Часто задаваемые вопросы по FreeBSD 12.X и 13.X

Аннотация

Этот документ является так называемым FAQ (Frequently Asked Questions), то есть списком Часто Задаваемых Вопросов по FreeBSD версий 13.X и 12.X. Мы прилагаем все усилия, чтобы сделать этот FAQ максимально информативным; если у вас есть идеи по его усовершенствованию, присылайте их на адрес Список рассылки Проекта Документации FreeBSD.

Самая последняя редакция этого документа всегда доступна на Web-сайте FreeBSD. Его также можно сгрузить в виде одного большого файла в формате HTML по HTTP или в различных других форматах с FTP-сервера FreeBSD.

Содержание

1. Вступление . . . 3
2. Документация и поддержка . . . 9
3. Установка . . . 12
4. Аппаратная совместимость . . . 16
5. Устранение некоторых проблем . . . 21
6. Прикладные программы . . . 28
7. Конфигурирование ядра . . . 31
8. Диски, файловые системы и начальные загрузчики . . . 33
9. ZFS . . . 42
10. Системное администрирование . . . 44
11. X Window System и виртуальные консоли . . . 53
12. Работа в сети . . . 61
13. Безопасность . . . 66
14. Коммуникационные адаптеры . . . 69
15. Разное . . . 72
16. Юмор от FreeBSD . . . 77
17. Сложные темы . . . 81
18. Наши благодарности . . . 86

Chapter 1. Вступление

1.1. FreeBSD - что это такое?

FreeBSD - это современная операционная система для настольных компьютеров, ноутбуков, серверов и встраиваемых систем с поддержкой большого количества платформ.

В основе FreeBSD лежит операционная система "4.4BSD-Lite" Калифорнийского Университета (Беркли) с некоторыми усовершенствованиями из "4.4BSD-Lite2". Также она косвенно базируется на 386BSD (BSD Net/2, перенесённой на платформу i386™ Уильямом Джолитцем (William Jolitz)), хотя от того первоначального кода осталось очень мало.

FreeBSD используется компаниями, Интернет-провайдерами, научными работниками, профессионалами в вычислительной технике, студентами и рядовыми пользователями по всему миру для работы, образования и отдыха.

Для более детального знакомства с FreeBSD обратитесь к Руководству по FreeBSD.

1.2. Какова цель FreeBSD?

Цель проекта FreeBSD - предоставить быструю и стабильную операционную систему общего назначения, которую можно использовать в любых целях без каких-либо ограничений.

1.3. Есть ли в лицензии FreeBSD какие-то ограничения?

Да. Эти ограничения не касаются аспектов использования кода, но главным образом описывают отношение к этому коду со стороны Проекта FreeBSD. Текст лицензионного соглашения доступен здесь, и вкратце он может быть изложен следующим образом:

- Не говорите, что это написано вами.
- Не судитесь с нами, если что-то не работает.
- Не удаляйте и не изменяйте лицензию.

Многие из нас вкладывают в проект значительные усилия и определённо были бы не против получения небольшой финансовой поддержки сейчас и в будущем, но мы на этом не настаиваем. Мы надеемся, что наша основная и самая значительная "миссия" - предоставить код всем желающим, для любых целей, так чтобы он нашел самое широкое применение и принёс наибольшую пользу. Это, на наш взгляд, одна из самых фундаментальных целей Free Software, которую мы с энтузиазмом поддерживаем.

Часть исходного кода нашей системы, подпадающая под действие GNU General Public License (GPL) или GNU Library General Public License (LGPL), имеет несколько больше ограничений, хотя и представляет собой навязывание доступа к исходным текстам, а не наоборот, как обычно. Из-за дополнительных сложностей, которые могут возникнуть в случае коммерческого использования программного обеспечения GPL, мы стараемся, где только

это возможно, заменить подобное программное обеспечение аналогичным, но подпадающим под менее строгую лицензию FreeBSD.

1.4. Может ли FreeBSD заменить операционную систему, используемую мною сейчас?

Для большинства людей это возможно. Но этот вопрос не так уж и однозначен.

Большинство пользователей на самом деле не используют операционную систему. Они работают с приложениями. Именно прикладные программы и используют операционную систему. FreeBSD разработана для того, чтобы дать надежное и полнофункциональное окружение для приложений. Она поддерживает широкий спектр Web-браузеров, офисных пакетов, программ для работы с электронной почтой, графических пакетов, программных сред, сетевых серверов и многое другое. Большинство этих приложений могут быть получено из Коллекции Портов.

Если приложение доступно только для одной операционной системы, то нельзя всего лишь заменить эту операционную систему. Однако есть вероятность, что похожая программа существует для FreeBSD. В качестве сервера для офиса, или сервера Интернет, или надежной рабочей станции FreeBSD практически всегда справится со всем, что вам нужно. Многие пользователи по всему миру, включая как новичков, так и опытных администраторов UNIX®, используют FreeBSD в качестве своей единственной настольной операционной системы.

Пользователям, переходящим на FreeBSD с другого варианта UNIX®-подобной системы, FreeBSD покажется очень знакомой. Пользователей Windows® и Mac OS®, напротив, может привлечь использование одного из трёх дистрибутивов GhostBSD, MidnightBSD или NomadBSD, предназначенных для настольных систем и построенных на базе FreeBSD. Пользователям, которые не использовали до этого UNIX®, нужно понимать, что понадобится потратить дополнительное время на изучение подхода UNIX® к организации работы. Этот FAQ и Руководство по FreeBSD являются прекрасными отправными точками.</para>

1.5. Почему система называется именно FreeBSD?

- Она может использоваться безо всяческих выплат, даже для извлечения выгоды.
- Все исходные тексты операционной системы свободно доступны, на её использование в других разработках (как коммерческих, так и некоммерческих) и дальнейшее распространение наложены минимальные ограничения.
- Любой, у кого есть усовершенствования или исправления, может предоставить свой код и он будет (правда, с парой оговорок) добавлен в исходные тексты системы.

Следует отметить, что слово "free" используется здесь в двух смыслах: один означает "бесплатно", а другой "делать всё, что хотите". За исключением пары вещей, которые вы *не можете* делать с FreeBSD, например, претендовать на то, что являетесь её разработчиком, на самом деле можно делать с ней всё, что вам заблагорассудится.

1.6. В чём заключается разница между FreeBSD и NetBSD, OpenBSD и другими операционными системами с открытым кодом семейства BSD?

Джеймс Ховард (James Howard) создал хорошее описание истории и отличий между разными проектами под названием Семейное древо BSD, в котором даётся подробный ответ на этот вопрос. Часть информации там устарела, однако историческая часть остаётся точной.

Многие из проектов семейства BSD обмениваются изменениями и готовым кодом даже сегодня. Все они происходят от общего предка.

Цели проекта FreeBSD описаны выше в Какова цель FreeBSD?. Цели других наиболее известных проектов семейства BSD можно кратко описать так:

- OpenBSD ориентируется на то, что превыше всего является безопасность операционной системы. Команда OpenBSD написала ssh(1) и pf(4), которые были перенесены во FreeBSD.
- NetBSD ориентируется на простое портирование на другие аппаратные платформы.
- DragonFly BSD отделилась от FreeBSD 4.8, и с тех пор в ней были разработаны многие интересные собственные функциональные возможности, включая файловую систему HAMMER и поддержку "vkernels" - запуска ядра в пользовательском режиме.

1.7. Какова последняя версия FreeBSD?

На любом этапе разработки FreeBSD может существовать несколько параллельных веток. Релизы 13.X выполняются из ветки 13-STABLE, а релизы 12.X выполняются из 12-STABLE.

Вплоть до версии 12.0 ветка 12.X была известна как *-STABLE*. Однако к моменту выхода 14.X ветка 12.X получит статус "extended support" (расширенная поддержка), и исправления будут вноситься только для серьёзных проблем, к примеру, связанных с безопасностью.

Релизы делаются раз в несколько месяцев. Хотя многие стараются отслеживать актуальное состояние исходных текстов FreeBSD (обратите внимание на вопросы о FreeBSD-CURRENT и FreeBSD-STABLE), делать это не обязательно, так как исходные тексты постоянно меняются.

Более полную информацию о релизах FreeBSD можно получить на странице Информации о подготовке релизов и на странице Справочника release(7).

1.8. Что такое FreeBSD-CURRENT?

FreeBSD-CURRENT - это версия операционной системы, находящаяся в стадии разработки, которая должна потом стать новой веткой FreeBSD-STABLE. Таким образом, она представляет реальный интерес только для разработчиков системы и её фанатов. Обратитесь к соответствующему разделу Руководства для прояснения деталей работы с *-CURRENT*.

Пользователям, не знакомым с FreeBSD, не следует использовать FreeBSD-CURRENT. Эта

ветка зачастую меняется очень быстро и иногда из-за ошибок может быть неработоспособной. Подразумевается, что те, кто используют FreeBSD-CURRENT, должны быть в состоянии изучить проблему, найти причину и сообщить о этом.

1.9. В чём смысл FreeBSD-STABLE?

FreeBSD-STABLE является веткой разработки, из которой выполняются основные релизы. В эту ветку изменения вносятся медленнее, и при этом предполагается, что до этого они были протестированы во FreeBSD-CURRENT. Несмотря на это, исходный код FreeBSD-STABLE в любой момент времени может быть пригоден, а может быть и непригоден к широкому использованию, так как может содержать скрытые ошибки и вырожденные случаи, которые ещё не были выявлены во FreeBSD-CURRENT. Пользователям, не имеющим возможностей для тестирования, следует работать с самым свежим релизом FreeBSD. С другой стороны, *FreeBSD-CURRENT* продолжает являться единой веткой, не разрываемой с момента выхода версии 2.0.

Для получения более подробной информации о ветках обратитесь к разделу статьи "Подготовка релизов FreeBSD: Создание ветки релиза", а состояние веток и расписание предстоящих релизов можно получить на странице Информация о подготовке релизов. Версия 12.3 является самым последним релизом в ветке 12-STABLE; она была выпущена December 7, 2021). Версия 13.1 является самым последним релизом в ветке 13-STABLE; она была выпущена May 16, 2022).

1.10. В какой момент выпускаются новые версии FreeBSD?

Группа Выпуска Релизов FreeBSD <re@FreeBSD.org> выпускает новую старшую версию FreeBSD в среднем каждые 18 месяцев и младшие версии каждые 8 месяцев. Даты релизов обычно объявляются заранее, так что те, кто работает над системой, знают, когда их проекты должны быть закончены и протестированы. Период тестирования предшествует выходу каждого релиза, для того, чтобы удостовериться в том, что добавление новых возможностей не повлияло на стабильность работы релиза. Многие пользователи расценивают такую осторожность как одну из приятнейших черт FreeBSD, хотя необходимость дожидаться *-STABLE* для получения всех этих новых возможностей может несколько разочаровывать.

Дополнительная информация о процессе подготовки релиза (в том числе планы выпуска последующих релизов) может быть найдена на страницах Web-сайта FreeBSD, посвящённых выпуску релизов.

Для тех, кому нужно или хочется, еженедельно выпускаются бинарные снапшоты, как описано выше.

1.11. Когда выпускаются снэпшоты FreeBSD?

Снэпшот-релизы FreeBSD выпускаются исходя из актуального состояния веток *-CURRENT* и *-STABLE*. Цели выпуска каждого такого снэпшот-релиза таковы:

- Протестировать самую последнюю версию программы установки системы.
- Дать людям, которые хотят работать с ветками *-CURRENT* или *-STABLE*, но не имеют достаточно времени или пропускной способности сети для отслеживания ежедневных изменений, простой способ их начальной установки.
- Зафиксировать состояние определённого кода на какой-то момент времени на случай, если позже мы что-нибудь очень сильно сломаем. (Хотя Git, как правило, не позволяет случиться ничему такому ужасному).
- Обеспечить все новые функции и исправления, которым требуется тестирование, как можно большим количеством потенциальных тестировщиков.

Не утверждается, что всякий снэпшот *-CURRENT* с любой точки зрения имеет качество "готового продукта". Если нужна стабильно работающая и полностью протестированная система, то придерживайтесь политики использования полноценных релизов.

Снэпшот-релизы доступны непосредственно отсюда.

Официальные снэпшоты регулярно выпускаются для всех активно разрабатываемых веток.

1.12. Кто отвечает за разработку FreeBSD?

Ключевые решения, касающиеся проекта FreeBSD, такие, как общее направление развития проекта или кто может добавлять код к дереву исходных текстов, принимаются управляющей командой разработчиков (Core Team), состоящей из 9 человек. Также существует многочисленная группа, состоящая из более чем 350 так называемых коммиттеров (committers), которые могут вносить изменения прямо в дерево исходных текстов FreeBSD.

Однако большинство нетривиальных изменений широко обсуждается в списках рассылки, и не существует никаких ограничений на участие в подобных дискуссиях.

1.13. Где можно найти FreeBSD?

Все поддерживаемые релизы FreeBSD доступны на странице поиска релизов FreeBSD:

- Для получения самого последнего релиза 13-STABLE, 13.2-RELEASE, перейдите по ссылке для выбора соответствующей архитектуры и режима установки для 13.2-RELEASE.
- Для получения самого последнего релиза 12-STABLE, 12.4-RELEASE, перейдите по ссылке для выбора соответствующей архитектуры и режима установки для 12.4-RELEASE.
- Снэпшот-релизы для веток -CURRENT и -STABLE выпускаются ежемесячно, но они нужны исключительно для разработчиков и тех, кто тестирует самые последние нововведения.

Информация о возможностях получения FreeBSD на CD, DVD и других носителях доступна в Руководстве.

1.14. Как можно получить доступ к базе сообщений о проблемах (Problem Report)?

База данных всех сообщений пользователей о проблемах может быть запрошена с помощью нашего Web-интерфейса.

Можно использовать Web-интерфейс для отсылки сообщений об ошибках через браузер.

Перед тем, как посылать сообщение об ошибке, прочтите статью Составление сообщений о проблеме во FreeBSD о том, как писать хорошие сообщения об ошибках.

Chapter 2. Документация и поддержка

2.1. Есть ли хорошие книги по FreeBSD?

В рамках проекта создаётся обширная документация, которая доступна в онлайне по следующей ссылке: https://www.FreeBSD.org/docs/.

2.2. Можно ли получить документацию в другом формате, например, в виде PDF?

Да. Документация доступна к сгрузке также и в других форматах на сайте FreeBSD.

Каталоги с документацией подразделяется на категории в зависимости от:

- Имени документа, например, `faq` или `handbook`.
- Языка документа на базе наименования региональных настроек, размещаемых в каталоге /usr/share/locale системы FreeBSD, без учёта кодировки, так как во всей документации сейчас используется UTF-8. На данный момент доступны следующие языки:

Кодировка	Язык
`en`	Английский
`bn-bd`	Бенгальский или Бангла (Бангладеш)
`da`	Датский
`de`	Немецкий
`el`	Греческий
`es`	Испанский
`fr`	Французский
`hu`	Венгерский
`it`	Итальянский
`ja`	Японский
`ko`	Корейский
`mn`	Монгольский
`nl`	Голландский
`pl`	Польский
`pt-br`	Португальский (Бразилия)
`ru`	Русский
`tr`	Турецкий
`zh-cn`	Упрощённый китайский (Китай)

Кодировка	Язык
zh-tw	Традиционный китайский (Тайвань)

Некоторые документы могут иметься не на всех языках.

- Формат документа. Мы распространяем документацию в нескольких различных форматах. У каждого из форматов имеются свои плюсы и свои минусы. Некоторые форматы лучше подходят для чтения в on-line, тогда как другие предназначены для получения эстетично выглядящей бумажной копии. Наличие документации во всех этих форматах обеспечивает возможность прочтения нашими пользователями любой её части как с экрана монитора, так и на бумаге после вывода на печать. На данный момент доступны следующие форматы;

Формат	Значение
html	В зависимости от документа: Один большой HTML-файл, содержащий документ полностью, или набор небольших связанных HTML-файлов, в обоих случаях содержащих изображения, таблицы стилей и код JavaScript
pdf	Adobe's Portable Document Format

- Способ сжатия и создания архива.
  a. В случае формата html, файлы пакетируются с помощью tar(1). Получающийся файл .tar затем сжимается утилитой gzip(1).
  b. При использовании формата PDF создаётся один файл. К примеру, explaining-bsd_en.pdf, faq_en.pdf и так далее.

