapper), таким образом типы данных разрабатываются как объекты Python. Для этих объектов предоставляется богатый интерфейс доступа к базам данных, при этом у разработчика не возникает необходимости написания SQL-запросов. Django также предоставляет расширяемую систему шаблонов, так что логика приложения отделена от его HTML-представления.

Для Django требуются следующие компоненты: mod_python, Apache и одна из нескольких возможных SQL СУБД. Укажите соответствующие опции сборки, и порт установит всё необходимое.

Пример 40. Установка Django совместно с Apache2, mod_python3 и PostgreSQL

```
# cd /usr/ports/www/py-django; make all install clean -DWITH_MOD_PYTHON3
-DWITH_POSTGRESQL
```

После установки Django и всех необходимых ему компонентов вам потребуется создать каталог для проекта Django. Далее потребуется настроить Apache для определенных URL адресов на вашем сайте выполнять ваше приложение встроенным интерпретатором

Python.

Пример 41. Конфигурация Apache для Django/mod_python

Чтобы настроить Apache отправлять запросы для определенных URL адресов вашему веб-приложению, вам потребуется внести несколько строк в конфигурационный файл httpd.conf:

```
<Location "/">
    SetHandler python-program
    PythonPath "['/dir/to/your/django/packages/'] + sys.path"
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE mysite.settings
    PythonAutoReload On
    PythonDebug On
</Location>
```

24.7.6.2. Ruby on Rails

Ruby on Rails это еще одна веб инфраструктура с открытым исходным кодом, которая предоставляет полный стек разработки и которая оптимизирована для продуктивного и быстрого создания мощных веб-приложений. Ruby on Rails может быть легко установлена из коллекции портов.

```
# cd /usr/ports/www/rubygem-rails; make all install clean
```

24.7.6.3. mod_perl

Проект интеграции Apache/Perl объединяет мощь языка программирования Perl и HTTP сервера Apache. С модулем mod_perl возможно написание модулей Apache полностью на Perl. Кроме того, постоянно запущенный встроенный в сервер интерпретатор позволяет не тратить ресурсы на запуск внешнего интерпретатора и время на запуск Perl.

mod_perl можно использовать различными способами. Помните, что mod_perl 1.0 работает только с Apache 1.3, тогда как mod_perl 2.0 совместим только с Apache 2.X. mod_perl 1.0 доступен как порт www/mod_perl, а также в виде статически скомпилированной версии в www/apache13-modperl. mod_perl 2.0 доступен как www/mod_perl2.

24.7.6.4. mod_php

PHP, также известный как "Препроцессор гипертекста" ("Hypertext Preprocessor"), - это скриптовый язык общего назначения, в основном предназначенный для веб разработки. Этот язык может быть встроен в HTML, его синтаксис заимствован из C, Java™ и Perl, и он позволяет веб разработчикам быстро писать динамически генерируемые страницы.

Добавление поддержки PHP5 к веб серверу Apache производится путем установки порта lang/mod_php5.

Если порт lang/php5 устанавливается впервые, то автоматически отобразятся все доступные опции (OPTIONS). Если меню не отображается, так как порт lang/php5 устанавливался ранее, всегда можно повторно вызвать диалог меню выполнив следующую команду в каталоге порта:

```
# make config
```

Выберите в меню опцию APACHE, тем самым вы построите загружаемый модуль mod_php5 для веб сервера Apache.

Множество сайтов по разным причинам (например, из-за проблем совместимости или из-за наличия уже развёрнутых веб приложений) всё еще используют PHP4. Если требуется mod_php4 вместо mod_php5, то воспользуйтесь портом lang/php4. Порт lang/php4 поддерживает многие из конфигурационных и установочных опций порта lang/php5.

Этот порт устанавливает и настраивает модули, необходимые для поддержки динамических PHP веб страниц. Убедитесь, что в файл /usr/local/etc/apache/httpd.conf были добавлены следующие секции:

```
LoadModule php5_module        libexec/apache/libphp5.so
```

```
AddModule mod_php5.c
    <IfModule mod_php5.c>
        DirectoryIndex index.php index.html
    </IfModule>
    <IfModule mod_php5.c>
        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
    </IfModule>
```

Для загрузки модуля PHP после этого просто вызовите команду apachectl с параметром graceful:

```
# apachectl graceful
```

При дальнейших обновлениях PHP команда make config больше не потребуется; выбранные опции сохраняются автоматически инфраструктурой портов FreeBSD

Поддержка PHP в FreeBSD построена по модульному принципу, поэтому базовая установка обладает очень ограниченной функциональностью. Дополнительная функциональность может быть легко добавлена при помощи порта lang/php5-extensions, управляющего набором расширений PHP через меню, либо просто путем установки дополнительных портов.

Например, для добавления поддержки MySQL к PHP5, просто установите порт databases/php5-mysql.

После установки новых расширений сервер Apache должен быть рестартован, чтобы изменения в конфигурации вступили в силу:

```
# apachectl graceful
```

24.8. Файл сервер и печать для Microsoft® Windows® клиентов (Samba)

24.8.1. Обзор

Samba это популярный пакет программ с открытыми исходными текстами, которая предоставляет файловые и принт-сервисы Microsoft® Windows® клиентам. Эти клиенты могут подключаться и использовать файловое пространство FreeBSD, как если бы это был локальный диск, или принтеры FreeBSD, как если бы это были локальные принтеры.

Пакет Samba должен быть включен в поставку FreeBSD. Если вы не установили Samba при первой установке системы, ее можно установить из порта или пакета net/samba34.

24.8.2. Настройка

Файл настройки Samba по умолчанию устанавливается в /usr/local/shared/examples/samba34/smb.conf.default. Этот файл необходимо скопировать в /usr/local/etc/smb.conf и отредактировать перед использованием Samba.

В файле smb.conf находится информация, необходимая для работы Samba, например определение принтеров и "общих каталогов", которые будут использоваться совместно с Windows® клиентами. В пакет Samba входит программа с веб интерфейсом, называемая swat, которая дает простой способ редактирования файла smb.conf.

24.8.2.1. Использование Samba Web Administration Tool (SWAT)

Программа веб администрирования Samba (Samba Web Administration Tool, SWAT) запускается как даемон из inetd. Следовательно, в /etc/inetd.conf необходимо снять комментарий перед тем, как использовать swat для настройки Samba:

```
swat   stream  tcp     nowait/400      root    /usr/local/sbin/swat    swat
```

Как описано в Перезагрузка конфигурационного файла inetd, после изменения настроек inetd необходимо перечитать конфигурацию.

Как только swat был включен inetd.conf, вы можете использовать браузер для подключения к http://localhost:901. Сначала необходимо зарегистрироваться с системной учетной записью `root`.

После успешного входа на основную страницу настройки Samba, вы можете просмотреть документацию или начать настройку, нажав на кнопку **Globals**. Раздел **Globals** соответствует переменным, установленным в разделе `[global]` файла /usr/local/etc/smb.conf.

24.8.2.2. Глобальные настройки

Независимо от того, используете ли вы swat, или редактируете /usr/local/etc/smb.conf непосредственно, первые директивы, которые вы скорее всего встретите при настройке Samba, будут следующими:

`workgroup`

Имя домена или рабочей группы NT для компьютеров, которые будут получать доступ к этому серверу.

`netbios name`

Устанавливает имя NetBIOS, под которым будет работать Samba сервер. По умолчанию оно устанавливается равным первому компоненту DNS имени хоста.

`server string`

Устанавливает строку, которая будет показана командой `net view` и некоторыми другими сетевыми инструментами, которые отображают строку описания сервера.

24.8.2.3. Настройки безопасности

Две из наиболее важных настроек в /usr/local/etc/smb.conf отвечают за выбор модели безопасности и за формат паролей для клиентов. Эти параметры контролируются следующими директивами:

`security`

Два наиболее часто используемых параметра это `security = share` и `security = user`. Если имена пользователей для клиентов совпадают с их именами на компьютере FreeBSD, вы возможно захотите включить безопасность уровня пользователя (user). Это политика безопасности по умолчанию, она требует, чтобы клиент авторизовался перед доступом к совместно используемым ресурсам.

На уровне безопасности share клиенту не требуется входить на сервер перед подключением к ресурсу. Эта модель безопасности использовалась по умолчанию в старых версиях Samba.

`passdb backend`

Samba поддерживает несколько различных подсистем аутентификации. Вы можете аутентифицировать клиентов с помощью LDAP, NIS+, базы данных SQL, или через модифицированный файл паролей. Метод аутентификации по умолчанию `smbpasswd`, и здесь рассматривается только он.

Предполагая, что используется подсистема по умолчанию `smbpasswd`, необходимо создать файл /usr/local/etc/samba/smbpasswd, чтобы Samba могла аутентифицировать клиентов. Если вы хотите разрешить к учетным записям UNIX® доступ с Windows® клиентов, используйте следующую команду:

```
# smbpasswd -a username
```

Ныне рекомендуемой подсистемой аутентификации является `tdbsam`, поэтому для добавления пользователей используйте следующую команду:

```
# pdbedit -a -u username
```

Пожалуйста, обратитесь к Official Samba HOWTO за дополнительной информацией о параметрах настройки. Основные настройки, рассмотренные здесь, достаточны для первого запуска Samba.

24.8.3. Запуск Samba

Порт net/samba34 добавляет новый стартовый сценарий, который может быть использован для контроля Samba. Для того, чтобы им можно было запускать, останавливать или перезапускать сервер Samba, добавьте следующую запись в файл /etc/rc.conf:

```
samba_enable="YES"
```

Или, для более тонкого контроля:

```
nmbd_enable="YES"
```

```
smbd_enable="YES"
```

Внесение этих записей в /etc/rc.conf также обеспечит автоматический запуск сервера Samba во время старта системы.

Теперь становится возможным запустить сервер Samba, для чего наберите следующую команду:

```
# /usr/local/etc/rc.d/samba start
Starting SAMBA: removing stale tdbs :
Starting nmbd.
Starting smbd.
```

За дальнейшей информацией об использовании rc скриптов обратитесь к Использование rc во FreeBSD 5.X и последующих версиях.

Samba состоит из трех отдельных даемонов. Вы можете видеть, что nmbd и smbd запускаются скриптом samba. Если вы включили сервис разрешения имен winbind в smb.conf, то увидите также запуск даемона winbindd.

Вы можете остановить Samba в любой момент, набрав:

```
# /usr/local/etc/rc.d/samba stop
```

Samba это сложный программный набор с функциональностью, позволяющей полную интеграцию в сети Microsoft® Windows®. За дальнейшей информацией о функциях, выходящих за рамки описанной здесь базовой установки, обращайтесь к http://www.samba.org.

24.9. Протокол передачи файлов (FTP)

24.9.1. Обзор

Протокол передачи файлов (File Transfer Protocol, FTP) дает пользователям простой путь передачи файлов на и с FTP сервера. В FreeBSD серверная программа FTP, ftpd, включена в базовую систему. Это упрощает настройку и администрирование FTP сервера в FreeBSD.

24.9.2. Настройка

Наиболее важный шаг заключается в определении того, каким учетным записям будет позволено получать доступ к FTP серверу. В обычной системе FreeBSD есть множество системных учетных записей, используемых различными даемонами, но пользователям должно быть запрещен вход с использованием этих учетных записей. В файле /etc/ftpusers находится список пользователей, которым запрещен доступ по FTP. По умолчанию он включает упомянутые системные учетные записи, но в него можно добавить и определенных пользователей, которым будет запрещен доступ по FTP.

Вам может понадобиться ограничить доступ определенных пользователей без полного запрета использования FTP. Это можно сделать через файл /etc/ftpchroot. В нем находится список пользователей и групп, к которым применяется ограничение доступа. На странице справочника ftpchroot(5) дана подробная информация, и она не будет дублироваться здесь.

Если вы захотите разрешить анонимный FTP доступ на сервер, в системе FreeBSD необходимо создать пользователя `ftp`. Этот пользователь сможет входить на FTP сервер с именем пользователя `ftp` или `anonymous`, с любым паролем (существует соглашение об использовании почтового адреса пользователя в качестве пароля). FTP сервер выполнит chroot(2) при входе пользователя anonymous для ограничения доступа только домашним каталогом пользователя `ftp`.

Существуют два текстовых файла, определяющих сообщение, отправляемое FTP клиентам. Содержимое файла /etc/ftpwelcome будет выведено пользователям перед приглашением на вход. После успешного входа будет выведено содержимое файла /etc/ftpmotd. Обратите внимание, что путь к этому файлу задается относительно домашнего каталога пользователя, так что анонимным пользователям будет отправляться ~ftp/etc/ftpmotd.

Как только FTP сервер был правильно настроен, он должен быть включен в /etc/inetd.conf. Все, что необходимо, это удалить символ комментария "#" из начала существующей строки

ftpd:

```
ftp stream  tcp nowait  root    /usr/libexec/ftpd   ftpd -l
```

Как описано в Перезагрузка конфигурационного файла inetd, inetd должен перечитать конфигурацию после того, как этот файл настройки был изменен. Пожалуйста обратитесь к Настройки за деталями по запуску inetd на вашей системе.

В качестве альтернативы, демон ftpd может быть запущен как самостоятельный сервер. В этом случае достаточно установить соответствующую переменную в файле /etc/rc.conf:

```
ftpd_enable="YES"
```

Демон будет запущен автоматически при следующей загрузке системы. Также демон можно запустить вручную, для чего выполните следующую команду как пользователь `root`:

```
# /etc/rc.d/ftpd start
```

Теперь вы можете войти на FTP сервер, введя:

```
% ftp localhost
```

24.9.3. Поддержка

Для протоколирования даемон ftpd использует сообщения syslog(3). По умолчанию, syslog(3) поместит сообщения, относящиеся к FTP, в файл /var/log/xferlog. Местоположение лог файла FTP может быть изменено путем изменения следующей строки в файле /etc/syslog.conf:

```
ftp.info      /var/log/xferlog
```

Учитывайте потенциальные проблемы, возникающие с анонимным FTP сервером. В частности, вы должны дважды подумать, прежде чем позволить анонимным пользователям загружать файлы на сервер. Вы можете обнаружить, что FTP сайт стал форумом, на котором происходит обмен нелицензионным коммерческим программным обеспечением или чем-то еще хуже. Если вам необходимо разрешить анонимную выгрузку файлов на FTP, права должны быть настроены таким образом, чтобы эти файлы не могли прочитать другие анонимные пользователи до их рассмотрения администратором.

24.10. Синхронизация часов через NTP

24.10.1. Обзор

С течением времени часы компьютера имеют тенденцию отставать. Network Time Protocol -

Сетевой Протокол Времени (NTP) является одним из способов вести точное время.

Многие сервисы Интернет опираются или сильно зависят от точности часов компьютеров. К примеру, веб-сервер может получать запрос на посылку файла, который был недавно модифицирован. В локальной сети необходимо, чтобы часы компьютеров, совместно использующих файлы, были синхронизированы, чтобы время модификации файлов устанавливалось правильно. Такие службы, как cron(8), также зависят от правильности установки системных часов, поскольку запускают команды в определенное время.

FreeBSD поставляется с сервером NTP ntpd(8), который можно использовать для опроса других серверов NTP для установки часов на вашей машине или предоставления услуг точного времени.

24.10.2. Выбор подходящих серверов NTP

Для синхронизации ваших часов вам нужно найти для использования один или большее количество серверов NTP. Ваш сетевой администратор или провайдер могут иметь сервер NTP для этой цели-обратитесь к ним, так ли это в вашем случае. Существует онлайн список общедоступных серверов NTP, которым можно воспользоваться для поиска ближайшего к вам сервера NTP. Не забудьте выяснить политику выбранного вами сервера и спросить разрешения, если это требуется.

Выбор нескольких несвязанных серверов NTP является хорошей идеей в том случае, если один из используемых вами серверов станет недоступным или его часы неточны. ntpd(8) использует ответы, которые он получает от других серверов с умом-он делает предпочтение надежным серверам.

24.10.3. Настройка вашей машины

24.10.3.1. Базовая конфигурация

Если вам нужно только синхронизировать ваши часы при загрузке машины, вы можете воспользоваться утилитой ntpdate(8). Это может подойти для некоторых настольных машин, которые часто перезагружаются и только требуют изредка синхронизироваться, но на большинстве машин должен работать ntpd(8).

Использование ntpdate(8) при загрузке также хорошо для машин, на которых запущен даемон ntpd(8). Программа ntpd(8) изменяет время постепенно, тогда как ntpdate(8) устанавливает время вне зависимости от того, насколько велика разница между текущим временем машины и точным временем.

Для включения ntpdate(8) во время загрузки, добавьте строчку `ntpdate_enable="YES"` в файл /etc/rc.conf. Вам также потребуется указать все серверы, с которыми вы хотите синхронизироваться, и все параметры, которые передаются в ntpdate(8), в `ntpdate_flags`.

24.10.3.2. Общие настройки

NTP настраивается в файле /etc/ntp.conf, формат которого описан в ntp.conf(5). Вот простой пример:

```
server ntplocal.example.com prefer
server timeserver.example.org
server ntp2a.example.net

driftfile /var/db/ntp.drift
```

Параметр `server` задает, какие серверы будут использоваться, по одному в каждой строке. Если сервер задан с аргументом `prefer`, как `ntplocal.example.com`, то этому серверу отдается предпочтение перед остальными. Ответ от предпочтительного сервера будет отброшен, если он значительно отличается от ответов других серверов, в противном случае он будет использоваться безотносительно к другим ответам. Аргумент `prefer` обычно используется для серверов NTP, о которых известно, что они очень точны, такими, на которых используется специальное оборудование точного времени.

Параметр `driftfile` задает файл, который используется для хранения смещения частоты системных часов. Программа ntpd(8) использует его для автоматической компенсации естественного смещения часов, позволяя ему поддерживать достаточно правильную настройку, даже если он на некоторый период отключается от внешнего источника информации о времени.

Параметр `driftfile` задает, какой файл используется для сохранения информации о предыдущих ответах от серверов NTP, которые вы используете. Этот файл содержит внутреннюю информацию для NTP. Он не должен изменяться никакими другими процессами.

24.10.3.3. Управление доступом к вашему серверу

По умолчанию ваш сервер NTP будет доступен всем хостам в Интернет. Параметр `restrict` в файле /etc/ntp.conf позволяет вам контролировать, какие машины могут обращаться к вашему серверу.

Если вы хотите запретить всем машинам обращаться к вашему серверу NTP, добавьте следующую строку в файл /etc/ntp.conf:

```
restrict default ignore
```

Эта строка конфигурации также предотвратит доступ вашего сервера к другим серверам, перечисленным в вашей локальной конфигурации. Если вам необходимо синхронизировать ваш сервер с внешним сервером NTP, вам необходимо будет изменить настройки относительно этого конкретного сервера. За более детальной информацией обратитесь к странице руководства ntp.conf(5).

Если вы хотите разрешить синхронизировать свои часы с вашим сервером только машинам в вашей сети, но запретить им настраивать сервер или быть равноправными участниками синхронизации времени, то вместо указанной добавьте строчку

```
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

где `192.168.1.0` является адресом IP вашей сети, а `255.255.255.0` её сетевой маской.

/etc/ntp.conf может содержать несколько директив `restrict`. Для получения подробной информации обратитесь к подразделу `Access Control Support` (Поддержка Управления Доступом) в ntp.conf(5).

24.10.4. Запуск сервера NTP

Для того, чтобы сервер NTP запускался при загрузке, добавьте строку `ntpd_enable="YES"` в файл /etc/rc.conf. Если вы хотите передать дополнительные опции в ntpd(8), то отредактируйте параметр `ntpd_flags` в файле /etc/rc.conf.

Для запуска сервера без перезагрузки вашей машины, выполните команду `ntpd`, не забыв задать дополнительные параметры из переменной `ntpd_flags` в файле /etc/rc.conf. К примеру:

```
# ntpd -p /var/run/ntpd.pid
```

24.10.5. Использование ntpd с временным подключением к Интернет

Для нормальной работы программе ntpd(8) не требуется постоянное подключение к Интернет. Однако если ваше временное подключение к Интернет настроено для дозвона по требованию, хорошо бы запретить трафику NTP вызывать дозвон или поддерживать соединение постоянно. Если вы используете пользовательский PPP, то можете воспользоваться директивами `filter` в файле /etc/ppp/ppp.conf. К примеру:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

Более подробную информацию можно найти в разделе `PACKET FILTERING` (ФИЛЬТРАЦИЯ ПАКЕТОВ) в ppp(8), а примеры в /usr/shared/examples/ppp/.

Некоторые провайдеры Интернет блокируют трафик по портам с маленькими номерами, что приводит к неработоспособности NTP, так как ответы никогда не достигают вашей машины.

24.10.6. Дополнительная литература

Документация по серверу NTP может быть найдена в каталоге /usr/shared/doc/ntp/ в формате HTML.

24.11. * Remote Host Logging with `syslogd`

Этот раздел не переведен.

Глава 25. Межсетевые экраны

25.1. Введение

Межсетевые экраны (firewall, брандмауэр) делают возможной фильтрацию входящего и исходящего трафика, идущего через вашу систему. Межсетевой экран использует один или более наборов "правил" для проверки сетевых пакетов при их входе или выходе через сетевое соединение, он или позволяет прохождение трафика или блокирует его. Правила межсетевого экрана могут проверять одну или более характеристик пакетов, включая но не ограничиваясь типом протокола, адресом хоста источника или назначения и портом источника или назначения.

Межсетевые экраны могут серьезно повысить уровень безопасности хоста или сети. Они могут быть использованы для выполнения одной или более нижеперечисленных задач:

- Для защиты и изоляции приложений, сервисов и машин во внутренней сети от нежелательного трафика, приходящего из внешней сети интернет.
- Для ограничения или запрещения доступа хостов внутренней сети к сервисам внешней сети интернет.
- Для поддержки преобразования сетевых адресов (network address translation, NAT), что дает возможность задействовать во внутренней сети приватные IP адреса и совместно использовать одно подключение к сети Интернет (либо через один выделенный IP адрес, либо через адрес из пула автоматически присваиваемых публичных адресов).

После прочтения этой главы вы узнаете:

- Как правильно задать правила фильтрации пакетов.
- Разницу между межсетевыми экранами, встроенными в FreeBSD
- Как использовать и настраивать межсетевой экран OpenBSD PF.
- Как использовать и настраивать IPFILTER.
- Как использовать и настраивать IPFW.

Перед прочтением этой главы вам потребуется:

- Ознакомиться с основами FreeBSD и интернет.

25.2. Принципы работы межсетевых экранов

Существует два основных способа создания наборов правил межсетевого экрана: "включающий" и "исключающий". Исключающий межсетевой экран позволяет прохождение всего трафика, за исключением трафика, соответствующего набору правил. Включающий межсетевой экран действует прямо противоположным образом. Он пропускает только трафик, соответствующий правилам и блокирует все остальное.

Включающий межсетевой экран обеспечивает гораздо большую степень контроля исходящего трафика. Поэтому включающий межсетевой экран является лучшим выбором

для систем, предоставляющих сервисы в сети Интернет. Он также контролирует тип трафика, порождаемого вне и направляющегося в вашу приватную сеть. Трафик, не попавший в правила, блокируется, а в файл протокола вносятся соответствующие записи. Включающие межсетевые экраны обычно более безопасны, чем исключающие, поскольку они существенно уменьшают риск пропуска межсетевым экраном нежелательного трафика.

Если не указано иначе, то все приведенные в этом разделе примеры наборов правил и конфигураций относятся к типу включающего межсетевого экрана.

Безопасность может быть дополнительно повышена с использованием "межсетевого экрана с сохранением состояния". Такой межсетевой экран сохраняет информацию об открытых соединениях и разрешает только трафик через открытые соединения или открытие новых соединений. Недостаток межсетевого экрана с сохранением состояния в том, что он может быть уязвим для атак DoS (Denial of Service, отказ в обслуживании), если множество новых соединений открывается очень быстро. Большинство межсетевых экранов позволяют комбинировать поведение с сохранением состояния и без сохранения состояния, что позволяет создавать оптимальную конфигурацию для каждой конкретной системы.

25.3. Пакеты межсетевых экранов

В базовую систему FreeBSD встроено три программных межсетевых экрана. Это *IPFILTER* (известный также как IPF), *IPFIREWALL* (известный также как IPFW) и *OpenBSD PacketFilter* (также известный как PF). Помимо этого, FreeBSD содержит два пакета ограничения трафика (по существу, предназначенных для контроля пропускной способности сетевого соединения): altq(4) и dummynet(4). Dummynet традиционно сильно связан с IPFW, а ALTQ с PF. В настоящее время IPFILTER не поддерживает ограничение пропускной способности сетевого соединения. Для реализации этой функции предлагается использовать IPFILTER совместно с одним из двух существующих пакетов ограничения трафика. Конфигурация следующая: IPFILTER задействуется для фильтрации и трансляции трафика, а IPFW с dummynet(4) _или_PF с ALTQ - для контроля пропускной способности сетевого соединения. IPFW и PF для контроля исходящих и входящих пакетов используют наборы правил, хотя и разными способами с разным синтаксисом правил.

Причина, по которой в FreeBSD включено более одного пакета межсетевых экранов, заключается в том, что разные пользователи выдвигают к ним различные требования и используют разные предпочтения. Нет одного пакета, который был бы очевидно лучше других.

Автор предпочитает IPFILTER, поскольку его правила с сохранением состояния гораздо проще использовать совместно с NAT; кроме того, в него встроен ftp прокси, что упрощает правила для фильтрации исходящих FTP соединений.

Поскольку все межсетевые экраны основаны на анализе значений выбранных полей заголовка пакета, для создания правил межсетевого экрана необходимо понимание принципов TCP/IP, того, что означают различные поля заголовка пакета, и как эти поля

используются в обычной сессии. Хорошим примером является: http://www.ipprimer.com/overview.cfm.

25.4. Packet Filter (PF, межсетевой экран OpenBSD) и ALTQ

В июле 2003 программный межсетевой экран OpenBSD, известный как PF, был портирован в FreeBSD и стал доступен из коллекции портов FreeBSD; первым релизом, где PF был интегрирован в основную систему, стала FreeBSD 5.3 в ноябре 2004. PF это полноценный межсетевой экран с широким набором возможностей, в котором есть опциональная поддержка ALTQ (Alternate Queuing). ALTQ предоставляет управление пропускной способностью Quality of Service (QoS).

Проект OpenBSD осуществляет замечательную работу по поддержке PF FAQ. Этот раздел руководства фокусируется на взаимосвязи PF и FreeBSD, предоставляя лишь общую информацию по его использованию. За более подробной информацией по использованию PF обратитесь к PF FAQ.

Дополнительные сведения о PF для FreeBSD можно получить с веб сайта: http://pf4freebsd.love2party.net/.

25.4.1. Использование модуля ядра PF

Чтобы загрузить PF как модуль ядра, добавьте следующую строку в /etc/rc.conf:

```
pf_enable="YES"
```

Далее, выполните стартовый скрипт:

```
# /etc/rc.d/pf start
```

Учтите, модуль PF не загрузится, если он не сможет найти конфигурационный файл с набором правил. По умолчанию размещение файла с правилами следующее: /etc/pf.conf. Если путь к файлу отличается от вышеприведённого, то внесите в /etc/rc.conf строку вида:

```
pf_rules="/path/to/pf.conf"
```

Файл с примерами конфигураций pf.conf находится в каталоге /usr/shared/examples/pf/.

Модуль PF можно также загрузить вручную:

```
# kldload pf.ko
```

Поддержка ведения логов для PF обеспечивается модулем `pflog.ko`, для загрузки которого

добавьте следующую строку в /etc/rc.conf:

```
pflog_enable="YES"
```

и запустите на выполнение скрипт:

```
# /etc/rc.d/pflog start
```

Если вам необходимы другие функциональные возможности PF, то придется добавить поддержку PF в ядро.

25.4.2. Параметры ядра

Включение PF путем компиляции с ядром FreeBSD не является обязательным требованием, однако вам может понадобиться одна из функциональных возможностей, которая не включена в загружаемый модуль. Например, pfsync(4) являет собой псевдоустройство, которое вносит определенные изменения в таблицу состояний, используемую PF. В дальнейшем, это псевдоустройство может быть скомпоновано с carp(4) чтобы создать отказоустойчивую систему межсетевых экранов на основе PF.

Пример параметров конфигурации ядра для включения PF находится в /usr/src/sys/conf/NOTES и показан здесь:

```
device pf
device pflog
device pfsync
```

`device pf` включает поддержку межсетевого экрана "Packet Filter" (pf(4)).

`device pflog` включает необязательное сетевое псевдоустройство pflog(4), которое может использоваться для протоколирования трафика через bpf(4). Даемон pflogd(8) может использоваться для сохранения протоколируемой информации на диск.

`device pfsync` включает необязательное сетевое псевдоустройство pfsync(4), используемое для отслеживания "изменений состояния".

25.4.3. Доступные параметры rc.conf

Для активации PF и pflog(4) во время загрузки в rc.conf(5) должны быть включены следующие переменные:

```
pf_enable="YES"                 # Включить PF (загрузить модуль если необходимо)
pf_rules="/etc/pf.conf"         # определение правил для pf
pf_flags=""                     # дополнительные флаги для запуска pfctl
pflog_enable="YES"              # запустить pflogd(8)
pflog_logfile="/var/log/pflog"  # где pflogd должен сохранять протокол
```

```
pflog_flags=""                  # дополнительные флаги для запуска pflogd
```

Если за межсетевым экраном находится локальная сеть и необходимо передавать пакеты для компьютеров этой сети, или использовать NAT, включите также следующий параметр:

```
gateway_enable="YES"            # Включить сетевой шлюз
```

25.4.4. Создание правил фильтрации

Пакет PF читает конфигурацию из файла pf.conf(5) (полный путь: /etc/pf.conf); пакеты отвергаются, пропускаются или модифицируются в соответствии с правилами и определениями из этого файла. В стандартную поставку FreeBSD входят несколько файлов с примерами конфигураций, которые находятся в каталоге /usr/shared/examples/pf/. За исчерпывающим описанием правил PF обратитесь к PF FAQ.

Изучая PF FAQ, имейте в виду, что различные версии FreeBSD могут содержать разные версии pf. В настоящий момент FreeBSD использует ту же версию PF, которая включена в OpenBSD 4.1.

Список рассылки, посвящённый FreeBSD packet filter является хорошим местом, чтобы задавать вопросы по конфигурации и использованию пакета PF. Не забудьте проверить архивы списка рассылки перед тем, как задавать вопрос.

25.4.5. Работа с PF

Для управления PF используйте утилиту pfctl(8). Ниже приведено несколько полезных команд (все возможные команды и опции приведены на странице справочника pfctl(8)):

Команда	Действие
`pfctl -e`	Включить PF
`pfctl -d`	Выключить PF
`pfctl -F all -f /etc/pf.conf`	Сбросить все правила (NAT, правила фильтрации, состояния соединений, таблицы и т.д.) и загрузить новые с файла /etc/pf.conf
`pfctl -s [ rules | nat | state ]`	Отобразить правила фильтрации, правила NAT или таблицу состояний соединений
`pfctl -vnf /etc/pf.conf`	Проверить /etc/pf.conf на наличие ошибок, но сами наборы правил не загружать

25.4.6. Включение ALTQ

ALTQ может быть включен только путем компилирования ядра FreeBSD с соответствующими параметрами. ALTQ поддерживается не всеми существующими

драйверами сетевых карт. Для просмотра списка поддерживаемых устройств в вашем релизе FreeBSD обратитесь к странице справочника altq(4).

Следующие параметры включат ALTQ и добавят дополнительную функциональность.

```
options         ALTQ
options         ALTQ_CBQ        # Class Bases Queuing (CBQ)
options         ALTQ_RED        # Random Early Detection (RED)
options         ALTQ_RIO        # RED In/Out
options         ALTQ_HFSC       # Hierarchical Packet Scheduler (HFSC)
options         ALTQ_PRIQ       # Priority Queuing (PRIQ)
options         ALTQ_NOPCC      # Required for SMP build
```

`options ALTQ` включает подсистему ALTQ.

`options ALTQ_CBQ` включает *Class Based Queuing* (CBQ). CBQ позволяет распределять пропускную способность соединений по классам или очередям для выставления приоритетов трафика на основе правил фильтрации.

`options ALTQ_RED` включает *Random Early Detection* (RED). RED используется для предотвращения перегрузки сети. RED вычисляет длину очереди и сравнивает ее с минимальным и максимальным значением длины очереди. Если очередь превышает максимум, все новые пакеты будут отброшены. В соответствии со своим названием, RED отбрасывает пакеты из различных соединений в произвольном порядке.

`options ALTQ_RIO` включает *Random Early Detection In and Out*.

`options ALTQ_HFSC` включает *Hierarchical Fair Service Curve Packet Scheduler*. Дополнительная информация о HFSC находится по адресу: http://www-2.cs.cmu.edu/~hzhang/HFSC/main.html.

`options ALTQ_PRIQ` включает *Priority Queuing* (PRIQ). PRIQ всегда первым пропускает трафик из очереди с более высоким приоритетом.

`options ALTQ_NOPCC` включает поддержку SMP для ALTQ. Эта опция необходима для SMP систем.

25.5. * IPFILTER (IPF)

Перевод раздела не завершен.

Этот раздел находится в процессе написания; содержание может не вполне соответствовать действительности.

Автором IPFILTER является Darren Reed. IPFILTER не зависит от операционной системы: это приложение с открытыми исходными текстами, которое было портировано на операционные системы FreeBSD, NetBSD, OpenBSD, SunOS™, HP/UX, и Solaris™. IPFILTER активно разрабатывается и поддерживается, регулярно выпускаются обновленные версии.

IPFILTER основан на межсетевом экране и механизме NAT уровня ядра, которые управляются и контролируются утилитами уровня пользовательских процессов. Правила межсетевого экрана могут устанавливаться или удаляться утилитой ipf(8). Правила NAT могут устанавливаться или удаляться утилитой ipnat(1). Утилита ipfstat(8) выводит статистику IPFILTER для ядра. Программа ipmon(8) может заносить действия IPFILTER в файлы системных протоколов.

IPF был первоначально написан с использованием правила "последнее совпадение применяется" и только с правилами без сохранения состояния. Со временем IPF был расширен и включает параметры "quick" и "keep state" (сохранение состояния), которые кардинальным образом изменяют логику обработки пакетов. Официальная документация IPF включает традиционные параметры правил с традиционной последовательностью обработки пакетов. Измененные функции включены в виде дополнительных параметров, они необходимы для создания эффективного межсетевого экрана.

Инструкции этого раздела подразумевают использование параметра "quick" и параметра сохранения состояния "keep state". Это основа для создания включающего межсетевого экрана.

Детальное описание традиционных методов обработки правил: http://www.obfuscation.org/ipf/ipf-howto.html#TOC_1 и http://coombs.anu.edu.au/~avalon/ip-filter.html.

IPF FAQ находится по адресу http://www.phildev.net/ipf/index.html.

Архив списка рассылки по IPFilter с возможностью поиска доступен по адресу http://marc.theaimsgroup.com/?l=ipfilter.

25.5.1. Включение IPF

IPF включен в базовую систему FreeBSD в качестве отдельного загружаемого модуля. Система динамически загрузит модуль IPF, если в rc.conf указана переменная `ipfilter_enable="YES"`. Модуль создается с включенным протоколированием и правилом по умолчанию `pass all` (пропускать все). Для изменения правила по умолчанию не обязательно собирать ядро с новыми параметрами. Просто добавьте в конец набора правило, блокирующее все пакеты.

25.5.2. Параметры ядра

Включение IPF в ядро FreeBSD не является обязательным требованием. Эта процедура представлена здесь в качестве дополнительной информации. При включении IPF в ядро загружаемый модуль не используется.

Пример параметров настройки ядра для IPF находится в /usr/src/sys/conf/NOTES и воспроизведен здесь:

```
options IPFILTER
options IPFILTER_LOG
```

```
options IPFILTER_DEFAULT_BLOCK
```

`options IPFILTER` включает поддержку межсетевого экрана "IPFILTER".

`options IPFILTER_LOG` включает протоколирование трафика через IPF путем записи его в псевдо-устройство протоколирования пакетов ipl для каждого правила, содержащего ключевое слово `log`.

`options IPFILTER_DEFAULT_BLOCK` изменяет поведение по умолчанию так, что блокируется каждый пакет, не соответствующий правилу `pass`.

Эти настройки будут работать только после сборки и установки нового ядра.

25.5.3. Доступные параметры rc.conf

Для активации IPF во время загрузки в /etc/rc.conf потребуется добавить следующие переменные:

```
ipfilter_enable="YES"             # Запуск межсетевого экрана ipf
ipfilter_rules="/etc/ipf.rules"   # Загрузка файла с правилами
ipmon_enable="YES"                # Включение протоколирования IP monitor
ipmon_flags="-Ds"                 # D = запуск в виде даемона
                                  # s = протоколирование в syslog
                                  # v = протоколирование tcp window, ack, seq
                                  # n = отображение имен IP и портов
```

Если за межсетевым экраном находится локальная сеть, использующая приватные IP адреса, для включения NAT потребуется добавить следующие переменные:

```
gateway_enable="YES"              # Включение шлюза для локальной сети
ipnat_enable="YES"                # Запуск функции ipnat
ipnat_rules="/etc/ipnat.rules"    # Определение файла правил для ipnat
```

25.5.4. IPF

Команда ipf(8) используется для загрузки файла с правилами. Обычно создается файл, содержащий подготовленный набор правил, который полностью замещает набор, используемый на данный момент:

```
# ipf -Fa -f /etc/ipf.rules
```

`-Fa` означает сброс всех внутренних таблиц правил.

`-f` указывает файл с правилами, который необходимо загрузить.

Это дает вам возможность отредактировать файл с правилами, запустить

вышеприведенную команду IPF, тем самым обновить набор правил работающего межсетевого экрана без перезагрузки системы. Для обновления правил такой подход очень удобен, поскольку команду можно выполнять столько раз, сколько нужно.

На странице справочной системы ipf(8) находится подробная информация по всем флагам этой команды.

Набор правил для команды ipf(8) должен быть в виде стандартного текстового файла. Правила, написанные в виде скрипта с символами подстановки, не принимаются.

Есть способ составления правил IPF, использующих символы подстановки. Обратитесь к Создание набора правил с использованием символьной подстановки.

25.5.5. IPFSTAT

По умолчанию ipfstat(8) получает и отображает суммарную статистику, полученную в результате применения действующих правил к пакетам, проходящим через межсетевой экран с момента его последнего запуска, или с того момента, когда статистика была последний раз обнулена командой `ipf -Z`.

Детальная информация приводится на странице справочника ipfstat(8).

Вывод команды ipfstat(8) по умолчанию выглядит примерно так:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
 output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
 input packets logged: blocked 99286 passed 0
 output packets logged: blocked 0 passed 0
 packets logged: input 0 output 0
 log failures: input 3898 output 0
 fragment state(in): kept 0 lost 0
 fragment state(out): kept 0 lost 0
 packet state(in): kept 169364 lost 0
 packet state(out): kept 431395 lost 0
 ICMP replies: 0 TCP RSTs sent: 0
 Result cache hits(in): 1215208 (out): 1098963
 IN Pullups succeeded: 2 failed: 0
 OUT Pullups succeeded: 0 failed: 0
 Fastroute successes: 0 failures: 0
 TCP cksum fails(in): 0 (out): 0
 Packet log flags set: (0)
```

При задании флага `-i` или `-o` соответственно для входящих или исходящих пакетов, команда извлечет и отобразит соответствующий список правил, установленных и используемых на данный момент.

`ipfstat -in` отображает правила, применяемые к входящим пакетам, вместе с номерами этих правил.

`ipfstat -on` отображает правила, применяемые к исходящим пакетам, вместе с номерами

этих правил.

Вывод команды будет выглядеть примерно так:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

`ipfstat -ih` отображает правила, применяемые к входящим пакетам, со счетчиком количества совпадений для каждого правила.

`ipfstat -oh` отображает правила, применяемые к исходящим пакетам, со счетчиком количества совпадений для каждого правила.

Вывод команды будет выглядеть примерно так:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Одна из наиболее важных функций команды `ipfstat` активируется флагом `-t`, правила отображаются подобно тому, как top(1) показывает таблицу запущенных процессов FreeBSD. Когда межсетевой экран подвергается атаке, эта функция позволяет обнаружить соответствующие пакеты. Дополнительные флаги дают возможность выбирать IP адрес назначения или источника, порт или протокол, которые будут отслеживаться в реальном времени. Подробная информация приведена на странице ipfstat(8).

25.5.6. IPMON

Для того, чтобы стало возможно использование команды `ipmon`, необходимо включить параметр ядра `IPFILTER_LOG`. Эта команда может использоваться в двух различных режимах. В основном режиме, который используется по умолчанию, она используется без флага `-D`.