Выбрав формат, сгрузите файлы, распакуйте их при необходимости, а затем скопируйте документацию в соответствующие места.

Например, HTML-версию FAQ можно найти в doc/en/books/faq/faq_en.tar.gz. Для сгрузки и распаковки этого файла выполните:

```
% fetch https://download.freebsd.org/doc/en/books/faq/faq_en.tar.gz
% tar xvf faq_en.tar.gz
```

Если файл сжат, tar автоматически определит подходящий формат и корректно его распакует, в результате чего появится набор файлов. Главным является index.html, и в нём находится всё содержимое документа, начиная с оглавления, ссылающегося на остальные части документа.

2.3. Где найти информацию по спискам рассылки FreeBSD? Какие существуют телеконференции по FreeBSD?

Исчерпывающая информация содержится в extref:https://docs.freebsd.org/ru/books/handbook/eresources/[разделе Руководства, eresources-mail], который посвящён спискам рассылки, и в разделе Руководства, касающемся новостных конференций.

2.4. Существуют ли каналы IRC (Internet Relay Chat) по FreeBSD?

Да, в большинстве крупнейших сетей IRC имеется канал для обсуждения FreeBSD, а на wiki-странице FreeBSD размещён актуальный список IRC-каналов.

Все эти каналы разные и не связаны друг к другом. Поскольку их манеры общения отличаются, попробуйте каждый, пока не найдёте соответствующий вашему стилю.

2.5. Есть ли какие-нибудь web-форумы для обсуждения FreeBSD?

Официальные форумы FreeBSD расположены по адресу https://forums.FreeBSD.org/.

2.6. Где можно пройти платные курсы по FreeBSD и получить поддержку?

iXsystems, Inc., дочерней компанией которой является FreeBSD Mall, оказывает услуги поддержки программного обеспечения FreeBSD на коммерческой основе, в дополнение к разработкам на платформе FreeBSD и решениям, требующим тонкой настройки.

BSD Certification Group, Inc. предоставляет сертификацию системного администрирования DragonFly BSD, FreeBSD, NetBSD и OpenBSD. Для получения дополнительной информации посетите их сайт.

Чтобы попасть в этот список, другие организации, осуществляющие обучение и поддержку, должны обратиться к нам в Проект.

Chapter 3. Установка

3.1. Какую архитектуру нужно загрузить? У меня есть 64-разрядный процессор Intel®, но я вижу только amd64.

amd64 - это термин, применяемый во FreeBSD для обозначения 64-разрядной архитектуры x86 (также известна как "x86-64" или "x64"). На большинстве современных компьютеров следует использовать amd64. Для более старых подойдёт i386. При установке системы на отличную от x86 архитектуру, выберите платформу, наиболее подходящую для оборудования.

3.2. Какой файл нужно скачать для установки FreeBSD?

На странице Получение FreeBSD выберите `[iso]` с соответствующей оборудованию архитектурой.

Можно использовать любой из:

файл	описание
disc1.iso	Содержит достаточно для установки FreeBSD и минимальный набор пакетов.
dvd1.iso	Наподобие disc1.iso, но с дополнительными пакетами.
memstick.img	Образ с автозагрузкой для записи на USB флешку.
bootonly.iso	Минимальный образ, требующий сетевое подключение для завершения установки FreeBSD.

Полные инструкции по этой процедуре, а также более подробную информацию по общим вопросам, возникающим при установке, можно найти в разделе Руководства об установке FreeBSD.

3.3. Что нужно делать, если установочный образ не запускается?

Это может быть вызвано тем, что образ был загружен по FTP не в режиме *binary*.

В некоторых клиентских программах FTP по умолчанию используется текстовый (*ascii*) режим передачи, в котором любые последовательности символов "конец строки" заменяются на используемые в системе клиента. В таком случае образ загрузочного диска

будет неизбежно испорчен. Проверьте контрольную сумму SHA-256 полученного файла: если он не *точно* такой же как на FTP-сервере, то ошибка произошла, скорее всего, в процессе передачи.

В случае использования командной строки FTP-клиента введите команду *binary* в командной строке FTP после подключения к серверу, но перед началом передачи файла.

3.4. Где находятся инструкции по установке FreeBSD?

Инструкции по установке можно найти в extref:https://docs.freebsd.org/ru/books/handbook/bsdinstall/[главе Руководства], посвящённой установке FreeBSD.

3.5. Как сделать собственный установочный диск?

Индивидуальный установочный носитель FreeBSD можно создать, запустив процедуру построения индивидуального релиза. Следуйте инструкциям в статье о подготовке релизов FreeBSD.

3.6. Может ли Windows® сосуществовать с FreeBSD? (специфично для x86)

Да, если Windows® установлена первой. Загрузчик FreeBSD будет управлять процессом выбора загрузки Windows® или FreeBSD. Если Windows® устанавливается следом, то это приведёт к перезаписи загрузчика. Если такое случится, обратитесь к следующему разделу.

3.7. Другая операционная система уничтожила мой загрузчик операционной системы! Как мне его вернуть? (специфично для x86)

Способ восстановления зависит от используемого загрузчика. Меню выбора загрузки, используемое во FreeBSD, можно переустановить с помощью boot0cfg(8). Пример для восстановления меню загрузки на диске *ada0*:

```
# boot0cfg -B ada0
```

Неинтерактивный загрузчик MBR можно установить с помощью gpart(8):

```
# gpart bootcode -b /boot/mbr ada0
```

Более сложные ситуации, включая использование дисков GPT, рассматриваются в gpart(8).

3.8. Нужно ли устанавливать исходные тексты системы?

В общем случае, нет. Для работы основной системы присутствие исходных текстов не требуется. Некоторые порты наподобие sysutils/lsof не будут собираться без установленных исходных текстов системы. В частности, если порт собирает модуль ядра или напрямую обращается к структурам ядра, в этом случае исходные тексты должны быть установлены.

3.9. Нужно ли перекомпилировать ядро?

Обычно нет. Поставляемое ядро `GENERIC` содержит драйвера, необходимые для типового компьютера. Инструмент freebsd-update(8) не может использоваться для обновления FreeBSD с собственным ядром, и это является ещё одной причиной для того, чтобы по возможности придерживаться использования ядра `GENERIC`. Для компьютеров с очень небольшим объёмом ОЗУ, таких как встраиваемые системы, может потребоваться собственное небольшое ядро, содержащее только необходимые драйверы.

3.10. Какой из методов шифрования паролей (DES, Blowfish или MD5) я должен использовать, и как указать, шифрование какого типа применяется пользователями?

Во FreeBSD по умолчанию используется метод шифрования *SHA512*. Пароли, зашифрованные методом DES, остаются доступными для обратной совместимости с операционными системами, в которых всё ещё используется менее защищённый метод шифрования паролей. FreeBSD также поддерживает пароли в форматах Blowfish и MD5. Управление выбором используемого метода для новых паролей осуществляется через параметр входа `passwd_format` в файле /etc/login.conf, принимающий значения `des`, `blf` (если они доступны) или `md5`. Подробная информация о параметрах входа находится на странице Справочника login.conf(5).

3.11. Какие существуют ограничения для файловой системы FFS?

Наибольший размер файловой системы FFS ограничен практически количеством памяти, которая требуется для работы fsck(8). fsck(8) использует 1 бит на фрагмент, и для стандартного размера фрагмента 4 Кбайт это эквивалентно использованию 32 Мбайт памяти на терабайт дискового пространства. Это означает, что на архитектурах с ограничением размера пользовательского процесса в 2 Гбайт (например, i386™) максимальный размер файловой системы, доступный для fsck(8), составляет ~60 Тбайт.

Без ограничения на память для fsck(8) максимальный размер файловой системы составляет 2 ^ 64 (блоков) * 32 Кбайт ⇒ 16 экса * 32 Кбайт ⇒ 512 зеттабайт.

Максимальный размер файла на FFS приблизительно равен 2 петабайт со стандартным размером блока 32 Кбайт. Каждый 32 Кбайтный блок может адресовать до 4096 блоков. С использованием тройной косвенной адресации это составляет 32 Кбайт * 12 + 32 Кбайт * 4096 + 32 Кбайт * 4096^2 + 32 Кбайт * 4096^3. Увеличение размера блока до 64 Кбайт увеличит максимальный размер файла в 16 раз.

3.12. Я скомпилировал новое ядро и при загрузке получил сообщение об ошибке readin failed.

Ядро и компоненты системы не синхронизированы. Такая конфигурация не поддерживается. Обязательно используйте команды `make buildworld` и `make buildkernel` для обновления ядра.

Загрузите систему, непосредственно указав ядро на втором этапе загрузки, нажав любую клавишу до запуска загрузчика при появлении символов `|`.

3.13. Существует ли инструмент для настройки системы после её установки?

Да, bsdconfig предоставляет замечательный интерфейс для пост-установочной настройки FreeBSD.

Chapter 4. Аппаратная совместимость

4.1. Вопросы общего характера

4.1.1. Я хочу приобрести некое оборудование для моей системы FreeBSD. Какая модель/производитель/тип лучше всего?

Это постоянно обсуждается в списках рассылки FreeBSD и является ожидаемым вопросом, так как аппаратура меняется очень быстро. Обратитесь к Hardware Notes для FreeBSD 12.3 или 13.1, а также поищите в архивах списков рассылки перед тем, как задавать вопросы о самом последнем и лучшем оборудовании. Весьма вероятно, что обсуждение касаемо этого типа оборудования велось как раз на прошлой неделе.

Перед приобретением лэптопа посмотрите архивы Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD или, возможно, более специфичные списки рассылки по данному типу оборудования.

4.1.2. Каковы ограничения на оперативную память?

FreeBSD как операционная система в целом поддерживает столько же физической памяти (ОЗУ), сколько аппаратная платформа, на которой она работает. Имейте в виду, что различные платформы имеют различные ограничения на память; например, i386™ без PAE поддерживает максимум 4 Гбайт памяти (и обычно ещё меньше из-за адресного пространства PCI), а i386™ с PAE поддерживает максимум 64 Гбайт. Во FreeBSD 10 для платформы AMD64 поддерживается до 4 Тбайт физической памяти.

4.1.3. Почему FreeBSD видит меньше 4 Гбайт памяти, когда система установлена на машину i386™?

Общее адресное пространство для машин i386™ является 32-разрядным; это означает, что адресоваться (т.е. быть получено) может не более 4 Гбайт памяти. Более того, некоторые адреса в этом диапазоне зарезервированы для различных целей аппаратным обеспечением, например, для использования и управления устройствами PCI, для доступа к видеопамяти, и так далее. Таким образом, общий объем памяти, используемой операционной системой для ядра и приложений, ограничен размером, значительно меньшим, чем 4 Гбайт. В такой конфигурации максимально доступная физическая память составляет от 3.2 Гбайт до 3.7 Гбайт.

Для преодоления ограничения в 3.2 Гбайт-3.7 Гбайт установленной памяти (т.е. для получения 4 Гбайт, но также более 4 Гбайт) должен использоваться специальный механизм, именуемый PAE. Сокращение PAE означает Physical Address Extension (расширение физического адреса) и предоставляет для 32-разрядных x86 процессоров способ адресовать более 4 Гбайт памяти. PAE переназначает память, которая иначе была бы перекрыта адресными резервациями для аппаратных устройств выше диапазона 4 Гбайт, и использует её как дополнительную физическую память (смотрите pae(4)). Использование PAE имеет свои недостатки; такая модель доступа к памяти является чуть более медленной по сравнению с обычным режимом (без PAE), и также не работают динамически загружаемые

модули (смотрите kld(4)). Это означает, что все драйверы должны присутствовать статически в самом ядре.

Самый распространённым способом включения PAE является сборка нового ядра со специальным уже подготовленным файлом конфигурации ядра, именуемым PAE, который уже сконфигурирован для сборки безопасного ядра. Имейте в виду, что некоторые строки в этом файле конфигурации ядра являются слишком консервативными, и некоторые драйверы, помеченные как неготовые для использования с PAE, на самом деле являются годными. На практике, если драйвер работает на 64-разрядной архитектуре (такой как AMD64), он также работает с PAE. При создании своего собственного файла конфигурации ядра можно включить PAE, добавив следующую строку:

```
options       PAE
```

PAE не является широко используемым в настоящее время, поскольку большинство нового x86 аппаратного обеспечения также поддерживает работу в 64-разрядном режиме, также известном как AMD64 или Intel® 64. Этот режим имеет большее адресное пространство и не нуждается в таких трюках. FreeBSD поддерживает AMD64, и рекомендуется использование этой версии FreeBSD вместо версии i386™, если требуется больше 4 Гбайт памяти.

4.2. Аппаратные платформы и процессоры

4.2.1. Поддерживает ли FreeBSD аппаратные платформы, отличные от x86?

Полноценно поддерживаются архитектуры первого класса, такие, как i386 или amd64. Архитектуры 2 и 3 класса поддерживаются, исходя из принципа наибольшего внимания при имеющихся возможностях. Полное описание классов доступно в extref:https://docs.freebsd.org/ru/articles/committers-guide/[Руководстве коммиттера, archs].

Полный список поддерживаемых архитектур находится на странице, посвящённой платформам.

4.2.2. Поддерживает ли FreeBSD многопроцессорные системы (SMP)?

FreeBSD поддерживает симметричное мультипроцессирование (SMP) на всех невстраиваемых платформах (например, i386, amd64 и так далее). SMP также поддерживается для arm и MIPS, хотя некоторые процессоры могут это не поддерживать. В реализации SMP во FreeBSD используется мелкодисперсная синхронизация, и производительность масштабируется почти линейно с ростом количества процессоров.

За подробной информацией обращайтесь к странице Справочника smp(4).

4.2.3. Что такое микрокод? Как установить обновление микрокода для процессоров AMD или Intel?

Микрокод является программным средством реализации аппаратных инструкций

процессора. Его использование позволяет исправлять ошибки процессора без замены микросхемы.

Установите sysutils/cpu-microcode, а затем добавьте:

```
microcode_update_enable="YES"
```

в /etc/rc.conf

4.3. Периферийные устройства

4.3.1. Какого рода периферийные устройства поддерживает FreeBSD?

За информацией о списке оборудования, о котором известно, что оно работоспособно, а также данными о каких бы то ни было ограничечниях, обратитесь к Hardware Notes для FreeBSD link:https://www.FreeBSD.org/releases/12.3R/hardware/ [12.3] или 13.1.

4.4. Клавиатуры и мыши

4.4.1. Можно ли использовать мышь вне X Window?

Используемый по умолчанию драйвер консоли syscons(4) предоставляет возможность использования указателя мыши в текстовых консолях для выделения и переноса текста. Запустите демон мыши moused(8) и включите отображение указателя мыши в виртуальной консоли:

```
# moused -p /dev/xxxx -t yyyy
# vidcontrol -m on
```

Здесь *xxxx* - это имя устройства мыши, а *yyyy* - тип протокола, используемого мышью. Даемон мыши может автоматически определять тип протокола большинства мышей, за исключением старых, работающих по последовательному интерфейсу. Для выполнения автоматического определения в качестве протокола укажите `auto`. Если автоматическое определение не работает, то обратитесь к справочным страницам по moused(8) для получения списка поддерживаемых типов протоколов.

Для мыши типа PS/2 добавьте строчку `moused_enable="YES"` в файл /etc/rc.conf для запуска даемона мыши во время загрузки системы. Кроме того, для использования даемона мыши во всех виртуальных терминалах, а не только на консоли, добавьте `allscreens_flags="-m on"` в файле /etc/rc.conf.

После запуска даемона мыши, доступ к мыши должен согласовываться между даемоном мыши и другими программами типа X Window. Обратитесь к вопросу из FAQПочему моя мышь не работает с X? для получения более полной информации по этому вопросу.

4.4.2. Как можно вырезать и копировать текст с помощью мыши в текстовой консоли?

Удалить данные с помощью мыши нельзя. Однако их можно скопировать и вставить. После запуска даемона мыши, как описано в ответе на предыдущий вопрос, нажмите кнопку 1 (левую) и двигайте мышь для выделения текста. Затем нажмите кнопку 2 (среднюю) для его вставки с позиции текстового курсора. Нажатие кнопки 3 (правой) "расширит" выбранную текстовую область.

Если у вашей мыши отсутствует средняя кнопка, её можно сэмулировать либо переназначить кнопки опциями даемона мыши. Обратитесь к справочным страницам по moused(8) для получения полной информации.

4.4.3. У моей мыши есть дополнительные колёсико и кнопочки. Можно ли их использовать во FreeBSD?

Ответ, к сожалению, "в зависимости от обстоятельств". Эти мышки с дополнительными возможностями, как правило, требуют наличия специальных драйверов. До тех пор, пока драйвер мыши или прикладная программа не будут иметь отдельную поддержку такой мыши, она будет работать как стандартная двух- или трёхкнопочная мышь.

Возможные способы использования колёсиков мыши при работе в X Window описаны в другом разделе.

4.4.4. Как использовать клавишу delete в sh и csh?

Для Bourne Shell добавьте следующие строки в ~/.shrc. Смотрите sh(1) и editrc(5).

```
bind ^[[3~ ed-delete-next-char # for xterm
```

Для C Shell добавьте следующие строки в ~/.cshrc. Смотрите csh(1).

```
bindkey ^[[3~ delete-char # for xterm
```

4.5. Другое оборудование

4.5.1. Есть ли решение проблемы отсутствия звука при использовании звуковых адаптеров pcm(4)?

Некоторые звуковые адаптеры при каждой загрузке устанавливают нулевой уровень громкости выводимого звука. При работе с FreeBSD 13 и более ранними версиями при каждой загрузке машины выполняйте следующую команду:

```
# mixer pcm 100 vol 100 cd 100
```

Используйте следующую команду при работе с FreeBSD 14 и более поздними версиями:

```
# mixer pcm.volume=100 vol.volume=100 cd.volume=100
```

4.5.2. Поддерживает ли FreeBSD управление энергосбережением на лэптопах?

FreeBSD поддерживает функции ACPI, реализованные в современном оборудовании. Дополнительная информация находится на странице Справочника acpi(4).

Chapter 5. Устранение некоторых проблем

5.1. Почему FreeBSD определяет неправильное количество памяти на аппаратуре i386™?

Наиболее вероятная причина заключается в различии между адресами физической и виртуальной памяти.

Существующее соглашение для большинства оборудования ПК заключается в использовании пространства памяти, лежащей в диапазоне между 3.5 ГБ и 4 ГБ для специальных нужд (обычно для нужд PCI). Это пространство адресов используется для доступа к PCI оборудованию. Как результат, реальная физическая память не может быть получена в данном адресном пространстве.

Какие действия выполняются с памятью в данном регионе, зависит от оборудования. К сожалению, некоторое оборудование ничего не выполняет и возможность использовать эти 500 МБ ОЗУ полностью потеряна.

К счастью, большинство оборудования перераспределяет память к более верхней позиции, так что она всё ещё может использоваться. Тем не менее, это может вызвать некоторое замешательство при просмотре сообщений, выдаваемых при загрузке.

На 32-битной версии FreeBSD кажется, что эта память потерялась, поскольку она переназначится в диапазон выше 4 ГБ, который не доступен для 32 битного ядра. В данном случае, решение заключается в сборке ядра с PAE. За дополнительной информацией обращайтесь к статье об ограничениях памяти.

На 64nbsp;битной версии FreeBSD или в случае использования ядра с включённым PAE FreeBSD корректно определит и перераспределит память, так, что она станет годной к использованию. Тем не менее, во время загрузки может показаться, что FreeBSD определяет больше памяти, чем реально имеется в системе из-за описанного перераспределения. Это нормально, и информация о доступной памяти будет скорректирована по окончанию процесса загрузки.

5.2. Программы аварийно завершают работу с ошибкой Signal 11.

Ошибки выполнения, связанные с сигналом 11, происходят, когда процесс пытается обратиться к области памяти, доступ к которой ему не был дан операционной системой. Если что-то подобное происходит в случайные, казалось бы, промежутки времени, следует начать поиск причины.

Эти проблемы могут быть классифицированы следующим образом:

1. Если проблема возникает только в определённом самостоятельно разработанном приложении, то скорее всего это ошибка в коде.

2. Если это проблема в части базового комплекта системы FreeBSD, то это тоже может быть ошибка в программном коде, хотя в большинстве случаев такие проблемы обнаруживаются и ошибки исправляются задолго до того, как обычным читателям FAQ доводится использовать этот код (именно для этого предназначена версия -CURRENT).

Вероятно, это не связано с ошибкой во FreeBSD, если проблема проявляется при компиляции программы, и при этом ошибка компилятора каждый раз разная.

Например, если запуск `make buildworld` завершился неудачей при попытке компиляции ls.c в ls.o и при повторном запуске компиляция снова прервалась на том же месте, то это ошибка процесса построения. Обновите исходные тексты и попробуйте снова. Если же компиляция прерывается в каком-то другом месте, то причина наиболее вероятно кроется в оборудовании.

В первом случае воспользуйтесь отладчиком, к примеру, gdb(1), для нахождения точки программы, в которой делается попытка доступа к неверному адресу, и исправьте эту ошибку.

Во втором случае проверьте, какой компонент вашего оборудования неисправен.

Среди часто приводящих к этому причин:

1. Жёсткие диски могут перегреваться. Проверьте работоспособность вентиляторов, так как жёсткие диски и другое оборудование могут перегреваться.
2. Процессор перегревается. Это может произойти при разгоне процессора или при поломке процессорного вентилятора. В любом случае убедитесь, что ваше оборудование работает в нормальном режиме, как ему и положено, по крайней мере, на момент поиска причин неисправности. В противном случае сбросьте частоту на настройки по умолчанию.)

   Что касается разгона, то медленная система обходится дешевле, чем сгоревшая система, требующая замены! К тому же проблемы на таких системах не находят понимания общественности.

3. Проблемная память. Если установлены различные микросхемы SIMM/DIMM, вытащите их все и попробуйте по одной до локализации проблемы в проблематичной микросхеме DIMM/SIMM, либо их комбинации.
4. Чересчур оптимистичные настройки материнской платы. Настройки BIOS и перемычки на материнской плате предоставляют возможность задавать различные частоты и задержки. Часто бывает достаточно настроек по умолчанию, но иногда установка слишком малых периодов ожидания для ОЗУ или установка параметра "RAM Speed: Turbo" вызывает странное поведение. Возможным решением может стать установка параметров BIOS по умолчанию с предварительной записью текущих значений.
5. Неустойчивое или недостаточное электропитание материнской платы. Уберите неиспользуемые адаптеры ввода/вывода, жёсткие диски и приводы компакт-дисков или отключите их от кабеля электропитания для проверки, что блок питания может работать с меньшей нагрузкой. Или попробуйте воспользоваться другим блоком питания, желательно большей мощности. Например, если имеющийся блок питания

рассчитан на 250 Ватт, попробуйте другой мощностью 300 Ватт.

Прочитайте раздел про Signal 11 для дальнейшего объяснения и обсуждения, как аппаратура или программное обеспечение для тестирования памяти могут пропускать сбойную память. Подробная информация по этому вопросу содержится в FAQ по проблеме SIG11.

Наконец, если ничего не помогает, то, возможно, это из-за ошибки во FreeBSD. Следуйте этим инструкциям для отправки сообщения о проблеме.

5.3. Моя система аварийно завершает работу с сообщениями Fatal trap 12: page fault in kernel mode либо panic:, и выдаёт много дополнительной информации. Что мне делать?

Разработчики FreeBSD интересуются такими ошибками, но им нужно больше информации, чем просто текст ошибки. Скопируйте весь текст сообщения. Затем обратитесь к разделу FAQ об аварийных завершениях работы ядра, постройте отладочное ядро и получите трассу вызовов. Это может звучать трудной задачей, зато не требует навыков программирования. Просто следуйте указаниям.

5.4. Что означает сообщение об ошибке maxproc limit exceeded by uid %i, please see tuning(7) and login.conf(5)?

Ядро FreeBSD позволяет одновременно существовать ограниченному числу процессов. Это зависит от значения переменной sysctl(8) `kern.maxusers`. `kern.maxusers` также влияет на другие ограничения ядра, такие как буферы работы с сетью. Если система сильно загружена, поднимите `kern.maxusers`. Кроме максимального числа процессов это также увеличит значения других параметров, ограничивающих систему.

Для корректировки значения `kern.maxusers` обратитесь к разделу Ограничения файлов/процессов Руководства. В нём говорится об открытых файлах, но те же ограничения касаются и процессов.

Если система загружена слабо, но в ней запущено слишком много процессов, поправьте параметр `kern.maxproc`, определив его значение в /boot/loader.conf. Изменение не вступит в силу до перезагрузки системы. За дополнительной информацией, касающейся настройки параметров, обращайтесь к странице Справочника loader.conf(5). Если эти процессы запущены одним и тем же пользователем, поправьте значение `kern.maxprocperuid`, чтобы оно было на единицу меньше, чем новое значение `kern.maxproc`. Оно должно быть меньше по крайней мере на единицу, потому что системная программа init(8) должна работать всегда.

5.5. Полноэкранные приложения на удалённой машине работают неправильно!

На удалённой машине тип терминала может отличаться от `xterm`, который требуется для использования консоли FreeBSD. Либо же ядро может иметь неправильные значения ширины и высоты терминала.

Проверьте, чтобы переменная окружения `TERM` имела значение `xterm`. Если удалённая машина его не поддерживает, попробуйте `vt100`.

Запустите `stty -a`, чтобы узнать, какие размеры терминала заданы в ядре. Если значения неправильные, их можно поменять командой `stty rows RR cols CC`.

Либо же, если на клиентской машине установлен x11/xterm, запуск `resize` позволит узнать у терминала правильные размеры и применить эти значения.

5.6. Почему подключение через ssh или telnet к моему компьютеру занимает так долго времени?

Симптом: между моментом установления TCP-соединения и выдачей клиентским программным обеспечением запроса на ввод пароля (или, в случае использования telnet(1), выдачей приглашения на вход) проходит большой промежуток времени.

Проблема: скорее всего, задержка вызвана программным обеспечением на стороне сервера, которое пытается преобразовать IP-адрес клиента в имя хоста. Многие серверы, включая Telnet и SSH, поставляемые с FreeBSD, делают это для того, чтобы, кроме всего прочего, записать имя хоста в файле журнала для справки администратора.

Лечение: Если проблема возникает при подключении клиента к любому серверу, то причина в клиенте. Если проблема возникает только при чьей-либо попытке подключиться к серверу, то проблема в сервере.

Если проблема с клиентом, то единственным методом ее решения является исправление DNS, чтобы сервер смог распознать вашу машину. Если это происходит в локальной сети, то предположите, что это проблема с сервером, и продолжайте чтение. Если это происходит в сети Интернет, обратитесь к вашему провайдеру.

Если проблема с сервером в локальной сети, настройте сервер для разрешения запросов на преобразование адреса в имя хоста в диапазоне локальных адресов. Обратитесь к страницам Справочника по hosts(5) и named(8) для получения более подробной информации. Если это происходит в сети Интернет, то проблема может заключаться в некорректной работе ресолвера локального сервера. Для проверки попробуйте получить адрес другого хоста, такого как `www.yahoo.com`. Если это не работает, то в этом и состоит проблема.

Из-за свежей установки FreeBSD, также возможно, что информация о домене и сервере имён отсутствует в /etc/resolv.conf. Это часто будет вызывать задержку в работе SSH, так как опция `UseDNS` по умолчанию установлена в значение `yes` в /etc/ssh/sshd_config. Если именно

это является причиной проблемы, то добавьте недостающую информацию в /etc/resolv.conf, либо в качестве временной меры установите `UseDNS` в `no` в файле sshd_config.

5.7. Почему в dmesg(8) регулярно выводятся сообщения file: table is full?

Такое сообщение об ошибке сигнализирует о том, что в системе закончились доступные файловые дескрипторы. Обратитесь к разделу kern.maxfiles главы о настройке ограничений ядра Руководства для выяснения всех подробностей и поиска решения.

5.8. Почему часы на моем компьютере показывают неправильное время?

На компьютере установлено по меньшей мере два таймера, и FreeBSD выбрала не тот.

Запустите dmesg(8) и посмотрите строки, содержащие слово `Timecounter`. FreeBSD выбирает таймер с наибольшим значением качества.

```
# dmesg | grep Timecounter
Timecounter "i8254" frequency 1193182 Hz quality 0
Timecounter "ACPI-fast" frequency 3579545 Hz quality 1000
Timecounter "TSC" frequency 2998570050 Hz quality 800
Timecounters tick every 1.000 msec
```

Удостоверьтесь в этом, проверив sysctl(3)-переменную `kern.timecounter.hardware`.

```
# sysctl kern.timecounter.hardware
kern.timecounter.hardware: ACPI-fast
```

Это может быть неработающий таймер ACPI. Самым простым решением будет отключить таймер ACPI в /boot/loader.conf:

```
debug.acpi.disabled="timer"
```

Либо же BIOS может изменить частоту таймера TSC - возможно, для изменения скорости работы процессора при работе от батареи или переводя в режим пониженного электропитания, но FreeBSD не отслеживает это, и в результате часы начинают спешить или отставать.

В этом примере имеется также и таймер `i8254`, и он может быть выбран записью его имени в sysctl(3)-переменную `kern.timecounter.hardware`.

```
# sysctl kern.timecounter.hardware=i8254
```

```
kern.timecounter.hardware: TSC -> i8254
```

Теперь компьютер должен аккуратнее следить за временем.

Чтобы это изменение вступало в силу во время загрузки системы, добавьте в файл /etc/sysctl.conf такую строчку:

```
kern.timecounter.hardware=i8254
```

5.9. Что означает сообщение swap_pager: indefinite wait buffer:?

Это значит, что процесс пытается считать страницу памяти с диска, но попытки сделать это оканчиваются неудачно в течение более 20 секунд. Это может быть вызвано повреждёнными блоками на диске, кабелями, подключением или другим оборудованием ввода/вывода. Если диск сам по себе повреждён, вы также увидите ошибки работы с диском в /var/log/messages и в выводе `dmesg`. В противном случае проверьте кабели и подключения.

5.10. Что такое lock order reversal?

Ядро FreeBSD использует несколько блокировок для арбитража доступа к соответствующим ресурсам. Когда несколько потоков в ядре пытаются захватить несколько блокировок подряд, всегда существует возможность появления мёртвой блокировки (deadlock), где два потока захватили по одной блокировке и заблокированы в ожидании освобождения другим потоком второй блокировки. Такой проблемы синхронизации можно избежать, если все потоки получают блокировки в одинаковом порядке.

Система диагностирования блокировок witness(4), которая по умолчанию включена во FreeBSD-CURRENT и выключена для стабильных веток и релизов, определяет возможность появления мёртвых блокировок из-за ошибок их использования, включая захват нескольких блокировок в различном порядке в разных частях ядра. Механизм witness(4) пытается обнаруживать эту проблему по мере её появления и сообщает о ней на системную консоль сообщением `lock order reversal` (которое также часто называют LOR).

В силу консервативности witness(4) возможны ложные срабатывания. При правильном срабатывании такое сообщение *не* означает, что система находится в состоянии мёртвой блокировки; его следует рассматривать как предупреждение о том, что в этом месте могла произойти мёртвая блокировка.

Плохие LOR обычно быстро исправляют, поэтому перед написанием сообщения в списки рассылки следует проверить архивы Список рассылки, посвящённый обсуждению FreeBSD-CURRENT.

5.11. Что означают сообщения `Called … with the following non-sleepable locks held`?

Это означает, что функция, которая может находиться в "спящем" состоянии была вызвана во время использования мьютекс (или другого не "засыпающего") блокирования.

Объяснением того, что это ошибка, является то, что мьютексы не предполагают нахождения в удерживаемом состоянии длительные промежутки времени, и блокируются только на короткие периоды синхронизации. Это правило позволяет драйверам устройств использовать мьютексы для синхронизации с остальной частью ядра во время прерываний. Прерывания (во FreeBSD) могут находиться не в "спящем состоянии". Следовательно необходимо, чтобы не было подсистем в ядре, которые бы занимались блокировкой длительный период, используя мьютекс.

Для нахождения таких ошибок в ядро могут быть добавлены assertions, которые будут взаимодействовать с подсистемой witness(4) для генерирования предупреждения или фатальной ошибки (в зависимости от системной конфигурации) в случаях когда производится потенциально блокирующий вызов с удержанием мьютекса.

В общем, такие предупреждения не критичны, но тем не менее, с неудачной синхронизацией (timing) они могут вызвать нежелательные эффекты, начиная от незначительной задержки в ответной реакции системы до полной её блокировки.