В режиме даемона создается непрерывный протокол, и возможен просмотр предыдущих событий. В этом режиме IPFILTER работает в FreeBSD. Поскольку в FreeBSD встроена функция ротации файлов протокола, лучше использовать syslogd(8), чем используемый по умолчанию вывод в обычный файл. В rc.conf по умолчанию `ipmon_flags` имеет значение `-Ds`:

```
ipmon_flags="-Ds" # D = start as daemon
                  # s = log to syslog
                  # v = log tcp window, ack, seq
                  # n = map IP & port to names
```

Описывать преимущества протоколирования излишне. Например, оно дает возможность отложенного просмотра информации об отброшенных пакетах, откуда они пришли и куда направлялись. Эта информация существенно помогает при отслеживании атак.

Даже с включенным протоколированием, IPF не ведет протокол для каждого правила. Администратор межсетевого экран должен решить, по каким правилам набора нужно вести протокол и добавить ключевое слово log к этим правилам. Обычно протоколируются только правила, отбрасывающие пакеты.

Включение в набор последнего правила, запрещающего прохождение пакетов, в сочетании с ключевым словом `log` является довольно распространённой практикой. Так вы можете увидеть все пакеты, не попадающие ни под одно правило набора.

25.5.7. Протоколирование IPMON

Для разделения собираемых данных syslogd использует свой собственный специальный метод. Он использует группировку по категории ("facility") и уровню ("level"). IPMON в режиме `-Ds` использует `local0` в качестве имени "категории". Для дальнейшего разделения протоколируемых данных, если такое необходимо, могут быть использованы следующие уровни:

```
LOG_INFO - packets logged using the "log" keyword as the action rather than pass or
block.
LOG_NOTICE - packets logged which are also passed
LOG_WARNING - packets logged which are also blocked
LOG_ERR - packets which have been logged and which can be considered short
```

Для указания IPFILTER протоколировать все данные в /var/log/ipfilter.log, создайте этот файл заранее, выполнив следующую команду:

```
# touch /var/log/ipfilter.log
```

Функционирование syslogd(8) управляется настройками в файле /etc/syslog.conf. Файл syslog.conf позволяет достаточно гибко настроить обработку системных сообщений, выдаваемых программами, такими как IPF.

Добавьте в /etc/syslog.conf следующую запись:

```
local0.* /var/log/ipfilter.log
```

`local0.*` означает запись всех протоколируемых сообщений в указанный файл.

Для применения внесенных в /etc/syslog.conf изменений вы можете перезагрузиться или заставить syslogd(8) перечитать /etc/syslog.conf, выполнив команду `/etc/rc.d/syslogd reload`.

Не забудьте отредактировать /etc/newsyslog.conf для ротации только что созданного лог файла.

25.5.8. Формат протоколируемых сообщений

Сообщения, генерируемые `ipmon`, состоят из полей данных, разделенных пробелами. Поля, общие для всех сообщений:

1. Дата получения пакета.
2. Время получения пакета. Формат времени HH:MM:SS.F для часов, минут, секунд и долей секунд (последнее поле может состоять из нескольких цифр).
3. Имя интерфейса, через который прошел пакет, например dc0.
4. Группа и номер правила, например `@0:17`.

Эти сообщения могут быть просмотрены командой `ipfstat -in`.

1. Действие: p для пропущенных, b для заблокированных, S для пакетов с неполным заголовком (short packet), n для пакетов, не соответствующих какому-либо правилу, L для соответствующих правилу протоколирования. Порядок следования по флагам: S, p, b, n, L. Знаки P или B в верхнем регистре означают, что пакет был протоколирован в соответствии с общими настройками, а не каким-то конкретным правилом.
2. Адреса. Всего три поля: адрес и порт источника (разделенные запятой), →, адрес и порт назначения. 209.53.17.22,80 → 198.73.220.17,1722.
3. `PR`, с последующим именем или номером протокола, например `PR tcp`.
4. `len`, с последующей длиной заголовка и общей длиной пакета, например `len 20 40`.

Для TCP пакетов добавляется дополнительное поле, начинающееся с дефиса, за которым следуют буквы, соответствующие установленным флагам. На странице справочника ipf(5) находится список букв и флагов.

Для пакетов ICMP, в конце находятся два поля, одно всегда "ICMP", а второе содержит тип и подтип ICMP сообщения (message и sub-message), разделенные символом косой черты, например ICMP 3/3 для сообщения "port unreachable".

25.5.9. Создание набора правил с использованием символьной подстановки

Некоторые опытные пользователи IPF создают файл правил, поддерживающий использование символьной подстановки. Основное преимущество использования такого подхода заключается в возможности изменения значения, присваиваемого символьному имени, в результате чего во всех правилах, содержащих эту символьную подстановку, будет использоваться новое значение. В начале скрипта вы можете поместить часто используемые переменные, а затем использовать их сразу в нескольких правилах. Ниже дан пример такого использования.

Синтаксис скрипта совместим с sh(1), csh(1), и tcsh(1).

Символьная подстановка предваряется знаком доллара: `$`.

Для присвоения значения символьным переменным знак $ не используется.

Присваиваемое символической переменной значение должно быть заключено в двойные кавычки (").

Начните файл правил примерно так:

```
############# Start of IPF rules script ########################

oif="dc0"            # name of the outbound interface
odns="192.0.2.11"    # ISP's DNS server IP address
myip="192.0.2.7"     # my static IP address from ISP
ks="keep state"
fks="flags S keep state"

# You can choose between building /etc/ipf.rules file
# from this script or running this script "as is".
#
# Uncomment only one line and comment out another.
#
# 1) This can be used for building /etc/ipf.rules:
#cat > /etc/ipf.rules << EOF
#
# 2) This can be used to run script "as is":
/sbin/ipf -Fa -f - << EOF

# Allow out access to my ISP's Domain name server.
pass out quick on $oif proto tcp from any to $odns port = 53 $fks
pass out quick on $oif proto udp from any to $odns port = 53 $ks

# Allow out non-secure standard www function
pass out quick on $oif proto tcp from $myip to any port = 80 $fks

# Allow out secure www function https over TLS SSL
pass out quick on $oif proto tcp from $myip to any port = 443 $fks
EOF
################## End of IPF rules script ########################
```

Это все, что требовалось сделать. В данном примере сами правила не важны; важно то, как используется символьная подстановка. Если вышеприведенный пример помещен в файл /etc/ipf.rules.script, то набор правил можно перезагрузить, введя следующую команду:

```
# sh /etc/ipf.rules.script
```

С использованием в правилах символьной подстановки связана одна проблема: IPF не понимает символьную подстановку и не может обработать такой скрипт непосредственно.

Скрипт может использоваться одним из следующих двух способов:

- Уберите комментарий перед строкой, начинающейся с `cat`, и закомментируйте строку,

начинающуюся с `/sbin/ipf`. Поместите строку `ipfilter_enable="YES"` в файл /etc/rc.conf как обычно, и запускайте скрипт после каждого его обновления для создания или обновления файла /etc/ipf.rules.

- Отключите IPFILTER в стартовых скриптах системы, поместив строку `ipfilter_enable="NO"` (это значение по умолчанию) в файл /etc/rc.conf.

  Поместите скрипт, подобный нижеприведенному, в каталог /usr/local/etc/rc.d/. У него должно быть однозначно говорящее о его назначении имя, например ipf.loadrules.sh. Расширение .sh обязательно.

  ```
  #!/bin/sh
  sh /etc/ipf.rules.script
  ```

  Права, установленные на этот файл, должны разрешать чтение, запись и выполнение владельцу `root`.

  ```
  # chmod 700 /usr/local/etc/rc.d/ipf.loadrules.sh
  ```

Теперь, правила IPF будут загружаться при загрузке системы.

25.5.10. Наборы правил IPF

Набор правил ipf это группа правил, составленных для пропускания или блокирования пакетов на основе их содержимого. Двусторонний обмен пакетами между хостами составляет сессию. Межсетевой экран обрабатывает как входящие из Интернет пакеты, так и исходящие пакеты, которые сгенерированы самой системой в ответ на входящий трафик. Для каждой службы TCP/IP (например, telnet, www, mail, и т.п.) назначен протокол и номер привилегированного (прослушиваемого) порта. Пакеты, предназначенные для определенного сервиса, порождаются с некоторым исходящим адресом и портом из непривилегированного диапазона и направляются на определенный адрес и определенный порт назначения. Все упомянутые параметры (номера портов и адреса) могут использоваться как критерии выбора в правилах, пропускающих или блокирующих доступ к службам TCP/IP.

IPF был первоначально написан с использованием логики "последнее совпадающее правило побеждает" и только с правилами без сохранения состояния. Со временем в IPF был включен параметр "quick" и параметр сохранения состояния "keep state", что существенно улучшило логику обработки правил.

Инструкции, помещенные в эту главу, созданы с использованием параметров "quick" и "keep state". Это основа для создания набора правил включающего межсетевого экрана.

При работе с правилами межсетевого экрана, будьте *очень осторожны*. Некоторые конфигурации *могут заблокировать вам доступ* к серверу. В целях предосторожности, первоначальную настройку межсетевого экрана вы можете выполнить с локальной консоли, а не через удаленное

подключение, такое как ssh.

25.6. IPFW

IPFIREWALL (IPFW) - представляет собой межсетевой экран, написанный и поддерживаемый добровольными участниками проекта FreeBSD. Он использует stateless правила, т.е. правила без учета состояния, и наследование техники кодирования правил для получения того, что называется простой логикой с сохранением состояния (stateful).

Пример простейшего набора правил IPFW (находится в /etc/rc.firewall и /etc/rc.firewall6) в стандартной установке FreeBSD достаточно прост и не рассчитан на непосредственное использование без изменений. В нём не используется фильтрация с сохранением состояния, которая даёт преимущества во многих конфигурациях, поэтому он не может быть взят за основу для этого раздела.

Синтаксис правил IPFW без сохранения состояния обеспечивает расширенные возможности фильтрации, которые намного превосходят уровень знаний обычного пользователя межсетевого экрана. IPFW рассчитан на профессиональных пользователей или технически продвинутых любителей, которые предъявляют повышенные требования к фильтрации пакетов. Чтобы использовать возможности IPFW в полную силу, необходимы углубленные знания того, как в различных протоколах формируются и используются заголовки пакетов. Углубленное изучение работы протоколов выходит за рамки этого раздела Руководства.

IPFW состоит из семи компонентов, главный из которых - процессор правил фильтрации уровня ядра и интегрированный в него механизм учета пакетов, а также средства протоколирования пакетов, правило `divert`, посредством которых вызывается функция NAT и другие возможности специального назначения, средства для ограничения скорости (шейпинга) трафика (dummynet), средства перенаправления `fwd`, средства организации сетевого моста bridge и механизм ipstealth. IPFW поддерживает протоколы IPv4 и IPv6.

25.6.1. Включение IPFW

IPFW включён в базовую установку FreeBSD в виде отдельного подгружаемого модуля. Система динамически загружает модуль ядра, когда в rc.conf присутствует строка `firewall_enable="YES"`. Если использовать функциональность NAT не планируется, то в этом случае дополнительно компилировать IPFW в состав ядра FreeBSD не требуется.

После перезагрузки системы с `firewall_enable="YES"` в rc.conf на экране в процессе загрузки отобразится выделенное белым сообщение:

```
ipfw2 initialized, divert disabled, rule-based forwarding disabled, default to deny,
logging disabled
```

Загружаемый модуль скомпилирован с возможностью протоколирования информации о трафике. Для включения протоколирования и установки уровня его детализации имеется переключатель, значение которого можно установить в конфигурационном файле

/etc/sysctl.conf. При добавлении следующих двух строк протоколирование будет включено при следующей загрузке системы:

```
net.inet.ip.fw.verbose=1
net.inet.ip.fw.verbose_limit=5
```

25.6.2. Параметры ядра

Включение следующих параметров в ядро FreeBSD не является обязательным, если дополнительно не требуется функциональность NAT. Эти параметры представлены здесь в качестве справочной информации для дальнейших примеров.

```
options    IPFIREWALL
```

Этот параметр включает IPFW в состав ядра.

```
options    IPFIREWALL_VERBOSE
```

Этот параметр включает протоколирование пакетов, которые проходят через IPFW по правилам с ключевым словом `log`.

```
options    IPFIREWALL_VERBOSE_LIMIT=5
```

Ограничение числа пакетов, прошедших через syslogd(8), отдельно для каждого правила. Этот параметр имеет смысл использовать в недружественной среде, когда необходимо отслеживать активность межсетевого экрана. Это закрывает возможность атак типа "отказ в обслуживании" через флуд сообщениями syslog.

```
options    IPFIREWALL_DEFAULT_TO_ACCEPT
```

Этот параметр включает для IPFW разрешающую политику по умолчанию. Это удобно на первых этапах настройки IPFW.

```
options    IPDIVERT
```

Включение функциональности NAT.

Межсетевой экран будет блокировать все входящие и исходящие пакеты, если отсутствует параметр ядра `IPFIREWALL_DEFAULT_TO_ACCEPT` или правило, явно разрешающее эти соединения.

25.6.3. Параметры /etc/rc.conf

Включение межсетевого экрана:

```
firewall_enable="YES"
```

Для выбора одного из стандартных режимов работы межсетевого экрана, предоставляемых FreeBSD, выберите наиболее подходящий в файле /etc/rc.firewall и разместите так, как указано ниже:

```
firewall_type="open"
```

Возможны следующие значения для этого параметра:

- `open` - пропускать весь трафик.
- `client` - защищать только эту машину.
- `simple` - защищать всю сеть.
- `closed` - полностью запретить IP трафик, за исключением loopback интерфейса.
- `UNKNOWN` - отключить загрузку правил межсетевого экрана.
- filename - абсолютный путь к файлу, содержащему правила межсетевого экрана.

Есть два варианта загрузки собственных правил в межсетевой экран ipfw. Первый способ - задать переменную `firewall_type` в виде абсолютного пути файла, содержащего *правила межсетевого экрана* без каких-либо параметров командной строки для самого ipfw(8). Ниже приведён простой пример набора правил, который блокирует весь входящий и исходящий трафик:

```
add deny in
add deny out
```

Второй способ - установить значение переменной `firewall_script` в виде абсолютного пути исполняемого скрипта, содержащего команды `ipfw`, которые будут выполнены во время загрузки операционной системы. Правильный формат правил исполняемого скрипта должен соответствовать формату файла, приведённому ниже:

```
#!/bin/sh

ipfw -q flush

ipfw add deny in
ipfw add deny out
```

Если переменной `firewall_type` присвоено значение `client` или `simple`, то

правила, расположенные по умолчанию в /etc/rc.firewall, должны быть приведены в соответствие с конфигурацией данной машины. Также заметим, что для используемых в этой главе примеров в качестве значения переменной `firewall_script` используется /etc/ipfw.rules.

Включение протоколирования:

```
firewall_logging="YES"
```

Единственное, что делает параметр `firewall_logging`, - присвоение логической единицы (1) переменной sysctl `net.inet.ip.fw.verbose` (смотрите Включение IPFW). В rc.conf нет переменной для ограничения протоколирования, но это можно сделать через переменную sysctl вручную либо используя файл /etc/sysctl.conf:

```
net.inet.ip.fw.verbose_limit=5
```

Если ваша машина выполняет роль шлюза, т.е. обеспечивает трансляцию сетевых адресов (NAT) с помощью natd(8), имеет смысл сразу перейти к чтению Даемон преобразования сетевых адресов (natd) для уточнения информации относительно параметров /etc/rc.conf.

25.6.4. Команда IPFW

Команда `ipfw` - это стандартный механизм для ручного добавления/удаления отдельных правил в активной цепочке правил межсетевого экрана. Основная проблема при использовании этого метода состоит в том, что при перезагрузке операционной системы все изменения, сделанные с помощью данной команды, будут утеряны. Вместо этого рекомендуется записать все правила в файл, из которого они будут считываться во время загрузки операционной системы, а также для полной замены текущего набора правил на содержимое из файла.

Тем не менее, команду `ipfw` удобно использовать для отображения текущей конфигурации правил на экране консоли. Учетный модуль IPFW динамически создаёт счётчики для каждого правила, которые подсчитывают количество пакетов, соответствующих условиям срабатывания правила. В процессе тестирования отображение правила со своим счётчиком является одним из способов проверки, срабатывает ли правило при прохождении через него пакета или нет.

Вывод полного списка правил:

```
# ipfw list
```

Вывод полного списка правил с маркером времени последнего срабатывания правила:

```
# ipfw -t list
```

Следующий пример выводит учетную информацию, количество совпавших пакетов и сами правила. Первым столбцом идет номер правила, за ним следует число совпавших исходящих пакетов, третий столбец - число соответствующих входящих пакетов, и затем само правило.

```
# ipfw -a list
```

Вывод динамических правил вместе со статическими:

```
# ipfw -d list
```

Отобразить статические и динамические правила, в т.ч. с истекшим временем действия:

```
# ipfw -d -e list
```

Обнуление счетчиков:

```
# ipfw zero
```

Обнулить счетчики для правила под номером *NUM*:

```
# ipfw zero NUM
```

25.6.5. Набор правил IPFW

Набор правил (ruleset) представляет собой группу правил IPFW, которые разрешают или запрещают прохождение пакета через межсетевой экран на основании значений, содержащихся в пакете. Двунаправленный обмен пакетов между машинами является сессией. Набор правил межсетевого экрана анализирует как пакеты, приходящие из глобальной сети, так и ответные пакеты, исходящие из системы. Каждый TCP/IP сервис (такой как telnet, www, mail, и т.д.) принадлежит определенному протоколу и привилегированному (прослушиваемому) порту. Пакеты, предназначенные для конкретного сервиса, передаются с непривилегированного (с высоким значением) порта по адресу назначения на указанный порт сервиса. Все эти параметры (т.е. порты и адреса) могут быть использованы в качестве критериев фильтрации при создании правил, которые пропускают или блокируют сервисы.

Когда пакет попадает в межсетевой экран, он сравнивается с каждым правилом, начиная с первого, двигаясь по множеству правил верху вниз в порядке увеличения номера правил. Когда пакет совпадает с критерием выбора правила, выполняется действие, указанное в

правиле, и на этом поиск правил прекращается. Такой метод поиска известен как "выигрыш первого совпадения", т.е. после срабатывания правила оставшиеся не просматриваются. Если содержимое пакета не соответствует ни одному из правил, он принудительно попадает на встроенное правило по умолчанию, заданное под номером 65535, которое запрещает и отбрасывает все пакеты без какого-либо отклика в сторону отправителя.

Поиск продолжается после правил `count`, `skipto` и `tee`.

Упомянутые здесь инструкции основаны на использовании правил, содержащих параметры с сохранением состояния `keep state`, `limit`, `in`, `out` и `via`. Это основной механизм для кодирования набора правил межсетевого экрана закрытого типа.

Будьте осторожны, когда работаете с правилами межсетевого экрана, так как вы можете легко заблокировать самого себя.

25.6.5.1. Синтаксис правил

Представленный здесь синтаксис правил был упрощен для создания стандартного набора правил межсетевого экрана закрытого типа. Для полного описания синтаксиса правил смотрите страницу Справочника ipfw(8).

Правила содержат ключевые слова: эти ключевые слова записываются в строке в определенном порядке слева направо. Ключевые слова выделены полужирным шрифтом. Некоторые ключевые слова имеют дополнительные параметры, которые могут являться ключевыми словами для них самих и также содержать вложенные дополнительные параметры.

Символ `#` используется для обозначения начала комментария и может быть расположен в конце строки с правилом или в начале строки над правилом. Пустые строки игнорируются.

CMD RULE_NUMBER ACTION LOGGING SELECTION STATEFUL

25.6.5.1.1. CMD

Каждое новое правило должно начинаться с префикса add для добавления во внутреннюю таблицу.

25.6.5.1.2. RULE_NUMBER

Каждое правило обозначено номером в диапазоне 1..65535.

25.6.5.1.3. ACTION

При соответствии пакета описанным в правиле критериям фильтрации будет выполнено одно из следующих действий.

allow | accept | pass | permit

Все эти действия означают одно и то же - пакеты, совпадающие с правилом, могут покинуть

обработку правил межсетевого экрана. На этом поиск прекращается.

check-state

Проверяет пакет на соответствие динамической таблице правил. Если совпадение найдено, выполняется действие, содержащееся в правиле, породившем данное динамическое правило, иначе выполняется переход к следующему правилу. Правило check-state не имеет критериев фильтрации. При отсутствии правила check-state в наборе правил проверка по динамической таблице происходит на первом правиле keep-state или limit.

deny | drop

Оба слова означают отбрасывание пакетов, совпавших с правилом. Поиск прекращается.

25.6.5.1.4. Протоколирование

log или logamount

Когда пакет совпадает с правилом, содержащим ключевое слово `log`, информация об этом событии записывается в syslogd(8) с пометкой SECURITY. Запись в журнал происходит только в том случае, если число срабатываний для данного правила не превышает значения параметра `logamount`. Если значение `logamount` не объявлено, то ограничение берется из значения переменной sysctl `net.inet.ip.fw.verbose_limit`. В обоих случаях обнуление значения отменяет ограничение. По достижению установленного лимита запись в журнал может быть повторно включена путем сброса счетчика срабатываний или счетчика пакетов для этого правила; смотрите описание команды `ipfw reset log`.

Протоколирование осуществляется после проверки на соответствие всем условиям в правиле и перед выполнением окончательного действия (accept, deny) над пакетом. Вы должны выбрать сами, какие действия правил вы хотите включить в журнал.

25.6.5.1.5. Условия отбора

Ключевые слова, представленные в этом разделе, используются для описания атрибутов пакета, по которым проверяется условие срабатывания того или иного правила. Для совпадения используется следующая последовательность атрибутов общего назначения:

udp | tcp | icmp

Также могут быть использованы имена протоколов, описанные в /etc/protocols. Указанное значение обозначает протокол для совпадения. Это является обязательным требованием.

from src to dst

Ключевые слова `from` и `to` служат для фильтрации по IP адресам. Обязательно должны быть указаны и источник, и получатель. `any` - это специальное ключевое слово, которое соответствует любому IP адресу. `me` - это специальное ключевое слово, которое соответствует любому из IP адресов, сконфигурированных на интерфейсе вашей системы FreeBSD, и служит для указания компьютера, на котором работает межсетевой экран (т.е. этот

компьютер), как показано на примерах `from me to any`, `from any to me`, `from 0.0.0.0/0 to any`, `from any to 0.0.0.0/0`, `from 0.0.0.0 to any`, `from any to 0.0.0.0` и `from me to 0.0.0.0`. IP адрес указывается в виде четырёх чисел, разделённых точками, или дополнительно с префиксом сети (нотация CIDR). Это является обязательным требованием. Для упрощения вычислений, связанных с IP адресами, используйте порт net-mgmt/ipcalc. Более подробную информацию можно посмотреть на странице программы: http://jodies.de/ipcalc.

port number

Для протоколов, работающих с портами (такие как TCP и UDP), обязательным требованием является указание номера порта соответствующего сервиса. Вместо номера порта можно использовать имя сервиса (из /etc/services).

in | out

Отбор соответственно по входящим и исходящим пакетам. Присутствие одного из этих ключевым слов в правиле обязательно для формирования критерия фильтрации.

via IF

Совпадает с пакетами, проходящими через указанный интерфейс. Ключевое слово `via` включает обязательную проверку на указанном интерфейсе в общий процесс поиска совпадений.

setup

Это обязательное ключевое слово определяет начало запроса сессии для TCP пакетов.

keep-state

Это обязательное ключевое слово. При совпадении межсетевой экран создает динамическое правило, которое по умолчанию будет совпадать с двунаправленным трафиком между отправителем и получателем для данной пары IP/порт по указанному протоколу.

limit {src-addr | src-port | dst-addr | dst-port}

Межсетевой экран разрешит только *N* соединений с одинаковым набором параметров, указанных в правиле. Можно задавать один или несколько адресов и портов отправителя и получателя. В одном и том же правиле использование `limit` и `keep-state` не допускается. Параметр `limit` предоставляет такую же функцию с сохранением состояний, что и `keep-state`, плюс свои собственные.

25.6.5.2. Параметры для правил с сохранением состояния

С точки зрения фильтрации по правилам с сохранением состояния весь трафик выглядит как двусторонний обмен пакетами, включая данные о сессиях. При такой фильтрации у нас есть средства сопоставления и определения корректности процедуры двустороннего обмена пакетами между стороной, породившей пакет, и стороной-получателем. Любые пакеты, которые не подходят под шаблон сессии, автоматически отбрасываются как злонамеренные.

Параметр `check-state` служит для указания места в наборе правил IPFW, в котором пакет будет передан на поиск соответствий динамическим правилам. В случае совпадения пакет пропускается, при этом создается новое динамическое правило для следующего пакета, принадлежащего данной двусторонней сессии. В противном случае пакет движется по обычным правилам, начиная со следующей позиции.

Динамические правила уязвимы к атаке SYN-пакетами, которые могут породить гигантское количество динамических правил. Для предотвращения такого рода атак во FreeBSD предусмотрен еще один параметр - `limit`. Этот параметр служит для ограничения количества одновременно установленных сессий путём проверки полей отправителя и получателя, в зависимости от параметра `limit`, с использованием IP адреса пакета для поиска открытых динамических правил, которые представляют собой счетчик количества совпадений для данного IP адреса и этого правила. Если это количество превышает значение, указанное в параметре `limit`, то такой пакет отбрасывается.

25.6.5.3. Протоколирование сообщений межсетевого экрана

Преимущества протоколирования очевидны: это предоставляет возможность отслеживать постфактум, прохождение каких пакетов было отклонено, откуда эти пакеты пришли и куда они назначались для тех правил, в которых включена функция записи в журнал. Это замечательный инструмент для отслеживания атак на вашу систему.

Даже при включенной функции ведения журнала само по себе оно производиться не будет. Администратор межсетевого экрана определяет, для каких правил будет включена функция ведения журнала, и добавляет к этим правилам `log`. Обычно в журнал пишутся только запрещающие правила, такие как правила deny для входящего ICMP ping. Довольно часто конец списка добавляют дублирующее правило вида "ipfw default deny everything" с приставкой `log`. Это позволяет отслеживать все пакеты, не совпадающие ни с одним из правил в вашем наборе.

Будьте крайне осмотрительны при использовании функции ведения журнала, так как это чревато несоразмерным разрастанием файла журнала, вплоть до полного заполнения им места на жестком диске. DoS атаки, направленные на переполнение свободного пространства жесткого диска, являются одними из самых старейших. Помимо заполнения жесткого диска это неприятно еще и тем, что сообщения журнала пишутся не только в syslogd, но также отображаются на экране системной консоли, и это вскоре начинает сильно раздражать.

Параметр ядра `IPFIREWALL_VERBOSE_LIMIT=5` ограничивает число идущих подряд сообщений в системный регистратор syslogd(8), касающихся пакетов, совпавших с правилом. Когда этот параметр включен в ядро, число последовательно идущих сообщений для определенного правила обрезается указанным числом. От записи 200 идентичных сообщений особого прока нет. В данном случае для сработавшего правила в журнале syslogd будут зафиксированы 5 сообщений подряд, остальные идентичные сообщения будут подсчитаны и отправлены в syslogd как одно сообщение такого вида:

```
last message repeated 45 times
```

Путь к файлу, в который пишутся сообщения, задается в файле /etc/syslog.conf. По умолчанию это файл /var/log/security.

25.6.5.4. Написание скрипта правил

Наиболее опытные пользователи IPFW создают скрипт, содержащий в себе правила, оформленные таким образом, что они могут быть исполнены как обыкновенный sh-скрипт. Основное преимущество такого подхода в том, что правила можно полностью заменить на новые без необходимости в перезагрузке системы для их активации. Это крайне удобно на этапе разработки и тестирования набора правил, т.к. перезагружать весь список правил можно сколь угодно часто. Помимо того, поскольку это скрипт, то здесь можно объявить некие часто используемые значения в виде переменной, и использовать её во множестве правил, как показано в примере ниже.

Синтаксис примера, приведенного ниже, совместим с тремя командными оболочками: sh(1), csh(1), tcsh(1). Для использования значения ранее объявленной переменной имя переменной предваряется символом $. Во время присвоения имя переменной не имеет префикса $, присваиваемое значение должно быть заключено в "двойные кавычки".

Так выглядит файл с правилами, с которого вы можете начать:

```
############### начало примера скрипта с правилами ipfw #############
#
ipfw -q -f flush    # Сброс всех правил.
# Установки по умолчанию
oif="tun0"      # наш интерфейс
odns="192.0.2.11"   # IP DNS сервера провайдера
cmd="ipfw -q add "  # префикс для создания правил
ks="keep-state"     # просто лень вводить каждый раз
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via $oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via $oif $ks
################### конец примера скрипта с правилами ipfw ############
```

Вот и все, что нужно сделать. Сами правила в этом примере не столь важны, они написаны ради того, чтобы продемонстрировать использование подстановки значения переменной по ее имени.

Если бы этот скрипт находился в файле /etc/ipfw.rules, то правила можно было бы перезагрузить следующей командой.

```
# sh /etc/ipfw.rules
```

Имя и расположение файла /etc/ipfw.rules могут быть какими угодно.

Такой же результат можно получить, выполнив вручную следующие команды:

```
# ipfw -q -f flush
# ipfw -q add check-state
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

25.6.5.5. Набор правил с сохранением состояния

Следующий набор правил, не включающий в себя правила трансляции адресов NAT, является примером того, как создавать правила для межсетевого экрана закрытого типа высокого уровня защиты. Закрытый межсетевой экран разрешает трафик, описанный в разрешающих правилах, и по умолчанию блокирует всё остальное. Межсетевой экран, предназначенный для защиты сегментов сети, имеет как минимум два интерфейса, для которых должны быть написаны правила для работы межсетевого экрана.

Все разновидности операционных систем UNIX®, включая FreeBSD, используют интерфейс lo0 и IP адрес `127.0.0.1` для передачи данных внутри операционной системы. Правила межсетевого экрана должны содержать в своем составе правила, разрешающие беспрепятственное прохождение трафика по этому интерфейсу.

Интерфейс, подключенный к Интернет, является местом для размещения правил авторизации и контроля доступа исходящих и входящих соединений. Это может быть туннельный интерфейс PPP tun0 или сетевой адаптер, подключенный к DSL или кабельному модему.

В случае, когда за межсетевым экраном один и более интерфейсов подсоединён к локальной сети, должны присутствовать правила для беспрепятственного прохождения исходящих пакетов с этих интерфейсов LAN.

Правила изначально разделяются на три основных раздела: интерфейсы, не ограниченные правилами, правила для исходящего трафика на внешнем интерфейсе и правила для входящего трафика на внешнем интерфейсе.

В каждом из разделов, относящихся к внешнему интерфейсу, правила должны быть упорядоченны по следующему принципу: наиболее используемые расположены в начале, наименее используемые - в конце. Последним должно идти правило блокирования и занесения в журнал информации о пакетах на этом интерфейсе, не попавших под предыдущие правила.

Раздел, описывающий правила для исходящего трафика на внешнем интерфейсе, содержит только разрешающие правила `allow`, состоящие из значений фильтрации, которые однозначно определяют сервис, которому разрешен доступ в Интернет. Все правила включают в себя поля `proto`, `port`, `in/out`, `via` и `keep state`. Правила, содержащие `proto tcp`, имеют также параметр `setup`, который служит для определения начала сессии, которое в дальнейшем передается как условие срабатывания в динамическую таблицу.

В разделе, описывающем правила для входящего трафика на внешнем интерфейсе, в самом начале должны стоять правила, блокирующие нежелательные пакеты. Так должно быть по двум причинам. Первая состоит в том, что пакеты, сформированные злоумышленником, могут частично или полностью соответствовать разрешающим правилам `allow`. Вторая причина состоит в том, что заведомо не интересующие нас пакеты могут быть просто отклонены, вместо того, чтобы быть перехваченными и записанными в файл журнала по последнему правилу. Последнее правило в каждом разделе блокирует и регистрирует в журнале все пакеты и может быть использовано для юридических обоснований в ходе разбирательств против злоумышленников, атаковавших вашу систему.

Также следует убедиться в том, что ваш сервер не отвечает ни на какие другие формы непредусмотренного трафика. Некорректные пакеты должны быть просто отброшены. В результате атакующие не получат информацию о том, достиг ли его пакет вашего сервера. Чем меньше атакующие будут знать о вашей системе, тем более она защищена. Назначение нераспознанного номера порта можно посмотреть в файле /etc/services/ или по адресу http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers. Рекомендуем ознакомиться с содержимым ссылки относительно номеров портов, используемых троянами: http://www.sans.org/security-resources/idfaq/oddports.php.

25.6.5.6. Пример правил для межсетевого экрана закрытого типа.

Следующие правила, не включающие поддержку NAT, являются логически полным набором правил для межсетевого экрана закрытого типа. При использовании этого набора правил вы вполне можете быть уверены в безопасности вашей системы. Просто закомментируйте некоторые из правил `pass` для тех служб, которые вам не требуются. Чтобы избежать занесения в журнал нежелательных сообщений, добавьте правило `deny` в раздел, описывающий входящий трафик на интерфейсе. Замените название интерфейса dc0, упоминающегося в правилах ниже, на название интерфейса (NIC), который соединяет вашу систему с глобальной сетью. Для PPP соединений это будет tun0.

Примечание по использованию этих правил.

- Все запросы начала сессии с внешней сетью используют параметр `keep-state`.
- Все разрешенные сервисы внешней сети имеют параметр `limit` для защиты от флуда.
- Все правила используют параметры `in` или `out` для указания направления трафика.
- Все правила используют параметр `via` *имя-интерфейса* для уточнения интерфейса, через который проходит пакет.

Следующие правила записываются в /etc/ipfw.rules.

```
################ Начало файла с правилами IPFW ################################
# Сброс всех правил перед началом работы скрипта.
ipfw -q -f flush

# Префикс для создания правил
cmd="ipfw -q add"
pif="dc0"       # название внешнего интерфейса,
            # принадлежащего глобальной сети
```

```
##################################################################
# Нет ограничений на внутреннем интерфейсе локальной сети
# Нет необходимости в этом, если у вас нет локальной сети.
# Замените xl0 на название интерфейса вашей локальной сети
##################################################################
#$cmd 00005 allow all from any to any via xl0

##################################################################
# Нет ограничений на интерфейсе Loopback
##################################################################
$cmd 00010 allow all from any to any via lo0

##################################################################
# Разрешить пакет, если он был ранее добавлен в "динамическую"
# таблицу при помощи выражения allow keep-state
##################################################################
$cmd 00015 check-state

##################################################################
# Раздел правил для исходящего трафика на внешнем интерфейсе
# Анализ запросов начала сессии, идущих из-за межсетевого экрана
# в локальную сеть или от этого шлюза в интернет.
##################################################################

# Разрешить исходящий трафик к DNS серверу провайдера
# x.x.x.x должен быть IP адресом DNS сервера вашего провайдера
# Продублируйте эти строки, если у вас больше одного DNS сервера
# Эти IP адреса можно взять из файла /etc/resolv.conf
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state

# Разрешить исходящий трафик к DHCP серверу провайдера для cable/DSL конфигураций.
# Это правило не нужно для .user ppp. соединений с глобальной сетью
# в этом случае вы можете удалить эти правила.
# Используйте это правило для записи необходимого нам IP адреса в лог-файл.
# Затем укажите IP адрес в закомментированном правиле и удалите первое правило.
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state

# Разрешить исходящий трафик для незащищенного www соединения
$cmd 00200 allow tcp from any to any 80 out via $pif setup keep-state

# Разрешить исходящий трафик для защищенного www соединения
# https с поддержкой TLS и SSL
$cmd 00220 allow tcp from any to any 443 out via $pif setup keep-state

# Разрешить исходящий POP/SMTP
$cmd 00230 allow tcp from any to any 25 out via $pif setup keep-state
$cmd 00231 allow tcp from any to any 110 out via $pif setup keep-state
```

```
# Разрешить исходящий трафик для FreeBSD (make install & CVSUP)
# По сути назначаем пользователю root полные привилегии.
$cmd 00240 allow tcp from me to any out via $pif setup keep-state uid root

# Разрешаем исходящий icmp ping
$cmd 00250 allow icmp from any to any out via $pif keep-state

# Разрешаем исходящий трафик Time
$cmd 00260 allow tcp from any to any 37 out via $pif setup keep-state

# Разрешаем исходящий трафик nntp news
$cmd 00270 allow tcp from any to any 119 out via $pif setup keep-state

# Разрешаем исходящий защищённый трафик FTP, Telnet и SCP
# Эта функция использует SSH (secure shell)
$cmd 00280 allow tcp from any to any 22 out via $pif setup keep-state

# Разрешаем исходящий трафик whois
$cmd 00290 allow tcp from any to any 43 out via $pif setup keep-state

# Запрещаем и заносим в журнал остальной исходящий трафик.
# Обеспечивает политику межсетевого экрана закрытого типа
$cmd 00299 deny log all from any to any out via $pif

################################################################
# Раздел правил для входящего трафика на внешнем интерфейсе
# Анализ пакетов, приходящих из глобальной сети,
# предназначенных для этого шлюза или локальной сети
#######################################################################

# Запрещаем весь входящий трафик с немаршрутизируемых сетей
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif  #RFC 1918 private IP
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif     #RFC 1918 private IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif          #RFC 1918 private IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif        #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif            #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif   #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif       #reserved for docs
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif         #Class D & E multicast

# Запрещаем пинг извне
$cmd 00310 deny icmp from any to any in via $pif

# Запрещаем ident
$cmd 00315 deny tcp from any to any 113 in via $pif

# Запрещаем все Netbios службы. 137=name, 138=datagram, 139=session
# Netbios это MS/Windows сервис обмена.
# Блокируем MS/Windows hosts2 запросы сервера имен на порту 81
$cmd 00320 deny tcp from any to any 137 in via $pif
```

```
$cmd 00321 deny tcp from any to any 138 in via $pif
$cmd 00322 deny tcp from any to any 139 in via $pif
$cmd 00323 deny tcp from any to any 81 in via $pif

# Запрещаем любые опоздавшие пакеты
$cmd 00330 deny all from any to any frag in via $pif

# Запрещаем ACK пакеты, которые не соответствуют динамической таблице правил.
$cmd 00332 deny tcp from any to any established in via $pif

# Разрешаем входящий трафик с DHCP сервера провайдера.  Это правило
# должно содержать IP адрес DHCP сервера вашего провайдера, поскольку
# только ему разрешено отправлять пакеты данного типа.  Необходимо только
# для проводных и DSL соединений.  Для 'user ppp' соединений с глобальной
# сетью использовать это правило нет необходимости.  Это тот же IP адрес,
# выбранный и используемый вами в разделе правил для исходящего трафика.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Разрешить входящий трафик для www, так как я использую сервер apache
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Разрешить входящий трафик безопасных FTP, Telnet и SCP из глобальной сети
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Разрешить входящий нешифрованный трафик Telnet из глобальной сети
# считается небезопасным, потому что ID и PW передаются через глобальную
# сеть в открытом виде.
# Удалите этот шаблон, если вы не используете telnet.
$cmd 00420 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Отбрасываем и заносим в журнал все входящие соединения снаружи
$cmd 00499 deny log all from any to any in via $pif

# Всё остальное запрещено по умолчанию
# Запрещаем и заносим в журнал все пакеты для дальнейшего анализа
$cmd 00999 deny log all from any to any
################ Конец файла правил IPFW ################################
```

25.6.5.7. Пример правил с сохранением состояний и поддержкой NAT

Здесь перечислены некоторые дополнительные конфигурационные параметры, которые нужно включить, чтобы активировать функцию NAT в IPFW. В файл конфигурации ядра к остальным параметрам IPFIREWALL нужно добавить строку `option IPDIVERT`.

В дополнение к обычным параметрам IPFW в /etc/rc.conf добавим следующее:

```
natd_enable="YES"       # Включить функцию NATD
natd_interface="rl0"        # Название внешнего сетевого интерфейса
natd_flags="-dynamic -m"    # -m = по возможности сохранить номера портов
```

Использование динамических правил с правилом `divert natd` (Network Address Translation) значительно затрудняет логику составления правил. Расположение `check-state` и `divert natd` в таблице правил влияет на поведение межсетевого экрана. Это уже не просто последовательный логический поток. При применении вышеозначенных параметров становится доступным новый тип действия `skipto`. При использовании `skipto` нумерация правил становится обязательной. В качестве аргумента `skipto` используется номер правила, к которому нужно перейти.

Ниже последует пример метода кодирования, не снабженный комментариями, приведенный здесь для внесения ясности относительно последовательности прохождения пакетов через набор правил.

Обработка правил начинается с первого по счету и идет последовательно, по правилу за раз, до достижения конца файла, либо если проверяемый пакет соответствует критериям фильтрации; в последнем случае пакет покидает межсетевой экран. Для правил под номерами 100, 101, 450, 500 и 510 важен порядок их расположения. Эти правила управляют трансляцией исходящих и входящих пакетов, таким образом в таблицу keep-state заносятся только приватные IP адреса локальной сети. Обратите внимание, что все правила allow и deny указывают направление, по которому передается пакет (исходящее или входящее) и сетевой интерфейс. Также стоит отметить, что все запросы начала исходящей сессии передаются с использованием `skipto rule 500` для трансляции адресов.

Предположим, что пользователь локальной сети запрашивает страницу через браузер. Веб-страницы передаются по порту 80. Пакет входит в межсетевой экран. Этот пакет не попадает под правило 100, потому что в критериях фильтрации этого правила указан параметр `in`. Этот пакет не попадает под правило 101, потому что это первый пакет сессии и он еще не был занесен в динамическую таблицу keep-state. Достигнув, наконец, правила 125, пакет удовлетворяет всем критериям фильтрации. Этот пакет является выходящим из интерфейса, взаимодействующим с глобальной сетью. На данном этапе у пакета в качестве исходящего адреса всё еще указан приватный IP адрес локальной сети. По условию этого правила к пакету применяются два действия. Параметр `keep-state` создаст новую запись в динамической таблице keep-state, и выполнится действие, указанное в правиле. Указанное действие является частью информации, заносимой в динамическую таблицу. В данном случае это `skipto rule 500`. Правило 500 транслирует (NAT) адреса пакета и отпускает его наружу. Данное замечание очень важно. Этот пакет идет к цели, где генерируется ответный пакет и отправляется обратно. Этот новый пакет входит в начало списка правил. На этот раз пакет соответствует правилу 100 и его IP адрес назначения транслируется обратно на соответствующий IP адрес локальной сети. Затем он обрабатывается правилом `check-state`, и поскольку для него уже присутствует в динамической таблице правило, соответствующее данной сессии, пакет пропускается в локальную сеть. Дальше пакет приходит к отправившему его компьютеру локальной сети, и генерируется новый пакет, запрашивающий новую порцию данных с удаленного сервера. На этот раз пакет сразу проверяется правилом `check-state`, и в случае присутствия исходящей записи данного пакета выполняется действие `skipto 500`. Пакет переходит к правилу 500, транслируется и пропускается во внешнюю сеть.

Для входящего трафика все пакеты, являющиеся частью уже установленной сессии, автоматически разбираются правилом `check-state` и правильно расположенными правилами `divert natd`. Всё, что нам остается сделать, это запретить все плохие пакеты и

разрешить прохождение внутрь сети пакетов только для разрешенных сервисов. Допустим, на сервере с межсетевым экраном запущен apache, и мы хотим разрешить людям из глобальной сети доступ на локальный веб-сайт. Новый входящий пакет, запрашивающий начало сессии, соответствует правилу 100, и его IP адрес транслируется как локальный IP системы с межсетевым экраном. Далее пакет проверяется на соответствие вредоносному трафику и в случае отсутствия соответствия попадает на правило 425. В случае соответствия данному правилу происходят две вещи. Пакет правил помещается в динамическую таблицу keep-state, но в этот раз любая новая сессия запросов, порожденных с этого IP, ограничена 2 одновременными соединениями. Это защищает от перегрузки сервис, работающей по указанному номеру порта. В качестве действия в правиле указан `allow`, следовательно пакет пропускается в локальную сеть. Пакет, сформированный в качестве ответа, попадает под `check-state` и распознается им как принадлежащий существующей сессии. Далее он передаётся на правило 500, где происходит обратная трансляция, после чего пакет пропускается на внешний интерфейс.

Пример файла правил #1:

```
#!/bin/sh
cmd="ipfw -q add"
skip="skipto 500"
pif=rl0
ks="keep-state"
good_tcpo="22,25,37,43,53,80,443,110,119"

ipfw -q -f flush

$cmd 002 allow all from any to any via xl0  # разрешаем трафик на локальном интерфейсе
$cmd 003 allow all from any to any via lo0  # разрешаем трафик на интерфейсе loopback

$cmd 100 divert natd ip from any to any in via $pif
$cmd 101 check-state

# Разрешенные исходящие пакеты
$cmd 120 $skip udp from any to xx.168.240.2 53 out via $pif $ks
$cmd 121 $skip udp from any to xx.168.240.5 53 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
$cmd 135 $skip udp from any to any 123 out via $pif $ks

# Запрещаем весь входящий трафик с немаршрутизируемых адресных пространств
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 для локальных IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 для локальных IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 для локальных IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP авто-конфигурации
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #Зарезервировано для
документации
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Class D & E multicast
```

```
# Разрешаем входящие пакеты
$cmd 400 allow udp from xx.70.207.54 to any 68 in $ks
$cmd 420 allow tcp from any to me 80 in via $pif setup limit src-addr 1

$cmd 450 deny log ip from any to any

# Раздел skipto для правил с сохранением состояния для исходящих пакетов
$cmd 500 divert natd ip from any to any out via $pif
$cmd 510 allow ip from any to any

######################### Окончание файла правил ##################
```

Следующий пример во многом повторяет то, что приведено выше, но использует самодокументирующий стиль записи с исчерпывающими комментариями для того, чтобы помочь начинающему составителю правил IPFW лучше понимать, для чего предназначено то или иное правило.

Пример файла правил #2:

```
#!/bin/sh
################ Начало файла правил IPFW ##############################
# Сброс всех правил перед началом работы скрипта.
ipfw -q -f flush

# Задание стандартных переменных
cmd="ipfw -q add"
skip="skipto 800"
pif="rl0"       # название внешнего интерфейса,
            # принадлежащего глобальной сети

#################################################################
# Нет ограничений на внутреннем интерфейсе локальной сети
# Замените xl0 на название интерфейса вашей локальной сети
#################################################################
$cmd 005 allow all from any to any via xl0

#################################################################
# Нет ограничений на интерфейсе Loopback
#################################################################
$cmd 010 allow all from any to any via lo0

#################################################################
# Трансляция адреса, если пакет является входящим
#################################################################
$cmd 014 divert natd ip from any to any in via $pif

#################################################################
# Разрешить пакет, если он был ранее добавлен в динамическую
# таблицу при помощи выражения allow keep-state
```

```
##################################################################
$cmd 015 check-state

##################################################################
# Раздел правил для исходящего трафика на внешнем интерфейсе
# Анализ запросов начала сессии, идущих из-за межсетевого экрана
# в локальную сеть или от этого шлюза в интернет.
##################################################################

# Разрешить исходящий трафик к DNS серверу провайдера
# x.x.x.x должен быть IP адресом DNS сервера вашего провайдера
# Продублируйте эти строки, если у вас больше одного DNS сервер
# Эти IP адреса можно взять из файла /etc/resolv.conf
$cmd 020 $skip tcp from any to x.x.x.x 53 out via $pif setup keep-state

# Разрешить исходящий трафик к DHCP серверу провайдера для cable/DSL конфигураций.
$cmd 030 $skip udp from any to x.x.x.x 67 out via $pif keep-state

# Разрешить исходящий трафик для незащищенного www соединения
$cmd 040 $skip tcp from any to any 80 out via $pif setup keep-state

# Разрешить исходящий трафик для защищенного www соединения
# https с поддержкой TLS и SSL
$cmd 050 $skip tcp from any to any 443 out via $pif setup keep-state

# Разрешить исходящий POP/SMTP
$cmd 060 $skip tcp from any to any 25 out via $pif setup keep-state
$cmd 061 $skip tcp from any to any 110 out via $pif setup keep-state

# Разрешить исходящий трафик для FreeBSD (make install & CVSUP)
# По сути назначаем пользователю root полные привилегии.
$cmd 070 $skip tcp from me to any out via $pif setup keep-state uid root

# Разрешаем исходящий icmp ping
$cmd 080 $skip icmp from any to any out via $pif keep-state

# Разрешаем исходящий трафик Time
$cmd 090 $skip tcp from any to any 37 out via $pif setup keep-state

# Разрешаем исходящий трафик nntp news (т.е. news groups)
$cmd 100 $skip tcp from any to any 119 out via $pif setup keep-state

# Разрешаем исходящий защищённый трафик FTP, Telnet и SCP
# Эта функция использует SSH (secure shell)
$cmd 110 $skip tcp from any to any 22 out via $pif setup keep-state

# Разрешаем исходящий трафик whois
$cmd 120 $skip tcp from any to any 43 out via $pif setup keep-state

# Разрешаем исходящий трафик ntp
$cmd 130 $skip udp from any to any 123 out via $pif keep-state
```

```
##################################################################
# Раздел правил для входящего трафика на внешнем интерфейсе
# Анализ пакетов, приходящих из глобальной сети,
# предназначенных для этого шлюза или локальной сети
##################################################################

# Запрещаем весь входящий трафик с немаршрутизируемых сетей
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 private IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 private IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 private IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #reserved for docs
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Class D & E multicast

# Запрещаем ident
$cmd 315 deny tcp from any to any 113 in via $pif

# Запрещаем все Netbios службы. 137=name, 138=datagram, 139=session
# Netbios это MS/Windows сервис обмена.
# Блокируем MS/Windows hosts2 запросы сервера имен на порту 81
$cmd 320 deny tcp from any to any 137 in via $pif
$cmd 321 deny tcp from any to any 138 in via $pif
$cmd 322 deny tcp from any to any 139 in via $pif
$cmd 323 deny tcp from any to any 81  in via $pif

# Запрещаем любые опоздавшие пакеты
$cmd 330 deny all from any to any frag in via $pif

# Запрещаем ACK пакеты, которые не соответствуют динамической таблице правил.
$cmd 332 deny tcp from any to any established in via $pif

# Разрешаем входящий трафик с DHCP сервера провайдера. Это правило
# должно содержать IP адрес DHCP сервера вашего провайдера, поскольку
# только ему разрешено отправлять пакеты данного типа. Необходимо только
# для проводных и DSL соединений. Для 'user ppp' соединений с глобальной
# сетью использовать это правило нет необходимости. Это тот же IP адрес,
# выбранный и используемый вами в разделе правил для исходящего трафика.
$cmd 360 allow udp from x.x.x.x to any 68 in via $pif keep-state

# Разрешить входящий трафик для www, т.к. я использую Apache сервер.
$cmd 370 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Разрешить входящий трафик безопасных FTP, Telnet и SCP из глобальной сети
$cmd 380 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Разрешить входящий нешифрованный трафик Telnet из глобальной сети
# считается небезопасным, потому что ID и PW передаются через глобальную
```

```
# сеть в открытом виде.
# Удалите этот шаблон, если вы не используете telnet.
$cmd 390 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Отбрасываем и заносим в журнал все неразрешенные входящие соединения из глобальной
сети
$cmd 400 deny log all from any to any in via $pif

# Отбрасываем и заносим в журнал все неразрешенные исходящие соединения в глобальную
сеть
$cmd 450 deny log all from any to any out via $pif

# Место для skipto в правилах с сохранением состояния для исходящих соединений
$cmd 800 divert natd ip from any to any out via $pif
$cmd 801 allow ip from any to any

# Всё остальное запрещено по умолчанию
# Запрещаем и заносим в журнал все пакеты для дальнейшего анализа
$cmd 999 deny log all from any to any
################ Окончание файла правил IPFW ################################
```

Глава 26. Сложные вопросы работы в сети

26.1. Краткий обзор

Эта глава охватывает множество различных сетевых тематик повышенной сложности.

После чтения этой главы вы будете знать:

- Основные понятия о маршрутизации и маршрутах.
- Как настроить IEEE 802.11 и Bluetooth®.
- Как заставить FreeBSD работать в качестве сетевого моста.
- Как настроить загрузку по сети для бездисковой машины.
- Как настроить трансляцию сетевых адресов.
- Как соединить два компьютера посредством PLIP.
- Как настроить IPv6 на машине FreeBSD.
- Как настроить ATM.

Перед чтением этой главы вы должны:

- Понимать основы работы скриптов /etc/rc.
- Свободно владеть основными сетевыми терминами.
- Знать как настраивать и устанавливать новое ядро FreeBSD (Настройка ядра FreeBSD).
- Знать как устанавливать дополнительное программное обеспечение сторонних разработчиков (Установка приложений. порты и пакеты).

26.2. Сетевые шлюзы и маршруты

Чтобы некоторая машина могла найти в сети другую, должен иметься механизм описания того, как добраться от одной машине к другой. Такой механизм называется *маршрутизацией*. "Маршрут" задаётся парой адресов: "адресом назначения" (destination) и "сетевым шлюзом" (gateway). Эта пара указывает на то, что если Вы пытаетесь соединиться с *адресом назначения*, то вам нужно устанавливать связь через "сетевой шлюз". Существует три типа адресов назначения: отдельные хосты, подсети и "маршрут по умолчанию" (default). "Маршрут по умолчанию" (default route) используется, если не подходит ни один из других маршрутов. Мы поговорим немного подробнее о маршрутах по умолчанию позже. Также имеется и три типа сетевых шлюзов: отдельные хосты, интерфейсы (также называемые "подключениями" (links)) и аппаратные адреса Ethernet (MAC-адреса).

26.2.1. Пример

Для иллюстрации различных аспектов маршрутизации мы будем использовать следующий пример использования команды `netstat`:

```
% netstat -r
Routing tables

Destination      Gateway            Flags     Refs     Use     Netif Expire

default      outside-gw     UGSc       37      418  ppp0
localhost    localhost      UH      0      181   lo0
test0        0:e0:b5:36:cf:4f   UHLW    5    63288   ed0     77
10.20.30.255     link#1         UHLW    1     2421
example.com  link#1         UC      0    0
host1        0:e0:a8:37:8:1e    UHLW    3     4601   lo0
host2        0:e0:a8:37:8:1e    UHLW    0    5   lo0 =>
host2.example.com link#1        UC      0    0
224      link#1         UC      0    0
```

В первых двух строках задаются маршрут по умолчанию (который будет описан в следующем разделе) и маршрут на `localhost`.

Интерфейс (колонка `Netif`), который указан в этой таблице маршрутов для использования с `localhost` и который назван lo0, имеет также второе название, устройство loopback. Это значит сохранение всего трафика для указанного адреса назначения внутри, без посылки его по сети, так как он все равно будет направлен туда, где был создан.

Следующими выделяющимися адресами являются адреса, начинающиеся с `0:e0:…`. Это аппаратные адреса Ethernet, или MAC-адреса. FreeBSD будет автоматически распознавать любой хост (в нашем примере это `test0`) в локальной сети Ethernet и добавит маршрут для этого хоста, указывающий непосредственно на интерфейс Ethernet, ed0. С этим типом маршрута также связан параметр таймаута (колонка `Expire`), используемый в случае неудачной попытки услышать этот хост в течении некоторого периода времени. Если такое происходит, то маршрут до этого хоста будет автоматически удалён. Такие хосты поддерживаются при помощи механизма, известного как RIP (Routing Information Protocol), который вычисляет маршруты к хостам локальной сети при помощи определения кратчайшего расстояния.

FreeBSD добавит также все маршруты к подсетям для локальных подсетей (`10.20.30.255` является широковещательным адресом для подсети `10.20.30`, а имя `example.com` является именем домена, связанным с этой подсетью). Назначение `link#1` соответствует первому адаптеру Ethernet в машине. Отметьте отсутствие дополнительного интерфейса для этих строк.

В обеих этих группах (хосты и подсети локальной сети) маршруты конфигурируются автоматически даемоном, который называется routed. Если он не запущен, то будут существовать только статически заданные (то есть введенные явно) маршруты.

Строка `host1` относится к нашему хосту, который известен по адресу Ethernet. Так как мы являемся посылающим хостом, FreeBSD знает, что нужно использовать loopback-интерфейс (lo0) вместо того, чтобы осуществлять посылку в интерфейс Ethernet.

Две строки `host2` являются примером того, что происходит при использовании алиасов в

команде ifconfig(8) (обратитесь к разделу об Ethernet для объяснения того, почему мы это делаем). Символ ⇒ после интерфейса lo0 указывает на то, что мы используем не просто интерфейс loopback (так как это адрес, обозначающий локальный хост), но к тому же это алиас. Такие маршруты появляются только на хосте, поддерживающем алиасы; для всех остальных хостов в локальной сети для таких маршрутов будут показаны просто строчки `link#1`.

Последняя строчка (подсеть назначения `224`) имеет отношение к многоадресной посылке, которая будет рассмотрена в другом разделе.

И наконец, различные атрибуты каждого маршрута перечисляются в колонке `Flags`. Ниже приводится краткая таблица некоторых из этих флагов и их значений:

U	Up: Маршрут актуален.
H	Host: Адресом назначения является отдельный хост.
G	Gateway: Посылать все для этого адреса назначения на указанную удаленную систему, которая будет сама определять дальнейший путь прохождения информации.
S	Static: Маршрут был настроен вручную, а не автоматически сгенерирован системой.
C	Clone: Новый маршрут сгенерирован на основе указанного для машин, к которым мы подключены. Такой тип маршрута обычно используется для локальных сетей.
W	WasCloned: Указывает на то, что маршрут был автоматически сконфигурирован на основе маршрута в локальной сети (Clone).
L	Link: Маршрут включает ссылку на аппаратный адрес Ethernet.

26.2.2. Маршруты по умолчанию

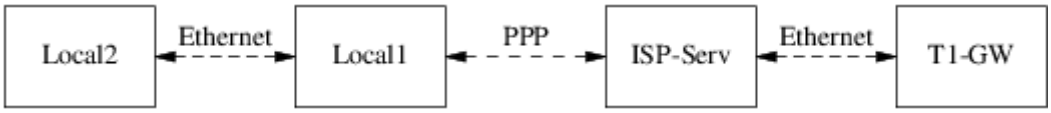
Когда локальной системе нужно установить соединение с удаленным хостом, она обращается к таблице маршрутов для того, чтобы определить, существует ли такой маршрут. Если удаленный хост попадает в подсеть, для которой известен способ ее достижения (маршруты типа Cloned), то система определяет возможность подключиться к ней по этому интерфейсу.

Если все известные маршруты не подходят, у системы имеется последняя возможность: маршрут "default". Это маршрут с особым типом сетевого шлюза (обычно единственным, присутствующим в системе), и в поле флагов он всегда помечен как `c`. Для хостов в локальной сети этот сетевой шлюз указывает на машину, имеющую прямое подключение к внешнему миру (неважно, используется ли связь по протоколу PPP, канал DSL, кабельный

модем, T1 или какой-то другой сетевой интерфейс).

Если вы настраиваете маршрут по умолчанию на машине, которая сама является сетевым шлюзом во внешний мир, то маршрутом по умолчанию будет являться сетевой шлюз у Вашего провайдера Интернет (ISP).

Давайте взглянем на примеры маршрутов по умолчанию. Вот типичная конфигурация:



Хосты `Local1` и `Local2` находятся в нашей сети. `Local1` подключён к ISP через коммутируемое соединение по протоколу PPP. Этот компьютер с сервером PPP подключён посредством локальной сети к другому шлюзовому компьютеру через внешний интерфейс самого ISP к Интернет.

Маршруты по умолчанию для каждой из ваших машин будут следующими:

Хост	Маршрут по умолчанию	Интерфейс
Local2	Local1	Ethernet
Local1	T1-GW	PPP

Часто задаётся вопрос "Почему (или каким образом) в качестве шлюза по умолчанию для машины `Local1` мы указываем `T1-GW`, а не сервер провайдера, к которому подключаемся?".

Запомните, что из-за использования PPP-интерфейсом адреса в сети провайдера Интернет с вашей стороны соединения, маршруты для всех других машин в локальной сети провайдера будут сгенерированы автоматически. Таким образом, вы уже будете знать, как достичь машины `T1-GW`, так что нет нужды в промежуточной точке при посылке трафика к серверу ISP.

В локальных сетях адрес `X.X.X.1` часто используется в качестве адреса сетевого шлюза. Тогда (при использовании того же самого примера) если пространство адресов класса C вашей локальной сети было задано как `10.20.30`, а ваш провайдер использует `10.9.9`, то маршруты по умолчанию будут такие:

Хост	Маршрут по умолчанию
Local2 (10.20.30.2)	Local1 (10.20.30.1)
Local1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

Вы можете легко задать используемый по умолчанию маршрутизатор посредством файла /etc/rc.conf. В нашем примере на машине `Local2` мы добавили такую строку в файл /etc/rc.conf:

```
defaultrouter="10.20.30.1"
```

Это также возможно сделать и непосредственно из командной строки при помощи

команды route(8):

```
# route add default 10.20.30.1
```

Для получения дополнительной информации об управлении таблицами маршрутизации обратитесь к справочной странице по команде route(8).

26.2.3. Хосты с двойным подключением

Есть еще один тип подключения, который мы должны рассмотреть, и это случай, когда хост находится в двух различных сетях. Технически, любая машина, работающая как сетевой шлюз (в примере выше использовалось PPP-соединение), считается хостом с двойным подключением. Однако этот термин реально используется для описания машины, находящейся в двух локальных сетях.

В одном случае у машины имеется два адаптера Ethernet, каждый имеющий адрес в разделенных подсетях. Как альтернативу можно рассмотреть вариант с одним Ethernet-адаптером и использованием алиасов в команде ifconfig(8). В первом случае используются два физически разделённые сети Ethernet, в последнем имеется один физический сегмент сети, но две логически разделённые подсети.

В любом случае таблицы маршрутизации настраиваются так, что для каждой подсети эта машина определена как шлюз (входной маршрут) в другую подсеть. Такая конфигурация, при которой машина выступает в роли маршрутизатора между двумя подсетями, часто используется, если нужно реализовать систему безопасности на основе фильтрации пакетов или функций брандмауэра в одном или обоих направлениях.

Если вы хотите, чтобы эта машина действительно перемещала пакеты между двумя интерфейсами, то вам нужно указать FreeBSD на включение этой функции. Обратитесь к следующей главе, чтобы узнать, как это сделать.

26.2.4. Построение маршрутизатора

Сетевой маршрутизатор является обычной системой, которая пересылает пакеты с одного интерфейса на другой. Стандарты Интернет и хорошая инженерная практика не позволяют Проекту FreeBSD включать эту функцию по умолчанию во FreeBSD. Вы можете включить эту возможность, изменив значение следующей переменной в `YES` в файле rc.conf(5):

```
gateway_enable=YES          # Set to YES if this host will be a gateway
```

Этот параметр изменит значение sysctl(8)-переменной `net.inet.ip.forwarding` в `1`. Если вам временно нужно выключить маршрутизацию, вы можете на время сбросить это значение в `0`.

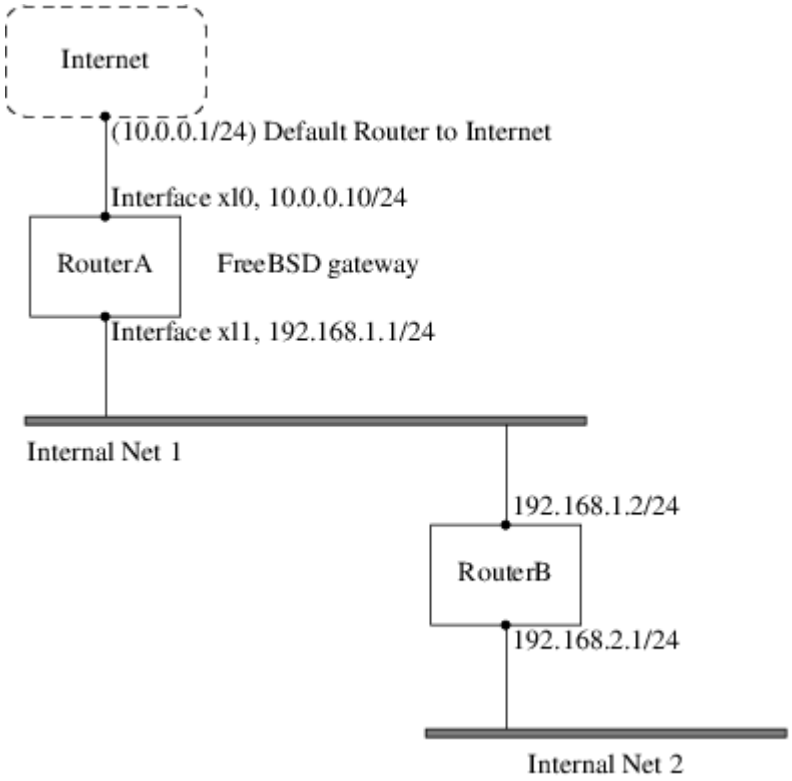
Вашему новому маршрутизатору нужна информация о маршрутах для того, чтобы знать, куда пересылать трафик. Если ваша сеть достаточно проста, то вы можете использовать статические маршруты. С FreeBSD также поставляется стандартный даемон BSD для

маршрутизации routed(8), который умеет работать с RIP (как версии 1, так и версии 2) и IRDP. Поддержка BGP v4, OSPF v2 и других сложных протоколов маршрутизации имеется в пакете net/zebra. Также существуют и коммерческие продукты, применяемые как более комплексное решение проблемы маршрутизации в сети, такие как GateD®.

26.2.5. Настройка статических маршрутов

26.2.5.1. Ручная настройка

Предположим, что у нас есть следующая сеть:



В этом сценарии, `RouterA` это наш компьютер с FreeBSD, который выступает в качестве маршрутизатора в сеть Интернет. Его маршрут по умолчанию настроен на `10.0.0.1`, что позволяет ему соединяться с внешним миром. Мы будем предполагать, что `RouterB` уже правильно настроен и знает все необходимые маршруты (на этом рисунке все просто; добавьте на `RouterB` маршрут по умолчанию, используя `192.168.1.1` в качестве шлюза).

Если мы посмотрим на таблицу маршрутизации `RouterA`, то увидим примерно следующее:

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0/24          link#1             UC          0        0    xl0
192.168.1/24       link#2             UC          0        0    xl1
```

С текущей таблицей маршрутизации `RouterA` не сможет достичь внутренней сети 2 (Internal Net 2). Один из способов обхода этой проблемы - добавление маршрута вручную. Следующая команда добавляет внутреннюю сеть 2 к таблице маршрутизации `RouterA` с `192.168.1.2` в качестве следующего узла:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Теперь `RouterA` сможет достичь любого хоста в сети `192.168.2.0/24`.

26.2.5.2. Постоянная конфигурация

Предыдущий пример прекрасно подходит для настройки статического маршрута в работающей системе. Однако, проблема заключается в том, что маршрутная информация не сохранится после перезагрузки FreeBSD. Способ сохранения добавленного маршрута заключается в добавлении его в файл /etc/rc.conf:

```
# Добавление статического маршрута в Internal Net 2
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

В переменной `static_routes` находятся строки, разделенные пробелами. Каждая строка означает имя маршрута. В примере выше в `static_routes` есть только одна строка, это *internalnet2*. Затем мы добавили переменную `route__internalnet2_`, куда помещены все параметры, которые необходимо передать команде route(8). В примере выше была использована команда:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

поэтому нам потребуется `"-net 192.168.2.0/24 192.168.1.2"`.

Как было сказано выше, мы можем добавить в `static_routes` более чем одну строку. Это позволит создать несколько статических маршрутов. В следующем примере показано добавление маршрутов для сетей `192.168.0.0/24` и `192.168.1.0/24` (этот маршрутизатор не показан на рисунке выше:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

26.2.6. Распространение маршрутов

Мы уже говорили о том, как мы задаем наши маршруты во внешний мир, но не упоминали о том, как внешний мир находит нас.

Мы уже знаем, что таблицы маршрутизации могут быть настроены так, что весь трафик

для некоторого диапазона адресов (в нашем примере это подсеть класса C) может быть направлен заданному хосту в той сети, которая будет перенаправлять входящие пакеты дальше.

При получении адресного пространства, выделенного Вашей сети, Ваш провайдер настроит свои таблицы маршрутизации так, что весь трафик для Вашей подсети будет пересылаться по PPP-соединению к Вашей сети. Но как серверы по всей стране узнают, что Ваш трафик нужно посылать Вашему ISP?

Существует система (подобная распределению информации DNS), которая отслеживает все назначенные пространства адресов и определяет точку подключения к магистрали Интернет. "Магистралью" называют главные каналы, по которым идет трафик Интернет внутри страны и по всему миру. Каждая магистральная машина имеет копию основного набора таблиц, согласно которой трафик для конкретной сети направляется по конкретному магистральному каналу, и затем, передаваясь по цепочке провайдеров, он достигает вашей сети.

Задачей вашего провайдера является объявить на магистрали о том, что он отвечает за подключение (и поэтому на него указывает маршрут) вашей сети. Этот процесс называется распространением маршрута.

26.2.7. Устранение неполадок

Иногда с распространением маршрута возникают проблемы, и некоторые сайты не могут к вам подключиться. Наверное, самой полезной командой для определения точки неверной работы маршрутизации является traceroute(8). Она также полезна и когда вы сами не можете подключиться к удаленной машине (то есть команда ping(8) не срабатывает).

Команда traceroute(8) запускается с именем удаленного хоста, с которым вы хотите установить соединение, в качестве параметра. Она показывает промежуточные сетевые шлюзы по пути следования, в конце концов достигая адрес назначения или прерывая свою работу из-за отсутствия соединения.

За дополнительной информацией обратитесь к странице Справочника по traceroute(8).

26.2.8. Маршрутизация многоадресного трафика

FreeBSD изначально поддерживает как приложения, работающие с многоадресным трафиком, так и его маршрутизацию. Такие приложения не требуют особой настройки FreeBSD; обычно они работают сразу. Для маршрутизации многоадресного трафика требуется, чтобы поддержка этого была включена в ядро:

```
options MROUTING
```

Кроме того, даемон многоадресной маршрутизации, mrouted(8), должен быть настроен посредством файла /etc/mrouted.conf на использование туннелей и DVMRP. Дополнительную информацию о настройки многоадресного трафика можно найти на страницах справочной системы, посвящённых даемону mrouted(8).

26.3. Беспроводные сети

26.3.1. Введение

Было бы весьма полезным иметь возможность использовать компьютер без хлопот, связанных с постоянно подключенным сетевым кабелем. FreeBSD может использоваться как клиент беспроводной сети, и даже в качестве "точки доступа" к ней.

26.3.2. Режимы работы беспроводной связи

Существуют два варианта конфигурации устройств беспроводного доступа 802.11: BSS и IBSS.

26.3.2.1. Режим BSS

Режим BSS является наиболее часто используемым. Режим BSS также называют режимом инфраструктуры. В этом режиме несколько точек доступа беспроводной сети подключаются к проводной сети передачи данных. Каждое беспроводная сеть имеет собственное имя. Это имя является идентификатором SSID сети.

Клиенты беспроводной сети подключаются к этим точкам доступа беспроводной сети. Стандарт IEEE 802.11 определяет протокол, используемый для связи в беспроводных сетях. Клиент сети беспроводного доступа может подключаться к некоторой сети, если задан её SSID. Клиент может также подключаться к любой сети, если SSID не задан.

26.3.2.2. Режим IBSS

Режим IBSS, также называемый ad-hoc, предназначен для соединений точка-точка. На самом деле существуют два типа режима ad-hoc. Один из них является режимом IBSS, называемый также режимом ad-hoc или IEEE ad-hoc. Этот режим определён стандартами IEEE 802.11. Второй режим называется демонстрационным режимом ad-hoc, или Lucent ad-hoc (или, иногда неправильно, режимом ad-hoc). Это старый, существовавший до появления 802.11, режим ad-hoc, и он должен использоваться только для старых сетей. В дальнейшем мы не будем рассматривать ни один из режимов ad-hoc.

26.3.3. Режим инфраструктуры

26.3.3.1. Точки доступа

Точки доступа представляют собой беспроводные сетевые устройства, позволяющие одному или большему количеству клиентов беспроводной сети использовать эти устройства в качестве центрального сетевого концентратора. При использовании точки доступа все клиенты работают через неё. Зачастую используются несколько точек доступа для полного покрытия беспроводной сетью некоторой зоны, такой, как дом, офис или парк.

Точки доступа обычно имеют несколько подключений к сети: адаптер беспроводной связи и один или большее количество сетевых ethernet-адаптеров для подключения к остальной части сети.

Точки доступа могут быть либо приобретены уже настроенными, либо вы можете создать собственную при помощи FreeBSD и поддерживаемого адаптера беспроводной связи. Несколько производителей выпускают точки беспроводного доступа и адаптеры беспроводной связи с различными возможностями.

26.3.3.2. Построение точки доступа с FreeBSD

26.3.3.2.1. Требования

Для того, чтобы создать беспроводную точку доступа на FreeBSD, вам нужно иметь совместимый адаптер беспроводной связи. На данный момент поддерживаются адаптеры только на основе набора микросхем Prism. Вам также потребуется поддерживаемый FreeBSD адаптер проводной сети (найти такой будет нетрудно, FreeBSD поддерживает множество различных устройств). В этом руководстве мы будем полагать, что вы будете строить сетевой мост (bridge(4)) для пропуска всего трафика между устройством беспроводной связи и сетью, подключенной к обычному Ethernet-адаптеру.

Функциональность hostap, которая используется FreeBSD для организации точки доступа, работает лучше всего с некоторыми версиями микрокода. Адаптеры Prism 2 должны использовать микрокод версии 1.3.4 или более новый. Адаптеры Prism 2.5 и Prism 3 должны использовать микрокод версии 1.4.9. Более старые версии микрокода могут работать нормально, а могут и некорректно. В настоящее время единственным способом обновления адаптеров является использование утилит обновления для Windows®, которые можно получить у производителя ваших адаптеров.

26.3.3.2.2. Настройка

Первым делом убедитесь, что ваша система распознаёт адаптер беспроводной связи:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
        inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
        ether 00:09:2d:2d:c9:50
        media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
        status: no carrier
        ssid ""
        stationname "FreeBSD Wireless node"
        channel 10 authmode OPEN powersavemode OFF powersavesleep 100
        wepmode OFF weptxkey 1
```

На данном этапе не беспокойтесь о деталях, просто убедитесь, что выдаётся нечто, указывающее на установленный адаптер беспроводной связи. Если при этом у вас есть проблемы с недоступностью интерфейса беспроводной связи, и вы используете PC Card, то обратитесь к страницам справочной системы, описывающим pccardc(8) и pccardd(8) для получения более полной информации.

Теперь вам нужно загрузить модуль для подготовки той части FreeBSD, что отвечает за организацию сетевых мостов, для работы с точкой доступа. Для загрузки модуля bridge(4)

просто выполните следующую команду:

```
# kldload bridge
```

При загрузке модуля никаких сообщений об ошибках быть не должно. Если это всё же произошло, вам может потребоваться вкомпилировать код для модуля bridge(4) в ядро. В этом вам должен помочь раздел этого Руководства об организации сетевых мостов.

Теперь, когда вы завершили с той частью, что касается организации сетевого моста, нам нужно указать ядру FreeBSD, какие интерфейсы должны объединяться в сетевом мосте. Это мы делаем при помощи sysctl(8):

```
# sysctl net.link.ether.bridge.enable=1
# sysctl net.link.ether.bridge.config="wi0 xl0"
# sysctl net.inet.ip.forwarding=1
```

В версиях FreeBSD, предшествующих 5.2, вместо указанных нужно использовать следующие параметры:

```
# sysctl net.link.ether.bridge=1
# sysctl net.link.ether.bridge_cfg="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

Теперь необходимо настроить адаптер беспроводной сети. Следующая команда заставит адаптер работать в режиме точки доступа:

```
# ifconfig wi0 ssid my_net channel 11 media DS/11Mbps mediaopt hostap up stationname
"FreeBSD AP"
```

Строчка ifconfig(8) активизирует интерфейс wi0, конфигурирует его SSID как *my_net*, а имя станции как *FreeBSD AP*. `media DS/11Mbps` переводит адаптер в режим 11Mbps и нужен только для того, чтобы сработал параметр `mediaopt`. Параметр `mediaopt hostap` переводит интерфейс в режим точки доступа. Параметр `channel 11` задаёт использование канала 802.11b. Страница справки по команде wicontrol(8) перечисляет корректные значения каналов для ваших нужд.

Теперь у вас должна получиться полнофункциональная работающая точка доступа. Настоятельно советуем прочесть страницы справочной по wicontrol(8), ifconfig(8), и wi(4) для получения дополнительной информации.

Также полагаем, что вы прочтёте следующий раздел о шифровании.

26.3.3.2.3. Информация о состоянии

После того, как точка доступа сконфигурирована и начала свою работу, операторам может понадобиться видеть клиентов, связанных с этой точкой. В любой момент оператор может

набрать:

```
# wicontrol -l
1 station:
00:09:b7:7b:9d:16  asid=04c0, flags=3<ASSOC,AUTH>, caps=1<ESS>, rates
=f<1M,2M,5.5M,11M>, sig=38/15
```

Это показывает, что имеется одна связанная станция с перечисленными характеристиками. Выдаваемое значение сигнала должно использоваться только как сравнительный индикатор его силы. Его перевод в dBm или другие единицы измерения различаются в разных версиях микрокода.

26.3.3.3. Клиенты

Клиент в беспроводной сети представляет собой систему, которая обращается к точке доступа или непосредственно к другому клиенту.

Как правило, клиенты беспроводной сети имеют только один сетевой адаптер, а именно адаптер беспроводной сети.

Существует несколько различных способов конфигурации клиента беспроводной сети. Они основаны на различных режимах работы в беспроводной сети, обычно BSS (режим инфраструктуры, который требует точки доступа) или IBSS (ad-hoc или режим одноранговой сети). В нашем примере мы будем использовать самый популярный их них, режим BSS, для связи с точкой доступа.

26.3.3.3.1. Требования

Существует только одно жёсткое условие для настройки FreeBSD в качестве клиента беспроводной сети. Вам нужен адаптер беспроводной связи, поддерживаемый FreeBSD.

26.3.3.3.2. Конфигурация FreeBSD как клиента беспроводной сети

Перед тем, как подключиться к беспроводной сети, вам нужно будет узнать о ней несколько вещей. В этом примере мы подключаемся к сети, которая называется *my_net*, и шифрование в ней отключено.

В этом примере мы не используем шифрование, но это небезопасно. В следующем разделе вы узнаете, как её включить, почему это так важно, и почему некоторые технологии шифрования всё же не могут полностью обеспечить вашу информационную безопасность.

Удостоверьтесь, что ваш адаптер распознаётся во FreeBSD:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
        inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
        ether 00:09:2d:2d:c9:50
```

```
        media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
        status: no carrier
        ssid ""
        stationname "FreeBSD Wireless node"
        channel 10 authmode OPEN powersavemode OFF powersavesleep 100
        wepmode OFF weptxkey 1
```

Теперь мы можем изменить настройки адаптера на те, что соответствуют нашей сети:

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ssid my_net
```

Замените `192.168.0.20` и `255.255.255.0` на правильные IP-адрес и сетевую маску в вашей проводной сети. Запомните, что наша точка доступа выступает в роли моста для данных между беспроводной и проводной сетями, так что они будут доступны для других устройств, находящихся в сети, как будто они тоже находятся в проводной сети.

Как только вы это выполнили, то сможете получить ping от хостов в проводной сети, как будто вы подключены посредством обычных проводов.

Если вы столкнулись с проблемами при работе в беспроводной сети, удостоверьтесь, что вы ассоциированы (подключены) с точкой доступа:

```
# ifconfig wi0
```

должна выдать некоторую информацию, и вы должны увидеть:

```
status: associated
```

Если статус не будет соответствовать `associated`, это может значить, что вы оказались вне зоны досягаемости точки доступа, включили шифрование или, возможно, имеются проблемы с конфигурацией.

26.3.3.4. Шифрование

Шифрование в беспроводной сети имеет важное значение, потому что у вас нет больше возможности ограничить сеть хорошо защищённой областью. Данные вашей беспроводной сети вещаются по всей окрестности, так что любой заинтересовавшийся может их считать. Вот здесь используется шифрование. Шифруя данные, посылаемые в эфир, вы делаете их прямой перехват гораздо более сложным для всех любопытных.

Двумя наиболее широко применяемыми способами шифрования данных между вашим клиентом и точкой доступа являются WEP и ipsec(4).

26.3.3.4.1. WEP

WEP является сокращением от Wired Equivalency Protocol (Протокол Соответствия

Проводной сети). WEP является попыткой сделать беспроводные сети такими же надёжными и безопасными, как проводные. К сожалению, он был взломан и сравнительно легко поддаётся вскрытию. Это означает также, что он не тот протокол, на который следует опираться, когда речь идёт о шифровании критически важных данных.

Он лучше, чем ничего, так что используйте следующую команду для включения WEP в вашей новой точке доступа FreeBSD:

```
# ifconfig wi0 inet up ssid my_net wepmode on wepkey 0x1234567890 media DS/11Mbps
mediaopt hostap
```

Вы можете включить WEP на клиенте следующей командой:

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ssid my_net wepmode on wepkey
0x1234567890
```

Отметьте, что вы должны заменить *0x1234567890* на более уникальный ключ.