Дополнительная информация о синхронизации во FreeBSD находится на странице Справочника locking(9).

5.12. Почему процесс buildworld/installworld завершается с сообщением touch: not found?

Эта ошибка не означает, что не найдена утилита touch(1). Ошибка наверняка появляется из-за того, что даты модификации файлов установлены в будущем. Если CMOS часы установлены на локальное время, отрегулируйте часовой механизм ядра, запустив команду `adjkerntz -i` в однопользовательском режиме.

Chapter 6. Прикладные программы

6.1. Где находятся все прикладные программы?

Обратитесь на страницу портов, содержащую информацию о программных продуктах, перенесённых во FreeBSD.

Большинство портов должно нормально работать во всех поддерживаемых версиях FreeBSD. Неработоспособные порты специально помечаются соответствующим образом. При выходе нового релиза FreeBSD в него в каталог ports/ также включается актуальная на момент выхода Коллекция Портов.

FreeBSD умеет работать со сжатыми двоичными пакетами для упрощения установки и удаления портов. Используйте pkg(7) для управления процессом установки пакетов.

6.2. Как загрузить дерево Портов? Следует ли использовать Git?

Обратитесь к разделу Установка Коллекции Портов.

6.3. Почему этот порт не компилируется на моей машине с 12.X - или 13.X -STABLE?

Если установленная версия FreeBSD значительно отстаёт от *-CURRENT* или *-STABLE*, обновите Коллекцию Портов, следуя указаниям раздела Использование Коллекции Портов. Если система находится в актуальном состоянии, то, возможно, кто-то внёс изменение в порт, который работоспособен в *-CURRENT*, но не работает в *-STABLE*. Отправьте сообщение об ошибке, так как предполагается, что Коллекция Портов работает как для ветки *-CURRENT*, так и *-STABLE*.

6.4. Я попытался сформировать файл INDEX командой make index, однако попытка окончилась неудачей. Почему?

Первым делом убедитесь, что Коллекция Портов находится в актуальном состоянии. Ошибки, которые отражаются на построении файла INDEX из актуальной копии Коллекции Портов, бросаются в глаза и поэтому практически всегда исправляются немедленно.

В редких случаях INDEX не перестраивается из-за странных комбинаций значений переменных `OPTIONS_SET`, заданных в файле make.conf. Если вы подозреваете, что дело именно в этом, то, прежде чем сообщать об этом в Список рассылки, посвящённый Портам FreeBSD, попытайтесь сформировать INDEX с отключенными значениями этих переменных.

6.5. Я обновил исходные тексты. Как теперь обновить установленные порты?

С FreeBSD не поставляется инструмент обновления портов, но есть несколько инструментов, немного облегчающих этот процесс. Кроме того, для упрощения работы с портами доступны дополнительные инструменты; смотрите раздел Руководства FreeBSD об обновлении портов.

6.6. Нужно ли мне перекомпилировать все порты каждый раз, когда я выполняю мажорное обновление (со сменой старшей версии FreeBSD)?

Да! Хотя на новой системе программное обеспечение, скомпилированное под более старый релиз, будет работать, но только до тех пор, пока вы не начнёте устанавливать другие порты или обновлять существующие.

Когда система обновляется, различные совместно используемые библиотеки, загружаемые модули и другие части системы замещаются более новыми версиями. Приложения, скомпонованные с более старыми версиями, могут перестать запускаться либо начнут функционировать неправильно.

Для получения дополнительной информации обращайтесь к разделу Руководства по FreeBSD.

6.7. Нужно ли мне перекомпилировать все порты каждый раз, когда я выполняю минорное обновление (без изменения старшей версии FreeBSD)?

В общем случае этого не требуется. Разработчики FreeBSD делают всё возможное для сохранения двоичной совместимости между всеми релизами в пределах одной старшей версии. Любые исключения из этого правила документируются в "Примечаниях к релизу", и там же даются советы, которых следует придерживаться.

6.8. Почему возможности /bin/sh так малы? Почему бы во FreeBSD не использовать bash или какой-либо другой командный процессор?

Многим требуется, чтобы разрабатываемые скрипты для командного процессора были переносимы между многими системами. Именно поэтому в POSIX® очень подробно описан командный процессор и набор утилит. Большинство скриптов пишутся на языке процессора Bourne shell (sh(1)), к тому же некоторые важные программные вызовы (make(1),

system(3), popen(3) и их аналоги на языках скриптов высокого уровня, таких как Perl или Tcl) предполагают для интерпретации команд использование именно Bourne shell. Так как Bourne shell используется столь широко и часто, то очень важно, чтобы он стартовал очень быстро, его поведение было строго регламентировано и при этом потребности в оперативной памяти были малы.

В имеющейся реализации мы приложили максимум усилий для воплощения в жизнь всех этих требований одновременно. Для того, чтобы сохранить `/bin/sh` небольшим по размеру, мы не включили многие из обычных возможностей, которые имеются в других командных процессорах. Для этого имеются в наличии командные процессоры, обладающие гораздо большими возможностями, такие как `bash`, `scsh`, tcsh(1) и `zsh`. Сравните использование памяти этими оболочками, посмотрев в колонки "VSZ" и "RSS" вывода команды `ps -u`.

Chapter 7. Конфигурирование ядра

7.1. Я хочу изменить настройки ядра. Это сложно?

Вовсе нет! Обратитесь к разделу о настройке ядра Руководства, который посвящён этому вопросу.

Новый файл kernel будет установлен в каталог /boot/kernel вместе со своими модулями, а старое ядро и его модули будут перемещены в каталог /boot/kernel.old. Если сделана ошибка в конфигурации, просто загрузите предыдущую версию ядра.

7.2. Почему моё ядро такое большое?

Ядра `GENERIC`, поставляемые в составе FreeBSD, компилируется в *режиме отладки*. Ядра, построенные в таком режиме, содержат отладочную информацию, которая хранится в отдельных файлах, используемых для отладки. Файлы для отладки размещаются в /usr/lib/debug/boot/kernel/. Заметьте, что снижения производительности при использовании отладочного ядра нет или оно незначительно, и отладочное ядро полезно иметь под рукой на случай аварийного завершения работы системы.

При нехватке дискового пространства существует несколько вариантов уменьшения размера /boot/kernel/ и /usr/lib/debug/.

Чтобы не устанавливать файлы с символьной информацией, убедитесь в наличии следующей строки в /etc/src.conf:

```
WITHOUT_KERNEL_SYMBOLS=yes
```

Для получения дополнительной информации смотрите страницу Справочника src.conf(5).

Если вы хотите совсем избежать создания файлов для отладки, убедитесь в выполнении следующих двух условий:

- В конфигурационном файле ядра нет такой строчки:

  ```
  makeoptions DEBUG=-g
  ```

- Вы не запускали утилиту config(8) с опцией `-g`.

В любой из вышеописанных ситуаций ядро будет построено с отладочным режимом.

Чтобы скомпилировать и установить только нужные модули, укажите их в /etc/make.conf:

```
MODULES_OVERRIDE= accf_http ipfw
```

Замените *accf_httpd ipfw* на нужный список модулей. Это уменьшит размер каталога с ядром, а также время сборки. Для получения дополнительной информации почитайте /usr/share/examples/etc/make.conf.

Для дальнейшего уменьшения размера также можно удалить ненужные устройства из ядра. Для получения дополнительной информации смотрите Я хочу изменить настройки ядра. Это сложно?.

Для вступления любого из этих действий в силу следуйте указаниям по сборке и установке нового ядра.

К вашему сведению, примерный размер ядра FreeBSD 11 amd64 (/boot/kernel/kernel) составляет 25 Мбайт.

7.3. Почему мне не удаётся откомпилировать ни один вариант ядра, даже GENERIC?

Есть несколько причин, приводящих к возникновению этой проблемы:

- Дерево исходных текстов отличается от того, что использовалось для построения работающей в данный момент системы. Перед обновлением прочитайте файл /usr/src/UPDATING, обращая особое внимание на раздел "COMMON ITEMS" в его конце.
- Выполнение команды `make buildkernel` не было успешно завершено. Корректное исполнение задачи `make buildkernel` зависит от файлов, полученных после выполнения `make buildworld`.
- Даже при построении FreeBSD-STABLE возможно, что дерево исходных текстов было загружено в тот момент, когда оно модифицировалось или было неработоспособно. Построение гарантируется только для релизов, хотя в большинстве случаев FreeBSD-STABLE строится без проблем. Попробуйте сгрузить дерево исходных текстов повторно и посмотрите, решилась ли проблема. Если с зеркалирующим сервером есть проблемы, попробуйте использовать другое зеркало.

7.4. Какой планировщик используется в работающей системе?

Название используемого планировщика доступно напрямую в виде значения sysctl-параметра `kern.sched.name`:

```
% sysctl kern.sched.name
kern.sched.name: ULE
```

7.5. Что такое kern.sched.quantum?

`kern.sched.quantum` определяет максимальное количество тактов, которое процесс может выполняться, не будучи прерванным.

Chapter 8. Диски, файловые системы и начальные загрузчики

8.1. Как добавить в систему новый диск?

Обратитесь к разделу Добавление дисков Руководства по FreeBSD.

8.2. Как перенести систему на большой новый диск?

Самый лучший способ заключается в переустановке операционной системы на новый диск с последующим переносом пользовательских данных. Это настоятельно рекомендуется при отслеживании ветки *-STABLE* в течение более одного релиза или при обновлении релиза вместо установки нового. Установите booteasy на оба диска с помощью boot0cfg(8) и выполняйте загрузку с любого из них, пока не будете довольны новой конфигурацией. Пропустите следующий абзац, чтобы перейти к вопросу переноса данных после этой операции.

Либо разбейте на разделы и разметьте новый диск с помощью sade(8) или gpart(8). Если диски отформатированы в MBR, то booteasy можно установить на оба диска с помощью boot0cfg(8) для того, чтобы иметь возможность выполнять загрузку как старой, так и новой системы после выполнения копирования.

После подготовки диска данные нельзя просто так взять и перенести. Используйте для этого инструменты, которые учитывают файлы устройств и системные флаги, например, dump(8). И хотя рекомендуется выполнять перенос данных в однопользовательском режиме, это не является обязательным условием.

Если на дисках стоит UFS, никогда не используйте ничего, кроме dump(8) и restore(8) для переноса корневой файловой системы. Эти команды также следует использовать при переносе отдельного раздела на другой пустой раздел. Последовательность шагов при использовании `dump` для переноса данных с раздела UFS на новый раздел:

1. выполните команду `newfs` над новым разделом.
2. командой `mount` смонтируйте его во временный каталог.
3. командой `cd` перейдите в этот каталог.
4. выполните команду `dump` над старым разделом, направив вывод в новый раздел.

Например, чтобы перенести корневую файловую систему на устройство /dev/ada1s1a с использованием каталога /mnt в качестве временной точки монтирования, наберите:

```
# newfs /dev/ad1as1a
# mount /dev/ad1as1a /mnt
```

```
# cd /mnt
# dump 0af - / | restore rf -
```

Переразбиение разделов с использованием `dump` требует несколько больше усилий. Для объединения раздела типа /var с его вышестоящим разделом, создайте новый раздел, достаточно большой для размещения их обоих, переместите вышестоящий раздел так, как это описано выше, а затем переместите нижестоящий раздел в пустой каталог, созданный при первом перемещении:

```
# newfs /dev/ada1s1a
# mount /dev/ad1as1a /mnt
# cd /mnt
# dump 0af - / | restore rf -
# cd var
# dump 0af - /var | restore rf -
```

Для отделения каталога от вышестоящего, скажем, для размещения /var в собственном разделе, которого не было, создайте оба раздела, затем смонтируйте нижестоящий раздел в подходящий каталог во временную точку монтирования, а затем переместите старый единый раздел:

```
# newfs /dev/ada1s1a
# newfs /dev/ada1s1d
# mount /dev/ada1s1a /mnt
# mkdir /mnt/var
# mount /dev/ada1s1d /mnt/var
# cd /mnt
# dump 0af - / | restore rf -
```

Для перемещения пользовательских данных также имеются программы cpio(1) и pax(1). Известно, что они теряют информацию о флагах файлов, так что используйте их с осторожностью.

8.3. На каких разделах можно без опаски использовать механизм Soft Updates? Я слышал, что использование Soft Updates на / могут приводить к проблемам. Что насчёт журналируемых Soft Updates?

Краткий ответ: обычно Soft Updates можно использовать без опаски на всех разделах.

Подробный ответ: Soft Updates имеют две характеристики, которые могут быть нежелательны на некоторых разделах. Во-первых, раздел с Soft Updates имеет небольшой шанс потери данных по время аварийного останова системы. Раздел не будет попорчен,

поскольку данные будет просто потеряны. Во-вторых, Soft Updates могут приводить к временной нехватке дискового пространства.

При использовании Soft Updates ядро может задерживать до тридцати секунд запись изменений на физический диск. При удалении большого файла он остается на диске, пока ядро не выполнит фактическое удаление. Это может привести к очень простой гонке. Предположим, удаляется один большой файл и тут же создаётся другой большой файл. Первый файл на самом деле ещё не удалён с диска, поэтому для второго файла на диске может не хватить места. Это приведёт к ошибке о том, что на разделе нет достаточного пространства, несмотря на то, что только что освободилось много места. Через пару секунд создание файла сработает, как и ожидалось.

Если система может аварийно остановиться после того, как ядро примет набор данных для записи на диск, но перед тем, как данные реально запишутся, то данные могут потеряться. Такой риск чрезвычайно мал, но в целом управляем.

Эти проблемы влияют на все разделы, использующие Soft Updates. Итак, что это означает для корневого раздела?

Жизненно важная информация на корневом разделе меняется очень редко. Если в системе произойдет сбой в период тридцатисекундного окна после выполнения такого изменения, возможно, что данные окажутся потерянными. Этот риск незначителен для большинства применений, но его нужно учитывать. Если система не может принять такой риск, не используйте Soft Updates с корневой файловой системой!

/ традиционно является одним из самых маленьких разделов. Если каталог /tmp размещён в / и у вас заполнен /tmp, то могут периодически возникать проблемы с дисковым пространством. Создание символической ссылки /tmp, указывающей на /var/tmp решит эту проблему.

В заключение, dump(8) не работает в режиме реального времени (-L) с файловой системой, для которой включены журналируемые Soft Updates (SU+J).

8.4. Можно ли смонтировать другие файловые системы?

FreeBSD поддерживает ряд других файловых систем.

UFS

Компакт-диски с файловой системой UFS могут быть смонтированы без всяких проблем. Монтирование файловых систем Digital UNIX или других систем, поддерживающих UFS, может быть более сложным, в зависимости от особенностей разбиения диска конкретной операционной системой.

ext2/ext3

FreeBSD поддерживает разделы `ext2fs`, `ext3fs` и `ext4fs`. За дополнительной информацией обратитесь к странице Справочника ext2fs(5).

NTFS

Поддержка NTFS, реализованная на базе FUSE, доступна в виде порта sysutils/fusefs-ntfs. Для получения более полной информации обратитесь к ntfs-3g(8).

FAT

Во FreeBSD имеется драйвер для работы с FAT в режиме чтения-записи. Для получения дополнительной информации обратитесь к странице справочника mount_msdosfs(8).

ZFS

FreeBSD включает в себя портированную из Sun™ реализацию файловой системы ZFS. В настоящее время рекомендуется использовать её только на платформе amd64 с достаточным объемом памяти. Для получения более полной информации обратитесь к странице Справочника zfs(8).

FreeBSD включает сетевую файловую систему NFS. В Коллекции портов FreeBSD имеется несколько приложений FUSE для поддержки многих других файловых систем.

8.5. Как смонтировать вторичный раздел DOS?

Вторичные разделы DOS находятся после *всех* первичных разделов. Например, если "E" является вторым разделом DOS на втором диске SCSI, то в каталоге /dev будет находиться файл устройства для "слайса 5". Чтобы его смонтировать, выполните следующую команду:

```
# mount -t msdosfs /dev/da1s5 /dos/e
```

8.6. Существует ли криптографическая файловая система для FreeBSD?

Да, gbde(8) и geli(8). Обратитесь к разделу Шифрование дисковых разделов Руководства по FreeBSD.

8.7. Как загрузить FreeBSD и Linux® с помощью GRUB?

Для загрузки FreeBSD с использованием GRUB добавьте следующие строки в /boot/grub/menu.lst или /boot/grub/grub.conf, в зависимости от используемого дистрибутива Linux®.

```
title FreeBSD 9.1
root (hd0,a)
kernel /boot/loader
```

Где *hd0,a* указывает на корневой раздел на первом диске. Чтобы указать номер слайса, напишите что-то вроде *(hd0,2,a)*. По умолчанию, если номер слайса не указан, GRUB ищет

первый слайс с разделом a.

8.8. Как загрузить FreeBSD и Linux® с помощью BootEasy?

Установите LILO в начало загрузочного раздела Linux®, а не в Master Boot Record. После этого загрузите LILO из BootEasy.

Это рекомендуется делать при одновременном использовании Windows® и Linux®, чтобы упростить восстановление работоспособности Linux® в случае переустановки Windows®.

8.9. Как сменить приглашение загрузчика с ??? на что-нибудь более значащее?

Этого нельзя сделать со стандартным загрузчиком, не переписав его. В категории sysutils Коллекции Портов есть ряд других менеджеров загрузки.

8.10. Как использовать устройство для чтения сменных дисков?

Если у вас уже есть файловая система на устройстве, то используйте такую команду:

```
# mount -t msdosfs /dev/da0s1 /mnt
```

Если это устройство будет использоваться только на системах FreeBSD, то разбейте его на разделы UFS или ZFS. Это обеспечит поддержку длинных имён файлов, увеличение производительности и надёжность. Если устройство будет использоваться с другими операционными системами, то лучше сделать более совместимый выбор, например, msdosfs.

```
# dd if=/dev/zero of=/dev/da0 count=2
# gpart create -s GPT /dev/da0
# gpart add -t freebsd-ufs /dev/da0
```

Наконец, остаётся создать новую файловую систему:

```
# newfs /dev/da0p1
```

и смонтировать её:

```
# mount /dev/da0s1 /mnt
```

Хорошо ещё добавить строку в файл /etc/fstab (прочтите справку по fstab(5)), чтобы в будущем можно было просто давать команду `mount /mnt`:

```
/dev/da0p1 /mnt ufs rw,noauto 0 0
```

8.11. Почему при монтировании компакт-диска выдаётся сообщение Incorrect super block?

Необходимо указать тип монтируемого устройства. Это описано в разделе Руководства Использование CD с данными.

8.12. При монтировании компакт-диска выдаётся сообщение Device not configured.

Обычно это означает, что в приводе нет компакт-диска либо устройство не обнаружено на шине. Обратитесь к разделу extref:https://docs.freebsd.org/ru/books/handbook/disks/[Использование CD с данными, mounting-cd] в Руководстве, где подробно обсуждается этот вопрос.

8.13. Почему при монтировании CD-ROM во FreeBSD все неанглийские символы в именах файлов отображаются как вопросительные знаки?

Скорее всего, на компакт-диске для хранения информации о файлах и каталогах используется расширение "Joliet". Это описано в разделе Руководства об использовании CD с данными.

8.14. Записанный во FreeBSD CD не читается ни в какой другой операционной системой. Почему?

Это означает, что на CD был записан непосредственно необработанный файл без создания файловой системы ISO 9660. Прочтите раздел Руководства об использовании CD с данными.

8.15. Как создать образ CD с данными?

Это описано в разделе Руководства о extref:https://docs.freebsd.org/ru/books/handbook/disks/[записи данных на файловую систему ISO, mkisofs]. Более полную информацию о работе с компакт-дисками можно найти в разделе о extref:https://docs.freebsd.org/ru/books/handbook/disks/[создании компакт-дисков, creating-cds] в главе Руководства об устройствах хранения данных.

8.16. Почему я не могу смонтировать аудио CD?

Попытка смонтировать аудио CD приведёт к сообщению об ошибке вида `cd9660: /dev/acd0c: Invalid argument`. Причина этого заключается в том, что команда `mount` работает только с файловыми системами. На аудио CD файловых систем нет; они содержат только данные. Используйте вместо этого программу, которая умеет читать аудио CD, например, порт audio/xmcd.

8.17. Как выполнить mount для многосеансового CD?

По умолчанию mount(8) будет пытаться смонтировать последнюю дорожку (сеанс) CD с данными. Для загрузки более раннего сеанса используйте параметр командной строки `-s`. За конкретными примерами обращайтесь к странице Справочника mount_cd9660(8).

8.18. Как разрешить обычным пользователям монтировать компакт-диски, DVD, USB-диски и другие сменные носители?

Как пользователь `root`, установите системную переменную `vfs.usermount` в значение `1`.

```
# sysctl vfs.usermount=1
```

Чтобы это срабатывало во время загрузки системы, добавьте строчку `vfs.usermount=1` в файл /etc/sysctl.conf.

Пользователи могут монтировать только те устройства, к которым у них имеется доступ на чтение. Чтобы разрешить пользователям монтировать устройство, должны быть заданы разрешения в /etc/devfs.conf.

Например, чтобы разрешить пользователям монтировать первое устройство USB, добавьте такую строчку:

```
# Allow all users to mount a USB drive.
        own       /dev/da0       root:operator
        perm      /dev/da0       0666
```

Теперь все пользователи могут монтировать устройства с правами доступа на чтение в собственные каталоги:

```
% mkdir ~/my-mount-point
% mount -t msdosfs /dev/da0 ~/my-mount-point
```

Размонтирование устройства осуществляется просто:

```
% umount ~/my-mount-point
```

Использование `vfs.usermount`, однако, имеет некоторые негативные стороны, связанные с вопросами безопасности. Более правильным способом работы с носителями в формате MS-DOS® является использование пакета emulators/mtools из Коллекции Портов.

Имя устройства, использованное в предыдущих примерах, должно быть изменено в соответствии с конфигурацией.

8.19. Команды du и df показывают разный объём доступного дискового пространства. Что происходит?

Это связано с тем, как эти команды на самом деле работают. `du` проходит по дереву каталогов, подсчитывая, насколько большой объём занимает каждый файл, и выдает общий объем. `df` просто запрашивает файловую систему об оставшемся объеме. Это выглядит как одно и то же, однако файл без записи в каталоге затронет `df`, но не повлияет на `du`.

Когда программа использует файл и этот файл удаляется, то на самом деле он не удаляется из файловой системы, пока программа не прекратит его использовать. Однако файл тут же удаляется из списка каталога. В качестве примера можно представить файл, размер которого достаточен, чтобы повлиять на результаты работы команд `du` и `df`. Любой файл, просматриваемый при помощи команды `more`, может быть удалён без выдачи сообщений об ошибке. Запись о файле просто удаляется из каталога, так что другие программы или пользователи не смогут к нему обратиться. Тем не менее, `du` покажет, что файл исчез, поскольку она просматривает дерево каталогов, а файла там нет. `df` показывает, что файл всё ещё здесь, так как файловая система знает, что `more` всё ещё использует это пространство. Как только закончится работа с `more`, команды `du` и `df` придут в соответствие.

Такая ситуация часто встречается на Web-серверах. Многие устанавливают Web-сервер на FreeBSD и забывают выполнять ротацию файлов протоколов. Журнал доступа заполняет /var. Новый администратор удаляет файл, но система всё ещё сообщает о том, что раздел заполнен. Остановка и перезапуск программы Web-сервера освободит файл, позволяя системе освободить дисковое пространство. Для предотвращения подобной ситуации настройте newsyslog(8).

Заметьте, что подсистема отложенных обновлений (Soft Updates) может задерживать освобождение дискового пространства, и может потребоваться до 30 секунд, чтобы изменения стали заметны.

8.20. Как добавить дополнительную виртуальную память?

В этом разделе Руководства описывается, как это сделать.

8.21. Почему FreeBSD считает, что размер моего диска меньше, чем заявляет его производитель?

Производители дисков считают гигабайт равным миллиарду байт, а FreeBSD приравнивает его к 1073741824 байт. Это объясняет, почему, к примеру, в сообщениях при загрузке FreeBSD указывается, что ёмкость диска, объём которого должен быть равным 80 Гбайт, составляет 76319 Мбайт.

Заметьте также, что FreeBSD будет (по умолчанию) резервировать 8% ёмкости диска.

8.22. Почему возможно заполнение раздела больше чем на 100%?

Часть каждого раздела UFS (по умолчанию 8%) зарезервировано для использования операционной системой и пользователем `root`. Утилита df(1) не учитывает это при подсчёте значения в колонке `Capacity`, так что оно может превышать 100%. Обратите внимание, что колонка `Blocks` всегда больше, чем сумма значений в колонках `Used` и `Avail`, обычно на 8%.

Для получения более подробной информации обратитесь к описанию опции `-m` в справке по tunefs(8).

Chapter 9. ZFS

9.1. Какой минимальный объём ОЗУ требуется для ZFS?

Для комфортного использования требуется 4 Гбайт ОЗУ, но конкретная нагрузка может сильно различаться.

9.2. Что такое ZIL и когда он используется?

ZIL (ZFS intent log) представляет собой журнал операций записи, используемый для реализации логики подтверждения записи на диск в стандарте posix, которая работает при сбоях. При нормальной работе отдельные операции записи объединяются в группу транзакций и записываются на диск при заполнении группы ("Transaction Group Commit"). Однако системные вызовы типа fsync(2) требуют подтверждения записи данных на надёжное устройство перед возвратом управления. ZIL нужен для работы с операциями записи, которые были подтверждены как выполненные, но в рамках транзакции на диск ещё не записаны. Группы транзакций помечаются меткой времени. В случае сбоя выполняется поиск последней корректной метки и из ZIL извлекаются недостающие данные.

9.3. Нужен ли мне SSD для ZIL?

По умолчанию ZFS хранит ZIL в пуле со всеми данными. Если приложение интенсивно выполняет операции записи, то размещение ZIL на отдельном устройстве, обладающем высокими показателями скорости синхронной последовательной записи, может улучшить общую производительность системы. При другом характере нагрузки SSD вряд ли сможет что-то сильно улучшить.

9.4. Что такое L2ARC?

L2ARC - это кэш на чтение, хранимый на быстром устройстве, таком как SSD. Этот кэш не сохраняется между перезагрузками. Следует заметить, что ОЗУ выступает как кэширующее устройство первого уровня, а L2ARC необходим только при нехватке оперативной памяти.

L2ARC требует пространства в ARC для его индексации. Таким образом, рабочий набор, который идеально помещается в ARC, не будет там помещаться при использовании L2ARC, поскольку часть ARC будет отведена под индекс L2ARC с вытеснением части рабочего набора в L2ARC, более медленный, чем ОЗУ.

9.5. Целесообразно ли включение дедупликации?

В общем случае нет.

Дедупликация занимает значительный объём ОЗУ и может увеличить задержки операций

ввода/вывода. Если только данные не являются сильно дублированными (сильно дублированными являются образы виртуальных машин или резервные копии пользовательских данных), то дедупликация может принести больше вреда, чем пользы. Другое соображение касается невозможности отменить статус дедупликации. Если данные записаны при включённой дедупликации, её выключение не приведёт к обратной репликации объединённых блоков до момента их модификации.

Дедупликация также может приводить к некоторым неожиданным ситуациям. В частности, удаление файлов может сильно замедлиться.

9.6. Я не могу создать или удалить файлы на пуле ZFS. Как я могу это исправить?

Такое может произойти при 100% заполненности пула. ZFS требуется свободное место на диске для записи метаданных транзакций. Для восстановления работоспособного состояния пула обрежьте файл перед его удалением.

```
% truncate -s 0 unimportant-file
```

Обрезка файла работает по той причине, что новая транзакция при этом не запускается, но вместо этого создаются новые свободные блоки.

На системах с дополнительными настройками наборов данных ZFS, такими, как дедупликация, свободное место может не быть доступно сразу.

9.7. Поддерживается ли TRIM в ZFS для твердотельных накопителей?

ZFS во FreeBSD 12.3 и 12.4: TRIM включен по умолчанию. Для отключения TRIM: добавьте строку ниже в файл /etc/sysctl.conf, затем перезапустите систему.

```
vfs.zfs.trim.enabled=0
```

OpenZFS во FreeBSD 13.0 и более старших версиях: обратитесь к zpool-trim(8) и изучите параметр `autotrim` в zpoolprops(7).

Chapter 10. Системное администрирование

10.1. Где расположены файлы конфигурации системы?

Основным конфигурационным файлом является /etc/defaults/rc.conf, который описан в rc.conf(5). Этот файл используют скрипты запуска системы, такие как /etc/rc и /etc/rc.d, которые описаны в rc(8). *Не редактируйте этот файл!* Вместо изменения параметра в /etc/defaults/rc.conf скопируйте соответствующую строку в /etc/rc.conf и поменяйте значение там.

К примеру, для запуска sshd(8), поставляемого с системой сервера OpenSSH:

```
# echo 'sshd_enable="YES"' >> /etc/rc.conf
```

Альтернативным способом является использование sysrc(8) для корректировки /etc/rc.conf:

```
# sysrc sshd_enable="YES"
```

Для запуска локальных сервисов поместите соответствующие скрипты в каталог /usr/local/etc/rc.d. У этих скриптов должен быть выставлен бит выполнимости, по умолчанию используются права доступа `555`.

10.2. Как проще всего добавить пользователя?

Используйте команду adduser(8) или pw(8) в случае выполнения более сложных операций.

Чтобы удалить пользователя, используйте команду rmuser(8) или, если это будет необходимо, pw(8).

10.3. Почему после редактирования моего файла crontab я получаю сообщения вида root: not found?

Обычно это случается при редактировании системного файла crontab.

Это неправильный подход, потому что системный crontab имеет формат, отличный от пользовательских crontab. Системный crontab имеет дополнительное поле, указывающее, под каким пользователем запускать команду. cron(8) полагает, что имя пользователя является первым словом в команде на выполнение. Поскольку такой команды не существует, отображается это сообщение об ошибке.

Чтобы удалить лишний неправильный crontab:

```
# crontab -r
```

10.4. Команда su выдаёт сообщение you are not in the correct group to su root, когда я пытаюсь сменить привилегии на root.

Это особенность работы системы защиты. Для того, чтобы сменить пользовательский идентификатор с помощью `su` на пользователя `root` или любого другого с привилегиями суперпользователя, учётная запись должна являться членом группы `wheel`. Если бы этого не было, то любой, имеющий доступ к системе и узнавший пароль пользователя `root`, смог бы получить в системе уровень доступа суперпользователя.

Чтобы разрешить кому-либо менять привилегии на `root`, включите его в группу `wheel` с помощью `pw`.

```
# pw groupmod wheel -m lisa
```

В примере выше пользователь `lisa` будет добавлен в группу `wheel`.

10.5. Я сделал ошибку в файле rc.conf или в каком-то другом файле начальной загрузки, и теперь не могу его отредактировать из-за того, что файловая система находится в режиме только для чтения. Что мне делать?

Перезапустите систему, используя в приглашении загрузчика команду `boot -s` для входа в однопользовательский режим. При получении приглашения на ввод полного пути до командного процессора нажмите `Enter`, а затем выполните команду `mount -urw /` для повторного монтирования корневой файловой системы в режиме чтения/записи. Вам может также потребоваться выполнить команду `mount -a -t ufs` для монтирования файловой системы, в которой расположен ваш любимый текстовый редактор. Если редактор расположен на сетевой файловой системе, выполните сетевые настройки вручную до монтирования сетевой файловой системы, либо воспользуйтесь редактором, находящимся в локальной файловой системе, таким, как ed(1).

Чтобы использовать полноэкранный редактор, такой как vi(1) или emacs(1), выполните команду `export TERM=xterm`, чтобы эти редакторы смогли получить корректную информацию из базы данных termcap(5).

После выполнения этих шагов отредактируйте файл /etc/rc.conf для исправления ошибки. Сообщение об ошибке, выводимое сразу же после сообщений при загрузке ядра, должно указать на номер строки в файле, которая содержит ошибку.

10.6. Почему у меня возникают проблемы с установкой принтера?

Обратитесь к соответствующему разделу Руководства, посвящённому печати, за советами по разрешению проблем.

10.7. Как я могу откорректировать раскладку клавиатуры для моей системы?

Обратитесь к разделу Руководства, посвящённому использованию локализации, а именно к части, описывающей настройку консоли.

10.8. Почему не получается заставить работать дисковые квоты?