26.3.3.4.2. IPsec

ipsec(4) является гораздо более надёжным и мощным средством шифрования данных в сети. Этот метод определённо является предпочтительным для шифрования данных в беспроводной сети. Более детально ознакомиться с безопасностью и применением ipsec(4) вы можете в разделе об IPsec этого Руководства.

26.3.3.5. Утилиты

Имеется несколько утилит, которые можно использовать для настройки и отладки вашей беспроводной сети, и здесь мы попытаемся описать некоторые из них и что они могут делать.

26.3.3.5.1. Пакет bsd-airtools

Пакет bsd-airtools представляет собой полный набор инструментов, включая инструменты для проверки беспроводной сети на предмет взлома WEP-ключа, обнаружения точки доступа и тому подобное.

Утилиты bsd-airtools можно установить из порта net-mgmt/bsd-airtools. Информацию об установке портов можно найти в Главе Установка приложений. порты и пакеты этого Руководства.

Программа `dstumbler` является инструментом, предназначенным для обнаружения точки доступа и выдачи отношения уровня сигнала к шуму. Если у вас с трудом получается запустить точку доступа, `dstumbler` может помочь вам начать.

Для тестирования информационной безопасности вашей беспроводной сети, вы можете воспользоваться набором "dweputils" (`dwepcrack`, `dwepdump` и `dwepkeygen`), который может помочь понять, является ли WEP подходящим решением для обеспечения ваших

потребностей в информационной безопасности.

26.3.3.5.2. Утилиты `wicontrol`, `ancontrol` и `raycontrol`

Это инструменты, которые могут быть использованы для управления поведением адаптера беспроводной связи в сети. В примере выше мы выбирали wicontrol(8), так как нашим адаптером беспроводной сети был интерфейс wi0. Если у вас установлено устройство беспроводного доступа от Cisco, этим интерфейсом будет an0, и тогда вы будете использовать ancontrol(8).

26.3.3.5.3. Команда `ifconfig`

Команда ifconfig(8) может использоваться для установки многих из тех параметров, что задаёт wicontrol(8), однако работа с некоторыми параметрами в ней отсутствует. Обратитесь к ifconfig(8) для выяснения параметров и опций командной строки.

26.3.3.6. Поддерживаемые адаптеры

26.3.3.6.1. Точки доступа

Единственными адаптерами, которые на данный момент поддерживаются в режиме BSS (как точка доступа), являются те устройства, что сделаны на основе набора микросхем Prism 2, 2.5 или 3). Полный список можно увидеть в wi(4).

26.3.3.6.2. Клиенты 802.11b

Практически все адаптеры беспроводной связи 802.11b на данный момент во FreeBSD поддерживаются. Большинство адаптеров, построенных на основе Prism, Spectrum24, Hermes, Aironet и Raylink, будут работать в качестве адаптера беспроводной сети в режиме IBSS (ad-hoc, одноранговая сеть и BSS).

26.3.3.6.3. Клиенты 802.11a и 802.11g

Драйвер устройства ath(4) поддерживает 802.11a и 802.11g. Если ваша карта основана на чипсете Atheros, вы можете использовать этот драйвер.

К сожалению, все еще много производителей, не предоставляющих схематику своих драйверов сообществу open source, поскольку эта информация считается торговым секретом. Следовательно, у разработчиков FreeBSD и других операционных систем остается два варианта: разработать драйверы долгим и сложным методом обратного инжиниринга, или использовать существующие драйверы для платформ Microsoft® Windows®. Большинство разработчиков FreeBSD выбрали второй способ.

Благодаря усилиям Билла Пола (wpaul), начиная с FreeBSD 5.3-RELEASE существует "прозрачная" поддержка Network Driver Interface Specification (NDIS). FreeBSD NDISulator (известный также как Project Evil) преобразует бинарный драйвер Windows® так, что он работает так же как и в Windows®. Эта возможность всё ещё относительно нова, но в большинстве тестов она работает адекватно.

Для использования NDISulator потребуются три вещи:

1. Исходные тексты ядра
2. Бинарный драйвер Windows® XP (расширение .SYS)
3. Файл конфигурации бинарного драйвера Windows® XP (расширение .INF)

Вам может потребоваться компиляция драйвера оболочки мини порта ndis(4). Под `root`:

```
# cd /usr/src/sys/modules/ndis
# make && make install
```

Определите местоположение файлов для вашей карты. Обычно их можно найти на входящем в комплект CD или на Web-сайте поставщика. В нашем примере используются файлы W32DRIVER.SYS и W32DRIVER.INF.

Следующий шаг это компиляция бинарного драйвера в загружаемый модуль ядра. Чтобы сделать это, сначала зайдите в каталог модуля if_ndis и с правами `root` скопируйте туда драйверы Windows®:

```
# cd /usr/src/sys/modules/if_ndis
# cp /path/to/driver/W32DRIVER.SYS ./
# cp /path/to/driver/W32DRIVER.INF ./
```

Теперь используйте утилиту `ndiscvt` для создания заголовка определения драйвера ndis_driver_data.h перед сборкой модуля:

```
# ndiscvt -i W32DRIVER.INF -s W32DRIVER.SYS -o ndis_driver_data.h
```

Параметры `-i` и `-s` задают соответственно файл настройки и бинарный файл. Мы используем параметр `-o ndis_driver_data.h`, поскольку Makefile при создании модуля будет обращаться именно к этому файлу.

Некоторым драйверам Windows® для работы требуются дополнительные файлы. Вы можете включить их параметром `ndiscvt -f`. Обратитесь к странице справочной системы ndiscvt(8) за дополнительной информацией.

Наконец, соберите и установите модуль драйвера:

```
# make && make install
```

Для использования драйвера необходимо загрузить соответствующие модули:

```
# kldload ndis
# kldload if_ndis
```

Первая команда загружает оболочку драйвера мини-порта NDIS, вторая загружает собственно сетевой интерфейс. Проверьте dmesg(8) на предмет ошибок загрузки. Если все прошло хорошо, вывод должен быть примерно таким:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

Начиная с этого момента вы можете использовать устройство ndis0 как любое другое беспроводное устройство (например, wi0); в этой ситуации применима информация, приведенная в начале этой главы.

26.4. Bluetooth

26.4.1. Введение

Bluetooth является беспроводной технологией для создания персональных сетей на расстоянии не более 10 метров, работающей на частоте 2.4 ГГц, которая не подлежит лицензированию. Обычно такие сети формируются из портативных устройств, таких, как сотовые телефоны, КПК и лэптопы. В отличие от Wi-Fi, другой популярной беспроводной технологии, Bluetooth предоставляет более высокий уровень сервиса, например, файловые серверы типа FTP, передачу файлов, голоса, эмуляцию последовательного порта и другие.

Стек протоколов Bluetooth во FreeBSD реализован на основе технологии Netgraph (обратитесь к netgraph(4)). Широкий спектр USB-устройств Bluetooth поддерживается драйвером ng_ubt(4). Устройства Bluetooth на основе набора микросхем Broadcom BCM2033 поддерживается драйвером ng_bt3c(4). Устройства Bluetooth, работающие через последовательные и UART-порты, поддерживаются драйверами sio(4), ng_h4(4) и hcseriald(8). В этом разделе описывается использование Bluetooth-устройств, подключаемых через USB.

26.4.2. Подключение устройства

По умолчанию драйверы устройств Bluetooth поставляются в виде модулей ядра. Перед подключением устройства вам необходимо подгрузить драйвер в ядро:

```
# kldload ng_ubt
```

Если Bluetooth-устройство в момент запуска системы подключено, то загружайте модуль из файла /boot/loader.conf:

```
ng_ubt_load="YES"
```

Подключите ваше USB-устройство. На консоли (или в журнале syslog) появится примерно

такое сообщение:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

Стек протоколов Bluetooth запускается вручную во FreeBSD 6.0, и во FreeBSD 5.X, перед 5.5. Это делается автоматически через devd(8) во FreeBSD 5.5, 6.1 и в более новых версиях.

Скопируйте файл /usr/shared/examples/netgraph/bluetooth/rc.bluetooth в какое-нибудь подходящее место, например, в файл /etc/rc.bluetooth. Этот скрипт используется для запуска и остановки работы Bluetooth-стека. Перед отключением устройства рекомендуется остановить его работы, хотя (обычно) это не фатально. При запуске стека вы получите сообщения, подобные следующим:

```
# /etc/rc.bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

26.4.3. Host Controller Interface (HCI)

Host Controller Interface (HCI) предоставляет интерфейс для управления контроллером передатчика и менеджером соединений, а также доступ к данным о состоянии оборудования и его управляющим регистрам. Этот интерфейс предоставляет унифицированный метод доступа к передающим возможностям Bluetooth. Уровень HCI на управляющей машине обменивается данными и командами с микрокодом HCI в оборудовании Bluetooth. Драйвер для Host Controller Transport Layer (то есть физической шины) предоставляет обоим слоям HCI возможность обмениваться данными друг с другом.

Для одного Bluetooth-устройства создаётся один узел Netgraph типа *hci*. HCI-узел обычно подключается к узлу драйвера устройства Bluetooth (входящий поток) и к узлу L2CAP (исходящий поток). Все операции с HCI должны выполняться на узле HCI, но не на узле драйвера устройства. В качестве имени по умолчанию для узла HCI используется "devicehci". Дополнительные подробности можно найти на справочной странице ng_hci(4).

Одной из самой часто выполняемой задач является обнаружение Bluetooth-устройств в радиусе RF-доступности. Эта операция называется *опросом* (inquiry). Опрос и другие операции, связанные с HCI, выполняются при помощи утилиты hccontrol(8). Пример ниже показывает, как найти доступные устройства Bluetooth. Список таких устройств должен быть получен в течение нескольких секунд. Заметьте, что удалённые устройства будут отвечать на опрос, если только они находятся в режиме *обнаруживаемости* (discoverable).

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
       BD_ADDR: 00:80:37:29:19:a4
       Page Scan Rep. Mode: 0x1
       Page Scan Period Mode: 00
       Page Scan Mode: 00
       Class: 52:02:04
       Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

BD_ADDR является уникальным адресом устройства Bluetooth, вроде MAC-адресов сетевых адаптеров. Этот адрес необходим для дальнейшей работы с устройством. Адресу BD_ADDR можно присвоить удобное для чтения имя. Файл /etc/bluetooth/hosts содержит информацию об известных хостах Bluetooth. В следующем примере показано, как получить имя, назначенное удалённому устройству:

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

Если вы выполните опрос на другом Bluetooth-устройстве, но ваш компьютер будет опознан как "your.host.name (ubt0)". Имя, назначаемое локальному устройству, может быть в любой момент изменено.

Система Bluetooth предоставляет услуги по соединениям типа точка-точка (при этом задействованы только два устройства Bluetooth) или точка-ко-многим-точкам. В последнем случае соединение используется совместно несколькими устройствам Bluetooth. В следующем примере показывается, как получить список активных для локального устройства соединений:

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

Идентификатор соединения (*connection handle*) полезен, когда необходимо прекратить соединение. Заметьте, что обычно нет нужды делать это вручную. Стек будет автоматически разрывать неактивные соединения.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Обратитесь к помощи посредством `hccontrol help` для получения полного списка доступных HCI-команд. Большинство команд HCI для выполнения не требуют прав администратора системы.

26.4.4. Logical Link Control and Adaptation Protocol (L2CAP)

Протокол L2CAP (Logical Link Control and Adaptation Protocol) предоставляет услуги по работе с данными, как ориентированные на соединения, так и без ориентации на них, протоколам более высокого уровня с возможностями мультиплексирования и обеспечением операций по сегментации и обратной сборке. L2CAP позволяет протоколам более высокого уровня и приложениям передавать и получать пакеты данных L2CAP длиной до 64 Кбайт.

L2CAP основан на концепции *каналов*. Каналом является логическое соединение поверх соединения по радиоканалу. Каждый канал привязан к некоторому протоколу по принципу многие-к-одному. Несколько каналов могут быть привязаны к одному и тому же протоколу, но канал не может быть привязан к нескольким протоколам. Каждый пакет L2CAP, получаемый каналом, перенаправляется к соответствующему протоколу более высокого уровня. Несколько каналов могут совместно использовать одно и то же радиосоединение.

Для одного Bluetooth-устройства создается один узел Netgraph типа *l2cap*. Узел L2CAP обычно подключается к узлу Bluetooth HCI (нижестоящий) и узлам Bluetooth-сокетов (вышестоящие). По умолчанию для узла L2CAP используется имя "devicel2cap". Для получения дополнительной информации обратитесь к справочной странице по ng_l2cap(4).

Полезной является программа l2ping(8), которая может использоваться для проверки связи с другими устройствами. Некоторые реализации Bluetooth могут не возвращать все данные, посылаемые им, так что `0 bytes` в следующем примере - это нормально.

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

Утилита l2control(8) используется для выполнения различных операций с узлами L2CAP. В этом примере показано, как получить список логических соединений (каналов) и перечень радиосоединений локального устройства:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
```

```
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

Ещё одним диагностическим инструментом является btsockstat(1). Она выполняет действия, подобные тем, что обычно выполняет netstat(1), но со структурами данных, связанных с работой в сети Bluetooth. В примере ниже описывается то же самое логическое соединение, что и с l2control(8) выше.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID   State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address     Foreign address   Chan DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6    OPEN
```

26.4.5. Протокол RFCOMM

Протокол RFCOMM эмулирует последовательные порты поверх протокола L2CAP. Он основан на ETSI-стандарте TS 07.10. RFCOMM представляет собой простой транспортный протокол, с дополнительными возможностями по эмуляции 9 цепей последовательных портов RS-232 (EIATIA-232-E). Протокол RFCOMM поддерживает одновременно до 60 соединений (каналов RFCOMM) между двумя устройствами Bluetooth.

В рамках RFCOMM полный коммуникационный маршрут включает два приложения, работающие на разных устройствах (конечные коммуникационные точки) с коммуникационным сегментом между ними. RFCOMM предназначен для сокрытия приложений, использующих последовательные порты устройств, в которых они расположены. Коммуникационный сегмент по сути является Bluetooth-связью от одного устройства к другому (прямое соединение).

RFCOMM имеет дело с соединением между устройствами в случае прямого соединения, или между устройством и модемом в сетевом случае. RFCOMM может поддерживать и другие конфигурации, такие, как модули, работающие через беспроводную технологию Bluetooth с одной стороны и предоставляющие проводное соединение с другой стороны.

Во FreeBSD протокол RFCOMM реализован на уровне сокетов Bluetooth.

26.4.6. Pairing of Devices

По умолчанию связь Bluetooth не аутентифицируется, поэтому любое устройство может общаться с любым другим. Устройство Bluetooth (например, сотовый телефон) может задать обязательность аутентификации для предоставления определённого сервиса (в частности, услугу доступа по коммутируемой линии). Bluetooth-аутентификация обычно выполняется

через *PIN-коды*. PIN-код представляет из себя ASCII-строку длиной до 16 символов. Пользователь обязан ввести один и тот же PIN-код на обоих устройствах. Как только он введёт PIN-код, оба устройства сгенерируют *ключ связи*. После этого ключ может быть сохранён либо в самом устройстве, либо на постоянном носителе. В следующий раз оба устройства будут использовать ранее сгенерированный ключ соединения. Процедура, описанная выше, носит название *подгонки пары* (pairing). Заметьте, что если ключ связи потерян любой из сторон, то подбор пары должен быть повторен.

За обработку всех запросов на Bluetooth-аутентификацию отвечает даемон hcsecd(8). По умолчанию файл конфигурации называется /etc/bluetooth/hcsecd.conf. Пример раздела, содержащего информацию о сотовом телефоне с явно заданным PIN-кодом "1234" приведен ниже:

```
device {
        bdaddr  00:80:37:29:19:a4;
        name    "Pav's T39";
        key     nokey;
        pin     "1234";
      }
```

Кроме длины, на PIN-коды не накладывается никаких ограничений. Некоторые устройства (например, Bluetooth-гарнитуры) могут иметь фиксированный встроенный PIN-код. Параметр `-d` позволяет запустить hcsecd(8) как нефоновый процесс, что облегчает просмотр происходящих событий. Задайте получение парного ключа на удалённом устройстве и инициируйте Bluetooth-соединение с этим устройством. Удалённое устройство должно подтвердить получение пары и запросить PIN-код. Введите тот же самый код, что находится в hcsecd.conf. Теперь ваш ПК и удалённое устройство спарены. Альтернативным способом является инициация процесса создания пары на удалённом устройстве.

Во FreeBSD 5.5, 6.1 и в более новых, следующая строка может быть добавлена к /etc/rc.conf, чтобы hcsecd запускался автоматически во время старта системы:

```
hcsecd_enable="YES"
```

Ниже даётся пример выдачи протокола команды hcsecd:

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
PIN code exists
```

```
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

26.4.7. Service Discovery Protocol (SDP)

Протокол обнаружения сервисов SDP даёт возможность клиентским приложениям осуществлять поиск услуг, предоставляемых серверными приложениями, а также характеристик этих услуг. В перечень атрибутов сервиса включается тип класса предлагаемого сервиса и информация о механизме или протоколе, требуемом для использования сервиса.

SDP подразумевает коммуникации между SDP-сервером и SDP-клиентом. Сервер поддерживает список сервисов, в котором описываются параметры сервисов, связанных с сервером. Каждая запись об услуге содержит информацию об одном сервисе. Клиент может запросить информацию об определённом сервисе, обслуживаемом SDP-сервером, выдавая SDP-запрос. Если клиент или приложение, связанное с клиентом, решат воспользоваться сервисом, то для его использования необходимо открыть отдельное соединение к устройству, предоставляющему сервис. SDP предоставляет механизм обнаружения услуг и их параметров, но не даёт механизма использования этих сервисов.

Обычно SDP-клиент выполняет поиск услуг на основе некоторых желаемых характеристик услуг. Однако иногда возникает необходимость выяснить полный перечень типов услуг, предоставляемых SDP-сервером, не имея никакой информации об имеющихся сервисах. Такой процесс всех предлагаемых сервисов называется *обзором* (browsing).

Bluetooth SDP сервер sdpd(8) и клиент с интерфейсом командной строки sdpcontrol(8) включены в стандартную поставку FreeBSD. В следующем примере показано, как выполнять запрос на SDP-обзор.

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
Service Class ID List:
        Service Discovery Server (0x1000)
Protocol Descriptor List:
        L2CAP (0x0100)
                Protocol specific parameter #1: u/int/uuid16 1
                Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
        Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
        LAN Access Using PPP (0x1102)
Protocol Descriptor List:
        L2CAP (0x0100)
        RFCOMM (0x0003)
                Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
```

```
        LAN Access Using PPP (0x1102) ver. 1.0
```

i. и так далее. Заметьте, что каждый сервис имеет перечень атрибутов (например, канал RFCOMM). В зависимости от сервиса вам может потребоваться где-то сохранить эти атрибуты. Некоторые реализации Bluetooth не поддерживают просмотр сервисов и могут возвращать пустой список. В этом случае возможен поиск конкретной услуги. В примере ниже показано, как выполнить поиск службы OBEX Object Push (OPUSH):

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Во FreeBSD предоставление сервисов клиентам Bluetooth осуществляется сервером sdpd(8). Во FreeBSD 5.5, 6.1 и в более новых, следующая строка может быть добавлена в файл /etc/rc.conf:

```
sdpd_enable="YES"
```

После этого sdpd даемон может быть запущен с помощью:

```
# /etc/rc.d/sdpd start
```

Во FreeBSD 6.0, и во FreeBSD 5.X перед 5.5, sdpd не интегрирован в скрипты загрузки системы. Он должен запускаться автоматически командой:

```
# sdpd
```

Приложение на локальном сервере, желающее предоставить сервис Bluetooth удаленным клиентам, регистрирует сервис через локального даемона SDP. Пример такого приложения - rfcomm_pppd(8). После запуска оно регистрирует Bluetooth LAN сервис через локального даемона SDP.

Список сервисов, зарегистрированных через локальный SDP сервер, может быть получен путем выдачи запроса на просмотр SDP через локальный контрольный канал:

```
# sdpcontrol -l browse
```

26.4.8. Доступ к сети по коммутируемой линии связи (DUN) и по протоколу PPP (LAN)

Модуль работы с коммутируемым доступом к сети (DUN - Dial-Up Networking) в большинстве случаев используется с модемами и сотовыми телефонами. Этот модуль покрывает следующие случаи:

- сотовый телефон или модем используется вместе с компьютером в качестве

беспроводного модема для подключения к серверу коммутируемого доступа в Интернет, или другой коммутируемой услуге;

- сотовый телефон или модем используется компьютером для приёма входящих соединений.

Модуль доступа к сети по протоколу PPP (Network Access with PPP - LAN) может использоваться в следующих ситуациях:

- доступ к ЛВС для одного Bluetooth-устройства;
- доступ к ЛВС для нескольких Bluetooth-устройств;
- связь между двумя ПК (при помощи протокола PPP поверх эмулируемого последовательного канала связи).

Во FreeBSD оба случая реализуются при помощи сервисных программ ppp(8) и rfcomm_pppd(8) - это обработчик, преобразующий RFCOMM-соединения Bluetooth в нечто, с чем может работать PPP. Перед тем, как использовать любой модуль, в файле /etc/ppp/ppp.conf должна быть создана новая PPP-метка. Примеры использования можно найти в справочной странице к rfcomm_pppd(8).

В следующем примере rfcomm_pppd(8) будет использоваться для открытия RFCOMM-соединения к удалённому устройству с BD_ADDR 00:80:37:29:19:a4 на DUN RFCOMM-канале. Реальный номер RFCOMM-канала будет получаться с удалённого устройства через SDP. Возможно указать RFCOMM-канал вручную, и в этом случае rfcomm_pppd(8) не будет выполнять SDP-запрос. Для нахождения RFCOMM-канала на удалённом устройстве используйте утилиту sdpcontrol(8).

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

Для того, чтобы организовать сервис Network Access with PPP (LAN), необходимо запустить сервер sdpd(8). В файле /etc/ppp/ppp.conf должна быть создана новая запись для клиентов LAN. Примеры можно найти в справке по rfcomm_pppd(8). Наконец, запустите RFCOMM PPP сервер на существующем номере канала RFCOMM. Сервер RFCOMM PPP автоматически зарегистрирует Bluetooth LAN сервис через локальный SDP даемон. В примере ниже показано, как запустить сервер RFCOMM PPP.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

26.4.9. OBEX Object Push (OPUSH) Profile

OBEX является широко используемым протоколом для простой передачи файлов между мобильными устройствами. В основном он используется в коммуникациях через инфракрасный порт для передачи файлов между ноутбуками или КПК, а также для пересылки визитных карточек или календарных планов между сотовыми телефонами и другими устройствами с персональными информационными менеджерами.

Сервер и клиент OBEX реализованы в виде пакета стороннего разработчика obexapp,

который доступен в виде порта comms/obexapp.

Клиент OBEX используется для посылки или приёма объектов с сервера OBEX. Объектом, к примеру, может быть визитная карточка или указание. Клиент OBEX может получить номер RFCOMM-канала, указав вместо него имя сервиса. Поддерживаются следующие имена сервиса: IrMC, FTRN и OPUSH. Канал RFCOMM можно задать его номером. Ниже даётся пример сеанса OBEX, где с сотового телефона забирается объект с информацией об устройстве, а новый объект (визитная карточка) передаётся в каталог сотового телефона.

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

Для того, чтобы предоставить сервис OBEX Push, должен быть запущен сервер sdpd(8). Должен быть создан корневой каталог, в котором будут сохраняться все поступающие объекты. По умолчанию корневым каталогом является /var/spool/obex. Наконец, запустите OBEX сервер на существующем номере канала RFCOMM. OBEX сервер автоматически зарегистрирует сервис OBEX Object Push через локального даемона SDP. В примере ниже показано, как запустить OBEX-сервер.

```
# obexapp -s -C 10
```

26.4.10. Профиль последовательного порта (SPP)

Профиль последовательного порта (SPP - Serial Port Profile) позволяет Bluetooth-устройствам осуществлять эмуляцию последовательного порта RS232 (или подобного). Этот профиль покрывает случаи, касающиеся работы унаследованных приложений с Bluetooth в качестве замены кабельному соединению, при это используется абстракция виртуального последовательного порта.

Утилита rfcomm_sppd(1) реализует профиль последовательного порта. В качестве виртуального последовательного порта используется псевдо-терминал. В примере ниже показано, как подключиться к сервису Serial Port удалённого устройства. Заметьте, что вы не указываете RFCOMM-канал - rfcomm_sppd(1) может получить его с удалённого устройства через SDP. Если вы хотите переопределить это, укажите RFCOMM-канал явно в командной строке.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/ttyp6
rfcomm_sppd[94692]: Starting on /dev/ttyp6...
```

После подключения псевдо-терминал можно использовать как последовательный порт:

```
# cu -l ttyp6
```

26.4.11. Решение проблем

26.4.11.1. Удалённое устройство не подключается

Некоторые старые Bluetooth-устройства не поддерживают переключение ролей. По умолчанию, когда FreeBSD подтверждает новое соединение, она пытается выполнить переключение роли и стать ведущим устройством. Устройства, которые это не поддерживают, не смогут подключиться. Заметьте, что переключение ролей выполняется при установлении нового соединения, поэтому невозможно выяснить, поддерживает ли удалённое устройство переключение ролей. На локальной машине имеется возможность отключить переключение ролей при помощи HCI-параметра:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

26.4.11.2. Что-то идёт не так, можно ли посмотреть, что в точности происходит?

Да, можно. Воспользуйтесь пакетом стороннего разработчика, hcidump который доступен в виде порта comms/hcidump. Утилита hcidump похожа на tcpdump(1). Она может быть использована для вывода на терминал содержимого Bluetooth-пакетов и сбрасывать пакеты Bluetooth в файл.

26.5. Мосты

26.5.1. Введение

Иногда полезно разделить одну физическую сеть (такую, как сегмент Ethernet) на два отдельных сегмента сети без необходимости создания подсетей IP и использования маршрутизатора для соединения сегментов. Устройство, которое соединяет две сети на такой манер, называется "сетевым мостом" ("bridge"). Система FreeBSD с двумя сетевыми адаптерами может выступать в роли моста.

Мост работает на основе изучения адресов уровня MAC (адресов Ethernet) устройств на каждом из своих сетевых интерфейсах. Он перенаправляет трафик между двумя сетями, только когда адреса отправителя и получателя находятся в разных сетях.

По многим параметрам мост работает также, как коммутатор Ethernet с малым количеством портов.

26.5.2. Ситуации, когда можно использовать мосты

На сегодняшний день есть две ситуации, когда можно использовать мост.

26.5.2.1. Большой трафик в сегменте

Первая ситуация возникает, когда ваша физическая сеть перегружена трафиком, но по каким-то соображениям вы не хотите разделять сеть на подсети и соединять их с помощью маршрутизатора.

Давайте рассмотрим в качестве примера газету, в которой редакторский и производственный отделы находятся в одной и той же подсети. Пользователи в редакторском отделе все используют сервер `A` для служб доступа к файлам, а пользователи производственного отдела используют сервер `B`. Для объединения всех пользователей используется сеть Ethernet, а высокая нагрузка на сеть замедляет работу.

Если пользователи редакторского отдела могут быть собраны в одном сегменте сети, а пользователи производственного отдела в другом, то два сетевых сегмента можно объединить мостом. Только сетевой трафик, предназначенный для интерфейсов с "другой" стороны моста, будет посылаться в другую сеть, тем самым снижая уровень нагрузки на каждый сегмент сети.

26.5.2.2. Сетевой экран с возможностями фильтрации/ограничения пропускной способности трафика

Второй распространённой ситуацией является необходимость в обеспечении функций сетевого экрана без трансляции сетевых адресов (NAT).

Для примера можно взять маленькую компанию, которая подключена к своему провайдеру по каналу DSL или ISDN. Для неё провайдер выделил 13 глобально доступных IP-адресов для имеющихся в сети 10 персональных компьютеров. В такой ситуации использование сетевого экрана на основе маршрутизатора затруднено из-за проблем с разделением на подсети.

Брандмауэр на основе моста может быть настроен и включен между маршрутизаторами DSL/ISDN без каких-либо проблем с IP-адресацией.

26.5.3. Настройка моста

26.5.3.1. Выбор сетевого адаптера

Для работы моста требуются по крайней мере два сетевых адаптера. К сожалению, не все сетевые адаптеры поддерживают функции моста. Прочтите страницу Справочника по bridge(4) для выяснения подробностей о поддерживаемых адаптерах.

Перед тем, как продолжить, сначала установите и протестируйте два сетевых адаптера.

26.5.3.2. Изменения в конфигурации ядра

Для включения поддержки функций сетевого моста в ядре, добавьте строчку

```
options BRIDGE
```

в файл конфигурации вашего ядра, и перестройте ядро.

26.5.3.3. Поддержка функций брандмауэра

Если вы планируете использовать мост в качестве брандмауэра, вам нужно также добавить опцию `IPFIREWALL`. Прочтите Межсетевые экраны, содержащую общую информацию о настройке моста в качестве брандмауэра.

Если вам необходимо обеспечить прохождение не-IP пакетов (таких, как ARP) через мост, то имеется опция брандмауэра, которую можно задать. Это опция `IPFIREWALL_DEFAULT_TO_ACCEPT`. Заметьте, что при этом правило, используемое брандмауэром по умолчанию, меняется на разрешительное для всех пакетов. Перед тем, как задавать эту опцию, убедитесь, что вы понимаете работу вашего набора правил.

26.5.3.4. Поддержка функций ограничения пропускной способности

Если вы хотите использовать мост в качестве машины, ограничивающей пропускную способность, то добавьте в файл конфигурации ядра опцию `DUMMYNET`. Дополнительную информацию можно почерпнуть из страницы Справочника по dummynet(4).

26.5.4. Включение функций моста

Добавьте строку

```
net.link.ether.bridge.enable=1
```

в файл /etc/sysctl.conf для включения функций моста во время работы системы, и строку:

```
net.link.ether.bridge.config=if1,if2
```

для включения функций моста для указанных интерфейсов (замените *if1* и *if2* на имена двух ваших сетевых интерфейсов). Если вы хотите, чтобы проходящие через мост пакеты фильтровались посредством ipfw(8), вы должны также добавить строчку:

```
net.link.ether.bridge.ipfw=1
```

Для версий FreeBSD, предшествующих FreeBSD 5.2-RELEASE, нужно использовать следующие строки:

```
net.link.ether.bridge=1
net.link.ether.bridge_cfg=if1,if2
net.link.ether.bridge_ipfw=1
```

26.5.5. Дополнительные замечания

Если вы хотите осуществлять удалённый доступ на мост через ssh(1) из сети, то корректно назначить одному из сетевых адаптеров IP-адрес. Общепринято, что назначение адреса обоим сетевым адаптерам является не самой хорошей идеей.

Если в вашей сети присутствует несколько мостов, не должно быть более одного маршрута между любыми двумя рабочими станциями. С технической точки зрения это означает отсутствие поддержки протокола spanning tree.

Сетевой мост может увеличить задержки в замерах командой ping(8), особенно для трафика между двумя разными сегментами.

26.6. Работа с бездисковыми станциями

Машина с FreeBSD может загружаться по сети и работать без наличия локального диска, используя файловые системы, монтируемые с сервера NFS. Кроме стандартных конфигурационных файлов, не нужны никакие модификации в системе. Такую систему легко настроить, потому что все необходимые элементы уже готовы:

- Имеется по крайней мере два возможных способа загрузки ядра по сети:
  - PXE: Система Intel® Preboot eXecution Environment является формой загрузочного ПЗУ, встроенного в некоторые сетевые адаптеры или материнские платы. Обратитесь к справочной странице по pxeboot(8) для получения более полной информации.
  - Порт Etherboot (net/etherboot) генерирует код, который может применяться в ПЗУ для загрузки ядра по сети. Код может быть либо прошит в загрузочный PROM на сетевом адаптере, либо загружен с локальной дискеты (или винчестера), или с работающей системы MS-DOS®. Поддерживаются многие сетевые адаптеры.
- Примерный скрипт (/usr/shared/examples/diskless/clone_root) облегчает создание и поддержку корневой файловой системы рабочей станции на сервере. Скрипт, скорее всего, потребует некоторых настроек, но он позволит вам быстро начать работу.
- Стандартные файлы начального запуска системы, располагающиеся в /etc, распознают и поддерживают загрузку системы в бездисковом варианте.
- Подкачка, если она нужна, может выполняться через файл NFS либо на локальный диск.

Существует много способов настройки бездисковой рабочей станции. При этом задействованы многие компоненты, и большинство из них могут быть настроены для удовлетворения ваших вкусов. Далее будет описаны варианты полной настройки системы, при этом упор будет делаться на простоту и совместимость с стандартной системой скриптов начальной загрузки FreeBSD. Описываемая система имеет такие характеристики:

- Бездисковые рабочие станции совместно используют файловую систему / в режиме только чтения, а также используют /usr совместно тоже в режиме только чтения.

  Корневая файловая система является копией стандартной корневой системы FreeBSD (обычно сервера), с некоторыми настроечными файлами, измененными кем-то специально для бездисковых операций или, возможно, для рабочей станции, которой

она предназначена.

Части корневой файловой системы, которые должны быть доступны для записи, перекрываются файловыми системами md(4). Любые изменения будут потеряны при перезагрузках системы.

- Ядро передается и загружается посредством Etherboot или PXE, и в некоторых ситуациях может быть использован любой из этих методов.

Как описано, эта система не защищена. Она должна располагаться в защищенной части сети, а другие хосты не должны на нее полагаться.

Вся информация этого раздела была протестирована с релизом FreeBSD 5.2.1-RELEASE.

26.6.1. Общая информация

Настройка бездисковых рабочих станций относительно проста, но в то же время легко сделать ошибку. Иногда сложно диагностировать эти ошибки по нескольким причинам. Например:

- Параметры компиляции могут по-разному проявлять себя во время работы.
- Сообщения об ошибках бывают загадочны или вовсе отсутствуют.

В данной ситуации некоторые знания, касающиеся используемых внутренних механизмов, очень полезны при разрешении проблем, которые могут возникнуть.

Для выполнения успешной загрузки необходимо произвести несколько операций:

- Компьютеру необходимо получить начальные параметры, такие как собственный IP адрес, имя исполняемого файла, корневой каталог. Для этого используются протоколы DHCP или BOOTP. DHCP это совместимое расширение BOOTP, используются те же номера портов и основной формат пакетов.

  Возможна настройка системы для использования только BOOTP. Серверная программа bootpd(8) включена в основную систему FreeBSD.

  Тем не менее, у DHCP есть множество преимуществ над BOOTP (лучше файлы настройки, возможность использования PXE, плюс многие другие преимущества, не относящиеся непосредственно к бездисковым операциям), и мы в основном будем описывать настройку DHCP, с эквивалентными примерами для bootpd(8), когда это возможно. Пример конфигурации будет использовать пакет ISC DHCP (релиз 3.0.1.r12 был установлен на тестовом сервере).

- Компьютеру требуется загрузить в локальную память одну или несколько программ. Используются TFTP или NFS. Выбор между TFTP или NFS производится во время компилирования в нескольких местах. Часто встречающаяся ошибка это указание имен файлов для другого протокола: TFTP обычно загружает все файлы с одного каталога сервера, и принимает имена файлов относительно этого каталога. NFS нужны абсолютные пути к файлам.

- Необходимо инициализировать и выполнить возможные промежуточные программы загрузки и ядро. В этой области существует несколько важных вариаций:
  - PXE загрузит pxeboot(8), являющийся модифицированной версией загрузчика третьей стадии FreeBSD. loader(8) получит большинство параметров, необходимых для старта системы, и оставит их в окружении ядра до контроля передачи. В этом случае возможно использование ядра GENERIC.
  - Etherboot, непосредственно загрузит ядро, с меньшей подготовкой. Вам потребуется собрать ядро со специальными параметрами.

    PXE и Etherboot работают одинаково хорошо; тем не менее, поскольку ядро обычно позволяет loader(8) выполнить больше предварительной работы, метод PXE предпочтителен.

    Если ваш BIOS и сетевые карты поддерживают PXE, используйте его.
- Наконец, компьютеру требуется доступ к файловым системам. NFS используется во всех случаях.

Обратитесь также к странице справочника diskless(8).

26.6.2. Инструкции по настройке

26.6.2.1. Конфигурация с использованием ISC DHCP

Сервер ISC DHCP может обрабатывать как запросы BOOTP, так и запросы DHCP.

ISC DHCP 3.0 не включается в поставку системы. Сначала вам нужно будет установить порт net/isc-dhcp3-server или соответствующий пакет.

После установки ISC DHCP ему для работы требуется конфигурационный файл (обычно называемый /usr/local/etc/dhcpd.conf). Вот прокомментированный пример, где хост `margaux` использует Etherboot, а хост `corbieres` использует PXE:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

option domain-name "example.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
  use-host-decl-names on; ①
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.4.255;

  host margaux {
    hardware ethernet 01:23:45:67:89:ab;
    fixed-address margaux.example.com;
```

```
    next-server 192.168.4.4; ②
    filename "/data/misc/kernel.diskless"; ③
    option root-path "192.168.4.4:/data/misc/diskless"; ④
  }
  host corbieres {
    hardware ethernet 00:02:b3:27:62:df;
    fixed-address corbieres.example.com;
    next-server 192.168.4.4;
    filename "pxeboot";
    option root-path "192.168.4.4:/data/misc/diskless";
  }
}
```

① Этот параметр указывает dhcpd посылать значения деклараций `host` как имя хоста для бездисковой машины. Альтернативным способом было бы добавление `option host-name margaux` внутри объявлений `host`.

② Директива `next-server` определяет сервер TFTP или NFS, используемый для получения загрузчика или файла ядра (по умолчанию используется тот же самый хост, на котором расположен сервер DHCP).

③ Директива `filename` определяет файл, который Etherboot или PXE будут загружать для следующего шага выполнения. Он должен быть указан в соответствии с используемым методом передачи. Etherboot может быть скомпилирован для использования NFS или TFTP. FreeBSD порт по умолчанию использует NFS. PXE использует TFTP, поэтому здесь применяются относительные пути файлов (это может зависеть от настроек TFTP сервера, но обычно довольно типично). Кроме того, PXE загружает pxeboot, а не ядро. Существуют другие интересные возможности, такие как загрузка pxeboot из каталога /boot FreeBSD CD-ROM (поскольку pxeboot(8) может загружать GENERIC ядро, это делает возможной загрузку с удаленного CD-ROM).

④ Параметр `root-path` определяет путь к корневой файловой системе, в обычной нотации NFS. При использовании PXE, можно оставить IP хоста отключенным, если параметр ядра BOOTP не используется. Затем NFS сервер может использоваться так же, как и TFTP.

26.6.2.2. Настройка с использованием BOOTP

Далее описана эквивалентная конфигурация с использованием bootpd (для одного клиента). Она будет располагаться в /etc/bootptab.

Пожалуйста, отметьте, что Etherboot должен быть откомпилирован с нестандартной опцией `NO_DHCP_SUPPORT` для того, чтобы можно было использовать BOOTP, и что для работы PXE_необходим_DHCP. Единственным очевидным преимуществом bootpd является его наличие в поставке системы.

```
.def100:\
  :hn:ht=1:sa=192.168.4.4:vm=rfc1048:\
  :sm=255.255.255.0:\
  :ds=192.168.4.1:\
  :gw=192.168.4.1:\
  :hd="/tftpboot":\
```

```
  :bf="/kernel.diskless":\
  :rp="192.168.4.4:/data/misc/diskless":

margaux:ha=0123456789ab:tc=.def100
```

26.6.2.3. Подготовка программы загрузки при помощи Etherboot

Сайт Etherboot содержит подробную документацию, в основном предназначенную для систем Linux, но несомненно, она полезна. Далее будет просто кратко описано, как вы должны использовать Etherboot в системе FreeBSD.

Сначала вы должны установить пакет или порт net/etherboot.

Вы можете изменить настройку Etherboot (например, для использования TFTP вместо NFS) путем редактирования файла Config в каталоге исходных текстов Etherboot.

В нашей ситуации мы будем использовать загрузочную дискету. Для других методов (PROM или программа MS-DOS®) пожалуйста, обратитесь к документации по Etherboot.

Для создания загрузочной дискеты, вставьте дискету в дисковод на машине, где установлен Etherboot, затем перейдите в каталог src в дереве Etherboot и наберите:

```
# gmake bin32/devicetype.fd0
```

devicetype зависит от типа адаптера Ethernet на бездисковой рабочей станции. Обратитесь к файлу NIC в том же самом каталоге для определения правильного значения для *devicetype*.

26.6.2.4. Загрузка с PXE

По умолчанию, pxeboot(8) загружает ядро через NFS. Он может быть скомпилирован для использования вместо него TFTP путем указания параметра `LOADER_TFTP_SUPPORT` в /etc/make.conf. Смотрите комментарии в файле /usr/shared/examples/etc/make.conf.

Есть два не документированных параметра make.conf, которые могут быть полезны для настройки бездискового компьютера с последовательной консолью: `BOOT_PXELDR_PROBE_KEYBOARD`, и `BOOT_PXELDR_ALWAYS_SERIAL`.

Для использования PXE при загрузке компьютера вам обычно потребуется выбрать параметр `Boot from network` (загрузка по сети) в настройках BIOS, или нажать функциональную клавишу во время загрузки PC.

26.6.2.5. Настройка серверов TFTP и NFS

Если вы используете PXE или Etherboot, настроенные для использования TFTP, вам нужно включить tftpd на файловом сервере:

1. Создайте каталог, файлы которого будет обслуживать tftpd, например, /tftpboot.
2. Добавьте в ваш /etc/inetd.conf такую строчку:

```
tftp    dgram   udp wait    root    /usr/libexec/tftpd  tftpd -l -s /tftpboot
```

Бывает, что некоторым версиям PXE требуется TCP-вариант TFTP. В таком случае добавьте вторую строчку, заменяющую `dgram udp` на `stream tcp`.

3. Сообщите inetd о необходимости перечитать свой файл конфигурации. Файл /etc/rc.conf должен содержать строку `inetd_enable="YES"` для корректного исполнения команды

```
# /etc/rc.d/inetd restart
```

Вы можете поместить каталог tftpboot в любом месте на сервере. Проверьте, что это местоположение указано как в inetd.conf, так и в dhcpd.conf.

Во всех случаях, вам также нужно включить NFS и экспортировать соответствующую файловую систему на сервере NFS.

1. Добавьте следующее в /etc/rc.conf:

```
nfs_server_enable="YES"
```

2. Экспортируйте файловую систему, в которой расположен корневой каталог для бездисковой рабочей станции, добавив следующую строку в /etc/exports (подправьте точку монтирования и замените *margaux corbieres* именами бездисковых рабочих станций):

```
/data/misc -alldirs -ro margaux corbieres
```

3. Заставьте mountd перечитать настроечный файл. На самом деле если вам потребовалось на первом шаге включить NFS в /etc/rc.conf, то вам нужно будет выполнить перезагрузку.

```
# /etc/rc.d/mountd restart
```

26.6.2.6. Построение ядра для бездисковой рабочей станции

При использовании Etherboot, вам потребуется создать конфигурационный файл ядра для бездискового клиента со следующими параметрами (вдобавок к обычным):

```
options     BOOTP           # Use BOOTP to obtain IP address/hostname
```

```
options     BOOTP_NFSROOT  # NFS mount root filesystem using BOOTP info
```

Вам может потребоваться использовать `BOOTP_NFSV3`, `BOOT_COMPAT` и `BOOTP_WIRED_TO` (посмотрите файл NOTES).

Эти имена параметров сложились исторически, и могут немного ввести в заблуждение, поскольку включают необязательное использование DHCP и BOOTP в ядре (возможно включение обязательного использования BOOTP или DHCP use).

Постройте ядро (обратитесь к Настройка ядра FreeBSD) и скопируйте его в каталог, указанный в dhcpd.conf.

При использовании PXE, сборка ядра с вышеприведенными параметрами не является совершенно необходимой (хотя желательна). Включение этих параметров приведет к выполнению большинства DHCP запросов во время загрузки ядра, с небольшим риском несоответствия новых значений и значений, полученных pxeboot(8) в некоторых особых случаях. Преимущество использования в том, что в качестве побочного эффекта будет установлено имя хоста. Иначе вам потребуется установить имя хоста другим методом, например в клиент-специфичном файле rc.conf.

Для включения возможности загрузки с Etherboot, в ядро необходимо включить устройство hints. Вам потребуется установить в файле конфигурации следующий параметр (см. файл комментариев NOTES):

```
hints       "GENERIC.hints"
```

26.6.2.7. Подготовка корневой файловой системы

Вам нужно создать корневую файловую систему для бездисковых рабочих станций, в местоположении, заданном как `root-path` в dhcpd.conf.

26.6.2.7.1. Использование процедуры `make world`

Этот метод установит новую систему (не только корневую) в `DESTDIR`. Все, что вам потребуется сделать, это просто выполнить следующий скрипт:

```
#!/bin/sh
export DESTDIR=/data/misc/diskless
mkdir -p ${DESTDIR}
cd /usr/src; make buildworld && make buildkernel
cd /usr/src/etc; make distribution
```

Как только это будет сделано, вам может потребоваться настроить /etc/rc.conf и /etc/fstab, помещенные в `DESTDIR`, в соответствии с вашими потребностями.

26.6.2.8. Настройка области подкачки

Если это нужно, то файл подкачки, расположенный на сервере, можно использовать посредством NFS.

26.6.2.8.1. Подкачка через NFS

На стадии загрузки ядро не поддерживает подкачку через NFS. Подкачка должна быть разрешена при помощи загрузочных скриптов, монтирующих файловую систему, пригодную для записи и создающих на ней файл подкачки. Для создания файла подкачки подходящего размера вы можете выполнить следующие команды:

```
# dd if=/dev/zero of=/path/to/swapfile bs=1k count=1 oseek=100000
```

Для активации этого файла подкачки следует добавить в файл rc.conf строку

```
swapfile=/path/to/swapfile
```

26.6.2.9. Различные проблемы

26.6.2.9.1. Работа с /usr, доступной только для чтения

Если бездисковая рабочая станция настроена на запуск X, вам нужно подправить настроечный файл для XDM, который по умолчанию помещает протокол ошибок в /usr.

26.6.2.9.2. Использование не-FreeBSD сервера

Если сервер с корневой файловой системой работает не под управлением FreeBSD, вам потребуется создать корневую файловую систему на машине FreeBSD, а затем скопировать ее в нужно место, при помощи `tar` или `cpio`.

В такой ситуации иногда возникают проблемы со специальными файлами в /dev из-за различной разрядности целых чисел для старшего/младшего чисел. Решением этой проблемы является экспортирование каталога с не-FreeBSD сервера, монтирование его на машине с FreeBSD и использование devfs(5) для создания файлов устройств прозрачно для пользователя.

26.7. ISDN

Полезным источником информации о технологии ISDN и его аппаратном обеспечении является Страница Дэна Кегела (Dan Kegel) об ISDN.

Быстрое введение в ISDN:

- Если вы живёте в Европе, то вам может понадобиться изучить раздел об ISDN-адаптерах.
- Если вы планируете использовать ISDN в основном для соединений с Интернет через провайдера по коммутируемому, не выделенному соединению, рекомендуется посмотреть информацию о терминальных адаптерах. Это даст вам самую большую

гибкость и наименьшее количество проблем при смене провайдера.

- Если вы объединяете две локальные сети или подключаетесь к Интернет через постоянное ISDN-соединение, рекомендуем остановить свой выбор на отдельном мосте/маршрутизаторе.

Стоимость является важным фактором при выборе вашего решения. Далее перечислены все возможности от самого дешевого до самого дорогого варианта.

26.7.1. Адаптеры ISDN

Реализация ISDN во FreeBSD поддерживает только стандарт DSS1/Q.931 (или Евро-ISDN) при помощи пассивных адаптеров. Поддерживаются некоторые активные адаптеры, прошивки которых поддерживают также другие сигнальные протоколы; также сюда включена поддержка адаптеров ISDN Primary Rate (PRI).

Пакет программ isdn4bsd позволяет вам подключаться к другим маршрутизаторам ISDN при помощи IP поверх DHLC, либо при помощи синхронного PPP; либо при помощи PPP на уровне ядра с `isppp`, модифицированного драйвера sppp(4), или при помощи пользовательского ppp(8). При использовании пользовательского ppp(8) возможно использование двух и большего числа B-каналов ISDN. Также имеется приложение, работающее как автоответчик, и много утилит, таких, как программный модем на 300 Бод.

Во FreeBSD поддерживается все возрастающее число адаптеров ISDN для ПК, и сообщения показывают, что они успешно используются по всей Европе и других частях света.

Из пассивных адаптеров ISDN поддерживаются в основном те, которые сделаны на основе микросхем Infineon (бывший Siemens) ISAC/HSCX/IPAC ISDN, а также адаптеры ISDN с микросхемами от Cologne Chip (только для шины ISA), адаптеры PCI с микросхемами Winbond W6692, некоторые адаптеры с набором микросхем Tiger300/320/ISAC и несколько адаптеров, построенных на фирменных наборах микросхем, такие, как AVM Fritz!Card PCI V.1.0 и AVM Fritz!Card PnP.

На данный момент из активных адаптеров ISDN поддерживаются AVM B1 (ISA и PCI) адаптеры BRI и AVM T1 PCI адаптеры PRI.

Документацию по isdn4bsd можно найти в каталоге /usr/shared/examples/isdn/ вашей системы FreeBSD или на домашней странице isdn4bsd, на которой также размещены ссылки на советы, замечания по ошибкам и более подробную информацию, например, на руководство по isdn4bsd.

Если вы заинтересованы в добавлении поддержки для различных протоколов ISDN, не поддерживаемых на данный момент адаптеров ISDN для PC или каких-то других усовершенствованиях isdn4bsd, пожалуйста, свяжитесь с Hellmuth Michaelis <hm@FreeBSD.org>.

Для обсуждения вопросов, связанных с установкой, настройкой и устранением неисправностей isdn4bsd, имеется список рассылки freebsd-isdn.

```
subscribe freebsd-isdn
```

26.7.2. Терминальные адаптеры ISDN

Терминальные адаптеры (TA) для ISDN выполняют ту же роль, что и модемы для обычных телефонных линий.

Большинство TA используют стандартный набор AT-команд Hayes-модемов, и могут использоваться в качестве простой замены для модемов.

TA будут работать точно так же, как и модемы, за исключением скорости соединения и пропускной способности, которые будут гораздо выше, чем у вашего старого модема. Вам потребуется настроить PPP точно также, как и в случае использования модема. Проверьте, что вы задали скорость работы последовательного порта максимально высокой.

Главным преимуществом использования TA для подключения к провайдеру Интернет является возможность использования динамического PPP. Так как пространство адресов IP истощается все больше, большинство провайдеров не хочет больше выдавать вам статический IP-адрес. Большинство же маршрутизаторов не может использовать динамическое выделение IP-адресов.

TA полностью полагаются на даемон PPP, который используете из-за его возможностей и стабильности соединения. Это позволяет вам при использовании FreeBSD легко заменить модем на ISDN, если у вас уже настроено соединение PPP. Однако, в тоже время любые проблемы, которые возникают с программой PPP, отражаются и здесь.

Если вы хотите максимальной надёжности, используйте PPP на уровне параметра ядра, а не пользовательский PPP.

Известно, что следующие TA работают с FreeBSD:

- Motorola BitSurfer и Bitsurfer Pro
- Adtran

Большинство остальных TA, скорее всего, тоже будут работать, производители TA прилагают все усилия для обеспечения поддержки практически всего набора стандартных AT-команд модема.

Как и в случае модемов проблемой использования внешнего TA является потребность в хорошем последовательном адаптере на вашем компьютере.

Вы должны прочесть учебник Последовательные устройства во FreeBSD для того, чтобы в деталях понять работу последовательных устройств и осознать различие между асинхронными и синхронными последовательными портами.

TA, работающий со стандартным последовательным (асинхронным) портом PC, ограничивает вас скоростью 115.2 Кбит/с, хотя реально у вас соединение на скорости 128 Кбит/с. Чтобы использовать 128 Кбит/с, которые обеспечивает ISDN, полностью, вы должны подключить TA к синхронному последовательному адаптеру.

Не обманывайте себя, думая, что покупка встроенного TA поможет избежать проблемы синхронности/асинхронности. Встроенные TA просто уже имеют внутри стандартный

последовательный порт PC. Все, что при этом достигается - это экономия дополнительных последовательного кабеля и электрической розетки.

Синхронный адаптер с TA по крайней мере так же быстр, как и отдельный маршрутизатор, а если он работает под управлением машины класса 386 с FreeBSD, то это гораздо более гибкое решение.

Выбор между использованием синхронного адаптера/TA или отдельного маршрутизатора в большей степени является религиозным вопросом. По этому поводу в списках рассылки была некоторая дискуссия. Рекомендуем поискать в архивах обсуждение полностью.

26.7.3. Отдельные мосты/маршрутизаторы ISDN

Мосты или маршрутизаторы ISDN не так уж специфичны для FreeBSD или для любой другой операционной системы. Для более подробного описания технологий маршрутизации и работы мостов, пожалуйста, обратитесь к справочникам по сетевым технологиям.

В контексте этого раздела термины маршрутизатор и сетевой мост будут использоваться как взаимозаменяемые.

Вместе с падением цен на простые мосты/маршрутизаторы ISDN, они становятся все более популярными. Маршрутизатор ISDN представляет собой маленькую коробочку, которая подключается непосредственно в вашу сеть Ethernet, и поддерживает связь с другим мостом/маршрутизатором. Всё программное обеспечение для работы по PPP и другим протоколам встроено в маршрутизатор.

Маршрутизатор обладает гораздо большей пропускной способностью, чем стандартный TA, так как он использует полное синхронное соединение ISDN.

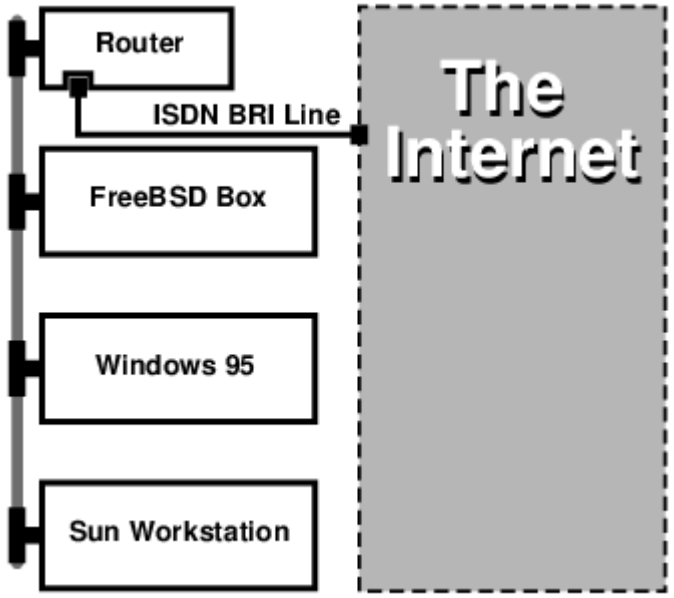
Основной проблемой с маршрутизаторами и мостами ISDN является то, что их совместная работа с оборудованием других производителей может оказаться под вопросом. Если вы собираетесь подключаться к провайдеру, то вы должны обсудить с ним то, что вам нужно.

Если вы планируете объединить два сегмента локальной сети, например, домашнюю сеть с сетью офиса, это самое простое решение с минимальными издержками на обслуживание. Так как вы покупаете оборудование для обоих сторон соединения, то можете быть уверены, что связь будет работать нормально.

Например, для соединения домашнего компьютера или сети подразделения к сети центрального офиса, может использоваться такая настройка:

Пример 42. Офис подразделения или домашняя сеть

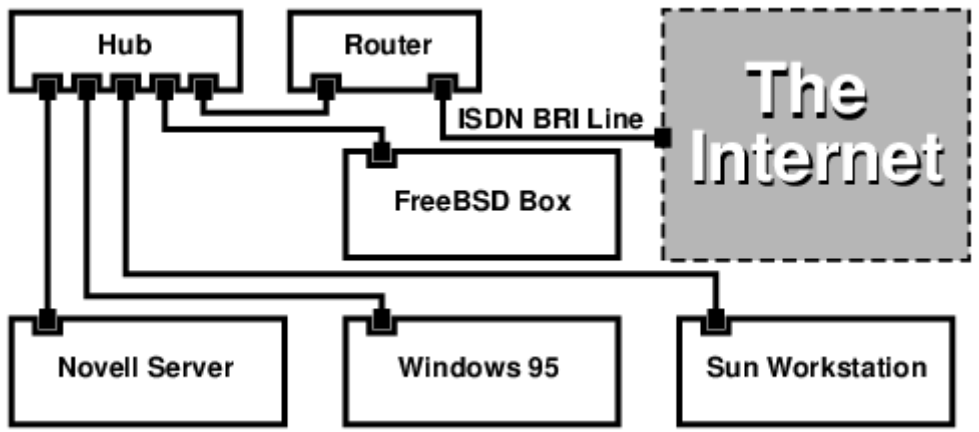
Сеть построена в топологии общей шины на основе 10 base 2 Ethernet ("thinnet" - "тонкий Ethernet"). Подключите маршрутизатор к сетевому кабелю с помощью трансивера AUI/10BT, если это нужно.



Если ваш домашний или удаленный офис представляет собой один компьютер, то для непосредственного подключения к маршрутизатору вы вы можете использовать витую пару с перекрестным соединениям.

Пример 43. Центральный офис или другая локальная сеть

Сеть построена в топологии звезды на основе 10 Base T Ethernet ("витая пара").



Одним большим преимуществом большинства маршрутизаторов/мостов является то, что они позволяют иметь 2 *отдельных независимых* соединения PPP к 2 различным сайтам *одновременно*. Это не поддерживается в большинстве TA, кроме специальных (обычно дорогих) моделей, имеющих по два последовательных порта. Не путайте это с балансировкой нагрузки, MPP и так далее.

Это может оказаться весьма полезной особенностью, например, если у вас имеется постоянное ISDN-соединение в вашем офисе, и вы хотите им воспользоваться, но не хотите задействовать дополнительный канал ISDN на работе. Маршрутизатор, расположенный в офисе, может использовать выделенное соединение по каналу B (64 Кбит/с) для Интернет, и одновременно другой канал B для отдельного соединения для передачи данных. Второй

канал B может использоваться для входящих, исходящих и динамически распределяемых соединений (MPP и так далее) совместно с первым каналом B для повышения пропускной способности.

Мост Ethernet также позволяет вам передавать больше, чем просто трафик IP. Вы сможете передавать IPX/SPX и любые другие протоколы, которые вы используете.

26.8. Даемон преобразования сетевых адресов (natd)

26.8.1. Обзор

Даемон преобразования сетевых адресов (Network Address Translation) во FreeBSD, широко известный как natd(8), является даемоном, который принимает входящие IP-пакеты, изменяет адрес отправителя на адрес локальной машины и повторно отправляет эти пакеты в потоке исходящих пакетов. natd(8) делает это, меняя IP-адрес отправителя и порт таким образом, что когда данные принимаются обратно, он может определить расположение источника начальных данных и переслать их машине, которая запрашивала данные изначально.

Чаще всего NAT используется для организации так называемого Совместного Использования Интернет.

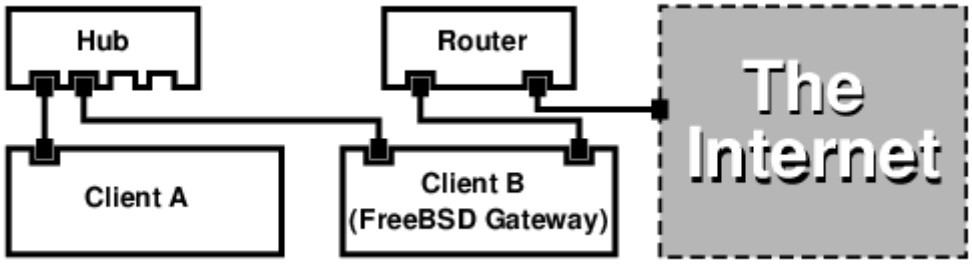
26.8.2. Настройка

Из-за исчерпания пространства адресов в IPv4 и увеличения количества пользователей высокоскоростных каналов связи, таких, как кабельное подключение или DSL, необходимость в решении по Совместному Использованию Интернет растёт. Возможность подключить несколько компьютеров через единственное соединение и IP-адрес делает natd(8) подходящим решением.

Чаще всего у пользователя имеется машина, подключенная к кабельному каналу или каналу DSL с одним IP-адресом и есть желание использовать этот единственный подключенный компьютер для организации доступа в Интернет другим компьютерам в локальной сети.

Для этого машина FreeBSD, находящаяся в Интернет, должна выступать в роли шлюза. Эта шлюзовая машина должна иметь два сетевых адаптера-один для подключения к маршрутизатору Интернет, а другой для подключения к ЛВС. Все машины в локальной сети подключаются через сетевой концентратор или коммутатор.

Существует много способов подсоединить локальную сеть к Internet через шлюз FreeBSD. Этот пример показывает шлюз с двумя сетевыми картами.



Подобная конфигурация часто используется для совместного использования доступа в Интернет. Одна из подключенных к локальной сети машин подключается к Интернет. Остальные машины работают с Интернет посредством этой "шлюзовой" машины.

26.8.3. Настройка

В файле конфигурации ядра должны присутствовать следующие параметры:

```
options IPFIREWALL
options IPDIVERT
```

Дополнительно, если это нужно, можно добавить следующее:

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

В файле /etc/rc.conf должны быть такие строки:

```
gateway_enable="YES" ①
firewall_enable="YES" ②
firewall_type="OPEN" ③
natd_enable="YES"
natd_interface="fxp0" ④
natd_flags="" ⑤
```

① Указывает машине выступать в качестве шлюза. Выполнение команды `sysctl net.inet.ip.forwarding=1` приведёт к тому же самому результату.

② При загрузке включает использование правил брандмауэра из файла /etc/rc.firewall.

③ Здесь задается предопределенный набор правил брандмауэра, который разрешает все. Посмотрите файл /etc/rc.firewall для нахождения дополнительных типов.

④ Указывает, через какой интерфейс передавать пакеты (интерфейс, подключенный к Интернет).

⑤ Любые дополнительный параметры, передаваемые при запуске даемону natd(8).

При использовании вышеуказанных параметров в файле /etc/rc.conf при загрузке будет запущена команда `natd -interface fxp0`. Эту команду можно запустить и вручную.

Если для передачи natd(8) набирается слишком много параметров, возможно также использовать конфигурационный файл. В этом случае имя настроечного файла должно быть задано добавлением следующей строки в /etc/rc.conf:

```
natd_flags="-f /etc/natd.conf"
```

Файл /etc/natd.conf будет содержать перечень конфигурационных параметров, по одному в строке. К примеру, для примера из следующего раздела будет использоваться такой файл:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcp 192.168.0.3:80 80
```

Для получения более полной информации о конфигурационном файле прочтите страницу справки по natd(8) относительно параметра `-f`.

Каждой машине и интерфейсу в ЛВС должен быть назначен IP-адрес из адресного пространства частных сетей, как это определено в RFC 1918, а в качестве маршрутизатора по умолчанию должен быть задан IP-адрес машины с natd из внутренней сети.

Например, клиенты `A` и `B` в ЛВС имеют IP-адреса `192.168.0.2` и `192.168.0.3`, а интерфейс машины с natd в локальной сети имеет IP-адрес `192.168.0.1`. Маршрутизатором по умолчанию для клиентов `A` и `B` должна быть назначена машина с natd, то есть `192.168.0.1`. Внешний, или Интернет-интерфейс машины с natd не требует особых настроек для работы natd(8).

26.8.4. Перенаправление портов

Минусом использования natd(8) является то, что машины в локальной сети недоступны из Интернет. Клиенты в ЛВС могут выполнять исходящие соединения во внешний мир, но не могут обслуживать входящие. Это является проблемой при запуске служб Интернет на клиентских машинах в локальной сети. Простым решением является перенаправление некоторых портов Интернет машины с natd на клиента локальной сети.

Пусть, к примеру, сервер IRC запущен на клиенте `A`, а Web-сервер работает на клиенте `B`. Чтобы это работало, соединения, принимаемые на портах 6667 (IRC) и 80 (Web), должны перенаправляться на соответствующие машины.

Программе natd(8) должна быть передана команда `-redirect_port` с соответствующими параметрами. Синтаксис следующий:

```
-redirect_port proto targetIP:targetPORT[-targetPORT]
            [aliasIP:]aliasPORT[-aliasPORT]
            [remoteIP[:remotePORT[-remotePORT]]]
```

В примере выше аргументы должен быть такими:

```
    -redirect_port tcp 192.168.0.2:6667 6667
    -redirect_port tcp 192.168.0.3:80 80
```

При этом будут перенаправлены соответствующие порты *tcp* на клиентские машины в локальной сети.

Аргумент `-redirect_port` может использоваться для указания диапазонов портов, а не конкретного порта. Например, *tcp 192.168.0.2:2000-3000 2000-3000* будет перенаправлять все соединения, принимаемые на портах от 2000 до 3000, на порты от 2000 до 3000 клиента `A`.

Эти параметры можно указать при непосредственном запуске natd(8), поместить их в параметр `natd_flags=""` файла /etc/rc.conf, либо передать через конфигурационный файл.

Для получение информации о других параметрах настройки обратитесь к справочной странице по natd(8)

26.8.5. Перенаправление адреса

Перенаправление адреса полезно, если имеется несколько адресов IP, и они должны быть на одной машине. В этой ситуации natd(8) может назначить каждому клиенту ЛВС свой собственный внешний IP-адрес. Затем natd(8) преобразует исходящие от клиентов локальной сети пакеты, заменяя IP-адреса на соответствующие внешние, и перенаправляет весь трафик, входящий на некоторый IP-адрес, обратно конкретному клиенту локальной сети. Это также называют статическим NAT. К примеру, пусть IP-адреса `128.1.1.1`, `128.1.1.2` и `128.1.1.3` принадлежат шлюзовой машине natd. `128.1.1.1` может использоваться в качестве внешнего IP-адреса шлюзовой машины natd, тогда как `128.1.1.2` и `128.1.1.3` будут перенаправляться обратно к клиентам ЛВС `A` и `B`.

Синтаксис для `-redirect_address` таков:

```
-redirect_address localIP publicIP
```

localIP	Внутренний IP-адрес клиента локальной сети.
publicIP	Внешний IP, соответствующий клиенту локальной сети.

В примере этот аргумент будет выглядеть так:

```
-redirect_address 192.168.0.2 128.1.1.2
-redirect_address 192.168.0.3 128.1.1.3
```

Как и для `-redirect_port`, эти аргументы также помещаются в строку `natd_flags=""` файла /etc/rc.conf или передаются через конфигурационный файл. При перенаправлении адресов

нет нужды в перенаправлении портов, потому что перенаправляются все данные, принимаемые для конкретного IP-адреса.

Внешние IP-адреса машины с natd должны быть активизированы и являться синонимами для внешнего интерфейса. Обратитесь к rc.conf(5), чтобы это сделать.

26.9. IP по параллельному порту (PLIP)

PLIP позволяет нам работать с TCP/IP по параллельному порту. Это полезно для машин без сетевых адаптеров или для установки на лэптопы. В этом разделе мы обсудим:

- создание кабеля для параллельного порта (laplink).
- Соединение двух компьютеров посредством PLIP.

26.9.1. Создание параллельного кабеля

Вы можете приобрести кабель для параллельного порта в большинстве магазинов, торгующих комплектующими. Если вы его не найдете, или же просто хотите знать, как он делается, то следующая таблица поможет вам сделать такой кабель из обычного принтерного кабеля для параллельного порта.

Таблица 17. Распайка кабеля для параллельного порта для сетевой работы

A-name	A-End	B-End	Описание	Post/Bit
.... DATA0 -ERROR	.... 2 15	.... 15 2	Data	.... 0/0x01 1/0x08 ....
.... DATA1 +SLCT	.... 3 13	.... 13 3	Data	.... 0/0x02 1/0x10 ....
.... DATA2 +PE	.... 4 12	.... 12 4	Data	.... 0/0x04 1/0x20 ....
.... DATA3 -ACK	.... 5 10	.... 10 5	Strobe	.... 0/0x08 1/0x40 ....
.... DATA4 BUSY	.... 6 11	.... 11 6	Data	.... 0/0x10 1/0x80 ....
GND	18-25	18-25	GND	-

26.9.2. Настройка PLIP

Прежде всего вы должны найти laplink-кабель. Затем удостоверьтесь, что на обоих компьютерах в ядро включена поддержка драйвера lpt(4):

```
# grep lp /var/run/dmesg.boot
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

Управление параллельным портом должно выполняться по прерываниям. Файл /boot/device.hints должен содержать следующие строки:

```
hint.ppc.0.at="isa"
hint.ppc.0.irq="7"
```

Затем проверьте, что файл конфигурации ядра имеет строку `device plip`, или загружен ли модуль ядра plip.ko. В обоих случаях интерфейс работы с сетью по параллельному порту должен присутствовать на момент использования команды ifconfig(8).

```
# ifconfig plip0
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

Подключите кабель laplink к параллельным интерфейсам на обоих компьютерах.

Настройте параметры сетевого интерфейса с обеих сторон, работая как пользователь `root`. К примеру, если вы хотите соединить хост `host1`, на котором работает FreeBSD 4.X, с хостом `host2` под управлением FreeBSD 5.X:

```
                host1 <-----> host2
IP Address    10.0.0.1      10.0.0.2
```

Настройте интерфейс на машине `host1`, выполнив:

```
# ifconfig plip0 10.0.0.1 10.0.0.2
```

Настройте интерфейс на машине `host2`, выполнив:

```
# ifconfig lp0 10.0.0.2 10.0.0.1
```

Теперь вы должны получить работающее соединение. Пожалуйста, прочтите страницы руководства по lp(4) и lpt(4) для выяснения деталей.

Вы должны также добавить оба хоста в /etc/hosts:

```
127.0.0.1               localhost.my.domain localhost
10.0.0.1                host1.my.domain host1
10.0.0.2                host2.my.domain
```

Чтобы проверить работу соединения, перейдите к каждому хосту и выполните тестирование соединения с другой машиной посредством команды ping. К примеру, на машине host1:

```
# ifconfig lp0
lp0: flags=8851<UP,POINTOPOINT,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 10.0.0.1 --> 10.0.0.2 netmask 0xff000000
# netstat -r
Routing tables

Internet:
Destination        Gateway           Flags     Refs     Use      Netif Expire
host2              host1               UH          0       0       lp0
# ping -c 4 host2
PING host2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=2.774 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=2.530 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=2.556 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=2.714 ms

--- host2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.530/2.643/2.774/0.103 ms
```

26.10. IPv6

IPv6 (также называемый IPng "IP next generation" - следующее поколение IP) является новой версией широко известного протокола IP (называемого также IPv4). Как и другие современные системы *BSD, FreeBSD включает эталонную реализацию IPv6 от KAME. Так что система FreeBSD поставляется со всем, что вам нужно для экспериментирования с IPv6. Этот раздел посвящён настройке и запуску в работу IPv6.

В начале 1990-х люди стали беспокоиться о быстро иссякающем адресном пространстве IPv4. Принимая во внимание темпы роста Интернет, имелись основные проблемы:

- Нехватка адресов. Сегодня это не такая большая проблема, так как стали применяться адресные пространства для частных сетей (RFC1918) (10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/24) и технология преобразования сетевых адресов (NAT - Network Address Translation).
- Таблицы маршрутов становятся чересчур большими. Это всё ещё является проблемой сегодня.

IPv6 решает эти и многие другие вопросы:

- 128-битное адресное пространство. Другими словами, теоретически доступны 340,282,366,920,938,463,463,374,607,431,768,211,456 адреса. Это означает плотность примерно в 6.67 * 10^27 адресов IPv6 на квадратный метр нашей планеты.
- Маршрутизаторы будут хранить в своих таблицах только агрегированные адреса сетей, что уменьшает средний размер таблицы маршрутизации до 8192 записей.

Имеется также множество других полезных особенностей IPv6, таких, как:

- Автоматическая настройка адреса (RFC2462)
- Групповые адреса ("один к нескольким из многих")
- Обязательные адреса множественной рассылки
- IPsec (IP security - безопасный IP)
- Упрощённая структура заголовка
- Мобильный IP
- Механизмы преобразования IPv6-в-IPv4

Для получения дополнительной информации посмотрите:

- Обзор IPv6 на сайте playground.sun.com
- KAME.net
- 6bone.net

26.10.1. Основы адресации IPv6

Существуют различные типы адресов IPv6: одноадресные (Unicast), групповые (Anycast) и многоадресные (Multicast).

Адреса типа Unicast хорошо всем известны. Пакет, посланный на такой адрес, достигает в точности интерфейса, который этому адресу соответствует.

Адреса типа Anycast синтаксически неотличимы от адресов Unicast, но они адресуют группу интерфейсов. Пакет, направленный такому адресу, попадёт в ближайший (согласно метрике маршрутизатора) интерфейс. Адреса Anycast могут использоваться только маршрутизаторами.

Адреса типа Multicast идентифицируют группу интерфейсов. Пакет, посланный на такой адрес, достигнет всех интерфейсов, привязанных к группе многоадресного вещания.

Широковещательные адреса IPv4 (обычно `xxx.xxx.xxx.255`) выражаются адресами многоадресного вещания IPv6.

Таблица 18. Зарезервированные адреса IPv6

IPv6 адрес	Длина префикса (биты)	Описание	Заметки
`::`	128 бит	нет описания	cf. `0.0.0.0` в IPv4

IPv6 адрес	Длина префикса (биты)	Описание	Заметки
`::1`	128 бит	loopback адрес	cf. `127.0.0.1` в IPv4
`::00:xx:xx:xx:xx`	96 бит	встроенный IPv4	Нижние 32 бита это адрес IPv4. Также называется "IPv4 совместимым IPv6 адресом"
`::ff:xx:xx:xx:xx`	96 бит	Адрес IPv6, отображенный на IPv4	Нижние 32 бита это адрес IPv4. Для хостов, не поддерживающих IPv6.
`fe80::` - `feb::`	10 бит	link-local	cf. loopback адрес в IPv4
`fec0::` - `fef::`	10 бит	site-local	
`ff::`	8 бит	широковещательный	
`001` (основание 2)	3 бит	global unicast	Все global unicast адреса присваиваются из этого пула. Первые три бита "001".

26.10.2. Чтение адресов IPv6

Каноническая форма представляется в виде `x:x:x:x:x:x:x:x`, где каждый символ "x" является 16-разрядным числом в шестнадцатеричной форме. К примеру, `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`

Часто в адресе присутствуют длинные строчки, заполненные нулями, поэтому одна такая последовательность на адрес может быть сокращена до "::". Кроме того, до трех ведущих "0" на шестнадцатеричную четверку могут быть пропущены. К примеру, `fe80::1` соответствует канонической форме `fe80:0000:0000:0000:0000:0000:0000:0001`.

В третьей форме последние 32 бита записываются в широко известном (десятичном) стиле IPv4 с точками "." в качестве разделителей. Например, `f2002::10.0.0.1` соответствует (шестнадцатеричному) каноническому представлению `2002:0000:0000:0000:0000:0000:0a00:0001`, которое, в свою очередь, равнозначно записи `2002::a00:1`.

Теперь читатель должен понять следующую запись:

```
# ifconfig
```

```
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
```

```
        inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
        inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
        ether 00:00:21:03:08:e1
        media: Ethernet autoselect (100baseTX )
        status: active
```

`fe80::200:21ff:fe03:8e1%rl0` является автоматически настроенным локальным адресом. Он генерируется из MAC адреса в процессе автоматической конфигурации.

Для получения дополнительной информации о структуре адресов IPv6 обратитесь к RFC3513.

26.10.3. Настройка подключения

На данный момент существуют четыре способа подключиться к другим хостам и сетям IPv6:

- Подключиться к экспериментальному 6bone
- Получить сеть IPv6 от вышестоящего провайдера. Для получения рекомендаций обратитесь к вашему провайдеру Интернет.
- Туннелировать посредством 6-в-4 (RFC3068)
- Использовать порт net/freenet6, если вы используете коммутируемое соединение.

Здесь мы будем рассматривать подключение к 6bone, так как на данный момент это является самым популярным способом.

Сначала взгляните на сайт 6bone и найдите ближайшую к вам точку подключения к 6bone. Напишите ответственному и при некоторой удаче вам дадут инструкции по настройке соединения. Обычно это касается настройки туннеля GRE (gif).

Вот типичный пример настройки туннеля gif(4):

```
# ifconfig gif0 create
# ifconfig gif0
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
# ifconfig gif0 tunnel MY_IPv4_ADDR  MY_IPv4_REMOTE_TUNNEL_ENDPOINT_ADDR
# ifconfig gif0 inet6 alias MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR
MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR
```

Замените слова, написанные заглавными буквами, информацией, которую вам дал вышестоящий узел 6bone.

При этом установится туннель. Проверьте работу туннеля утилитой ping6(8) с адресом `ff02::1%gif0`. Вы должны получить два положительных ответа.

Если вы заинтригованы адресом `ff02:1%gif0`, скажем, что это адрес многоадресного вещания. `%gif0` указывает на использование такого адреса с сетевым интерфейсом gif0. Так как мы выполняем `ping` над адресом

многоадресного вещания, то другая сторона туннеля также должна ответить.

Теперь настройка маршрута к вашей вышестоящей точке подключения 6bone должна быть весьма проста:

```
# route add -inet6 default -interface gif0
# ping6 -n MY_UPLINK
```

```
# traceroute6 www.jp.FreeBSD.org
(3ffe:505:2008:1:2a0:24ff:fe57:e561) from 3ffe:8060:100::40:2, 30 hops max, 12 byte
packets
     1  atnet-meta6  14.147 ms  15.499 ms  24.319 ms
     2  6bone-gw2-ATNET-NT.ipv6.tilab.com  103.408 ms  95.072 ms *
     3  3ffe:1831:0:ffff::4  138.645 ms  134.437 ms  144.257 ms
     4  3ffe:1810:0:6:290:27ff:fe79:7677  282.975 ms  278.666 ms  292.811 ms
     5  3ffe:1800:0:ff00::4  400.131 ms  396.324 ms  394.769 ms
     6  3ffe:1800:0:3:290:27ff:fe14:cdee  394.712 ms  397.19 ms  394.102 ms
```

Эта выдача будет отличаться от машины к машине. Теперь вы должны суметь достигнуть сайта IPv6 www.kame.net и увидеть танцующую черепаху - в случае, если ваш браузер поддерживает IPv6, как, например, www/mozilla или Konqueror, который входит в x11/kdebase3, или www/epiphany.

26.10.4. DNS в мире IPv6

Для IPv6 использовались два типа записей DNS. IETF объявил записи A6 устаревшими. Стандартом на данный момент являются записи AAAA.

Использование записей AAAA достаточно просто. Назначение вашему имени хоста нового адреса IPv6 достигается просто добавлением:

```
MYHOSTNAME          AAAA    MYIPv6ADDR
```

к вашему первичному файлу DNS зоны. В случае, если вы не обслуживаете собственные зоны DNS, обратитесь к вашему провайдеру DNS. Имеющиеся версии bind (версий 8.3 и 9) и dns/djbdns (с патчем IPv6) поддерживают записи AAAA.

26.10.5. Внесение необходимых изменений в /etc/rc.conf

26.10.5.1. Настройки клиентов IPv6

Эти установки помогут вам настроить компьютер, который будет работать в сети как клиент, а не как маршрутизатор. Для включения настройки интерфейсов через rtsol(8) при загрузке, все, что вам потребуется, это добавить следующую строку:

```
ipv6_enable="YES"
```

Для статического присвоения IP адреса, такого как `2001:471:1f11:251:290:27ff:fee0:2093`, интерфейсу fxp0, добавьте:

```
ipv6_ifconfig_fxp0="2001:471:1f11:251:290:27ff:fee0:2093"
```

Для назначения маршрутизатором по умолчанию `2001:471:1f11:251::1`, добавьте следующую строку к /etc/rc.conf:

```
ipv6_defaultrouter="2001:471:1f11:251::1"
```

26.10.5.2. Настройки маршрутизатора/шлюза IPv6

Этот раздел поможет вам использовать инструкции, которые выдал провайдер туннеля, например, 6bone, и сделать эти настройки постоянными. Для восстановления туннеля при загрузке системы используйте в /etc/rc.conf нижеприведенные настройки.

Задайте список туннельных интерфейсов (Generic Tunneling interfaces), которые необходимо настроить, например gif0:

```
gif_interfaces="gif0"
```

Для настройки интерфейса с локальным подключением на *MY_IPv4_ADDR* к удаленной точке *REMOTE_IPv4_ADDR*:

```
gifconfig_gif0="MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

Для включения IPv6 адреса, который был вам присвоен для использования в подключении к туннелю IPv6, добавьте:

```
ipv6_ifconfig_gif0="MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Затем все, что вам потребуется сделать, это добавить маршрут по умолчанию для IPv6. Это другая сторона туннеля IPv6:

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

26.10.5.3. Настройка туннелирования IPv6

Если сервер будет обеспечивать маршрутизацию между вашей сетью и остальным миром, то в файле /etc/rc.conf понадобится следующая строка:

```
ipv6_gateway_enable="YES"
```

26.10.6. Распространение маршрутов и автоматическая настройка хостов

Этот раздел поможет вам настроить rtadvd(8) для распространения маршрута IPv6 по умолчанию.