1. Возможно, что ядро не сконфигурировано должным образом для работы с квотами. В этом случае добавьте следующую строчку в конфигурационный файл ядра и пересоберите ядро:

   ```
   options QUOTA
   ```

   Прочтите главу Руководства по квотам, quotas для получения полной информации.

2. Не включайте квотирование на разделе /.
3. Помещайте файл с квотами в ту файловую систему, которую он обслуживает:

Файловая система	Файл квот
/usr	/usr/admin/quotas
/home	/home/admin/quotas
...	...

10.9. Поддерживает ли FreeBSD вызовы IPC из System V?

Да, во FreeBSD в ядро GENERIC включена поддержка IPC в стиле System V, в том числе совместно используемой памяти, сообщений и семафоров. В нестандартном ядре поддержка может быть включена посредством загрузки модулей ядра sysvshm.ko, sysvsem.ko и sysvmsg.ko или добавлением в конфигурационный файл ядра следующих строк:

```
options    SYSVSHM    # enable shared memory
options    SYSVSEM    # enable for semaphores
```

```
options    SYSVMSG    # enable for messaging
```

Перекомпилируйте и переустановите ядро.

10.10. Какое другое программное обеспечение для почтового сервера можно использовать вместо Sendmail?

Сервер Sendmail является программным обеспечением для работы почтового сервера во FreeBSD, используемым по умолчанию, но его можно заменить другим MTA, установленным из Коллекции Портов. В дереве портов имеется mail/exim, mail/postfix и mail/qmail. Проверьте архивы списков рассылки на предмет обсуждения достоинств и недостатков имеющихся MTA.

10.11. Я забыл пароль пользователя root! Что делать?

Без паники! Перезапустите систему, наберите `boot -s` в приглашении `Boot:` для входа в однопользовательский режим. На вопрос об используемой оболочке нажмите `Enter` для получения # в качестве приглашения. Выполните команду `mount -urw /`, чтобы перемонтировать корневую файловую систему в режиме чтения/записи, после чего выполните команду `mount -a` для монтирования всех файловых систем. Запустите команду `passwd root`, чтобы сменить пароль пользователя `root`, а затем exit(1) для продолжения процесса загрузки.

> Если при входе в однопользовательский режим предлагается ввести пароль пользователя `root`, это означает, что консоль была помечена как `insecure` в /etc/ttys. В этом случае потребуется загрузиться с установочного диска FreeBSD, выбрать Live CD или Shell в начале процесса установки и выполнить указанные выше команды. В этом случае смонтируйте нужный раздел и выполните туда chroot. Например, замените команду `mount -urw /` на `mount /dev/ada0p1 /mnt; chroot /mnt` для системы, расположенной на *ada0p1*.

> Если корневой раздел не получается смонтировать в однопользовательском режиме, то возможно, что разделы являются зашифрованными, и смонтировать их без ключей доступа не представляется возможным. За дополнительной информацией обратитесь к разделу Руководства, посвящённому шифрованию дисков во FreeBSD.

10.12. Как запретить перезагрузку по нажатию Control Alt Delete?

При использовании стандартного драйвера консоли vt(4) этого можно добиться, задав следующий системный параметр sysctl(8):

```
# sysctl kern.vt.kbd_reboot=0
```

10.13. Как преобразовать текстовые файлы DOS в формат UNIX®?

Воспользуйтесь следующей командой perl(1):

```
% perl -i.bak -npe 's/\r\n/\n/g' file(s)
```

где *file(s)* - это один или несколько файлов для обработки. Преобразование делается в том же самом файле, оригинальные файлы сохраняются с расширением .bak.

Либо используйте tr(1):

```
% tr -d '\r' < dos-text-file > unix-file
```

где *dos-text-file* - это имя файла, содержащего текст DOS, а в файл *unix-file* будет помещён уже преобразованный текст. Этот способ может работать гораздо быстрее, чем при использовании `perl`.

Еще один способ отформатировать тестовые файлы DOS состоит в использовании converters/dosunix из Коллекции Портов. Для получения дополнительной информации ознакомьтесь с документацией порта.

10.14. Как перечитать содержимое /etc/rc.conf и перестартовать /etc/rc без перезагрузки системы?

Перейдите в однопользовательский режим, а затем возвратитесь обратно в многопользовательский.

```
# shutdown now
# return
# exit
```

10.15. Я пытался обновить мою систему до последней -STABLE, а получил -BETAx, -RC или -PRERELEASE! Что происходит?

Краткий ответ: это же просто название. *RC* означает "Release Candidate". Это значит, что вскоре произойдет выход релиза. Во FreeBSD появление *-PRERELEASE*, как правило, равнозначно прекращению внесения изменений в код системы перед появлением релиза. (Для некоторых релизом метка *-BETA* использовалась точно так же, как и *-PRERELEASE*.)

Подробный ответ: во FreeBSD релизы выпускаются из одного из двух мест. Крупные релизы, точка-ноль, такие, как 9.0-RELEASE и 10.0-RELEASE, ответвляются от основного потока разработки, более известного как -CURRENT. Мелкие релизы, такие, как 6.3-RELEASE или 5.2-RELEASE, являлись снэпшотами активной ветки -STABLE. Начиная с 4.3-RELEASE, каждый релиз также имеет свою ветвь, которой могут следовать те, кому необходим сверхконсервативный метод обновления (как правило, внесение только тех исправлений, которые касаются вопросов обеспечения безопасности).

Когда делается релиз, то ветвь, из которой он выпускается, подвергается некоторой подготовке. Частью этого процесса является замораживание кода. Когда инициируется замораживание кода, то имя ветки изменяется для того, чтобы отразить факт близости релиза. Например, если ветка называлась 6.2-STABLE, то её имя будет изменено на 6.3-PRERELEASE, чтобы обозначить момент прекращения внесения изменений в код системы и период дополнительного тестирования перед выходом релиза. В это время исправления ошибок могут быть внесены в код системы для того, чтобы быть включенными в релиз. Когда исходный код подготовлен к выпуску релиза, имя будет изменено на 6.3-RC для обозначения того, что релиз будет сделан, скорее всего, именно из этого кода. Когда код находится на этапе RC, в нём могут исправляться только самые критичные ошибки. Как только релиз (в нашем примере 6.3-RELEASE) и ветка релиза будут созданы, ветвь будет переименована в 6.3-STABLE.

Для получения дополнительной информации о номерах версий и различных ветках Git обратитесь к статье о выпуске релизов.

10.16. Я попытался установить новое ядро, однако утилита chflags(1) не сработала. Как это обойти?

Краткий ответ: Режим безопасности имеет значение больше нуля. Для установки ядра перезагрузите машину и войдите в однопользовательский режим.

Подробный ответ: FreeBSD запрещает менять системные флаги при работе на уровнях безопасности, превышающих 0. Чтобы проверить текущий уровень безопасности:

```
# sysctl kern.securelevel
```

Уровень безопасности нельзя понизить в многопользовательском режиме, поэтому для

установки ядра загрузитесь в однопользовательский режим, или измените уровень безопасности в /etc/rc.conf, а затем выполните перезагрузку. Обратитесь к странице Справочника по init(8) за подробной информацией о `securelevel` и посмотрите /etc/defaults/rc.conf и справочную страницу по rc.conf(5) для выяснения подробностей о файле rc.conf.

10.17. Не получается изменить системное время больше чем на одну секунду! Как это обойти?

Краткий ответ: Система работает на уровне безопасности со значением выше 1. Для смены даты перезагрузите машину и войдите в однопользовательский режим.

Подробный ответ: FreeBSD запрещает менять системное время больше чем на одну секунду на уровне безопасности выше 1. Чтобы определить уровень безопасности:

```
# sysctl kern.securelevel
```

Уровень безопасности нельзя понизить в многопользовательском режиме. Для изменения даты перезагрузите систему в однопользовательский режим, либо измените уровень безопасности в /etc/rc.conf, а затем выполните перезагрузку. Обратитесь к странице Справочника по init(8) за подробной информацией о `securelevel` и посмотрите /etc/defaults/rc.conf и справочную страницу по rc.conf(5) для выяснения подробностей о файле rc.conf.

10.18. Для чего rpc.statd использует 256 Мбайт оперативной памяти?

Нет, это не ошибка утечки памяти и он не использует 256 Мбайт оперативной памяти. Для удобства `rpc.statd` отображает большой объём памяти в своё адресное пространство. Здесь нет ничего ужасно неправильного с технической точки зрения; просто это сбивает с толку программы вроде top(1) и ps(1).

rpc.statd(8) отображает свой статусный файл (/var/db/statd.status) в сообственное адресное пространство; для того, чтобы избежать последующих беспокойств о повторном отображении статусного файла в память, когда его размер нужно будет увеличить, под этот файл выделяется щедро заданный объём.

10.19. Почему я не могу снять с файла флаг schg?

Система работает на уровне защиты выше нуля. Понизьте уровень защиты и попробуйте ещё раз. Для получения более подробной информации обратитесь к разделу FAQ об уровне защиты и справочной странице init(8).

10.20. Что такое vnlru?

`vnlru` сбрасывает и освобождает vnode, когда система достигает ограничения по параметру `kern.maxvnodes`. Этот поток ядра в основном работает вхолостую и активируется только при наличии огромного объема ОЗУ и обращении к десяткам тысяч файлов небольшого размера.

10.21. Что означают различные состояния памяти, показываемые утилитой top?

- `Active`: по статистике страницы недавно использовались.
- `Inactive`: по статистике страницы недавно не использовались.
- `Laundry`: страницы, которые, согласно статистике, давно не использовались, но содержат данные, то есть их содержимое необходимо сохранить перед тем, как повторно использовать.
- `Free`: страницы, не содержащие данных, и которые можно повторно использовать прямо сейчас.
- `Wired`: страницы, зафиксированные в памяти, обычно для использования ядром, а также иногда для специального использования процессами.

Страницы чаще всего записываются на диск (типа синхронизации VM), когда они находятся в состоянии Laundry, однако синхронизироваться могут также и страницы в состоянии Active и Inactive. Это зависит от возможности отслеживания центральным процессором бита 'модифицированности', и в некоторых ситуациях это может быть лучше для блока синхронизируемых страниц VM, вне зависимости от их принадлежности к той или иной очереди. В большинстве встречающихся ситуаций лучше всего представлять очередь Laundry как очередь условно неиспользуемых страниц, которые могут быть, а могут и не быть в процессе записи на диск. Очередь Inactive содержит как чистые, так и грязные страницы; чистые страницы ближе к голове очереди потребляются сразу же для восполнения нехватки свободных страниц памяти, а грязные страницы перемещаются в очередь Laundry для последующей обработки.

Есть ещё некоторые другие флаги (например, флаг занятости или счётчик занятости), которые могут влиять на описанные правила.

10.22. Сколько свободной памяти доступно?

Есть пара определений "свободной памяти". В наиболее распространённом случае это объём памяти, доступной к немедленному потреблению без высвобождения уже используемой. Этот объём равен объёму очереди свободных страниц вместе с некоторыми другими зарезервированными страницами. Этот размер доступен в виде sysctl(8) -переменной `vm.stats.vm.v_free_count`, которая показывается, например, утилитой top(1). Во втором случае "свободная память" обозначает общий объём виртуальной памяти, доступной пользовательским процессам, который зависит от объёма раздела подкачки и доступной к использованию оперативной памяти. Другие определения "свободной памяти"

также возможны, но они достаточно бесполезны, ведь в любом случае важно сохранять низкий уровень подкачки и избегать исчерпания раздела подкачки.

10.23. Что такое /var/empty?

/var/empty представляет собой каталог, который используется в программе sshd(8) при выполнении разделения полномочий. Каталог /var/empty пуст, его владельцем является `root`, и на нём установлен флаг `schg`. Этот каталог не должен удаляться.

10.24. Я поменял /etc/newsyslog.conf. Как проверить правильность изменений?

Чтобы посмотреть, что будет делать newsyslog(8), используйте следующую команду:

```
% newsyslog -nrvv
```

10.25. Как поправить часовой пояс?

Используйте tzsetup(8).

Chapter 11. X Window System и виртуальные консоли

11.1. Что такое X Window System?

X Window System (обычно `X11`) является наиболее общедоступной оконной системой, которая может работать на UNIX® и UNIX®-подобных системах, в том числе и во FreeBSD. Разработкой стандартов на используемый X-протокол занимается организация The X.Org Foundation, с текущей эталонной реализацией version 11 release 7.7, поэтому название часто сокращается до `X11`.

Для разных архитектур и операционных систем существует множество реализаций этой системы. Реализацию кода для серверной части называют `X-сервером`.

11.2. Я хочу запустить Xorg, как это сделать?

Для установки Xorg выполните одно из действий:

Используйте мета-порт x11/xorg, который выполняет построение и установку всех компонентов Xorg.

Используйте x11/xorg-minimal, который выполняет построение и установку только необходимых компонентов Xorg.

Установите Xorg из пакетов FreeBSD.

```
# pkg install xorg
```

После установки Xorg следуйте указаниям в разделе Конфигурация X11 Руководства по FreeBSD.

11.3. Я попытался запустить X, но получил сообщение 'No devices detected' после ввода команды startx. Что мне теперь делать?

Вероятно, в системе установлен повышенный уровень безопасности (`securelevel`). При повышенном уровне защиты систему X запустить невозможно, потому что X требуются права на операции записи в устройство io(4). Дополнительная информация находится на странице Справочника init(8).

Существует два решения проблемы: возвратить нулевое значение `securelevel` или запускать xdm(8) (или любой другой менеджер дисплеев) во время загрузки, до того, как будет повышено значение `securelevel`.

Обратитесь к Как запустить XDM во время загрузки? для получения более полной информации о запуске xdm(8) во время загрузки.

11.4. Почему моя мышь не работает с X?

При использовании стандартного драйвера консоли vt(4) во FreeBSD можно включить поддержку указателя мыши во всех виртуальных экранах. Во избежание конфликтов с X, драйвер vt(4) поддерживает виртуальное устройство /dev/sysmouse. Все события от реального устройства мыши пишутся в устройство sysmouse(4) через moused(8). Чтобы использовать мышь на одной и более виртуальных консолях, и при этом продолжать использовать X, посмотрите Можно ли использовать мышь вне X Window? и настройте moused(8).

Затем отредактируйте /etc/X11/xorg.conf, чтобы в нём были следующие строки:

```
Section "InputDevice"
   Option     "Protocol" "SysMouse"
   Option     "Device" "/dev/sysmouse"
.....
```

Начиная с версии Xorg 7.4 раздел `InputDevice` в файле xorg.conf игнорируется, и вместо него используется механизм автоматически определяемых устройств. Чтобы восстановить прежнее поведение, добавьте в раздел `ServerLayout` или `ServerFlags` такую строку:

```
Option "AutoAddDevices" "false"
```

Некоторые предпочитают использовать в X устройство /dev/mouse. Чтобы оно работало, файл устройства /dev/mouse должен являться ссылкой на /dev/sysmouse (посмотрите справку по sysmouse(4)). Это можно сделать, добавив следующую строку в /etc/devfs.conf (посмотрите справку по devfs.conf(5)):

```
link    sysmouse    mouse
```

Ссылка может быть создана путем перезапуска devfs(5) с использованием следующей команды (из под пользователя `root`):

```
# service devfs restart
```

11.5. У моей мыши есть колёсико. Могу ли я его использовать при работе в X?

Да, если X настроена для использования 5-кнопочной мыши. Для этого добавьте строчки `Buttons 5` и `ZAxisMapping 4 5` в раздел "InputDevice" файла /etc/X11/xorg.conf как показано в

этом примере:

```
Section "InputDevice"
   Identifier      "Mouse1"
   Driver     "mouse"
   Option     "Protocol" "auto"
   Option     "Device" "/dev/sysmouse"
   Option     "Buttons" "5"
   Option     "ZAxisMapping" "4 5"
EndSection
```

Использование мыши может быть активировано в Emacs путём добавления в ~/.emacs следующих строк:

```
;; wheel mouse
(global-set-key [mouse-4] 'scroll-down)
(global-set-key [mouse-5] 'scroll-up)
```

11.6. Как заставить работать тачпад Synaptics в X?

Для его работы понадобится настроить некоторые вещи.

Чтобы использовать драйвер synaptics из Xorg, для начала удалите строку moused_enable из rc.conf.

Для включения synaptics добавьте следующую строку в /boot/loader.conf:

```
hw.psm.synaptics_support="1"
```

Добавьте следующее в /etc/X11/xorg.conf:

```
Section "InputDevice"
Identifier  "Touchpad0"
Driver      "synaptics"
Option      "Protocol" "psm"
Option      "Device" "/dev/psm0"
EndSection
```

Добавьте в раздел "ServerLayout" вот что:

```
InputDevice    "Touchpad0" "SendCoreEvents"
```

11.7. Как использовать удалённые X-дисплеи?

Из соображений обеспечения информационной безопасности открывать удалённые окна на машине по умолчанию запрещено.

Для включения этой возможности запустите X с аргументом `-listen_tcp`:

```
% startx -listen_tcp
```

11.8. Что такое виртуальные консоли и как изменить их количество?

Виртуальные консоли предоставляют несколько одновременных сеансов работы с той же самой машиной без установки какой бы то ни было сети или запуска X.

При запуске системы после вывода сообщений этапа загрузки на консоль выдаётся приглашение для входа в систему. Введите своё имя и пароль, чтобы начать работу на первой виртуальной консоли.

Чтобы запустить ещё один сеанс, скажем, чтобы заглянуть в документацию по программе или для чтения электронной почты во время ожидания завершения передачи данных по FTP, нажмите `F2`, удерживая `Alt`. Это отобразит приглашение на второй виртуальной консоли. Чтобы вернуться к первоначальному сеансу, нажмите `Alt` + `F1`.

По умолчанию во FreeBSD задействованы восемь виртуальных консолей, а комбинации клавиш `Alt` + `F1`, `Alt` + `F2`, `Alt` + `F3` и далее служат для переключения между ними.

Чтобы увеличить количество виртуальных консолей, отредактируйте /etc/ttys (смотрите страницу Справочника ttys(5)), добавив туда записи для терминалов с именами от ttyv8 до ttyvc после комментария про "Virtual terminals":

```
# Edit the existing entry for ttyv8 in /etc/ttys and change
# "off" to "on".
ttyv8   "/usr/libexec/getty Pc"  xterm  on secure
ttyv9   "/usr/libexec/getty Pc"  xterm  on secure
ttyva   "/usr/libexec/getty Pc"  xterm  on secure
ttyvb   "/usr/libexec/getty Pc"  xterm  on secure
```

Чем больше виртуальных терминалов, тем больше ресурсов используется. Это может привести к проблемам на системах с 8 Мбайт ОЗУ или меньше. Подумайте о смене статуса консолей с `secure` на `insecure`.

Чтобы запустить сервер X, нужно зарезервировать под него хотя бы один виртуальный терминал со значением `off`. Это означает, что под виртуальные консоли можно отвести только одиннадцать функциональных клавиш, и ещё одна остаётся за X-сервером.

Например, чтобы запустить X и 11 виртуальных консолей, нужно настроить двенадцатый виртуальный терминал:

```
ttyvb   "/usr/libexec/getty Pc"  xterm  off secure
```

Самым простым способом активировать виртуальные консоли является перезагрузка.

11.9. Как осуществляется доступ к виртуальным консолям из X?

Используйте комбинацию клавиш `Ctrl` + `Alt` + `Fn` для переключения обратно в виртуальную консоль. Нажмите `Ctrl` + `Alt` + `F1`, чтобы вернуться на первую виртуальную консоль.

После того, как вы оказались в текстовой консоли, используйте комбинации `Alt` + `Fn` для переключения между ними.

Чтобы вернуться в сеанс работы X, переключитесь в виртуальную консоль, на которой запущена X Window. Если X была запущена из командной строки с использованием команды `startx`, то сеанс работы X будет привязан к следующей неиспользуемой виртуальной консоли, а не к той текстовой консоли, с которой она была запущен. В случае восьми активных виртуальных терминалов X будет работать на девятом, поэтому используйте комбинацию `Alt` + `F9`.

11.10. Как запустить XDM во время загрузки?

Есть две философские школы, проповедующие различные методы запуска xdm(8). Последователи одного течения запускают `xdm` из /etc/ttys (посмотрите ttys(5)), используя приводимый там пример, тогда как другие устанавливают значение параметра `xdm_enable=yes` в /etc/rc.conf. Оба метода равноправны, и один из них может работать в ситуациях, с которыми не справляется другой, и наоборот. В обоих случаях результат один и тот же: X выводит графическое приглашение для входа в систему.

Плюсом метода с использованием ttys(5) является документирование того, на каком vty будет запущен X и то, что ответственность за перезапуск X-сервера при завершении сеанса работы лежит на процессе init(8). Метод с использованием rc(8) позволяет просто прекратить работу xdm командой `kill xdm`, если при запуске X возникли какие-нибудь проблемы.

При использовании метода с rc(8) значение `xdm_tty` (которое по умолчанию определено как `ttyv8`) может быть задано в файле /etc/rc.conf для указания виртуального терминала, на котором запускается xdm(8).

11.11. При запуске xconsole выдаётся сообщение Couldn’t open console.

Если X запускается с помощью startx, права на устройство /dev/console*не* изменяются, поэтому такие программы как xterm -C и xconsole не будут работать.

Это зависит от прав доступа, установленных для консоли по умолчанию. В многопользовательской системе вовсе не нужно, чтобы любой пользователь мог выводить информацию на системную консоль. Для пользователей, вошедших в систему через VTY, для решения этой проблемы существует файл fbtab(5).

В общем, раскомментируйте строчку в файле /etc/fbtab (посмотрите справку по fbtab(5)):

```
/dev/ttyv0 0600 /dev/console
```

Этого будет достаточно для того, чтобы всякий, кто вошёл в систему с терминала /dev/ttyv0, имел доступ к консоли.

11.12. Моя мышь PS/2 в X работает неправильно.

Мышь и драйвер могли рассинхронизироваться. В редких случаях драйвер может ошибочно сообщать о проблемах синхронизации:

```
psmintr: out of sync (xxxx != yyyy)
```

Если это случилось, отмените проверку согласования, установив значение флага для драйвера мыши PS/2 в `0x100`. Проще всего это сделать добавлением `hint.psm.0.flags="0x100"` в /boot/loader.conf с перезагрузкой.

11.13. Как поменять местами кнопки мыши?

Наберите `xmodmap -e "pointer = 3 2 1"`. Добавьте эту команду в ~/.xinitrc или ~/.xsession для автоматического запуска.

11.14. Как установить экранную заставку и где такие заставки можно найти?

Подробный ответ находится в разделе extref:https://docs.freebsd.org/ru/books/handbook/[Загрузочные экранные заставки, boot-splash] Руководства FreeBSD.

11.15. Можно ли в X задействовать `Windows`-клавиши на клавиатуре?

Да. Воспользуйтесь xmodmap(1) для привязки функций к этим клавишам.

Если все клавиатуры Windows стандартны, то эти три клавиши имеют следующие клавиатурные коды:

- 115 - клавиша `Windows` между клавишами `Ctrl` и `Alt` с левой стороны
- 116 - клавиша `Windows` справа от `AltGr`
- 117 - клавиша `Menu`, слева от клавиши `Ctrl`, находящейся справа

Чтобы заставить левую клавишу `Windows` набирать запятую, попробуйте выполнить такую команду:

```
# xmodmap -e "keycode 115 = comma"
```

Для того, чтобы переопределения клавиш `Windows` выполнялись автоматически каждый раз при запуске X, поместите команды `xmodmap` в ~/.xinitrc либо, что предпочтительней, создайте файл ~/.xmodmaprc и включите в него параметры `xmodmap` по одному на строку, затем добавьте в ~/.xinitrc такую строку:

```
xmodmap $HOME/.xmodmaprc
```

Например, чтобы переопределить эти 3 клавиши так, чтобы они выполняли функции клавиш `F13`, `F14` и `F15`. Это позволит легко привязать их к полезным функциям в приложениях или менеджере окон.

Чтобы сделать это, поместите такие строки в файл ~/.xmodmaprc:

```
keycode 115 = F13
keycode 116 = F14
keycode 117 = F15
```

При использовании менеджера рабочего стола x11-wm/fvwm2 клавиши можно переопределить так, чтобы нажатие `F13` сворачивало в иконку (и восстанавливало предыдущий размер) того окна, на которое указывает курсор, `F14` перемещало окно с курсором на передний план или, если оно уже впереди, возвращало обратно, а `F15` вызывало главное меню Workplace, даже если курсор находится не на рабочем столе, что бывает полезно, когда рабочий стол совсем не виден.

Следующие записи в ~/.fvwmrc позволяют достичь описанных выше функций:

```
Key F13 FTIWS   A   Iconify
Key F14 FTIWS   A   RaiseLower
```

```
Key F15 A   A   Menu Workplace Nop
```

11.16. Как активировать аппаратное ускорение 3D-графики для OpenGL®?

Наличие 3D-ускорения зависит от версии сервера Xorg и типа графического адаптера. Для адаптера nVidia используйте предскомпилированный драйвер для FreeBSD, установив один из нижеследующих портов:

Последние версии адаптеров nVidia поддерживаются портом x11/nvidia-driver.

Более старые драйверы доступны в следующем виде:

- x11/nvidia-driver-390
- x11/nvidia-driver-340
- x11/nvidia-driver-304

nVidia предоставляет подробную информацию о том, какие адаптеры поддерживаются тем или иным драйвером, на своём сайте: http://www.nvidia.com/object/IO_32667.html.

Для адаптеров Matrox G200/G400 следует попробовать порт x11-drivers/xf86-video-mga.

Для ATI Rage 128 и Radeon посмотрите страницы Справочника ati(4), r128(4) и radeon(4).

Chapter 12. Работа в сети

12.1. Где можно найти информацию о бездисковой загрузке?

"Бездисковая загрузка" означает, что машина с FreeBSD загружается по сети и читает необходимые файлы с сервера, а не со своего диска. Подробное описание есть в extref:https://docs.freebsd.org/ru/books/handbook/advanced-networking/[соответствующей главе, network-diskless] Руководства.

12.2. Может ли машина с FreeBSD использоваться как выделенный маршрутизатор?

Да. Обратитесь к разделу Руководства, посвящённому сложным вопросам работы в сети, а именно к той части, где рассказывается о маршрутизации и шлюзах маршрутизации.

12.3. Поддерживает ли FreeBSD технологию NAT или Masquerading?

Да. Для получения указаний по использованию NAT через подключение PPP обратитесь к разделу Руководства о PPP. Чтобы использовать NAT вместе с другим типом сетевого подключения, взгляните на раздел Руководства, посвящённый extref:[natd, network-natd].

12.4. Как настроить алиас в сети Ethernet?

Если алиас находится в той же самой сети, что и уже настроенный на интерфейсе адрес, допишите к этой команде `netmask 0xffffffff`:

```
# ifconfig ed0 alias 192.0.2.2 netmask 0xffffffff
```

В противном случае укажите сетевой адрес и маску обычным образом:

```
# ifconfig ed0 alias 172.16.141.5 netmask 0xffffff00
```

Дополнительная информация может быть найдена в extref:https://docs.freebsd.org/ru/books/handbook/config/[Руководстве, configtuning-virtual-hosts] по FreeBSD.

12.5. Почему я не могу смонтировать диск Linux® по NFS?

Некоторые версии NFS для Linux® поддерживают запросы на монтирование только с привилегированного порта; попробуйте выполнить следующую команду:

```
# mount -o -P linuxbox:/blah /mnt
```

12.6. Почему mountd продолжает выдавать сообщения can’t change attributes и bad exports list на моём сервере NFS, работающем под управлением FreeBSD?

В большинстве случаев проблема заключается в недостаточном понимании корректного формата файла /etc/exports. Просмотрите ещё раз справочную информацию по exports(5) и раздел об NFS Руководства, особенно в части настройки NFS.

12.7. Как включить поддержку multicast IP?

Установите пакет net/mrouted и добавьте `mrouted_enable="YES"` в /etc/rc.conf для запуска этого сервиса во время загрузки.

12.8. Почему я должен использовать FQDN для хостов не в моей сети?

За ответом на этот вопрос обратитесь к extref:https://docs.freebsd.org/ru/books/handbook/mail/[Руководству, mail-trouble] по FreeBSD.

12.9. Почему при выполнении любых сетевых операций выдаётся сообщение Permission denied?

Если ядро скомпилировано с параметром `IPFIREWALL`, имейте в виду, что политикой по умолчанию является запрет прохождения любых пакетов, которые не разрешены явным образом.

Если межсетевой экран был случайно сконфигурирован неверным образом, то для восстановления работоспособности сети наберите такую команду из-под пользователя `root`:

```
# ipfw add 65534 allow all from any to any
```

Рассмотрите использование `firewall_type='open'` в файле /etc/rc.conf.

Дополнительная информация о настройке данного межсетевого экрана находится в соответствующей главе Руководства.

12.10. Почему моё правило “fwd” для `ipfw` по перенаправлению сервиса на другую машину не работает?

Возможно, потому что вместо простого перенаправления пакетов нужна трансляция сетевых адресов (NAT). Правило "fwd" только перенаправляет пакеты и данные внутри него не меняет. Рассмотрим такое правило:

```
01000 fwd 10.0.0.1 from any to foo 21
```

Когда пакет с адресом назначения *foo* достигает машины с этим правилом, пакет перенаправляется на *10.0.0.1*, но в нём остаётся адрес назначения *foo*. Адрес назначения пакета не меняется на *10.0.0.1*. Большинство машин, скорее всего, отбросят полученный пакет, имеющий адрес назначения, им не соответствующий. Таким образом, правило "fwd" часто работает не так, как ожидает пользователь. Описанное поведение является особенностью, а не ошибкой.

Обратитесь к FAQ о перенаправлении сервисов, руководству по natd(8) или одной из нескольких утилит для перенаправления портов из Коллекции Портов для того, чтобы сделать это правильно.

12.11. Как можно перенаправить запросы сервисов с одной машины на другую?

Запросы FTP и других сервисов можно перенаправить с помощью порта sysutils/socket. Замените запись для этого сервиса в /etc/inetd.conf на вызов `socket`, как показано в этом примере для ftpd:

```
ftp stream tcp nowait nobody /usr/local/bin/socket socket ftp.example.com ftp
```

где *ftp.example.com* и *ftp* являются соответственно хостом и портом для перенаправления.

12.12. Где можно найти средства управления сетевым трафиком?

Для FreeBSD имеются три средства управления трафиком. dummynet(4) интегрирован в систему FreeBSD как составная часть ipfw(4). ALTQ включен во FreeBSD как составная часть pf(4). Bandwidth Manager компании Emerging Technologies является коммерческим продуктом.

12.13. Почему появляются сообщения /dev/bpf0: device not configured?

Для работы приложения требуется Berkeley Packet Filter (bpf(4)), однако это устройство удалено из вашего ядра. Постройте новое ядро с добавлением в его конфигурационный файл следующей строки:

```
device bpf  # Berkeley Packet Filter
```

12.14. Как смонтировать диск Windows®-машины в моей локальной сети, как это делает smbmount в Linux®?

Используйте пакет SMBFS. В него включён набор изменений в ядре и пользовательские программы. Программы и информация доступны как mount_smbfs(8) и входят в состав базовой системы.

12.15. Что значат сообщения Limiting icmp/open port/closed port response в файле журнала?

Данное сообщение ядра означает, что имеет место некоторая активность, приводящая к отправке большого количества ответных пакетов ICMP или сбросов TCP (RST). Ответы ICMP часто генерируются в результате попыток подключения к незанятым портам UDP. Сбросы TCP генерируются в результате попыток подключения к закрытым портам TCP. Помимо всего прочего, такие сообщения могут быть вызваны следующими действиями:

- Лобовая атака типа отказ в обслуживании DoS (в отличие от атак в один пакет, которые используют конкретную брешь в защите).
- Сканирование портов в попытке осуществить подключение к большому количеству портов (в отличие от проб нескольких известных портов).

Первое число в сообщении показывает количество пакетов, которое ядро отправило бы при отсутствии ограничений, а второе число отражает лимит. Этот лимит управляется при помощи переменной `net.inet.icmp.icmplim`. В этом примере устанавливается лимит на `300` пакетов в секунду:

```
# sysctl net.inet.icmp.icmplim=300
```

Для выключения подобных сообщений без отключения самого ограничения используйте `net.inet.icmp.icmplim_output`, чтобы подавить вывода:

```
# sysctl net.inet.icmp.icmplim_output=0
```

И наконец, чтобы полностью выключить это ограничение, сделайте `net.inet.icmp.icmplim` равным `0`. Выключение этого лимита не приветствуется по причинам, изложенным выше.