Для включения rtadvd(8) вам понадобится добавить в /etc/rc.conf следующую строку:

```
rtadvd_enable="YES"
```

Важно указать интерфейс, на котором выполняется запрос маршрутизатора IPv6. Например, для указания rtadvd(8) использовать fxp0:

```
rtadvd_interfaces="fxp0"
```

Теперь мы должны создать файл настройки, /etc/rtadvd.conf. Вот пример:

```
fxp0:\
        :addrs#1:addr="2001:471:1f11:246::":prefixlen#64:tc=ether:
```

Замените fxp0 на интерфейс, который вы будете использовать.

Затем, замените `2001:471:1f11:246::` на префикс вашего размещения.

Если у вас выделенная подсеть `/64`, больше ничего менять не потребуется. Иначе, вам потребуется изменить `prefixlen#` на корректное значение.

26.11. Асинхронный режим передачи (ATM)

26.11.1. Классическая настройка IP через ATM (PVC)

Классический IP через ATM (CLIP) это простейший метод использования асинхронного режима передачи (Asynchronous Transfer Mode, ATM) с IP. Он может быть использован с коммутируемыми подключениями (switched connections, SVC) и с постоянными подключениями (permanent connections, PVC). В этом разделе будет описано как настроить сеть на основе PVC.

26.11.1.1. Полностью объединенные конфигурации

Первый метод для настройки CLIP с PVC это подключение каждого компьютера к каждому в сети с выделенным PVC. Хотя настройка проста, она непрактична для большого количества компьютеров. В примере предполагается, что в сети есть четыре компьютера, каждый

подключенный к ATM сети с помощью карты ATM адаптера. Первый шаг это планирование IP адресов и ATM подключений между компьютерами. Мы используем:

Хост	IP адрес
hostA	192.168.173.1
hostB	192.168.173.2
hostC	192.168.173.3
hostD	192.168.173.4

Для сборки полностью объединенной сети нам потребуется по одному ATM соединению между каждой парой компьютеров:

Компьютеры	VPI.VCI соединение
hostA - hostB	0.100
hostA - hostC	0.101
hostA - hostD	0.102
hostB - hostC	0.103
hostB - hostD	0.104
hostC - hostD	0.105

Значения VPI и VCI на каждом конце соединения конечно могут отличаться, но для упрощения мы предполагаем, что они одинаковы. Затем нам потребуется настроить ATM интерфейсы на каждом хосте:

```
hostA# ifconfig hatm0 192.168.173.1 up
hostB# ifconfig hatm0 192.168.173.2 up
hostC# ifconfig hatm0 192.168.173.3 up
hostD# ifconfig hatm0 192.168.173.4 up
```

предполагая, что ATM интерфейс называется hatm0 на всех хостах. Теперь PVC необходимо настроить на hostA (мы предполагаем, что ATM коммутаторы уже настроены, вам необходимо свериться с руководством на коммутатор за информацией по настройке).

```
hostA# atmconfig natm add 192.168.173.2 hatm0 0 100 llc/snap ubr
hostA# atmconfig natm add 192.168.173.3 hatm0 0 101 llc/snap ubr
hostA# atmconfig natm add 192.168.173.4 hatm0 0 102 llc/snap ubr

hostB# atmconfig natm add 192.168.173.1 hatm0 0 100 llc/snap ubr
hostB# atmconfig natm add 192.168.173.3 hatm0 0 103 llc/snap ubr
hostB# atmconfig natm add 192.168.173.4 hatm0 0 104 llc/snap ubr

hostC# atmconfig natm add 192.168.173.1 hatm0 0 101 llc/snap ubr
hostC# atmconfig natm add 192.168.173.2 hatm0 0 103 llc/snap ubr
hostC# atmconfig natm add 192.168.173.4 hatm0 0 105 llc/snap ubr
```

```
hostD# atmconfig natm add 192.168.173.1 hatm0 0 102 llc/snap ubr
hostD# atmconfig natm add 192.168.173.2 hatm0 0 104 llc/snap ubr
hostD# atmconfig natm add 192.168.173.3 hatm0 0 105 llc/snap ubr
```

Конечно, вместо UBR может быть использован другой тип, если ATM адаптер поддерживает это. В этом случае имя типа дополняется параметрами трафика. Помощь по atmconfig(8) может быть получена командой:

```
# atmconfig help natm add
```

или на странице справочника atmconfig(8).

Та же настройка может быть выполнена через /etc/rc.conf. Для `hostA` это будет выглядеть примерно так:

```
network_interfaces="lo0 hatm0"
ifconfig_hatm0="inet 192.168.173.1 up"
natm_static_routes="hostB hostC hostD"
route_hostB="192.168.173.2 hatm0 0 100 llc/snap ubr"
route_hostC="192.168.173.3 hatm0 0 101 llc/snap ubr"
route_hostD="192.168.173.4 hatm0 0 102 llc/snap ubr"
```

Текущий статус всех маршрутов CLIP может быть получен командой:

```
hostA# atmconfig natm show
```

Часть V: Приложения

Приложение A: Получение FreeBSD

A.1. Наборы CD и DVD

Наборы FreeBSD CD и DVD доступны у нескольких онлайн поставщиков:

- FreeBSD Mall, Inc.
  2420 Sand Creek Rd C-1 #347
  Brentwood, CA
  94513
  США
  Телефон: +1 925 240-6652
  Факс: +1 925 674-0821
  Email: <info@freebsdmall.com>
  WWW: http://www.freebsdmall.com/
- Getlinux
  78 Rue de la Croix Rochopt
  'Epinay-sous-S'enart
  91860
  Франция
  Email: <contact@getlinux.fr>
  WWW: http://www.getlinux.fr/
- Dr. Hinner EDV
  Kochelseestr. 11
  D-81371 M"unchen
  Германия
  Телефон: (0177) 428 419 0
  Email: <infow@hinner.de>
  WWW: http://www.hinner.de/linux/freebsd.html
- Linux Center
  ул. Галерная, 55
  Санкт-Петербург
  190000
  Россия
  Телефон: +7-812-3125208
  Email: <info@linuxcenter.ru>
  WWW: http://linuxcenter.ru/shop/freebsd

A.2. FTP сайты

Официальные исходные тексты FreeBSD доступны через анонимные FTP зеркала по всему миру. Сайт ftp://ftp.FreeBSD.org/pub/FreeBSD/ имеет хорошее подключение и поддерживает большое количество одновременных соединений, но для вас возможно потребуется найти "ближайшее" зеркало (особенно если вы решили настроить у себя какой-то из видов зеркал).

Кроме того, FreeBSD доступна через анонимный FTP со следующих зеркал. Если вы выбрали получение FreeBSD через анонимный FTP, пожалуйста, выберите ближайший к вам сайт. Зеркала из списка "Основных зеркал" обычно содержат полный архив FreeBSD (все доступные на данный момент версии для каждой архитектуры), но скорость загрузки возможно будет больше с зеркала, расположенного в вашей стране или регионе. Сайты каждой страны содержат последнюю версию для наиболее популярных архитектур, но на них может не быть полного архива FreeBSD. Все сайты предоставляют доступ через анонимный FTP, а некоторые предоставляют доступ и другими методами. Для каждого сайта приведен список методов доступа в скобках после имени хоста.

Central Servers, Primary Mirror Sites, Armenia, Australia, Austria, Brazil, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hong Kong, Ireland, Japan, Korea, Latvia, Lithuania, Netherlands, New Zealand, Norway, Poland, Russia, Saudi Arabia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Ukraine, United Kingdom, United States of America.

(as of UTC)

Central Servers

ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.FreeBSD.org/pub/FreeBSD/ / http://ftp.FreeBSD.org/pub/FreeBSD/)

Primary Mirror Sites

In case of problems, please contact the hostmaster `<mirror-admin@FreeBSD.org>` for this domain.

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.FreeBSD.org/pub/FreeBSD/ / http://ftp4.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp10.FreeBSD.org/pub/FreeBSD/ / http://ftp10.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.FreeBSD.org/pub/FreeBSD/)

Armenia

In case of problems, please contact the hostmaster `<hostmaster@am.FreeBSD.org>` for this domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.am.FreeBSD.org/pub/FreeBSD/ / rsync)

Australia

In case of problems, please contact the hostmaster <hostmaster@au.FreeBSD.org> for this domain.

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

Austria

In case of problems, please contact the hostmaster <hostmaster@at.FreeBSD.org> for this domain.

- ftp://ftp.at.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.at.FreeBSD.org/pub/FreeBSD/ / http://ftp.at.FreeBSD.org/pub/FreeBSD/)

Brazil

In case of problems, please contact the hostmaster <hostmaster@br.FreeBSD.org> for this domain.

- ftp://ftp2.br.FreeBSD.org/FreeBSD/ (ftp / http://ftp2.br.FreeBSD.org/)
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)

Czech Republic

In case of problems, please contact the hostmaster <hostmaster@cz.FreeBSD.org> for this domain.

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.cz.FreeBSD.org/pub/FreeBSD/)

Denmark

In case of problems, please contact the hostmaster <staff@dotsrc.org> for this domain.

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.dk.FreeBSD.org/pub/FreeBSD/ / http://ftp.dk.FreeBSD.org/pub/FreeBSD/)

Estonia

In case of problems, please contact the hostmaster <hostmaster@ee.FreeBSD.org> for this domain.

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

Finland

In case of problems, please contact the hostmaster <hostmaster@fi.FreeBSD.org> for this domain.

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

France

In case of problems, please contact the hostmaster <hostmaster@fr.FreeBSD.org> for this domain.

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.fr.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

Germany

In case of problems, please contact the hostmaster <de-bsd-hubs@de.FreeBSD.org> for this domain.

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.de.FreeBSD.org/freebsd/ (ftp / http://www1.de.FreeBSD.org/freebsd/ / rsync://rsync3.de.FreeBSD.org/freebsd/)
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.de.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp4.de.FreeBSD.org/FreeBSD/ (ftp / http://ftp4.de.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp7.de.FreeBSD.org/pub/FreeBSD/)

Greece

In case of problems, please contact the hostmaster <hostmaster@gr.FreeBSD.org> for this domain.

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

Hong Kong

ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

Ireland

In case of problems, please contact the hostmaster <hostmaster@ie.FreeBSD.org> for this domain.

- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Japan

In case of problems, please contact the hostmaster <hostmaster@jp.FreeBSD.org> for this domain.

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

Korea

In case of problems, please contact the hostmaster <hostmaster@kr.FreeBSD.org> for this domain.

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.kr.FreeBSD.org/pub/FreeBSD/)

Latvia

In case of problems, please contact the hostmaster <hostmaster@lv.FreeBSD.org> for this domain.

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lv.FreeBSD.org/pub/FreeBSD/)

Lithuania

In case of problems, please contact the hostmaster <hostmaster@lt.FreeBSD.org> for this domain.

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lt.FreeBSD.org/pub/FreeBSD/)

Netherlands

In case of problems, please contact the hostmaster <hostmaster@nl.FreeBSD.org> for this domain.

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nl.FreeBSD.org/os/FreeBSD/ / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

New Zealand

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nz.FreeBSD.org/pub/FreeBSD/)

Norway

In case of problems, please contact the hostmaster <hostmaster@no.FreeBSD.org> for this domain.

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Poland

In case of problems, please contact the hostmaster <hostmaster@pl.FreeBSD.org> for this domain.

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp2.pl.FreeBSD.org

Russia

In case of problems, please contact the hostmaster <hostmaster@ru.FreeBSD.org> for this domain.

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ru.FreeBSD.org/FreeBSD/ / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp5.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

Saudi Arabia

In case of problems, please contact the hostmaster <ftpadmin@isu.net.sa> for this domain.

- ftp://ftp.isu.net.sa/pub/ftp.freebsd.org (ftp)

Slovenia

In case of problems, please contact the hostmaster <hostmaster@si.FreeBSD.org> for this domain.

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)

South Africa

In case of problems, please contact the hostmaster <hostmaster@za.FreeBSD.org> for this domain.

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

Spain

In case of problems, please contact the hostmaster <hostmaster@es.FreeBSD.org> for this domain.

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.es.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

Sweden

In case of problems, please contact the hostmaster <hostmaster@se.FreeBSD.org> for this domain.

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.se.FreeBSD.org/)
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.se.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.se.FreeBSD.org/pub/FreeBSD/)

Switzerland

In case of problems, please contact the hostmaster <hostmaster@ch.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ch.FreeBSD.org/pub/FreeBSD/)

Taiwan

In case of problems, please contact the hostmaster <hostmaster@tw.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.tw.FreeBSD.org/ / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp11.tw.FreeBSD.org/FreeBSD/)
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

Ukraine

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ua.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.ua.FreeBSD.org/pub/FreeBSD / rsync://ftp6.ua.FreeBSD.org/FreeBSD/)
- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

United Kingdom

In case of problems, please contact the hostmaster <hostmaster@uk.FreeBSD.org> for this domain.

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.uk.FreeBSD.org/ftp.freebsd.org/pub/FreeBSD/)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

United States of America

In case of problems, please contact the hostmaster <hostmaster@us.FreeBSD.org> for this domain.

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.us.FreeBSD.org/pub/FreeBSD/ / http://ftp4.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp13.us.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

A.3. Использование CTM

CTM это метод синхронизации удаленного дерева каталогов с центральным. Он встроен во FreeBSD и может использоваться для синхронизации системы с репозиториями исходных текстов FreeBSD. Он поддерживает синхронизацию всего репозитория или только заданного набора ветвей.

CTM создан специально для использования в условиях некачественного соединения по TCP/IP или его отсутствия и предоставляет возможность автоматической отправки изменений по электронной почте. Пользователю требуется загружать до трех изменений в день для наиболее активных ветвей. Размеры обновлений всегда поддерживаются настолько малыми, насколько это возможно, и обычно составляют меньше 5 Кб. Примерно на каждое десятое приходится по 10-50 Кб, и иногда случаются обновления больше 100 Кб.

При использовании CTM для отслеживания процесса разработки FreeBSD требуется учитывать особенности в работе напрямую с исходных текстов, находящихся в разработке, по сравнению с использованием готовых релизов. Эти вопросы рассматриваются в разделе Использование ветви разработки.

Существует немного документации по процессу создания дельта-файлов (delta, разница между имеющимися и актуальными исходными текстами) или использованию CTM в других целях. Обратитесь в список рассылки ctm-users-desc для получения ответов на вопросы по использованию CTM.

A.3.1. Получение дельта-файлов

"Дельта-файлы" для использования в CTM можно получить двумя способами: через анонимный FTP или по электронной почте.

Дельта-файлы по FTP можно получить со следующих сайтов зеркал. При использовании анонимного FTP для получения дельта-файлов CTM выберите географически ближайшее зеркало. По случаю возникновения проблем сообщайте в список рассылки ctm-users-desc.

США, Калифорния, Bay Area, официальный источник

- ftp://ftp.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ftp.FreeBSD.org/pub/FreeBSD/CTM/

Южная Африка, запасной сервер для старых дельт

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/CTM/

Тайвань/R.O.C.

- ftp://ctm.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm2.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm3.tw.FreeBSD.org/pub/FreeBSD/development/CTM/

Для получения дельта-файлов по почте подпишитесь на один из списков распространения `ctm-src`, доступных на http://lists.freebsd.org/mailman/listinfo. Например, ctm-src-cur-desc поддерживает главную ветвь разработки, а ctm-src-9-desc поддерживает ветвь выпуска релизов 9.X.

Как только вы получаете обновления CTM по почте, используйте программу `ctm_rmail` для их распаковки и применения. Эта программа может выполняться непосредственно из записи в /etc/aliases для автоматизации процесса. Обращайтесь к странице справочника ctm_rmail(1) для получения дополнительной информации.

Вне зависимости от способа получения дельта-файлов, пользователям CTM следует подписаться на список рассылки ctm-announce-desc, поскольку это единственный механизм публикации объявлений CTM.

A.3.2. Использование CTM

Перед началом использования дельта-файлов CTM потребуется определить исходную точку для последующего их применения.

Один из способов состоит в применении "стартового" дельта-файла к пустому каталогу. В имени такого файла присутствует `Xempty`, например, src-cur.3210XEmpty.gz. Обозначение перед `X` соответствует происхождению первоначального источника. Empty означает пустой каталог. Как правило, файл с `Empty` создается через каждые 100 дельта-файлов. Обратите внимание, что стартовые дельта-файлы имеют большой размер, и от 70 до 80 мегабайт сжатых в `gzip` данных для XEmpty является обычным делом.

Другой способ заключается в получении первоначального источника с -RELEASE CD. Это может существенно снизить объём передаваемых данных по сети.

Когда основной дельта-файл создан, примените все дельта-файлы с последующими номерами. Чтобы применить дельта-файлы:

```
# cd /directory/to/store/the/stuff
# ctm -v -v /directory/which/stores/the/deltas/src-xxx.*
```

Можно применять несколько дельт одной командой, по мере их последовательной обработки уже применённые дельты игнорируются. CTM работает с дельта-файлами, сжатыми с помощью `gzip`, что позволяет сэкономить на используемом дисковом пространстве.

Для проверки дельта-файла без его применения используйте параметр командной строки `-c`. CTM не будет модифицировать локальное дерево, а только проверит целостность дельта-файла на предмет его применимости без ошибок. Обращайтесь к ctm(1) для получения дополнительной информации по имеющимся параметрам и понимания процесса применения дельт, который используется в CTM.

Для поддержания исходных текстов в актуальном состоянии каждый раз, когда становится доступна новая дельта, применяйте её с использованием CTM.

Рекомендуется не удалять дельты после применения, если их сложно загрузить повторно. В этом случае будет доступна локальная копия, которая может понадобиться при восстановлении после сбоя.

A.3.3. Сохранение локальных изменений

Разработчики часто экспериментируют и изменяют файлы в локальном дереве исходных текстов. CTM имеет ограниченную поддержку локальных изменений: перед проверкой наличия файла сначала проверяется файл тем же именем и расширением .ctm. Если такой файл присутствует, CTM будет работать с ним вместо исходного файла.

Такое поведение обеспечивает простой путь поддержки локальных изменений. Перед изменением файла скопируйте его с расширением .ctm. Вносите любые изменения в исходный файл, зная что CTM будет применять обновления только к файлу с расширением .ctm.

A.3.4. Другие возможности CTM

Определение файлов, которые будут затронуты обновлением

Для определения списка изменений, которые CTM внесет в локальный репозиторий исходных текстов, используйте параметр `-l`. Этот параметр используется для записи лога изменений или выполнения предварительной или последующей обработки какого-либо подмножества изменяемых файлов.

Создание резервных копий перед обновлением

Для создания резервной копии всех файлов, которые будут изменены обновлением CTM, укажите параметр `-B backup-file`. С этим параметром CTM выполняет сохранение в backup-file всех файлов, которые затрагиваются применяемыми дельтами CTM.

Ограничение обновлений для определенных файлов

Для ограничения набора файлов, обновляемых CTM, или для извлечения лишь нескольких файлов из последовательности дельт можно указать фильтрующие регулярные выражения с использованием `-e`, который указывает, какие файлы обрабатывать, или `-x`, который указывает, какие файлы игнорировать.

Пример извлечения свежей копии lib/libc/Makefile из коллекции сохраненных дельт CTM:

```
# cd /directory/to/extract/to/
# ctm -e '^lib/libc/Makefile' /directory/which/stores/the/deltas/src-xxx.*
```

Для каждого файла, указанного в CTM дельте, параметры `-e` и `-x` применяются в порядке их задания в командной строке. Файл обрабатывается CTM, только если он помечается как подходящий после обработки всех параметров `-e` и `-x`.

A.4. Использование Subversion

A.4.1. Введение

По состоянию на июль 2012 года FreeBSD использует Subversion в качестве основной системы контроля версий для хранения всего исходного кода FreeBSD, документации и Коллекции Портов.

Subversion в основном является инструментом разработчика. Большинству пользователей следует использовать `freebsd-update` (Обновление FreeBSD) для обновления основной системы FreeBSD и `portsnap` (Использование Коллекции Портов) для обновления Коллекции Портов FreeBSD.

В этом разделе демонстрируется, как устанавливать Subversion в системе FreeBSD и затем использовать его для создания локальной копии репозитория FreeBSD. Здесь приводится список доступных зеркал Subversion для FreeBSD, а также ссылки на дополнительную информацию по использованию Subversion.

A.4.2. Установка

Subversion должен быть установлен до его использования для получения содержимого любого из репозиториев. Если уже имеется копия дерева портов, Subversion можно установить следующим образом:

```
# cd /usr/ports/devel/subversion
# make install clean
```

Если дерево портов недоступно, Subversion можно установить из пакета:

```
# pkg_add -r subversion
```

Если для управления пакетов используется pkgng, то Subversion можно установить с его помощью:

```
# pkg install devel/subversion
```

A.4.3. Работа с Subversion

Команда `svn` используется для извлечения чистой копии исходных кодов в локальный каталог. Файлы в этом каталоге называются *локальной рабочей копией*.

Если локальный каталог уже существует, но не был создан с помощью `svn`, переименуйте его или удалите перед загрузкой. Загрузка в существующий не-`svn` каталог может вызвать конфликты между существующими файлами и получаемыми из репозитория.

Subversion для обозначения репозитория использует URL, которые имеют вид *протокол://имя/путь*. Зеркала могут поддерживать различные протоколы как указано ниже. Первый компонент пути обозначает используемый репозиторий. Существует три различных репозитория: `base` для исходного кода основной системы FreeBSD, `ports` для Коллекции Портов и `doc` для документации. Например, URL `svn://svn0.us-east.FreeBSD.org/ports/head/` указывает на главную ветвь репозитория портов на зеркале `svn0.us-east.FreeBSD.org` с использованием протокола `svn`.

Загрузка из данного репозитория выполняется следующей командой:

```
# svn checkout svn-mirror/repository/branch lwcdir
```

где:

- *svn-mirror* - URL для одного из сайтов зеркала Subversion.
- *repository* - один из репозиториев проекта, т.е. `base`, `ports` или `doc`.
- *branch* зависит от используемого репозитория. `ports` и `doc` в основном обновляются в ветви `head`, в то время как `base` содержит последнюю версию -CURRENT в `head` и соответственно последние версии ветви -STABLE в `stable/8` (для 8.*x*), `stable/9` (9.*x*) и `stable/10` (10.*x*).
- *lwcdir* - каталог для размещения содержимого указанной ветви. Обычно это /usr/ports для `ports`, /usr/src для `base` и /usr/doc для `doc`.

В этом примере загружается Коллекция Портов с западного репозитория США с использованием протокола HTTPS и размещением локальной рабочей копии в /usr/ports. Если /usr/ports уже присутствует, но не был создан с помощью `svn`, не забудьте его переименовать или удалить перед загрузкой.

```
# svn checkout https://svn0.us-west.FreeBSD.org/ports/head /usr/ports
```

Поскольку на первоначальном этапе с удалённого репозитория загружается вся ветвь целиком, на это может уйти некоторое время. Пожалуйста, будьте терпеливы.

После первоначальной загрузки локальную рабочую копию можно обновить:

```
# svn update lwcdir
```

Для обновления /usr/ports, созданного в вышеприведённом примере, используйте:

```
# svn update /usr/ports
```

Обновление намного быстрее загрузки, т.к. передаются только файлы с изменениями.

Альтернативный способ обновления локальной рабочей копии после загрузки обеспечивается в Makefile в каталогах /usr/ports, /usr/src и /usr/doc. Используйте цель `update` с заданной переменной `SVN_UPDATE`. Пример для обновления /usr/src:

```
# cd /usr/src
# make update SVN_UPDATE=yes
```

A.4.4. Сайты зеркала Subversion

Все зеркала покрывают все репозитории.

Главный сервер Subversion FreeBSD `svn.FreeBSD.org` является общедоступным для чтения. Это может измениться в будущем, поэтому пользователям рекомендуется использовать одно из официальных зеркал. Для просмотра репозиториев Subversion FreeBSD через браузер используйте http://svnweb.FreeBSD.org/.

Сеть зеркал Subversion FreeBSD находится на раннем этапе развития и скорее всего будет меняться. Не полагайтесь на неизменность этого списка. В частности, серверные сертификаты SSL скорее всего изменятся.

Название	Протоколы	Местоположение	SSL Fingerprint
svn0.us-west.FreeBSD.org	svn, http, https	США, Калифорния	SHA1 1C:BD:85:95:11:9F:EB:75:A5:4B:C8:A3:FE:08:E4:02:73:06:1E:61
svn0.us-east.FreeBSD.org	svn, http, https, rsync	Сша, Нью Джерси	SHA1 1C:BD:85:95:11:9F:EB:75:A5:4B:C8:A3:FE:08:E4:02:73:06:1E:61
svn0.eu.FreeBSD.org	svn, http, https, rsync	Великобритания	SHA1 39:B0:53:35:CE:60:C7:BB:00:54:96:96:71:10:94:BB:CE:1C:07:A7

Название	Протоколы	Местоположение	SSL Fingerprint
svn0.ru.FreeBSD.org	svn, http, https, rsync	Россия, Москва	SHA1 F6:44:AA:B9:03:89:0E:3E:8C:4D:4D:14:F0:27:E6:C7:C1:8B:17:C5

Предпочтительным протоколом является HTTPS, который обеспечивает защиту от других компьютеров, маскирующихся под зеркало FreeBSD (известно как атака "человек посредине"), и прочих, пытающихся послать плохое содержимое конечному пользователю.

При первом соединении с зеркалом по HTTPS пользователю будет предложено проверить *fingerprint* (отпечаток) сервера:

```
Error validating server certificate for 'https://svn0.us-west.freebsd.org:443':
 - The certificate is not issued by a trusted authority. Use the
   fingerprint to validate the certificate manually!
 - The certificate hostname does not match.
Certificate information:
 - Hostname: svnmir.ysv.FreeBSD.org
 - Valid: from Jul 29 22:01:21 2013 GMT until Dec 13 22:01:21 2040 GMT
 - Issuer: clusteradm, FreeBSD.org, (null), CA, US (clusteradm@FreeBSD.org)
 - Fingerprint: 1C:BD:85:95:11:9F:EB:75:A5:4B:C8:A3:FE:08:E4:02:73:06:1E:61
(R)eject, accept (t)emporarily or accept (p)ermanently?
```

Сравните отпечаток с вышеуказанными в таблице. Если отпечаток совпадает, сертификат безопасности сервера можно принять на временной или постоянной основе. Временный сертификат действует до конца сессии с сервером, и при следующем соединении этап верификации будет повторён. Постоянное принятие сертификата сохраняет параметры аутентификации в ~/.subversion/auth/, и пользователю не придётся проверять отпечаток снова до истечения сертификата.

Если `https` не получается использовать из-за фаервола или иных проблем, `svn` - следующий выбор с чуть более быстрой передачей. Если ни один из них не может быть использован, используйте `http`.

A.4.5. Дополнительная информация

Для получения другой информации по использованию Subversion смотрите "книгу Subversion" по названию Version Control with Subversion или Документацию Subversion.

A.5. Использование rsync

Следующие сайты организуют доступ к FreeBSD через протокол rsync. Утилита rsync работает в основном таким же способом, что и команда rcp(1), но поддерживает больше параметров и использует протокол удаленного обновления rsync, который передает только разницу между двумя наборами файлов, что значительно повышает скорость синхронизации по сети. Это особенно полезно, если вы поддерживаете зеркало сервера FreeBSD FTP или репозитория CVS. Пакет rsync доступен для многих операционных систем; в

FreeBSD смотрите порт net/rsync или используйте пакет.

Чешская республика

rsync://ftp.cz.FreeBSD.org/

Доступные коллекции:

- ftp: Частичное зеркало FreeBSD FTP сервера.
- FreeBSD: Полное зеркало FreeBSD FTP сервера.

Нидерланды

rsync://ftp.nl.FreeBSD.org/

Доступные коллекции:

- FreeBSD: Полное зеркало FreeBSD FTP сервера.

Россия

rsync://ftp.mtu.ru/

Доступные коллекции:

- FreeBSD: Полное зеркало FreeBSD FTP сервера.
- FreeBSD-gnats: База данных системы отслеживания ошибок GNATS.
- FreeBSD-Archive: Зеркало FreeBSD архивного FTP сервера.

Швеция

rsync://ftp4.se.freebsd.org/

Доступные коллекции:

- FreeBSD: Полное зеркало FreeBSD FTP сервера.

Тайвань

rsync://ftp.tw.FreeBSD.org/

rsync://ftp2.tw.FreeBSD.org/

rsync://ftp6.tw.FreeBSD.org/

Доступные коллекции:

- FreeBSD: Полное зеркало FreeBSD FTP сервера.

Великобритания

rsync://rsync.mirrorservice.org/

Доступные коллекции:

- ftp.freebsd.org: Полное зеркало FreeBSD FTP сервера.

Соединенные Штаты Америки

rsync://ftp-master.FreeBSD.org/

Этот сервер может использоваться только основными зеркалами FreeBSD.

Доступные коллекции:

- FreeBSD: Основной архив FreeBSD FTP сервера.
- acl: Основной ACL список FreeBSD.

  rsync://ftp13.FreeBSD.org/

  Доступные коллекции:

- FreeBSD: Полное зеркало FreeBSD FTP сервера.

Приложение B: Библиография

Так как страницы Справочника FreeBSD предоставляют лишь описание отдельных частей операционной системы FreeBSD, они не очень удобны для иллюстрации объединения этих частей вместе для того, чтобы настроить ОС и сделать ее работу более гладкой. Для этого незаменимы хорошая книга по системному администрированию UNIX® и хорошее руководство пользователя.

B.1. Книги и журналы, специализирующиеся на FreeBSD

Международные книги и журналы:

- Using FreeBSD (на китайском).
- FreeBSD Unleashed (перевод на китайский), опубликовано China Machine Press. ISBN 7-111-10201-0.
- FreeBSD From Scratch First Edition (на китайском), опубликовано China Machine Press. ISBN 7-111-07482-3.
- FreeBSD From Scratch Second Edition (на китайском), опубликовано China Machine Press. ISBN 7-111-10286-X.
- FreeBSD Handbook (на китайском), опубликовано Posts & Telecom Press. ISBN 7-115-10541-3.
- FreeBSD 3.x Internet (на китайском), опубликовано Tsinghua University Press. ISBN 7-900625-66-6.
- FreeBSD & Windows (на китайском), ISBN 7-113-03845-X
- FreeBSD Internet Services HOWTO (на китайском), ISBN 7-113-03423-3
- FreeBSD for PC 98’ers (на японском), выпущено SHUWA System Co, LTD. ISBN 4-87966-468-5 C3055 P2900E.
- FreeBSD (на японском), выпущено CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- Complete Introduction to FreeBSD (на японском), выпущено Shoeisha Co., Ltd. ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (на японском), выпущено ASCII. ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (японский перевод), выпущено ASCII. ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (на немецком), выпущено Computer und Literatur Verlag/Vertrieb Hanser, 1998. ISBN 3-932311-31-0.
- FreeBSD 4 - Installieren, Konfigurieren, Administrieren (на немецком), выпущено Computer und Literatur Verlag, 2001. ISBN 3-932311-88-4.
- FreeBSD 5 - Installieren, Konfigurieren, Administrieren (на немецком), выпущено Computer und Literatur Verlag, 2003. ISBN 3-936546-06-1.
- FreeBSD de Luxe (на немецком), выпущено Verlag Modere Industrie, 2003. ISBN 3-8266-1343-

0.

- FreeBSD Install and Utilization Manual (на японском), выпущено Mainichi Communications Inc..
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo *Создание Интернет Сервера с использованием FreeBSD* (на Индонезийском языке), выпущено Elex Media Komputindo.

Книги и журналы на английском языке:

- Absolute BSD: The Ultimate Guide to FreeBSD, выпущено No Starch Press, 2002. ISBN: 1886411743
- The Complete FreeBSD, выпущено O'Reilly, 2003. ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide, выпущено Addison-Wesley, 2000. ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer, выпущено The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, выпущено Sams, 2002. ISBN: 0672324245
- FreeBSD unleashed, выпущено Sams, 2006. ISBN: 0672328755
- FreeBSD: The Complete Reference, выпущено McGrawHill, 2003. ISBN: 0072224096

B.2. Руководства для пользователей

- Computer Systems Research Group, UC Berkeley. *4.4BSD User's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-075-9
- Computer Systems Research Group, UC Berkeley. *4.4BSD User's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-076-7
- *UNIX in a Nutshell*. O'Reilly & Associates, Inc., 1990. ISBN 093717520X
- Mui, Linda. *What You Need To Know When You Can't Find Your UNIX System Administrator*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-104-6
- Ohio State University написал Ознакомительный Курс UNIX который доступен в Online в HTML и PS форматах.

  Итальянский перевод этого документа доступен как часть FreeBSD Italian Documentation Project.

- Jpman Project, Japan FreeBSD Users Group. FreeBSD User's Reference Manual (Японский перевод). Mainichi Communications Inc., 1998. ISBN4-8399-0088-4 P3800E.
- Эдинбургский Университет составил Online Путеводитель для новичков в UNIX.

B.3. Руководства для администраторов

- Albitz, Paul and Liu, Cricket. *DNS and BIND*, 4th Ed. O'Reilly & Associates, Inc., 2001. ISBN 1-59600-158-4
- Computer Systems Research Group, UC Berkeley. *4.4BSD System Manager's Manual*. O'Reilly &

Associates, Inc., 1994. ISBN 1-56592-080-5

- Costales, Brian, et al. *Sendmail*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-222-0
- Frisch, Æleen. *Essential System Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-127-5
- Hunt, Craig. *TCP/IP Network Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-322-7
- Nemeth, Evi. *UNIX System Administration Handbook*. 2nd Ed. Prentice Hall, 2000. ISBN 0-13-020601-6
- Stern, Hal *Managing NFS and NIS* O'Reilly & Associates, Inc., 1991. ISBN 0-937175-75-7
- Jpman Project, Japan FreeBSD Users Group. FreeBSD System Administrator's Manual (Japanese translation). Mainichi Communications Inc., 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD 2nd Ed. (на французском), Eyrolles, 2004. ISBN 2-212-11463-X

B.4. Руководства для программистов

- Asente, Paul, Converse, Diana, and Swick, Ralph. *X Window System Toolkit*. Digital Press, 1998. ISBN 1-55558-178-1
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language*. 2nd Ed. PTR Prentice Hall, 1988. ISBN 0-13-110362-8
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7
- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9
- Spinellis, Diomidis. Code Reading: The Open Source Perspective. Addison-Wesley, 2003. ISBN 0-201-79940-5
- Spinellis, Diomidis. Code Quality: The Open Source Perspective. Addison-Wesley, 2003. ISBN 0-201-79940-5
- Stevens, W. Richard. *Advanced Programming in the UNIX Environment*. 2nd Ed. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X
- Wells, Bill. "Writing Serial Drivers for UNIX". *Dr. Dobb's Journal*. 19(15), December 1994. pp68-71, 97-99.

B.5. Внутренности операционной системы

- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. "Porting UNIX to the 386". *Dr. Dobb's Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1
- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4

  (глава 2 этой книги доступна онлайн как часть Проекта документирования FreeBSD, и глава 9 доступна здесь.)

- Marshall Kirk McKusick, George V. Neville-Neil *The Design and Implementation of the FreeBSD Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3
- Vahalia, Uresh. *UNIX Internals — The New Frontiers*. Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

B.6. Безопасность

- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4
- Garfinkel, Simson and Gene Spafford. *Practical UNIX & Internet Security*. 2nd Ed. O'Reilly & Associates, Inc., 1996. ISBN 1-56592-148-8
- Garfinkel, Simson. *PGP Pretty Good Privacy* O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

B.7. Оборудование

- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5
- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7

- Intel Corporation publishes documentation on their CPUs, chipsets and standards on their developer web site, usually as PDF files.
- Shanley, Tom. *80486 System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass: Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

B.8. История UNIX®

- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Also known as the Jargon File
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Не печаталась, доступна по адресу online.
- Don Libes, Sandy Ressler *Life with UNIX* - special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *The BSD family tree*. http://www.FreeBSD.org/cgi/cvsweb.cgi/src/shared/misc/bsd-family-tree или /usr/shared/misc/bsd-family-tree на компьютере FreeBSD.
- *The BSD Release Announcements collection*. 1997. http://www.de.FreeBSD.org/de/ftp/releases/
- *Networked Computer Science Technical Reports Library*. http://www.ncstrl.org/
- *Old BSD releases from the Computer Systems Research group (CSRG)*. http://www.mckusick.com/csrg/: В набор на 4CD включены все версии BSD от 1BSD до 4.4BSD и 4.4BSD-Lite2 (но к сожалению нет 2.11BSD). На последнем диске находится также последняя версия исходных текстов и файлы SCCS.

B.9. Прочие издания

- *The C/C++ Users Journal*. R&D Publications Inc. ISSN 1075-2838
- *Sys Admin - The Journal for UNIX System Administrators* Miller Freeman, Inc., ISSN 1061-2688
- *freeX - Das Magazin für Linux - BSD - UNIX* (на немецком) Computer- und Literaturverlag GmbH, ISSN 1436-7033

Приложение C: Ресурсы в интернет

Высокая скорость прогресса FreeBSD делает непрактичным использование печатных изданий для информирования о последних разработках. Электронные ресурсы это лучший, а зачастую и единственный способ информирования о новых возможностях. Поскольку FreeBSD основывается на усилиях добровольцев, сообщество пользователей само по себе зачастую выполняет роль "службы технической поддержки", а электронная почта, веб форумы и новости USENET это наиболее эффективный способ обращения к этому сообществу.

Наиболее важная контактная информация сообщества пользователей FreeBSD показана ниже. Если вам известно о других ресурсах, не упомянутых здесь, пожалуйста отправьте информацию о них в Список рассылки Проекта Документации FreeBSD, чтобы мы могли включить в этот документ.

C.1. Списки рассылки

Обращение в списки рассылки является наиболее простым способом задания вопросов или открытия технических дискуссий в сосредоточенной аудитории пользователей и разработчиков FreeBSD. Существует большое разнообразие списков по десяткам разных тем FreeBSD. Обращение в наиболее подходящий список рассылки обеспечит вас более быстрым и более точным ответом.

Все сообщения в приведенные ниже списки рассылки должны быть составлены *только на английском языке*.

Описание каждой рассылки дано в конце этого документа. *Пожалуйста, прочтите описание перед подпиской или отправкой почты в любой из списков*. Большинство наших подписчиков получают многие сотни относящихся к FreeBSD сообщений каждый день, и определяя правила использования рассылок мы стремимся удержать высокое соотношение "сигнал к шуму". При меньшем соотношении списки рассылки перестанут быть эффективной средой общения участников проекта.

Если вы желаете проверить возможность отправки сообщений в списки рассылки FreeBSD, пошлите тестовое сообщение на Тестовый список рассылки FreeBSD. Пожалуйста, не шлите тестовые сообщения в любой другой список рассылки.

Если вы сомневаетесь, в какой именно список рассылки нужно направить вопрос, то прочтите статью Как эффективно использовать список рассылки FreeBSD-questions.

Перед тем, как направлять сообщение в любой список рассылки, пожалуйста, выясните, как лучше всего их использовать, например, как избежать частого повторения одних и тех же дискуссий, прочитав документ Часто задаваемые вопросы о списках рассылки (FAQ).

Архивы поддерживаются для всех списков рассылки и поиск по ним организован на WWW сервере FreeBSD. Поиск в архиве по ключевым словам дает отличный способ получения

ответов на часто задаваемые вопросы и должен быть выполнен перед отправкой вопроса. Следует также отметить то, что отправленные в списки рассылок сообщения сохраняются в архивах навсегда. Если есть необходимость сохранить приватность, то задействуйте любой доступный вам второстепенный адрес электронной почты, предоставляя в сообщении лишь публичную информацию.

C.1.1. Списки рассылки

Общие списки: Ниже представлены общие списки рассылки, к которым каждый может (и приглашается) присоединиться:

Список	Назначение
freebsd-advocacy	В защиту FreeBSD
Список рассылки анонсов FreeBSD	Важные события и вехи проекта
freebsd-arch	Обсуждения архитектуры и дизайна системы
freebsd-bugbusters	Обсуждения, относящиеся к поддержке базы данных сообщений о проблемах FreeBSD и соответствующим инструментам
freebsd-bugs	Сообщения о проблемах
freebsd-chat	Не-технические темы, относящиеся к сообществу FreeBSD
freebsd-chromium	Вопросы, относящиеся к Chromium во FreeBSD
Список рассылки, посвящённый обсуждению FreeBSD-CURRENT	Обсуждения, относящиеся к использованию FreeBSD-CURRENT
freebsd-isp	Вопросы использования FreeBSD провайдерами
freebsd-jobs	Вакансии и резюме, относящиеся к FreeBSD, с полной и частичной занятостью
freebsd-policy	Публикация правил FreeBSD Core team. Только для чтения, малое количество сообщений
freebsd-questions	Вопросы пользователей и техническая поддержка
Список рассылки FreeBSD, посвящённый срочным сообщениям, связанным с безопасностью	Уведомления безопасности
Список рассылки, посвящённый обсуждению FreeBSD-STABLE;	Обсуждения, относящиеся к использованию FreeBSD-STABLE
Тестовый список рассылки FreeBSD	Рассылка для отправки тестовых сообщений (вместо обычных списков рассылки

Технические списки: Следующие списки предназначены для технических обсуждений. Вам необходимо внимательно прочитать описание перед подпиской или отправкой почты в один из этих списков, поскольку они предназначены для использования внутри проекта.

Список	Назначение
Список рассылки FreeBSD ACPI	Разработка ACPI и системы управления энергопотреблением
freebsd-afs	Портирование AFS на FreeBSD
freebsd-aic7xxx	Разработка драйверов для Adaptec® AIC 7xxx
freebsd-amd64	Портирование FreeBSD на системы AMD64
freebsd-apache	Обсуждение портов, относящихся к Apache
freebsd-arm	Портирование FreeBSD на процессоры ARM®
freebsd-atm	Использование ATM сетей с FreeBSD
freebsd-audit	Проект аудита исходных текстов
freebsd-binup	Дизайн и разработка системы бинарных обновлений
freebsd-bluetooth	Использование технологии Bluetooth® во FreeBSD
freebsd-cluster	Использование FreeBSD в кластерах
freebsd-cvsweb	Поддержка CVSweb
freebsd-database	Обсуждение использования и разработки баз данных под FreeBSD
freebsd-doc	Создание относящихся к FreeBSD документов
freebsd-drivers	Создание драйверов устройств для FreeBSD
freebsd-eclipse	Использование в FreeBSD Eclipse IDE, а также приложений и портов для нее.
freebsd-eol	Взаимная поддержка относящегося к FreeBSD программного обеспечения, более не поддерживаемого проектом FreeBSD.
freebsd-embedded	Использование FreeBSD во встроенных системах
freebsd-emulation	Эмуляция других систем, таких как Linux/MS-DOS®/Windows®
freebsd-firewire	Техническое обсуждение FreeBSD FireWire® (iLink, IEEE 1394)
freebsd-fs	Файловые системы
freebsd-gecko	Проблемы со средствами визуализации Gecko

Список	Назначение
freebsd-geom	Относящиеся к GEOM обсуждения и реализации
freebsd-gnome	Портирование GNOME и приложений GNOME
freebsd-hackers	Общее техническое обсуждение
freebsd-hardware	Общее обсуждение оборудования для FreeBSD
freebsd-i18n	Интернационализация FreeBSD
freebsd-ia32	FreeBSD на платформе IA-32 (Intel® x86)
freebsd-ia64	Портирование FreeBSD на будущие системы Intel® IA64
freebsd-ipfw	Технические обсуждения, относящиеся к переработке кода IP брандмауэра
freebsd-isdn	Разработчики ISDN
freebsd-jail	Обсуждение jail(8)
freebsd-java	Разработчики Java™ и те, кто занимается переносом JDK™ на FreeBSD
freebsd-kde	Портирование KDE и приложений KDE
freebsd-lfs	Портирование LFS на FreeBSD
freebsd-libh	Второе поколение программы установки системы и пакетов
freebsd-mips	Портирование FreeBSD на MIPS®
freebsd-mobile	Обсуждение портативных компьютеров
freebsd-mono	Mono и C# приложения на FreeBSD
freebsd-mozilla	Портирование Mozilla на FreeBSD
Список рассылки, посвящённый поддержке средств мультимедиа под FreeBSD	Мультимедиа приложения
freebsd-new-bus	Технические обсуждения архитектуры шины
freebsd-net	Обсуждения, относящиеся к сети и исходному тексту TCP/IP
freebsd-office	Офисные приложения во FreeBSD
freebsd-performance	Вопросы оптимизации производительности для быстрых/работающих под большой нагрузкой серверов
freebsd-perl	Поддержка различных портов, относящихся к Perl

Список	Назначение
freebsd-pf	Обсуждение брандмауэра на базе packet filter
freebsd-platforms	Относится к портам для платформ не-Intel® архитектуры
freebsd-ports	Обсуждения Коллекции Портов
freebsd-ports-bugs	Обсуждения относящихся к портам ошибок/PR
freebsd-ppc	Портирование FreeBSD на PowerPC®
freebsd-proliant	Работа FreeBSD на серверной платформе HP ProLiant
freebsd-python	Вопросы, специфичные для использования Python во FreeBSD
freebsd-rc	Обсуждения, касающиеся системы rc.d и работы над ней
freebsd-realtime	Разработка расширений реального времени для FreeBSD
freebsd-ruby	Обсуждения, касающиеся специфики использования Ruby во FreeBSD
freebsd-scsi	Подсистема SCSI
Список рассылки FreeBSD, посвящённый информационной безопасности	Сообщения безопасности, касающиеся FreeBSD
freebsd-small	Использование FreeBSD во встроенных приложениях (устарел; следует использовать freebsd-embedded)
freebsd-sparc64	Портирование FreeBSD на системы, основанные на SPARC®
freebsd-standards	Соответствие FreeBSD стандартам C99 и POSIX®
freebsd-sysinstall	Разработка sysinstall(8)
freebsd-threads	Потоки в FreeBSD
freebsd-testing	Тестирование производительности и стабильности FreeBSD
freebsd-tilera	Портирование FreeBSD на процессоры Tilera
freebsd-tokenring	Поддержка Token Ring в FreeBSD
freebsd-toolchain	Поддержка встроенных инструментальных средств FreeBSD
freebsd-usb	Обсуждение поддержки USB в FreeBSD

Список	Назначение
freebsd-virtualization	Обсуждение различных техник виртуализации, поддерживаемых FreeBSD
link:Обсуждение инфраструктуры VuXML[freebsd-vuxml]	Обсуждение инфраструктуры VuXML
freebsd-x11	Сопровождение и поддержка X11 в FreeBSD
freebsd-xen	Обсуждение портирования FreeBSD на Xen™ - реализация и использование

Ограниченные списки: Следующие списки рассылки предназначены для более специализированной (и более официальной) аудитории и вероятно не могут заинтересовать широкую публику. Вероятно хорошей идеей будет сначала наладить общение в технических списках рассылки перед присоединением к ограниченным спискам, так вы сможете освоить этику общения.

Список	Назначение
freebsd-hubs	Люди, поддерживающие зеркала (поддержка инфраструктуры)
freebsd-user-groups	Координация групп пользователей
freebsd-vendors	Координация поставщиков перед релизом
freebsd-wip-status	Статус начатых работ (Work-In-Progress) во FreeBSD
freebsd-wireless	Обсуждение разработки стека 802.11, утилит, драйверов устройств
freebsd-www	Ответственные за www.FreeBSD.org

Дайджест рассылки: Все вышеприведенные списки доступны в формате дайджеста. После подписки на рассылку, вы можете изменить параметры дайджеста в разделе настроек учетной записи.

CVS и SVN рассылки: Следующие рассылки предназначены для людей, заинтересованных в просмотре сообщений об изменении в различных областях дерева исходных текстов. Это списки *только для чтения* и вы не должны отправлять туда почту.

Рассылка	Область исходного текста	Описание области исходного текста
cvs-all	/usr/(CVSROOT|doc|ports)	Все изменения в любой области дерева исходных текстов (надмножество других списков рассылки CVS)

Рассылка	Область исходного текста	Описание области исходного текста
cvs-doc	/usr/(doc|www)	Все изменения в дереве исходных текстов документации и Web-сервера
cvs-ports	/usr/ports	Все изменения в дереве портов
cvs-projects	/usr/projects	Все изменения в дереве проектов
cvs-src	/usr/src	Все изменения в дереве исходных текстов системы (создаваемых во время импорта из SVN в CVS)
Список рассылки сообщений об изменениях в репозитории SVN для всего дерева исходных текстов (за исключением <quote>user</quote> и <quote>projects</quote>)	/usr/src	Все изменения в репозитории Subversion (за исключением user и projects)
Список рассылки сообщений об изменениях в репозитории SVN для ветки head/-current дерева исходных текстов	/usr/src	Все изменения в ветке "head" репозитория Subversion (ветка FreeBSD-CURRENT)
svn-src-projects	/usr/projects	Все изменения в части projects дерева исходных текстов репозитория Subversion
svn-src-release	/usr/src	Все изменения в части release дерева исходных текстов репозитория Subversion
svn-src-releng	/usr/src	Все изменения в части releng дерева исходных текстов репозитория Subversion
svn-src-stable	/usr/src	Все изменения во всех ветках stable дерева исходных текстов репозитория Subversion
svn-src-stable-6	/usr/src	Все изменения в ветке stable/6 дерева исходных текстов репозитория Subversion

Рассылка	Область исходного текста	Описание области исходного текста
svn-src-stable-7	/usr/src	Все изменения в ветке stable/7 дерева исходных текстов репозитория Subversion
svn-src-stable-8	/usr/src	Все изменения в ветке stable/8 дерева исходных текстов репозитория Subversion
svn-src-stable-other	/usr/src	Все изменения в предыдущих ветках stable дерева исходных текстов репозитория Subversion
svn-src-svnadmin	/usr/src	Все изменения в административных скриптах, хуках (hook) и других конфигурационных параметрах дерева исходных текстов репозитория Subversion
svn-src-user	/usr/src	Все изменения в экспериментальной части user дерева исходных текстов репозитория Subversion
svn-src-vendor	/usr/src	Все изменения в части дерева исходных текстов репозитория Subversion, выделенной для работы поставщиков (vendor)

C.1.2. Как подписаться

Для подписки на рассылку, нажмите на название списка рассылки выше или воспользуйтесь ссылкой https://lists.freebsd.org и нажмите на имя рассылки, которой вы заинтересовались. Страница списка рассылки содержит все необходимые инструкции по подписке.

Для отправки сообщения в выбранный список рассылки, отправьте письмо в listname@FreeBSD.org. Это письмо будет разослано участникам рассылки по всему миру.

Для отписки от рассылки, нажмите на ссылку, находящуюся внизу каждого письма, отправляемого через список рассылки. Возможна также отписка путем отправки письма на listname-unsubscribe@FreeBSD.org.

Напоминаем, что обсуждение в технических списках рассылки должно оставаться в рамках

технической темы. Если вас интересует только получение важных анонсов, мы предлагаем подписаться на рассылку с небольшим трафиком Список рассылки анонсов FreeBSD.

C.1.3. Описание рассылок

Все списки рассылки FreeBSD имеют определенные основные правила, которых должен придерживаться каждый использующий их. Несоблюдение этих правил приведет к отправлению двух (2) предупреждений от FreeBSD Postmaster postmaster@FreeBSD.org, после которых, после третьего нарушения, подписчик будет удален из всех списков рассылки FreeBSD и дальнейшие его сообщения будут отфильтровываться. Мы сожалеем, что эти правила и меры вообще необходимы, но современный интернет это довольно суровая среда и многие его механизмы довольно слабы.

Основные правила:

- Тема любого сообщения должна соответствовать назначению списка рассылки, в который это сообщение отправляется. Например, если список рассылки посвящен техническим вопросам, сообщение должно быть техническим. Продолжающееся обсуждение вне темы, или флейм только понижают ценность рассылки для всех ее участников и поэтому не разрешаются. Для обсуждений вне какой-либо определенной темы необходимо использовать Список рассылки, посвящённый неформальным беседам о FreeBSD, специально для этого предназначенный.
- Ни одно сообщение не должно отправляться более чем в 2 рассылки, отправка сообщения в 2 рассылки должна выполняться только при наличии простой и очевидной причины для дублирования сообщения. В большинстве рассылок подписчикам уже приходит много избыточного материала, и за исключением редких случаев (скажем, "-stable & -scsi"), на самом деле нет причины отправлять сообщение более чем в один список рассылки. Если сообщение отправлено вам так, что в поле `Cc` находятся несколько списков рассылки, необходимо урезать поле `Cc` перед отправкой ответа. *Именно вы отвечаете за собственные сообщения, независимо от того, кто был автором исходного письма.*
- Персональные нападки и профанация (в контексте аргументов) не разрешены, это относится и к пользователям, и к разработчикам. Грубые нарушения сетевой этики, такие как цитирование или пересылка личной переписки без специального на то разрешения, осуждаются но специальные меры в этом случае не принимаются. *Однако*, существует несколько специальных случаев, когда такие письма не отвечают назначению списка рассылки и, следовательно, могут повлечь отправку предупреждения (или исключение из списка рассылки).
- Реклама не-FreeBSD продуктов или сервисов строго запрещена и исключение из списка рассылки последует незамедлительно, если станет очевидным, что это спам.

Описания рассылок:

Список рассылки FreeBSD ACPI

Разработка ACPI и системы управления энергопотреблением

freebsd-afs

Andrew File System

Этот список предназначен для обсуждения портирования и использования AFS от CMU/Transarc

Список рассылки анонсов FreeBSD

Важные события / вехи проекта

Этот список рассылки предназначен для тех, кто интересуется только периодическими анонсами значительных событий FreeBSD. Сюда включаются анонсы снэпшотов и других релизов, а также новых возможностей FreeBSD. Рассылка может содержать призыв к добровольцам и т.п. Это строго модерируемый список рассылки с малым объемом трафика.

freebsd-arch

Обсуждение архитектуры и дизайна системы

Эта рассылка предназначена для обсуждения архитектуры FreeBSD. Сообщения в основном строго технические. Примеры подходящих тем:

- Как изменить систему сборки для одновременной сборки нескольких по-разному настроенных систем.
- Что необходимо исправить в VFS для включения слоев Heidemann.
- Как необходимо изменить интерфейс драйверов устройств для использования одних и тех же драйверов на множестве шин и архитектур.
- Как написать сетевой драйвер.

freebsd-audit

Проект аудита исходных текстов

Это список рассылки для проекта аудита исходных текстов FreeBSD. Хотя первоначально он предназначался для изменений, связанных с безопасностью, его назначение было расширено для пересмотра всех изменений кода.

В эту рассылку отправляется большой объем исправлений, и она вероятно не представляет интереса для обычного пользователя FreeBSD. Обсуждения безопасности, не относящиеся к определенному изменению в коде, ведутся в freebsd-security. Разработчикам предлагается отправлять изменения в этот список рассылки для просмотра, особенно если эти изменения затрагивают части кода, ошибки в которых могут повлечь нарушение целостности системы.

freebsd-binup

Проект бинарного обновления FreeBSD

Этот список предназначен для обсуждений системы бинарного обновления системы, или binup. В этой рассылке обсуждаются вопросы дизайна, детали реализации, исправления, сообщения об ошибках, сообщения о статусе, запросы на расширение

функциональности, протоколы коммитов, и все, что относится к binup.

freebsd-bluetooth

Bluetooth® во FreeBSD

Это форум, где собираются пользователи Bluetooth® во FreeBSD. Обсуждения касаются вопросов архитектуры, деталей реализации, патчей, сообщений об ошибках, состояния работы, запросов на добавление функций и всего, что относится к Bluetooth®.

freebsd-bugbusters

Координация усилий по обработке сообщений о проблемах

Назначение этой рассылки в координации и предоставлении места для обсуждения для лиц, обслуживающих базу данных сообщений о проблемах (bugmeister, bugbusters) и для всех сторон, интересующихся базой данных PR. Эта рассылка не предназначена для обсуждения отдельных проблем, исправлений или PR.

freebsd-bugs

Сообщения об ошибках

Этот список рассылки предназначен для отправки сообщений об ошибках в FreeBSD. Когда это возможно, сообщения должны отправляться с использованием send-pr(1) или через WEB интерфейс к `send-pr`.

freebsd-chat

Не-технические темы, относящиеся к сообществу FreeBSD

В эту рассылку входят все темы, не подходящие для других рассылок, с не-технической, социальной информацией. Она включает обсуждения на темы: кто пьет слишком много кофе, где варят лучшее пиво, кто варит пиво в своем подвале, и так далее. Нерегулярные анонсы важных событий (такие как будущие встречи, свадьбы, дни рождения, новая работа и т.д.) могут быть опубликованы в технических рассылках, но ответы должны отправляться в -chat.

freebsd-chromium

Вопросы, относящиеся к Chromium во FreeBSD

Список рассылки, предназначенный для обсуждения поддержки Chromium во FreeBSD. Это технический список рассылки, в котором обговаривается разработка и установка Chromium.

Core Team

Команда FreeBSD core

Это внутренний список рассылки, используемый членами core. Сообщения в эту рассылку могут быть отправлены по серьезной, имеющей отношение к FreeBSD причине, которая требует рассмотрения на самом высоком уровне.

Список рассылки, посвящённый обсуждению FreeBSD-CURRENT

Обсуждения, касающиеся использования FreeBSD-CURRENT

Это список рассылки для пользователей FreeBSD-CURRENT. Он включает предупреждения о новых возможностях, вносимых в -CURRENT, влияющих на пользователей, и инструкции относительно действий, которые должны быть предприняты для поддержки -CURRENT. Всякий, работающий с "CURRENT", должен подписаться на эту рассылку. Это технический список рассылки, все сообщения должны быть строго техническими.

freebsd-cvsweb

FreeBSD CVSweb Project

Технические обсуждения использования, разработки и поддержки FreeBSD-CVSweb.

freebsd-doc

Проект документирования

Этот список рассылки предназначен для обсуждения вопросов и проектов, относящихся к созданию документации для FreeBSD. Члены этой рассылки все вместе обозначаются как "The FreeBSD Documentation Project". Это открытая рассылка; присоединяйтесь и участвуйте!

freebsd-drivers

Создание драйверов устройств для FreeBSD

Этот список рассылки предназначен для технических дискуссий, относящихся к написанию драйверов устройств для FreeBSD. Это наилучшее место для того, чтобы задать вопросы по форматам и протоколам общенияі (API) драйверов устройств с ядром FreeBSD.

freebsd-eclipse

Список рассылки для пользователей системы Eclipse IDE под FreeBSD, а также ее приложений и портов.

Этот список рассылки призван оказать помощь тем, кто выбирает, устанавливает, использует, разрабатывает и поддерживает работу Eclipse IDE под FreeBSD, а также портирует их под FreeBSD.

Кроме того, в данном списке для общего блага пересекаются и обмениваются информацией сообщества Eclipse и FreeBSD.

Хотя данный список предназначен главным образом для тех, кто использует Eclipse, в нем также можно обсуждать средства разработки приложений для FreeBSD при помощи среды Eclipse.

freebsd-embedded

Использование FreeBSD во встроенных системах

Этот список рассылки, предназначенный для технических обсуждений, рассматривает работу FreeBSD в особо стесненных условиях, в частности, во встроенных (embedded)

системах. В данном случае встроенными считаются вычислительные устройства, отличные от настольных компьютеров и выполняющие, как правило, специализированные задачи. Примером могут служить смартфоны, сетевые устройства, такие как маршрутизаторы, коммутаторы и офисные АТС, телеметрические системы, КПК, кассовые терминалы и т.п.

freebsd-emulation

_Эмуляция других систем, таких как Linux/MS-DOS®/Windows® _

Этот список рассылки предназначен для обсуждения вопросов запуска и эксплуатации под FreeBSD программ, предназначенных для работы под другими операционными системами.

freebsd-eol

Взаимная поддержка относящегося к FreeBSD программного обеспечения, более не поддерживаемого проектом FreeBSD.

Этот список рассылки предназначен для интересующихся предоставлением или использованием взаимной поддержки относящегося к FreeBSD программного обеспечения, для которого проект FreeBSD более не предоставляет официальной поддержки (например, в виде сообщений безопасности или патчей).

freebsd-firewire

FireWire® (iLink, IEEE 1394)

Это список рассылки, предназначенный для обсуждения дизайна и реализации подсистемы FireWire® (также известной как IEEE 1394 или iLink) в FreeBSD. Соответствующие темы относятся к стандартам, устройствам шины и их протоколам, наборам плат/карт/чипов адаптера, а также архитектуре и реализации кода для их правильной поддержки.

freebsd-fs

Файловые системы

Обсуждения, относящиеся к файловым системам FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-gecko

Средства визуализации Gecko

Дискуссия о приложениях Gecko, используемых на FreeBSD.

Обсуждение сосредоточено вокруг портированных приложений Gecko, их установки, разработки и поддержки во FreeBSD.

freebsd-geom

GEOM

Обсуждения, относящиеся к GEOM и связанным с GEOM реализациям. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-gnome

GNOME

Обсуждения, относящиеся к графической среде GNOME для системы FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-ipfw

IP брандмауэр

Это форум для технических обсуждений, относящихся к перепроектированию кода межсетевого экрана IP во FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-ia64

Портирование FreeBSD на IA64

Это технический список рассылки для тех, кто активно работает над портированием FreeBSD на платформу IA-64 от Intel®, предназначенный для поднятия вопросов или обсуждения альтернативных решений. Те, кто интересуется обсуждаемыми проблемами, также приглашаются к участию в рассылке.

freebsd-isdn

ISDN соединения

Это список рассылки для обсуждения разработки поддержки ISDN для FreeBSD.

freebsd-java

Разработка Java™

Этот список рассылки предназначен для обсуждения ключевых приложений Java™ для FreeBSD, а также портирования и поддержки JDK™.

freebsd-jobs

Предложение и поиск работы

Это форум для публикации вакансий и резюме, относящихся к FreeBSD. Например, если вы ищете работу, относящеюся к FreeBSD, или у вас есть работа, связанная с FreeBSD, вы можете разместить соответствующую информацию именно здесь. Эта рассылка *не* предназначена для обсуждения общих вопросов о приеме на работу, поскольку форумы на соответствующие темы уже существуют на других сайтах.

Имейте ввиду, что эта рассылка, как и другие рассылки `FreeBSD.org`, распространяется по всему миру. Поэтому вам необходимо чётко указать свое местоположение и область, с которой возможны телекоммуникации или помощь в перемещении.

Письма должны быть составлены только в открытых форматах - предпочтителен чистый текст, но Portable Document Format (PDF), HTML, и некоторые другие форматы могут быть прочитаны многими. Закрытые форматы, такие как Microsoft® Word (.doc) будут отброшены сервером почтовой рассылки.

freebsd-kde

KDE

Обсуждения, относящиеся к KDE в системах FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-hackers

Технические обсуждения

Это форум для технических обсуждений, относящихся к FreeBSD. Это в основном технический список рассылки. Он предназначен для тех, кто активно работает над FreeBSD, и служит для поднятия вопросов или обсуждения альтернативных решений. Те, кто интересуется обсуждаемыми вопросами, также приглашаются к участию в обсуждении. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-hardware

Общее обсуждение оборудования FreeBSD

Общее обсуждение типов оборудования, на котором работает FreeBSD, различных проблем и предложений относительно того, какое оборудование можно покупать а какое нет.

freebsd-hubs

Сайты зеркал

Анонсы и обсуждения для поддерживающих зеркала FreeBSD.

freebsd-isp

Вопросы использования FreeBSD провайдерами

Этот список рассылки предназначен для обсуждения тем, имеющих значение для провайдеров, использующих FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-mono

Mono и C# приложения на FreeBSD

Этот список рассылки посвящен обсуждениям разработки инфраструктуры Mono на FreeBSD. Это технический список рассылки, который предназначен для людей, активно работающих над портированием Mono или C# приложений на FreeBSD, для освещения проблем или обсуждения альтернативных решений. Также к дискуссии приглашаются все заинтересованные данной темой.

freebsd-office

Офисные приложения во FreeBSD

Тема этого списка рассылки - офисные приложения, установка офисных приложений, их разработка и поддержка во FreeBSD.

freebsd-performance

Обсуждения оптимизации или повышения скорости FreeBSD

Этот список рассылки существует как место для обсуждения тем, имеющих отношение к производительности FreeBSD, хакерами, администраторами, и/или заинтересованными сторонами. Приемлемые темы включают обсуждения установок FreeBSD, которые находятся под высокой нагрузкой и сталкиваются с проблемами производительности, или преодоление ограничений FreeBSD. Заинтересованным сторонам, собирающимся работать над улучшением производительности FreeBSD, настоятельно рекомендуется подписаться на эту рассылку. Это техническая рассылка, идеально подходящая для пользователей, хакеров или администраторов, заинтересованных в скорости, стабильности и расширяемости FreeBSD. Это не рассылка вопросов-и-ответов, заменяющая чтение документации, а место, где можно внести свой вклад или получить информацию по еще незатронутой теме, связанной с производительностью.

freebsd-pf

Обсуждение брандмауэра на базе packet filter

Обсуждения, касающиеся работы пакетного фильтра pf под FreeBSD. Допускаются как вопросы пользователей, так и технические дискуссии. Помимо этого, в данном списке уместно обсуждать инфраструктуру ALTQ QoS.

freebsd-platforms

Портирование на не-Intel® платформы

Кросс-платформенные вопросы FreeBSD, общее обсуждение и предложения для не-Intel® портов FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-policy

Правила core team

Это рассылка с малым количеством сообщений, только для чтения, предназначенная для публикации решений FreeBSD Core Team.

freebsd-ports

Обсуждения "ports"

Обсуждения, относящиеся к "коллекции портов" FreeBSD, (/usr/ports), инфраструктуры портов и общих усилий по координации портов. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-ports-bugs

Обсуждение проблем в "ports"

Обсуждения, относящиеся к сообщениям о проблемах для "коллекции портов" FreeBSD (/usr/ports), предлагаемых портов, или изменений к портам. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-proliant

Работа FreeBSD на серверной платформе HP ProLiant

Этот список используется для обсуждения технических аспектов использования FreeBSD на серверах HP ProLiant, в том числе для обсуждения специфичных для ProLiant драйверов, управляющего ПО, систем конфигурации и обновлений BIOS. В частности, это основное место для обсуждения модулей hpasmd, hpasmcli и hpacucli.

freebsd-python

Python во FreeBSD

Этот список рассылки посвящён обсуждениям, связанным с улучшением поддержки Python во FreeBSD. Это технический список рассылки. Он предназначен тем, кто работает над портированием во FreeBSD языка Python, модулей сторонних разработчиков для него и Zope. К участию приглашаются также все, кому интересны технические вопросы.

freebsd-questions

Вопросы пользователей

Это список рассылки по вопросам о FreeBSD. Вы не должны отправлять вопросы "как сделать" в технические рассылки, если только не уверены, что ваш вопрос чисто технический.

freebsd-ruby

Обсуждения, касающиеся специфики использования Ruby во FreeBSD

Список рассылки по вопросам поддержки Ruby на FreeBSD. Это технический список рассылки, который предназначен для людей, работающих над портами Ruby, над инфраструктурой и библиотеками от третьих сторон.

Также к обсуждению приглашаются все, кто заинтересован этой технической дискуссией.

freebsd-scsi

Подсистема SCSI

Это список рассылки для тех, кто работает над подсистемой SCSI для FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

Список рассылки FreeBSD, посвящённый информационной безопасности

Вопросы безопасности

Вопросы безопасности FreeBSD (DES, Kerberos, известные проблемы безопасности и исправления, и т.п.). Это технический список рассылки, предназначенный только для технических обсуждений. Обратите внимание, что это не рассылка вопросов-и-ответов, но дополнения в FAQ (И вопрос И ответ) приветствуются.

Список рассылки FreeBSD, посвящённый срочным сообщениям, связанным с безопасностью

Уведомления безопасности

Уведомления о проблемах безопасности FreeBSD и исправления. Эта рассылка не предназначена для обсуждений. Для обсуждения предназначена рассылка FreeBSD-security.

freebsd-small

Использование FreeBSD во встроенных приложениях

В этой рассылке обсуждаются темы, связанные с необычно малыми и встроенными установками FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

Этот список рассылки устарел; следует использовать freebsd-embedded.

Список рассылки, посвящённый обсуждению FreeBSD-STABLE;

Обсуждения, касающиеся использования FreeBSD-STABLE

Этот список рассылки предназначен для пользователей FreeBSD-STABLE. Он включает предупреждения о новых возможностях, добавляемых в -STABLE, и влияющих на пользователей, и инструкции по действиям, которые необходимы для поддержки системы в состоянии -STABLE. Всякий, использующий "STABLE", должен подписаться на эту рассылку. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-standards

Соответствие C99 и POSIX

Это форум для технических обсуждений, относящихся к соответствию FreeBSD стандартам C99 и POSIX.

freebsd-toolchain

Поддержка встроенных инструментальных средств FreeBSD

Список рассылки для технических дискуссий, относящихся к поддержке инструментальных средств, поставляемых с FreeBSD. Сюда включено состояние Clang и GCC, а также части программного обеспечения, такого как ассемблеры, компоновщики и отладчики.

freebsd-usb

Обсуждение поддержки USB в FreeBSD

Это форум для технических обсуждений, относящихся к поддержке в FreeBSD устройств с интерфейсом USB.

freebsd-user-groups

Список координации групп пользователей

Этот список рассылки предназначен для обсуждения вопросов координаторами каждой

группы пользователей и назначенным членом Core Team. Обсуждения в этой рассылке ограничены темой встреч и координацией проектов, относящихся к группам пользователей.

freebsd-vendors

Поставщики

Обсуждения, относящиеся к координации между FreeBSD Project и поставщиками программного и аппаратного обеспечения для FreeBSD.

freebsd-virtualization

Обсуждение различных техник виртуализации, поддерживаемых FreeBSD

Список рассылки, который предназначен для обсуждений различных техник виртуализации, поддерживаемых FreeBSD. С одной стороны фокус сосредоточен на реализации базовой функциональности, также, как и на добавлении новых возможностей. С другой стороны, пользователи получают возможность обратиться за помощью в случае возникновения проблем или обсудить их конкретные варианты использования.

freebsd-wip-status

Статус начатых работ (Work-In-Progress) во FreeBSD

Этот список рассылки может быть использован для анонсов начала и прогресса вашей работы над FreeBSD. Сообщения модерируются. Предполагается, что сообщение посылается ("To:") в тематический список рассылки FreeBSD, а в этот список отправляется копия ("BCC:"). Таким образом ваша работа может обсуждаться в тематическом списке рассылки, так как в этом списке дискуссии не разрешены.

Выборочный обзор сообщений из этого списка рассылки может публиковаться на сайте FreeBSD как часть Status Reports .

Если вам нужны подходящие примеры сообщений, загляните в архивы рассылки или на страницу Status Reports.

freebsd-wireless

Обсуждение разработки стека 802.11, утилит, драйверов устройств

Список рассылки FreeBSD-wireless предназначен для обсуждения разработки стека 802.11 (sys/net80211), утилит и драйверов устройств. Сообщения о проблемах и новых функциональных возможностях также направляются в эту рассылку.

freebsd-xen

Обсуждение портирования FreeBSD на Xen™ - реализация и использование

Список рассылки, предназначенный для дискуссий на тему портирования FreeBSD на Xen™. Ожидаемый объем сообщений довольно невелик, поэтому эта рассылка объединяет как технические обсуждения реализации и деталей дизайна, так и административные вопросы развертывания.

C.1.4. Фильтрация списков рассылки

Списки рассылки FreeBSD фильтруются различными способами для предотвращения распространения спама, вирусов, и другой нежелательной почты. Действия по фильтрации, описанные в этом разделе, не включают всех используемых для фильтрации списков рассылки проекта действий.

Только определенные типы вложений разрешены в списках рассылки. Все вложения с типами MIME содержимого, не входящие в список ниже, будут вырезаться перед тем, как письмо будет отправлено в список рассылки.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch

Некоторые из списков рассылки могут пропускать вложения других типов MIME, но список выше применим к большинству рассылок.

Если письмо содержит как HTML, так и только текстовую версию, версия HTML будет удалена. Если письмо содержит только HTML версию, она будет конвертирована в простой текст.

C.2. Новостные группы Usenet

В дополнение к двум относящимся к FreeBSD группам новостей, существуют множество других, где обсуждается FreeBSD или куда помещается другая информация, относящаяся к пользователям FreeBSD. Архивы с поиском по ключевому слову доступны для некоторых из этих новостных групп благодаря Warren Toomey wkt@cs.adfa.edu.au.

C.2.1. Относящиеся к BSD новостные группы

- comp.unix.bsd.freebsd.announce
- comp.unix.bsd.freebsd.misc
- de.comp.os.unix.bsd (German)

- fr.comp.os.bsd (French)
- it.comp.os.freebsd (Italian)
- tw.bbs.comp.386bsd (Traditional Chinese)

C.2.2. Другие интересные UNIX® новостные группы

- comp.unix
- comp.unix.questions
- comp.unix.admin
- comp.unix.programmer
- comp.unix.shell
- comp.unix.user-friendly
- comp.security.unix
- comp.sources.unix
- comp.unix.advocacy
- comp.unix.misc
- comp.bugs.4bsd
- comp.bugs.4bsd.ucb-fixes
- comp.unix.bsd

C.2.3. X Window System

- comp.windows.x.i386unix
- comp.windows.x
- comp.windows.x.apps
- comp.windows.x.announce
- comp.windows.x.intrinsics
- comp.windows.x.motif
- comp.windows.x.pex
- comp.emulators.ms-windows.wine

C.3. Серверы World Wide Web

Central Servers, Armenia, Australia, Austria, Czech Republic, Denmark, Finland, France, Germany, Hong Kong, Ireland, Japan, Latvia, Lithuania, Netherlands, Norway, Russia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, United Kingdom, United States of America.

(as of UTC)

Central Servers

- https://www.FreeBSD.org/

Armenia

- http://www.at.FreeBSD.org/ (IPv6)

Australia

- http://www.au.FreeBSD.org/
- http://www2.au.FreeBSD.org/

Austria

- http://www.at.FreeBSD.org/ (IPv6)

Czech Republic

- http://www.cz.FreeBSD.org/ (IPv6)

Denmark

- http://www.dk.FreeBSD.org/ (IPv6)

Finland

- http://www.fi.FreeBSD.org/

France

- http://www1.fr.FreeBSD.org/

Germany

- http://www.de.FreeBSD.org/

Hong Kong

- http://www.hk.FreeBSD.org/

Ireland

- http://www.ie.FreeBSD.org/

Japan

- http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)

Latvia

- http://www.lv.FreeBSD.org/

Lithuania

- http://www.lt.FreeBSD.org/

Netherlands

- http://www.nl.FreeBSD.org/

Norway

- http://www.no.FreeBSD.org/

Russia

- http://www.ru.FreeBSD.org/ (IPv6)

Slovenia

- http://www.si.FreeBSD.org/

South Africa

- http://www.za.FreeBSD.org/

Spain

- http://www.es.FreeBSD.org/
- http://www2.es.FreeBSD.org/

Sweden

- http://www.se.FreeBSD.org/

Switzerland

- http://www.ch.FreeBSD.org/ (IPv6)
- http://www2.ch.FreeBSD.org/ (IPv6)

Taiwan

- http://www.tw.FreeBSD.org/
- http://www2.tw.FreeBSD.org/
- http://www4.tw.FreeBSD.org/
- http://www5.tw.FreeBSD.org/ (IPv6)

United Kingdom

- http://www1.uk.FreeBSD.org
- http://www3.uk.FreeBSD.org/

United States of America

- http://www5.us.FreeBSD.org/ (IPv6)

C.3.1. Форумы, блоги и социальные сети

- Страница форумов FreeBSD предоставляет веб форум для технических дискуссий и вопросов о FreeBSD.
- Страница Planet FreeBSD содержит коллекцию ссылок на десятки блогов, ведомых разработчиками FreeBSD. Многие разработчики используют свои блоги для размещения коротких сообщений о своих текущих занятостях, о новых патчах и о прогрессе в своих работах.
- Страница Конференций BSD на YouTube предоставляет коллекцию видеозаписей с Конференций BSD со всего мира. Это прекрасная возможность ознакомиться с презентациями от ключевых разработчиков, освещающих новые работы во FreeBSD.

C.3.2. Официальные зеркала

C.4. Адреса Email

Следующие группы пользователей предоставляют для своих участников почтовые адреса. Приведенные в списке администраторы оставляют за собой право удалить адреса при любом злоупотреблении.

Домен	Возможности	Группа пользователей	Администратор
ukug.uk.FreeBSD.org	Только пересылка	ukfreebsd@uk.FreeBSD.org	Lee Johnston lee@uk.FreeBSD.org

Приложение D: PGP ключи

В случае, если вам нужно проверить подпись или послать зашифрованное электронное письмо одному из офицеров или разработчиков, то для вашего удобства здесь представлено некоторое количество ключей. Полный список ключей пользователей `FreeBSD.org` доступен для скачивания с pgpkeyring.txt.

D.1. Офицеры

D.1.1. Группа Офицеров Безопасности `<security-officer@FreeBSD.org>`

```
pub   rsa4096/D9AD2A18057474CB 2022-12-11 [C] [expires: 2026-01-24]
      Key fingerprint = 0BE3 3275 D74C 953C 79F8  1107 D9AD 2A18 0574 74CB
uid                           FreeBSD Security Officer <security-officer@freebsd.org>
sub   rsa4096/6E58DE901F001AEF 2022-12-11 [S] [expires: 2026-01-15]
sub   rsa4096/46DB26D62F6039B7 2022-12-11 [E] [expires: 2026-01-15]
```


```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGOVdeUBEADHF5VGg1iPbACB+7lomX6aDytUf0k2k2Yc/Kp6lfYv7JKU+1nr
TcNF7Gt1YkajPSeWRKNZw/X94g4w5TEOHbJ6QQWx9g+N7RjEq75actQ/r2N5zY4S
ujfFTepbvgR55mLTxlxGKFBmNrfNbpHRyh4GwFRgPlxf5Jy9SB+0m54yFS4QlSd0
pIzO0CLkjHUFy/8S93oSK2zUkgok5gLWruBXom+8VC3OtBElkWswPkE1pKZvMQCv
VyM+7BS+MCFXSdZczDZZoEzpQJGhUYFsdg0KqlLv6z1rP+HsgUYKTkRpcrumDQV0
MMuCE4ECU6nFDDTnbR8Wn3LF5oTt0GtwS0nWf+nZ1SFTDURcSPR4Lp/PKjuDAkOS
P8BaruCNx1ItHSwcnXw0gS4+h8FjtWNZpsawtzjjgApcl+m9KP6dkBcbN+i1DHm6
NG6YQVtVWyN8aOKmoC/FEm1CWh1bv+ri9XOkF2EqT/ktbjbT1hFoFGBkS9/35y1G
3KKyWtwKcyF4OXcArl6sQwGgiYnZEG3sUMaGrwQovRtMf7le3cAYsMkXyiAnEufa
deuabYLD8qp9L/eNo+9aZmhJqQg4EQb+ePH7bGPNDZ+M5oGUwReX857FoWaPhs4L
dAKQ1YwASxdKKh8wnaamjIeZSGP5TCjurH7pADAIaB3/D+ZNl2a7od+C1wARAQAB
tDdGcmVlQlNEIFNlY3VyaXR5IE9mZmljZXIgPHNlY3VyaXR5LW9mZmljZXJAZnJl
ZWJzZC5vcmc+iQJSBBMBCgA8AhsBBAsJCAcEFQoJCAUWAgMBAAIeBQIXgBYhBAvj
MnXXTJU8efgRB9mtKhgFdHTLBQJjlXeQBQkF3u+rAAoJENmtKhgFdHTLOVoQALS3
cj7rqYkHiV4zDYrgPEp9O1kAyGI8VdfGAMkDVTqr+wP4v/o7LIUrgwZl5qxesVFB
VknFr0Wp5g9h0iAjasoI5sDd6tH2SmumhBHXFVdftzDQhrugxH6fWRhHs0SaFYCk
Qt5nFbcpUfWgtQ35XTbsL8iENdYpjKXsSFQrJneGSwxIjWYTFn6ps/AI3gwR8+Bn
OffEFdYugJ049O6Vu6YBFJHrnMO7NbF4v95dVYuLtpMIaXWM+V9KITmhaBzFz5fM
Q7UOzcLlbxOYKNIWcp8QQk429mayKW5VUeUExUD1ZzBHn+P6ZG7QTMDu/RmBqiHo
ewCMVz4n9uXT5BiOngE4CvS0WQwHzK+k9MLpG2u/Bo9+LT0Ceh9Ou1rfU5+0tRwl
GyOFFjf3INS7I7gkcAwxQ7dzDItN/UQPZpg8y9mABU2x4enz0AvTnb61d/1dnTEr
tdNgU433he0ZnD1HurZCjBEWC656wv6iMdWcD8gjhMbmEpPmjvXcYlTO6zhEygSM
DiwdQCWK2W4++YJerA6ULBi3niNWBpofOFH8XylV56ruhjtHCo7+/3carcMoPOJv
lVZ1zCKxLro3TRBT15JTFBGqblRyTopFK3PuxW//GTnZOtpQEOV6yL4RAXcWeC1d
1hb5k/YxUmRF6XsDNEH4b08T8ZO8dV3dAV43Wh1oiQEzBBABCAAdFiEEuyjUCzYO
7pNq7RVv5fe8y6O93fgFAmObXVYACgkQ5fe8y6O93fiBlwf/W8y1XXJIx1ZA3n6u
f7aS70rbP9KFPr4U0dixwKE/gbtIQ9ckeNXrDDWz0v0NCz4qS+33IPiJg1WcY3vR
W90e7QgAueCo5TdZPImPbCs42vadpa5byMXS4Pw+xyT+d/yp2oLKYbj3En4bg1GM
```

```
w71DezIjvV+e01UR++u1t9yZ8LOWM5Kumz1zyQLZDZ8qIKt1bBfpa+E0cEqtNQWu
iGhQE3AHI8eWV+jBkg5y2zHRIevbWb1UPsj43lgkFtAGHk9rrM8Rmgr4AXr531iD
srBwauKZ/MElcF3MINuLH+gkPPaFHw/YIpLRLaZXZVsw3Xi1RNXI2n2ea29dvs/C
Lcf1vYkCMwQQAQgAHRYhBPwOh4rlr+eIAo1jVdOXkvSep+XCBQJjm14FAAoJENOX
kvSep+XC0DcP/1ZB7k9p1T+9QbbZZE1PJiHby3815ccH3XKexbNmmakHIn3L6Cet
F891Kqt9ssbhFRMNtyZ/k/8y8Hv5bKxVep5/HMyK+8aqfDFN0WMrqZh0/CiR6DJh
gnAmPNw/hAVHMHaYGII9kCrFfPFJ02FKoc81g9F08odb7TV+UlvRjkErhRxF+dGS
wQoO0RCbf0Z1cs7nd0Vb2z4IJh4XMxBjWc/uQ2Q9dH/0uRzwpAnR4YX+MG5YrX7Z
zBvDyR0r76iQwRSDKgioNgkr6R3rq1NZGdaj+8b0LzdOqtzKJ/eupDe3+H67e/EN
qymtreGjrubpiU9bKvYArisUqhE5KtguryvR6Qz9bj87nPg33DT3WWGVrwFRxBox
dbWzjQFv0wug8m4GAwVF7fPR5/eW7IHw8zvgn0vSPcZz7MZ4e6Y5jN4kA5/xWJYZ
Sps54qQWB+FA30unIXN68KqdIzONIbtaY3W4/JjJUCm4T+wEjKaH+wJX8w1DMjlg
mkTmGh/UrTyC1vXbPgk9Sy3cRTICR1T9z7W8UlmTtnKrUklrjlFR7SXzrEXzLGOX
Fm+NEHpHNXqzcm6c3QfzY/yQ9HSAQ/t7SUQ9caRePbDz3/msyPxtGFor9roQv6VN
wRXCyRgkH4Y5tPhJAQ8G/FxX+VXFb93QL0lfelb23/BBu6cUwW63SRn5iHUEExYI
AB0WIQQeB2Johg/5/ikUnJwDU9SVF1S13AUCZISO3wAKCRADU9SVF1S13NnqAP95
LA1Om9XSAKl76VtV+L3JPDdAwIdbNa0OsRT4Wm7U3wD/YoFrdHXVHHQFKwYeUUhj
XZcxnZLe9Ixo0/JP+RVFVw65Ag0EY5V2yQEQAOqjzPpMUCGu8eElXnAd2PruC6hi
+lc/yC90KqizxIuW6qLQBaAkTCWq7suYpDqoygn7YM3rL50S285WAECAXrcst/cV
Aqr0UH/e6p4iJCUIiXcfjd/wq20RnN/+VuvLhjpCFLY5czfVS31D7Uh9MbC+zUTz
8nVTiNCsAao0qSdfJDIzB4nSO+9xIsme/dLsI5QlU5PdxOBV6HdEhCUXOoratJCb
KAOlLxtPwyMKxmv4oZ7Mqlt10peKjhpBb97qcIzJhHxujQZD0OmzIA6xoQ2eSCGd
xCEDsZ09kr3Esw1AwKnQ51xmWpFWNFk6627M1bo8+hzOz81CrTZhYrgE+1JXv6V1
L2A9lMsimdE1BHNycDS+dBOpIB9qxXCwAab4ykfvNxox/ZPDUrTy7v7mDI5uDNTN
CYYsKCj1UidycOKzSziB9Oa2uvmMJ5XstgNBf7Z8Cky1dtVd4ol6bU9L5nos9tbY
eSXFi4bmcWB7AJiVCMq6N+LBbUKWGLglB4TU1qhttpqv31X9V6ges5gARY/RuRTK
sVyhwsn7SDcqmNKRy0im2AYakwEp7hT07ulahOSLxjP+5hCf+nSJlwbxJ8ozwjjb
zeN2yLlJSI00klkIFBNUDt3wzFRW/n6qlf+/lepgzekfNrYMtfPB8AT07Z2A3U4x
lgiV346dZymbY/EjABEBAAGJBHIEGAEKACYWIQQL4zJ110yVPHn4EQfZrSoYBXR0
ywUCY5V2yQIbAgUJAgIpAAJACRDZrSoYBXR0y8F0IAQZAQoAHRYhBLYVJ36BCH33
XIGDO25Y3pAfABrvBQJjlXbJAAoJEG5Y3pAfABrvuBsQAOlQFPXhx6whO4yw5Ziz
IS02YHhSVMVYKS2T9jPIKi1qxnEiEw9eKH0bW0Oj0TEhZPyM2NJID7DRWK5r8+Ks
Mu8jwm1fUmIrefAx6fCVfCWRECT1MlbL3jhh6AcX/nK2e3Bn8vgExhzczO3JlvD6
wPCc0FkpiY7yDB9ihu1+gbE5Hg6dvftttRXDrbEdAifbNp9KYxDigxdlOb0S14hj
CBysLWH5Su/khcIlkeuqZcI8TmDldnUb2OqTCVpFhaNwsPSrHBzmb0s2sXo4FL03
pLsOdwhi31W6kjk4KvW5FKrOpoEwUMKVNMf50DHdvonUoUHRSIc/cV5NqUWHwvc0
T5031qk0CCRRa/+/iij/p2RG7c1mx7ZECj+jZfmvjSqT+WHJ1BFlNJMWyK4fdVRZ
WyCaoAecdbukwzDwUCUqHJFIWeFtbut7SOPxcwg7sbnKNApAKdi491dvH75s/U/O
wRYO/2P+ymHlqtyix2jq0ReSVYcQPXswQ8i2ifX41F+xTSl4RWCBBeXB1Nxk3+Hs
V4Jnnp1zAJZ0KlKW/oJxbNFdI1TImkpr2p8ioFf+aiePLvDkgeaG8vABgjoihPXW
HVAMR8Z+GvBY/A6OdexpibkTvC/zDr0/Exs4lsylZKDwvvFbctcpHVXBeCBQLX5v
fLrsTkaCLWF/SV9OdMykvYKU7ZAP+gKEwhp+HPFuOHZbOBhqFUdkfeCkdzX/QGdz
Tuz349roRhgz2vRfN7MtbtuzA6NWWhEWt5DcUgX/Y5I3Q4Z2bt3JiXQ6WJMgMMOX
Ar+XxtxyRRykc1HV3DQ/cq8OWYubNnIbgebPNIFr2OIWKsR9yDaucZzpmLfzaMZU
Au5hWmU9fIw5SIKGnQABBnNMhilfD+CkETp6baTvjTK4rpaobjJdeCTrsWgfXRNC
8x3hDvcrjPD70MylOGVQdx8GYChWJnCKXsLTGX7KwdfxkjclTyzWvdcCemp0eLha
mLGb9y1dtWdNIDcVCvZJy0lipHVUdFYYxb4iLZJANL631tlPM6AA8sO1/L4mqEGn
AIHVrUQd+2QkSiOl9mKlpgaR/fJz683BR5Qen9ywX0JPtBupqPW3t9VbO/uNxUql
HCeAhPi9NLOpujpYLfgW5QAfS3u0nkp5nrbkCoQUua2q0Oj7J0mFmtWtcE1c9+TH
mFJVb8j2G9yQw3ADe3Qp9ALazP5nVDVri8NZBhHK1/KuBmRYZtcyfqXUnKoiiWAl
m5rHaRiztW7e3wqm2oJu/RkEAagybutEuBWh2Ej2+gDxjEKKtIKGu54lif4kqTww
```

```
jKTcN1ekGihwwgCMUkBSBeNXk1ClkzLFHwESJCcFwdEgpVYQTKFsu0emYISyco3I
pUajGzfUiQRyBBgBCgAmAhsCFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmWFy28F
CQPyfaYCQMF0IAQZAQoAHRYhBLYVJ36BCH33XIGDO25Y3pAfABrvBQJjlXbJAAoJ
EG5Y3pAfABrvuBsQAOlQFPXhx6whO4yw5ZizIS02YHhSVMVYKS2T9jPIKi1qxnEi
Ew9eKH0bW0Oj0TEhZPyM2NJID7DRWK5r8+KsMu8jwm1fUmIrefAx6fCVfCWRECT1
MlbL3jhh6AcX/nK2e3Bn8vgExhzczO3JlvD6wPCc0FkpiY7yDB9ihu1+gbE5Hg6d
vftttRXDrbEdAifbNp9KYxDigxdlOb0S14hjCBysLWH5Su/khcIlkeuqZcI8TmDl
dnUb2OqTCVpFhaNwsPSrHBzmb0s2sXo4FL03pLsOdwhi31W6kjk4KvW5FKrOpoEw
UMKVNMf50DHdvonUoUHRSIc/cV5NqUWHwvc0T5031qk0CCRRa/+/iij/p2RG7c1m
x7ZECj+jZfmvjSqT+WHJ1BFlNJMWyK4fdVRZWyCaoAecdbukwzDwUCUqHJFIWeFt
but7SOPxcwg7sbnKNApAKdi491dvH75s/U/OwRYO/2P+ymHlqtyix2jq0ReSVYcQ
PXswQ8i2ifX41F+xTSl4RWCBBeXB1Nxk3+HsV4Jnnp1zAJZ0KlKW/oJxbNFdI1TI
mkpr2p8ioFf+aiePLvDkgeaG8vABgjoihPXWHVAMR8Z+GvBY/A6OdexpibkTvC/z
Dr0/Exs4lsylZKDwvvFbctcpHVXBeCBQLX5vfLrsTkaCLWF/SV9OdMykvYKUCRDZ
rSoYBXR0yyOqEACitDvbkbfjaton6izr4T8QU2yvhJHkf4B6KeVDbKY1J4784OxX
p2bJgPeF53SYBe8gm3YHjp8ULh4A/19U4hswyE8ymcm5nIs8OLyBdxkuBZJGEnzx
H3woiyYqWH7991kzhEjUkuMgKLuTI1HiO0oLMuPQNhUHOnWafSVPC0XO/tIL12Om
oUuc7ligY9Z9AceFjTZOuHamixHAAc6hpxdIW+yhC/qTpc2VK0niWeuQfq3453iR
Tf9MnR5Beztl3ZYRWcx7UiFuKGwZwBibNnNmUs6GyQcJ5UTa1oeJcLqHi0Lf/r0j
Xo3wgJq7EZjjVyU+GI2ZVoDOaS6c4/OvLm62XoeSlnn/dQxUcjUki+x8lb69IxSF
1xAgsC/oNtFZYd5rHdlnqIBUYK0lLtSCXBkzVeivSiQa0hL5on8LDu1nw2bXyW61
yt/YxVb4FanMxAqdYVBhOfU0RaPNifH01rbb4TwC9bTZN1LQ1KI/Swb/SruUE0Ry
T28fhYRtsReS2PnUODghJSFDJbwFbBZf6RKI16q1xqKRRvxIWPm+lMOi1NLOKR9P
+OKy9HmChMw0UJUcVl1cJ2xtRl3wi5t6AA6HoNv/TrLeYVgMR9wYmKlpvjTQ5jTd
rbHD1XP5jGsp8QsJMGja1m/7cryReCpcVxvImeReaOdgz+zDmQqq3O5zuIkEcgQY
AQoAJgIbAhYhBAvjMnXXTJU8efgRB9mtKhgFdHTLBQJnhEEJBQkF0/JAAkDBdCAE
GQEKAB0WIQS2FSd+gQh991yBgztuWN6QHwAa7wUCY5V2yQAKCRBuWN6QHwAa77gb
EADpUBT14cesITuMsOWYsyEtNmB4UlTFWCktk/YzyCotasZxIhMPXih9G1tDo9Ex
IWT8jNjSSA+w0Viua/PirDLvI8JtX1JiK3nwMenwlXwlkRAk9TJWy944YegHF/5y
tntwZ/L4BMYc3MztyZbw+sDwnNBZKYmO8gwfYobtfoGxOR4Onb37bbUVw62xHQIn
2zafSmMQ4oMXZTm9EteIYwgcrC1h+Urv5IXCJZHrqmXCPE5g5XZ1G9jqkwlaRYWj
cLD0qxwc5m9LNrF6OBS9N6S7DncIYt9VupI5OCr1uRSqzqaBMFDClTTH+dAx3b6J
1KFB0UiHP3FeTalFh8L3NE+dN9apNAgkUWv/v4oo/6dkRu3NZse2RAo/o2X5r40q
k/lhydQRZTSTFsiuH3VUWVsgmqAHnHW7pMMw8FAlKhyRSFnhbW7re0jj8XMIO7G5
yjQKQCnYuPdXbx++bP1PzsEWDv9j/sph5arcosdo6tEXklWHED17MEPIton1+NRf
sU0peEVggQXlwdTcZN/h7FeCZ56dcwCWdCpSlv6CcWzRXSNUyJpKa9qfIqBX/mon
jy7w5IHmhvLwAYI6IoT11h1QDEfGfhrwWPwOjnXsaYm5E7wv8w69PxMbOJbMpWSg
8L7xW3LXKR1VwXggUC1+b3y67E5Ggi1hf0lfTnTMpL2ClAkQ2a0qGAV0dMsNxRAA
suW1aLh+hgydW+iH6DmdQRMESsB1kE02kO1462TAQaziIAvNoxw5h48xvyEnrDA8
d+9IDMyxdrLmAbndUlSveMa9+EPiGHwr6VTyFL8nA5F7DcFi4mjEyGKe18JcaAlY
UtvHgWH6EjiX2iSXpsrJFEhtfFNolZ5sp9LFI6hOBihSJxZK4sbMR7Q6IkDuAVpT
FLiejBRlsXpFvTGL6040CtxbL5cqkVMYP38rFMTuc3pGGJA4wb5EC1dGjUi6XjbY
H7kuCAFyXqV9eQQP61x7K9W8qnXW+weCIMKfSX7AcCtH1jXBAM6lqpPrh6amc+/r
bg2eNA7DmgJnEY4apIcDB/b4khRMga2ozeGWWyIvOaVvR2R7ALQ+Rgut85cM+4+V
l2PHmOzW/yYdHVb5REQItFR5COb/mGUqYhkCtiV3nXo/K0uOQKu5SBbNzLuNvuwd
n+Eimxjl18VnrGG7sjtUa0MLmtr62GiEVrhrDqa/biHp8LdWkAQjLZ4aTRh2XZig
gaVFZHmkw3ILPyKKM21UXdM0YRk3TGVK8ODQy58ebPS4v9yYT9gUA9UDkDYeGcF2
qjoDPVNvcG6H8jCSsPRl1KZwtqqITCOSAIAPI4Nu97kO6nbOyQpYlwjd1MhvVXnP
66mHSvmqaxbNGX1mF9B/yERkBkooNZrKuJSvBTC2J1q5Ag0EY5V3BwEQAMpFVczZ
o9ZPNsgW791UW5o6wnrnd1nIO+S4rc37q2TEz8KGHCuxo5NwffZ2t6Ln04BI54pb
apg17b7a0hPka37HFkL28n4VyMdx0CsAm3QEfUsdK6xwKV2SucYeVcrV1upcN4Pd
```

```
XD7su1I7/A4CWXFJG047zJ0Z89lJZiQEiAq7ghvEoinC0sm+0a6ao/ocqCgWCKM1
yCPOyzJXleRrv29SRnYziMR+q2U0x9xg9Xl6GMwUmFwbJc9nORVvLH7fbU6/du8E
goAYrglFOFZG/TSolSGWRSMiavz0JSD/i+rEN4aIT4WfBe+L9Wy1AmrNxiAO+zKm
zHQu3JSxDncr+y+hcd+W0gqw10FoI9jWLcL7kR+6a0iOjuJSXSopq2l3DafiPxtC
Fmr4CGQhzBHM6e4/v/NNd3F0XpVbJ6RQph7lkfvfz8q2lvUlHhezJ0p1xXmhff9C
HjdVMhmAmz5+imBAXk2mottNfKb0pFEen1xY3K/UPA4g+oPsSj495MsvIg9eIMCc
C3/z0SEUMWH/styyJzPqfpyfGwZeTcIj9vg2o+RnGvmcLVYA/EGToPk905kv/cK7
3oy8bZyOB0zMg7T9PaWgLUO0sqjqo0Mw3knFySg3oRXlcilPQvfPdX0JvwLpc9DW
lr1+1GkCXJ08lWugJc96CJQupKRb1IbC0oUXABEBAAGJAjwEGAEKACYWIQQL4zJ1
10yVPHn4EQfZrSoYBXR0ywUCY5V3BwIbDAUJAgIpAAAKCRDZrSoYBXR0ywwtD/wI
DmEcHdFlyFRTomUBjbeK2uzcZIhkkgL58lc63UPle5iJ2FBvmYS+0rQS53sVEscc
n5KfkOwTryKllvWbl0IzuiqfawxALcfWpfZJHzTMSnDHfgXvOOyFMQruqRDAHAr7
PNC0CnbT0sEF2ZFzad8M9fLqtkXUx4mgECNGJ4CVqg75KY8uUzv/BmRwEf587FT5
/iAIed5MjFB2VFDX9GABcvTTbHxCZIxnxl3cs15SxT0lAofZ2ueU6kWYWZSXFeaE
M/4ymPJws2mmV0AkbJghLXCn9Mx3nX6NTZZ9Harbru+RzW3/Hg3DZd0J9vko8Paf
P0l1NWtgyX74CqvTgjzTxXTnqrRXzcczK7fhcC2u4i0prPtXXcyyi7SwpoLikaZC
LFFhUmOx+mS5TjtgFyFZBNxnO7iAwkzfcTcC9sPoWaFmiQf6q5EIYzG+WQpncj80
mxl3HWOP6oFj/hZJRYseKeMkvJzLTo87rFdM6CsMrLwETR6e+aWM0btPFil1rXVA
CNOjsy0bxTV8OJEfyxnYmyjvnBvB0kdiaVEDdVhxgSqzLAX4mgXa49/V6M/uzMr+
n3/A1Jdk4V6fVm8S5cFIXxoUat3cB4xGaT9OWD3o1NPr6eS9Vo0EsJlRl81SG68f
S+Qtk2fX27T68YG4Aa3zMfZxUsVuFLtTuQbRC+fJpIkCPAQYAQoAJgIbDBYhBAvj
MnXXTJU8efgRB9mtKhgFdHTLBQJlhcuqBQkD8n1oAAoJENmtKhgFdHTLoO0QAJsT
E9fkleb7YzPEuP9GJ3jx8PGdWm7n+8UNdr24kS6gOXVUfPZrWa5So21hcIwZb4PZ
DqHSVSQnRciKhSnG7gplYPNGZ4+FWbLr/mBRYarjkVFLUuCPexSIjxV1KSGJnWs9
YTVAKZAz75GpCML6jD6biCOQCQ86wqOdWvZIZR8YvurrxR64ABB0rjbsaG8cNOUX
1cwAfdLwthf64dS+2m3lqNGDHkP5eNL0RIxC5gXYEp0lvmlMH3ZuO5WrfH73PTDg
89bxXeuhrFmSEwf4xWm603oi8/2qQvR9/7jb0o+t71NQuWrWIFONZWWgZBUGso+u
yT3XgY4YqKGR3z2QzKHYnJ6M7SvSYpqS7RtcxcCXF0HGNfES8cAgtKVpFtbtSwXX
p8O8oLyjmVIO/NjUpbLOGdFIsarsezLFV9f2fqZ63J34hyUSg8LrYVV1fA5DJUpe
bbX4hLpdk0MMtgG43BwKIGlJTpL5RkQ/uQU3YW2kairy7o+1imDD0TRzQxtdjVOI
5vnlTNcfJZIIfLx4drABA12OvpX3dfPV62R+8BAlJFT430CG6AISJIBqJRFvuikm
nZGUvEHmOUs/FLbbaXTPkKc7tR2WIwljRvMV+Qk84cWcX6YchMslMuiDM1mtlQZi
g34WHGSE+zCWnXAslIHlSwox7qfdO0Kz2XncSbIAiQI8BBgBCgAmAhsMFiEEC+My
dddMlTx5+BEH2a0qGAV0dMsFAmeEQS4FCQXT8gIACgkQ2a0qGAV0dMsm1Q//V09x
OutSbWU44KRurdnGKsk56DFlqXtjGYJqDPrODpX3M8IDf2MuTIN2yfPMv984bAbO
A9RL7EaGVlQUW9QWPURMsZKEFQljhfxRJO9JoGDYI7uRDnSEi6WjVvgUk5OiIh0K
EI4jEcaLCzveIEcswrVDSAn+7nGvewP8Rrx7qMUNvLAltxiMyfGXneavRs3sfusz
db9LTTY8lCU0xrslaXrrvfCkaRbskFi3S31I+1ZB/ewuAhHqfc13eRBjPwQOanJF
epAzP4GF41fVQN9GtssATCD+dV6OFhYjfWJbOIcPv277wCvGIFucM9XRjbkCIYFQ
E5W+1O/act3Obj1sB90C+cVOgCng37YqfObYLF19RE4+a7NUAh/GxHj+8TxUyvvr
aWWyfNqTMDjHMShNDjG0qSzFX7vfyUpmfAfz+6ad78aks9LMf+86iGkvBhXFs7cz
Vv4PWWYV1+WShNU2Y9yMaH3zpWUaREdB07HKbLva4Y1icqWVx+z6xs3PvsqbTei/
moXiZk7ohpbBm0htJlki22ARYrGXSK6w5RQtCZoBW0DEj5JNBjkK6XbAW3VFuAA1
J5wLS2z0eIR5adP+/SxQUbTq+ZFiOGdBP1g/783e7zEdyA0YfA2KU9OdLzupUix/
x+JYcxmrZXndSMObd0IiWOhwXlgarbJJMReOORg=
=cYaK
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.2. Секретарь CORE <core-secretary@FreeBSD.org>

```
pub   rsa4096/4D632518C3546B05 2024-02-17 [SC] [expires: 2028-02-08]
      Key fingerprint = 1A23 6A92 528D 00DD 7965  76FE 4D63 2518 C354 6B05
uid                           FreeBSD Core Team Secretary <core-
secretary@FreeBSD.org>
sub   rsa4096/CABFDE12CA516ED2 2024-02-17 [E] [expires: 2028-02-08]
```


```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXQ1o8BEAC+Rcg8cmVxuP17Vu+q5KgCx/XiulQuqKXAqqBlYCH2jqk6DINP
yFrREGBhzd/qNmlAYEahQ4Zgl0bUZNTrZVDyzicOvPP0jH+KSTQwRs7NOawEdlVO
cyHrwDCPEqf5ZzD4NhfTriEOw+j0pEH/onitUGvoQRtx15xWyaJQxDEBMTYMLewE
86D1bltwnTNczE3UZb7oQLJXkAX5hcLtou70XJGgZITvJkK+kp/xot2eFjnqRz/u
WeXnKhYAmC07EKwZ1uw047eHKwMMRBYqzApLwoQtfE430Kxf2q8de64x8zDbi6YM
1J4r8OAxOtHVyfJ0j7Q23DEZz0VVb4b1Tx5OG2Re/KSNvqI0awJO4TcRmOR88OyY
dzyXgnX6Sa7GVQY1FXvn7vtFuDAt7egZOzeomSHL9bdX07LTQ4UtM88EV9wm3q4q
smoatV9jsvPQ1zxCU3aQD/5eWTJH2/kz1LIuBL/Qi5XQpJn91lBtUWJrCgkHWPGu
f//rnnXmsG7DAChW+yZ7cFO8lfNa8sFhPqSxCYphWmJTrvadyQtDngB8JakWdnmK
pfGS6y5lel+181vw38ZZKt04AKM+nDY8051lBM7Q9Q6kTLI33UZeImndx5xYukVD
kV6aQ31HYfEark15c7iEz+OAcwFnM2ntXMt7kKGd40CqzusiPcQkPqPbAQARAQAB
tDhGcmVlQlNEIENvcmUgVGVhbSBTZWNyZXRhcnkgPGNvcmUtc2VjcmV0YXJ5QEZy
ZWVCU0Qub3JnPokCVwQTAQoAQQIbAwgLCQ0IDAcLAwUVCgkICwUWAwIBAAIeBQIX
gBYhBBojapJSjQDdeWV2/k1jJRjDVGsFBQJnp8PiBQkHeofTAAoJEE1jJRjDVGsF
GMIQALhj+mNpH8OmTFeihQ6t9P8un3llz6Wmqe/Q+ULWeqJvV/uC1J5T9fnoGhwF
MgECuguXYJtoYQ16KXnsSOs1tcqMOK9GtEtFJTGe2DtfLiBednwUvu9j3HmTlwLN
M+7rqkiC0HCg2qSjcMjxbVbA5BSgNkgfyTSO2YdjfaZ+ceiHwo/qa5xWE6i2dMR1
PLGMMHTeELdtdSH6VR9/3h0qt3qzwdMBRCVAQHbim7CqwjUH9jg+IOySXl81jNoB
xuVZR3pKshyY4Oxo1dK+W86Uiff/+c4jCAxokGWIR7C4MkZZWUQqV7920gkZFC/5
qzpT1A1/sFUg8HfFTr0vCoPSVFWn2+Tto3vr78DICVaAf02aYAFlyKK18BMCoHkS
hDDO+/JQZmvHOIgEYK+T2WN1c9gm9IDJzGZuH0X2C/vkREnJKkccJfB4pXuN10wt
fqyP9fn8h6+/t+sv3qNLm4d8fkmLXofulG3WB4i/F+Hip2rjPvvBCF7zl4xy6cJ2
xVY5HUOBTqmIwVhYwUpXaqNoWa/qJBLTuot3z7ciKmKX6Lq+Dze5dzhrPNl/CalC
HBf3miHRK3TZbYLooG3bcEWgxU2BnBi3vl5NpCoUOKkPYRlALXi2TyHmPxO7oT4e
mIzS6BgnXOqO+AXvbKKfayuSePdBqkNMK/SMC4Dylkf6Xj25iQIzBBABCgAdFiEE
EBpxaxYrAOVb7eoFrbv4YQo3ibcFAmbu7x4ACgkQrbv4YQo3ibcr5A//TIcbD/EW
YJz0zrUQdc9xG3UNfU6uHmQzAuUy5ginevyqv0TSso5qvSkOHvdxbi41rfMiB2RJ
V3q0n0PSvHFld89fOTZUMZXaPvozCiBYWScrt+KA/2pq3K8mUumHS+IFtpHLL2Tu
+gI8XCHUFxO9HUThM/rFlIyEdzoctgmqQ7IG4uZG+o2J3w09llhDAUe0vraJK1On
p9yFACnRrhqvl41JeUWcv9MH9JcwHUqtUo9WLTCIb+hkByTOyRfHBYpYw//bdXdf
6rkCwKVWpyMDbk61zq2VsS1kqbP9IH/A8CsBnA6mg+zPq2i7HIFw8Swj4OGJcIvG
a9ubUYJRjDhX/vBpNrtncANZ88FQmA+Maq0vu0LS5IIGyIKkvd1fKIsyvBDDa9kE
nfCW0XMkJA0Gf+kxdB+elXQHBwKO2sr5BiKnJT5lJq3Yu9fxxGBnf93yiN4E9bmF
gG7cZxpyb1Bp76TJhLcANyybOTjCtiNRrgqeaSx9/6hSPfPigGXIne0H2lmJO6oq
jUrsYmFiWvU9sc7AcPVw/eHG8FgW35TuwKX71z8w994iaahUPNcSVyXOUD+QNROv
HhGUXrc81t613rivh22N0NZpNubVatq43KV7+/bnPyWBIi1Awh3vIFsNNSQSrYxF
lUuQaAHQXTeZMZ/7npE4t86seMt0T7BGn765Ag0EZdDWjwEQAL3VwFifpnRCYzQ4
VZ1dAjp8w542iRprqeA+C3tvNbk2OvKpN3DIc49l2bgNZ/VI7/T58lEKrFsgLK4w
AWtv18OV7xuh0AuOblmq5MvcPrUelkPj5HA7M6Ng/rAubuHfwdP/IjIrzzY6+XDh
```

```
fP9N5KIv9VRJYT23BbvLPeit4J4tQEB/NpxL3L6zI75qq4R3T/EkHP6Y9Buup9Lr
isJlcxkSK+CyORCgTpEz6fWsXiTDgS7cTaQ969XCygBpj4wRQzwkBdokxo1wsiFt
4zLt07/PrPYjeHltTDkrF9XDNhLNJ5GliHnd01oHg1j5/n+9Osvh1maoFwuBxXTN
nTZ2P+7RKLBaVQVSkUa5KJbXoM3v7bMbXM5aLs2XjglrAzrZOV+Y7zOu5UYQdpXd
7x8XAEtEozRJzt7dosQIhKx9h4LlltFFulDUpf5VCvcUEoAzMbiX0aju9n5Rwsql
DHatU9Mm3W8OdFXj1foIXZD+VX2Jp9DgxPLoAJ0CWhPXJ8f+WFSJZCjWoPJEJKWY
0EOxiXyAuvAyniAZAC/eKZkfGckXmu7edRgYbRTTwPmZ/axa/k9aHsHLwmbxuZo/
xxQXzqU70EleHZ31A3mOqsC1epjN6dn4AFKgvVesP/K/fbvSsfsfiABS2A/68ne4
zJG73Tnk4L6J79vrXc2iMJbmdg3ZABEBAAGJAjwEGAEKACYCGwwWIQQaI2qSUo0A
3Xlldv5NYyUYw1RrBQUCZ6fEFQUJB3qIBgAKCRBNYyUYw1RrBQd9EACMVckxzy4w
aUGlERguJ+kslS8MkMjNqnfDPLRDVxbUxNvbbhw7/u9blM65DCGCeNLc2n4oiu5C
E3I095AKmvq/0dOa81mEEdZkC1CVc64bXWbEyz5AtSHUpgdxRso6C+YopndSiz1T
WcIagQRXfWaw3FBWPooA87gmibmSmCegCtqx+uyc5QxX2eFI8mRK7vlfnGpYKHs0
D1/yUSGQOwoNRJ5FYm+ynfDE3FzHEQL7lv1vpv1k3xNKfBziMMg4IMEBKNHV4VKN
qJpa8UCodeTGSWQdNnCeCWPsxz5oQjCcVH9Z3e8sMpWLHhRcBZzSwXUOws2GbRMH
xHwrfrpHJcrBhe64pgfGOvZLUJ9BDs+8egTHsqRFacipbtTR+hhVhuJEHDaQQWuM
8IRHj9HIuTAczET8JTDHTMIoo8DZdOtiW/YgqCDwYghkI77d12oNQqYoeJ2HiqbK
cGzwCpsR0A+p/iOAxJG13tsxqZV8TQ8iTokWG6ACtZ7sfeWEhxqMbUKUMgogZn0I
3n1kV+tUZC6BQrIYI7TiKg95wLZsIydeoIsQoNZwvyKAXfVmQ62YjIX8njZwN+07
8/ipUPJxCYa8zL8BZyDmoFJqa3y9z+11+vtiZ9t+aTwGvjpHDwyeCJco7go9cU3m
GRFZYciqIoG4n3tl8Pob15vFlVqk47rRqg==
=8TzT
-----END PGP PUBLIC KEY BLOCK-----
```


D.1.3. Секретарь Группы менеджеров дерева портов <portmgr-secretary@FreeBSD.org>

```
pub   ed25519/E3C401F60D709D59 2023-03-06 [SC] [expires: 2027-03-05]
      Key fingerprint = BED4 A1D3 6555 B681 2E9F  ABDA E3C4 01F6 0D70 9D59
uid                           FreeBSD Ports Management Team Secretary <portmgr-
secretary@FreeBSD.org>
sub   cv25519/2C92B55E27A641C3 2023-03-06 [E] [expires: 2027-03-05]
```


```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mDMEZAXJvxYJKwYBBAHaRw8BAQdASFAC20WL3R1T6uNyGMZbfJCxDkcP4C5vi3Op
tcZ2fbq0R0ZyZWVCU0QgUG9ydHMgTWFuYWdlbWVudCBUZWFtIFNlY3JldGFyeSA8
cG9ydG1nci1zZWNyZXRhcnlARnJlZUJTRC5vcmc+iJYEExYKAD4WIQS+1KHTZVW2
gS6fq9rjxAH2DXCdWQUCZAXJvwIbAwUJB4TOAAULCQgHAwUVCgkICwUWAwIBAAIe
BQIXgAAKCRDjxAH2DXCdWYN1AP43TjyfZtZ3DLYT++g0+SuPsoO/3yWVybA+UmFL
zb8MngEA+LLNUfvEwCuXS/soh+ww5bpfmi3UUmeGiQEAXug3iA+JATMEEAEKAB0W
IQT7N0XIbxXo7ayBMvzYKU7Du8TX1QUCZAXLkwAKCRDYKU7Du8TX1XHMB/9R1MX4
6zMgpKqPPt76GOI+eGEdBK6bY8aJZjQGdqTh9f6VtXVoTGIG7cvhc9X8tDBoB0PT
2KZWheF51AV1+NHU4HwLAQ1BMebrFvWSfkw4xg4fBGwDhz9/GN85No+Js772V5ey
8lRiL6meRVWxMlLyWcxGd8JjcC5yX/iAUQ3SBGCLqW7unWjjg7CTd+AMBwcqPGrv
ax8q6eFVguJcHJAjMnKf6HAy4cpK3s+uMoUBCGnszSN12B3ysKfyC4pNO/pix5tA
Q5v8aRqTeFPh5zmNhWo0KGPzplTPqRQSHDl7GDQC8Ru3MhzFkeWzHsexjZVwS6W2
DPcYpuuAsA0XOZIZiQIzBBABCgAdFiEEEBpxaxYrAOVb7eoFrbv4YQo3ibcFAmQF
```

```
0u0ACgkQrbv4YQo3ibccwg/9F2Xuic3nhKxRbB3mJeDo6SYQETa/Gh1qQ34+8zlt
8UMazOx67gnYQfy+pXjro6eQ2up0a4eUYezcNOudqAQD21nRz3HA6EQVNcE/TzEA
xl5CJntTaLOt7S+EDXFW5BuQIvhhoMGgm8+WNVgA0EJ7tfL0OcYBSvr19fqwChEn
9c14cSk6mgHSsleP5NvskYN053pxHwy0LTSb8YBBv52th37t/CRFC1363rS5q+D7
JixFopd1O5pKpA5ipvE4gGgRjPtwjx0SjjepwK/3fuhEJQQyKzTIKlMfu2Dj/iR2
Li1Sfccau5LQXOj9fUITU3u1YG7yrm8VGzT7ao4d+KRwgMLjd2pLqiGIbbJwGBiP
FRmtilWQoeIlmSlFX4obAA517DOK0pW1mH8+eEn4EJd3SekT3yzFyKTASv0J48Z8
3F928xg+eZvHxVC0t1J+J5IG0gt3EEncuWKIPQGR7PiQbti6R3FQVTz6WfMWOebP
Qi0E9F/Aqakr6Vj2sKGrDq+ebpaF5G8Yw1YrUl2IDiPzkCegp3ZbI0wh11Xvzhi8
LXPQGK4jBQas4G8cegfitzmtdGRHYrbMv0R9I4mvaL+WlOuD2AvyVG28lguqVhnN
AZP+ohdquYyX2CNCVvbKWAtXo6Ur0vWG8BL8m6defAtEkIwVBALaOHQOSI3aNUz4
lwy4OARkBcm/EgorBgEEAZdVAQUBAQdAsefmSfxEOdOr02+K/6noYCuJ1FeAWVz6
jFYQ+9w6jggDAQgHiH4EGBYKACYWIQS+1KHTZVW2gS6fq9rjxAH2DXCdWQUCZAXJ
vwIbDAUJB4TOAAAKCRDjxAH2DXCdWRl4AP9h5ot212BK29S6ZcMBhHvmtF5PG1oD
c7LnZycSRmbFiwEAndCMpAGOhDW8iVgDd0wLQq/ZMPe+xccfG1b3zFH2EgE=
=iiAT
-----END PGP PUBLIC KEY BLOCK-----
```


D.1.4. <doceng-secretary@FreeBSD.org>

```
pub   rsa2048/E1C03580AEB45E58 2019-10-31 [SC] [expires: 2022-10-30]
      Key fingerprint = F24D 7B32 B864 625E 5541  A0E4 E1C0 3580 AEB4 5E58
uid                           FreeBSD Doceng Team Secretary <doceng-
secretary@freebsd.org>
sub   rsa2048/9EA8D713509472FC 2019-10-31 [E] [expires: 2022-10-30]
```


```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBF27FFcBCADeoSsIgyQUY8vREwkTikwFFlNg31MVy5s/Nq1cNK1PRfRMnprS
yfB62KqbYuz16bmQKaA9zHN4FGfiTvR6tl66LVHm1s/5HPiLv8sP14GsruLro9zN
v72dO7a9i68bMw+jarPOnu9dGiDFEI0dACOkdCGEYKEUapQeNpmWRrQ46BeXyFwF
JcNx76bJJUkwk6fWC0W63D762e6lCEX6ndoaPjjLBnFvtx13heNGUc8RukBwe2mA
U5pSGHj47J05bdWiRSwZaXa8PcW+20zTWaP755w7zWe4h60GANY7OsT9nuOqsioJ
QonxTrJuZweKRV8fNQ1EfDws3HZr7/7iXvO3ABEBAAG0PEZyZWVCU0QgRG9jZW5n
IFRlYW0gU2VjcmV0YXJ5IDxkb2Nlbmctc2VjcmV0YXJ5QGZyZWVic2Qub3JnPokB
VAQTAQoAPhYhBPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsDBQkFo5qABQsJ
CAcDBRUKCQgLBRYDAgEAAh4BAheAAAoJEOHANYCutF5YB2IIALw+EPYmOz9qlqIn
oTFmk/5MrcdzC5iLEfxubbF6TopDWsWPiOh5mAuvfEmROSGf6ctvdYe9UtQV3VNY
KeeyskeFrIBOFo2KG/dFqKPAWef6IfhbW3HWDWo5uOBg01jHzQ/pB1n6SMKiXfsM
idL9wN+UQKxF3Y7S/bVrZTV0isRUolO9+8kQeSYT/NMojVM0H2fWrTP/TaNEW4fY
JBDAl5hsktzdl8sdbNqdC0GiX3xb4GvgVzGGQELagsxjfuXk6PfOyn6Wx2d+yRcI
FrKojmhihBp5VGFQkntBIXQkaW0xhW+WBGxwXdaAl0drQlZ3W+edgdOl705x73kf
Uw3Fh2a5AQ0EXbsUVwEIANEPAsltM4vFj2pi5xEuHEcZIrIX/ZJhoaBtZkqvkB+H
4pu3/eQHK5hg0Dw12ugffPMz8mi57iGNI9TXd8ZYMJxAdvEZSDHCKZTX9G+FcxWa
/AzKNiG25uSISzz7rMB/lV1gofCdGtpHFRFTiNxFcoacugTdlYDiscgJZMJSg/hC
GXBdEKXR5WRAgAGandcL8llCToOt1lZEOkd5vJM861w6evgDhAZ2HGhRuG8/NDxG
r4UtlnYGUCFof/Q4oPNbDJzmZXF+8OQyTNcEpVD3leEOWG1Uv5XWS2XKVHcHZZ++
ISo/B5Q6Oi3SJFCVV9f+g09YF+PgfP/mVMBgif2fT20AEQEAAYkBPAQYAQoAJhYh
```

```
BPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsMBQkFo5qAAAoJEOHANYCutF5Y
kecIAMTh2VHQqjXHTszQMsy3NjiTVVITI3z+pzY0u2EYmLytXQ2pZMzLHMcklmub
5po0X4EvL6bZiJcLMI2mSrOs0Gp8P3hyMI40IkqoLMp7VA2LFlPgIJ7K5W4oVwf8
khY6lw7qg2l69APm/MM3xAyiL4p6MU8tpvWg5AncZ6lxyy27rxVflzEtCrKQuG/a
oVaOlMjH3uxvOK6IIxlhvWD0nKs/e2h2HIAZ+ILE6ytS5ZEg2GXuigoQZdEnv71L
xyvE9JANwGZLkDxnS5pgN2ikfkQYlFpJEkrNTQleCOHIIIp8vgJngEaP51xOIbQM
CiG/y3cmKQ/ZfH7BBvlZVtZKQsI=
=MQKT
-----END PGP PUBLIC KEY BLOCK-----
```