12.16. Что это за сообщения arp: unknown hardware address format?

Это означает, что какое-то устройство в локальной сети Ethernet использует MAC-адрес в формате, неизвестном FreeBSD. Вероятно, это происходит из-за того, что кто-то в сети экспериментирует с сетевым адаптером. Чаще всего это происходит в сетях с кабельными модемами. Это безобидно и не должно влиять на производительность системы FreeBSD.

12.17. Почему я постоянно вижу сообщения вида 192.168.0.10 is on fxp1 but got reply from 00:15:17:67:cf:82 on rl0 и как мне их отключить?

Это так, потому что пакет приходит извне сети, чего не должно быть. Чтобы отключить эти сообщения, установите `net.link.ether.inet.log_arp_wrong_iface` в значение `0`.

12.18. Как скомпилировать ядро, поддерживающее только IPv6?

Выполните конфигурацию ядра со следующими параметрами:

```
include GENERIC
ident GENERIC-IPV6ONLY
makeoptions MKMODULESENV+="WITHOUT_INET_SUPPORT="
nooptions INET
nodevice gre
```

Chapter 13. Безопасность

13.1. Что означает термин sandbox (песочница)?

"Sandbox" - это термин, используемый при обеспечении безопасности. Он имеет два значения:

- Процесс, помещённый внутрь некоторых виртуальных стен, которые предназначены для предотвращения взлома всей системы в результате взлома этого конкретного процесса.

  Процесс может работать только в границах этих стен. Поскольку, что бы этот процесс ни делал, он эти стены разрушить не может, особый аудит его кода не нужен для того, чтобы с уверенностью сказать, насколько его работа безопасна для системы.

  Стеной может служить, например, идентификатор пользователя. Вот определение, даваемое на страницах Справочника security(7) и named(8).

  Рассмотрим, например, службу `ntalk` (смотрите inetd(8)). Раньше эта служба запускалась с полномочиями пользователя `root`. Теперь она запускается с полномочиями пользователя с идентификатором `tty`. Пользователь `tty` является песочницей, предназначенной для того, чтобы пользователю, которому удалось проникнуть в систему через `ntalk`, было сложнее взломать систему и получить полномочия больше, чем обладает этот идентификатор.

- Процесс, помещённый внутрь симулируемой машины. Это означает, что некто, взломавший процесс, может думать, что может сломать и систему в целом, однако фактически может сломать только симулятор этой машины и не может модифицировать никаких реальных данных.

  Самым распространённым способом достигнуть такого результата является построение имитирующего окружения в каталоге и затем запуск процессов в этом каталоге через chroot (т.е. задав этот каталог в качестве / для этого процесса, а не реальный / всей системы).

  Другим часто используемым методом является монтирование низлежащей файловой системы в режиме "только для чтения" и затем создание уровня файловой системы поверх неё, что даёт процессу видимость доступа по записи на ту файловую систему. Процесс будет полагать, что может записывать в те файлы, но это будет единственный процесс, который увидит результат - другие процессы не будут этого делать ни в коем случае.

  Попытка сделать такой тип песочницы настолько прозрачна, что пользователь (или взломщик) даже не поймёт, что он в ней находится.

В UNIX® реализованы два типа "песочниц". Один на уровне процесса, и один на уровне идентификаторов пользователей.

Каждый процесс в UNIX® полностью защищён от других процессов. Никакой процесс не

может модифицировать адресное пространство другого процесса.

В UNIX® каждым процессом владеет некоторый идентификатор пользователя. Если этот пользователь не `root`, он ограждает процесс от других, владельцами которых являются другие пользователи. Этот идентификатор используется также для защиты данных на диске.

13.2. Что такое уровень безопасности (securelevel)?

`securelevel` является механизмом обеспечения безопасности, который реализован в ядре. Когда уровень защиты больше нуля, ядро ограничивает выполнение некоторых операций; даже суперпользователю `root` запрещается их выполнять. Механизм уровня защиты ограничивает возможности по:

- снятию некоторых флагов с файлов, таких, как `schg` (системный флаг неизменяемости),
- записи в память ядра через устройства /dev/mem и /dev/kmem,
- загрузке модулей ядра и
- изменению правил сетевого экрана.

Для выяснения состояния уровня защиты в работающей системе:

```
# sysctl -n kern.securelevel
```

Результат содержит текущее значение уровня защиты. Если оно больше нуля, то по крайней мере некоторые из защит этого механизма включены.

Уровень защиты работающей системы не может быть понижен, поскольку это противоречит назначению этого механизма. Если для задачи требуется неположительный уровень защиты, измените значения переменных `kern_securelevel` и `kern_securelevel_enable` в файле /etc/rc.conf, а затем перезагрузите систему.

Более подробная информация об уровнях защиты и о том, какие специфические действия выполняют все уровни, может быть найдена на справочных страницах о init(8).

Уровень защиты не является панацеей; в нём есть много недостатков. Зачастую он даёт обманчивое чувство безопасности.

Одной из самых больших проблем является то, что для его эффективной работы все файлы, используемые в процессе загрузки, должны быть защищены. Если атакующий сможет заставить систему выполнять свой код до установки уровня защиты (что происходит достаточно поздно во время процесса загрузки, так как некоторые вещи, выполняемые системой в это время, не могут быть сделаны при повышенном уровне защиты), то эта защита может быть отключена. Хотя такая задача по защите всех файлов, используемых в процессе загрузки, технически вполне осуществима, если это будет сделано, то поддержка системы станет кошмаром, так как для изменения конфигурационного файла придётся останавливать систему,

переводя её по крайней мере в однопользовательский режим.

Это обстоятельство, а также ряд других, часто обсуждаются в списках рассылки, в частности, во Список рассылки FreeBSD, посвящённый информационной безопасности. Поищите в архивах более подробное обсуждение. Предпочтителен более гибкий механизм.

13.3. Что это за пользователь toor с UID 0? Я подвергся взлому?

Не волнуйтесь, `toor` является "альтернативной" учётной записью суперпользователя (toor - это root, записанный задом наперёд). Его предлагается использовать с нестандартным командным интерпретатором, так чтобы не нужно было менять используемый по умолчанию командный процессор для `root`. Это важно, так как оболочки, не являющиеся частью дистрибутива системы, но установленные из портов или пакаджей, размещаются в каталоге /usr/local/bin, который по умолчанию располагается в другой файловой системе. Если командный процессор для пользователя `root` располагается в /usr/local/bin и файловая система, содержащая /usr/local/bin, не смонтирована, то пользователь `root` не сможет войти в систему для исправления проблемы и понадобится перезагрузиться в однопользовательском режиме, чтобы указать путь до командного процессора.

Некоторые используют `toor` для выполнения повседневных административных работ с нестандартным командным процессором, оставляя `root` со стандартной оболочкой для работы в однопользовательском режиме или выполнения аварийных работ. По умолчанию ни один пользователь не сможет войти в систему как `toor`, потому что для этой учётной записи не указан пароль, поэтому войдите из-под `root` и установите пароль для `toor` до того, как использовать его для входа в систему.

Chapter 14. Коммуникационные адаптеры

В этом разделе освещены вопросы о работе последовательных адаптеров во FreeBSD.

14.1. Как сделать так, чтобы приглашение boot: выводилось на консоль на последовательном порту?

Подробная информация находится в этом разделе Руководства.

14.2. Как узнать, обнаружила ли FreeBSD последовательные порты или внутренние модемы?

В процессе своей загрузки ядро FreeBSD будет пытаться найти последовательные порты, с поддержкой которых ядро сконфигурировано. Внимательно просмотрите сообщения загрузки либо выполните такую команду после того, как система запустилась и работает:

```
% grep -E '^(sio|uart)[0-9]' < /var/run/dmesg.boot
uart0: <16550 or compatible> port 0x3f8-0x3ff irq 4 flags 0x10 on acpi0
uart0: console (115200,n,8,1)
uart1: <16550 or compatible> port 0x2f8-2x3ff irq 3 on acpi0
```

В этом примере присутствуют два последовательных порта. Первый находится на IRQ4, порт ввода/вывода `0x3f8`, и построен на микросхеме UART типа 16550. Второй использует тот же тип микросхемы, но находится на IRQ3 и использует адрес порта ввода/вывода `0x2f8`. Внутренние модемы выглядят точно также, как последовательные порты, за исключением того, что модем к ним подключен всегда.

В ядро GENERIC встроена поддержка двух последовательных портов, с теми же IRQ и адресами портов ввода/вывода, как указано в примере выше. Если эти настройки не соответствуют системе или имеется больше внутренних модемов или последовательных портов, чем описано в ядре, переконфигурируйте его, следуя инструкциям в разделе о построении ядра.

14.3. Как осуществляется доступ к последовательным портам во FreeBSD? (специфично для x86)

Третий последовательный порт, sio2 или COM3, обозначается как /dev/cuad2 для устройств, выполняющих исходящие звонки, и /dev/ttyd2 для устройств, принимающих входящие звонки. Какая разница между этими двумя классами устройств?

При открытии /dev/ttydX в блокирующем режиме процесс будет ожидать неактивности соответствующего устройства cuadX, а затем появления сигнала о наличии несущей. При открытии устройства cuadX он проверяет, что последовательный порт не занят устройством ttydX. Если порт доступен, он похищает его у устройства ttydX. Также устройство cuadX не следит за наличием несущей. С такой схемой работы и модемом в режиме автоответа удалённые пользователи могут входить в систему, а локальные пользователи через тот же модем могут по прежнему осуществлять исходящие звонки, а система позаботится о возможных конфликтах.

14.4. Как включить поддержку многопортовых последовательных адаптеров?

Информация о конфигурировании ядра содержится в соответствующем разделе, посвящённом этому вопросу. Для многопортовых последовательных адаптеров добавьте в файл device.hints(5) по строке sio(4) на каждый порт. Но IRQ должен быть указан только у одного порта. Все порты на адаптере должны использовать одно и то же значение IRQ. Для обеспечения согласованности используйте для указания IRQ последний последовательный порт. Также укажите следующую опцию в файле конфигурации ядра:

```
COM_MULTIPORT
```

В следующем примере указано содержимое /boot/device.hints для 4-портового последовательного адаптера AST на IRQ 12:

```
hint.sio.4.at="isa"
hint.sio.4.port="0x2a0"
hint.sio.4.flags="0x701"
hint.sio.5.at="isa"
hint.sio.5.port="0x2a8"
hint.sio.5.flags="0x701"
hint.sio.6.at="isa"
hint.sio.6.port="0x2b0"
hint.sio.6.flags="0x701"
hint.sio.7.at="isa"
hint.sio.7.port="0x2b8"
hint.sio.7.flags="0x701"
hint.sio.7.irq="12"
```

Флаги указывают, что управляющий порт имеет младший номер устройства `7` (`0x700`), и все порты совместно используют один и тот же номер IRQ (`0x001`).

14.5. Можно ли настроить для последовательного порта режим работы по умолчанию?

Смотрите раздел Руководства по FreeBSD, посвящённый последовательным соединениям.

14.6. Почему не удаётся запустить tip или cu?

Встроенные утилиты tip(1) и cu(1) могут получить доступ к каталогу /var/spool/lock только из-под пользователя `uucp` и членов группы `dialer`. Используйте группу `dialer` для управления доступом к модему или удалённым системам посредством добавления в неё пользовательских учётных записей.

Либо же можно разрешить всем запускать tip(1) и cu(1):

```
# chmod 4511 /usr/bin/cu
# chmod 4511 /usr/bin/tip
```

Chapter 15. Разное

15.1. Почему FreeBSD использует много места в разделе подкачки даже при большом объёме свободной памяти?

FreeBSD активно перемещает неиспользуемые страницы памяти, к которым не было обращений, в раздел подкачки, чтобы увеличить объём доступной физической памяти для активного использования. Такое активное использование раздела подкачки компенсируется использованием дополнительной свободной оперативной памяти для кеширования.

Заметьте, что хотя FreeBSD предпочитает использовать раздел подкачки, страницы не перемещаются произвольно при полностью неактивной системе. По этой причине система не будет находиться целиком в разделе подкачки после ночного простаивания.

15.2. Почему утилита top(1) показывает очень маленький объём свободной памяти, даже когда запущено всего лишь несколько приложений?

Просто дело в том, что под свободной памятью подразумевается никак не используемая память. Вся память, которая программам явно не выделялась, используется ядром FreeBSD для дискового кэша. Значения, показываемые утилитой &man.top.1; с метками `Inact` и `Laundry`, являются кэшированными данными разных степеней устаревания. То, что данные находятся в кэше, означает, что система не будет обращаться к медленному диску снова за теми данными, обращение к которым было недавно, повышая таким образом общую производительность. В общем случае маленькие значения в пункте `Free`, показываемые утилитой top(1) для свободной памяти - это хорошо, если, конечно они не *очень* маленькие.

15.3. Почему командой chmod невозможно изменить права на символические ссылки?

Символические ссылки не имеют атрибутов доступа, и по умолчанию утилита chmod(1) следует по символической ссылке, чтобы по возможности изменить права доступа на исходный файл. Для файла foo с символической ссылкой bar на этот файл эта команда всегда будет выполняться успешно.

```
% chmod g-w bar
```

Однако права на файл bar не изменятся.

Чтобы это работало, используйте опцию `-H` или `-L` вместе с опцией `-R`. Обратитесь к страницам Справочника по команде chmod(1) и по symlink(7).

Опция `-R` выполняет chmod(1) *рекурсивно*. Будьте внимательны, задавая каталоги или символические ссылки на каталоги в параметрах chmod(1). Чтобы изменить права на каталог, на который указывает символическая ссылка, используйте chmod(1) без опций и следуйте символической ссылке с помощью лидирующего слэша (/). Например, если foo является символической ссылкой на каталог bar, то чтобы изменить права на foo (на самом деле на bar), выполните такую команду:

```
% chmod 555 foo/
```

Если задан ведущий слэш, то chmod(1) будет следовать символической ссылке foo, меняя права на каталог bar.

15.4. Могу ли я запускать программы для DOS во FreeBSD?

Да. Программа эмуляции DOS, emulators/doscmd, доступна в Коллекции Портов FreeBSD.

Если doscmd не достаточно, emulators/pcemu эмулирует 8088 и набор сервисов BIOS, достаточный для запуска многих приложений текстового режима DOS. Требуется X Window System.

В Коллекции Портов FreeBSD также имеется emulators/dosbox. Программа в основном предназначена для эмуляции старых игр, написанных под DOS, для хранения файлов которых используется локальная файловая система.

15.5. Что мне нужно сделать, чтобы перевести документацию FreeBSD на мой родной язык?

Ознакомьтесь с FAQ по Переводам из FreeBSD Documentation Project Primer.

15.6. Почему возвращается моя электронная почта, отправленная на любой из адресов FreeBSD.org?

В почтовой системе `FreeBSD.org` в Postfix применяются некоторые проверки входящей почты, и отвергаются сообщения, которые были неправильно сформированы при пересылке либо как-то иначе похожи на спам. Вот некоторые из требований:

- IP-адрес клиента SMTP должен иметь обратное преобразование в символическое имя.
- Полное имя хоста, указанное на этапе EHLO/HELO в процессе обмена сообщениями SMTP, должно разрешаться в IP-адрес клиента.

Дополнительные советы по доставке письма:

- Письмо должно быть отправлено в текстовом формате. Сообщение в почтовый список рассылки, как правило, не должно иметь размер больше 200 Кбайт.
- Избегайте избыточного кросспостинга. Выберите *один* список рассылки, который кажется наиболее подходящим.

Если у вас всё ещё остались трудности при работе с почтовой инфраструктурой `FreeBSD.org`, отправьте сообщение с подробным описанием на адрес postmaster@freebsd.org. Укажите в нём временной интервал для проверки логов - и обратите внимание, что мы держим журнал почтовых логов всего за неделю. (Обязательно укажите часовой пояс или разницу в UTC.)

15.7. Где можно получить бесплатный доступ к FreeBSD?

Хотя FreeBSD не предоставляет бесплатный доступ ни к одному из своих серверов, другие компании предоставляют UNIX®-системы с открытым доступом. Стоимость этой услуги различна, также как и ограниченный набор услуг.

Arbornet, Inc, также известный как *M-Net*, предоставляет свободный доступ к UNIX®-системам с 1983 года. Начиная на платформе Altos с работающей System III, сайт перешёл на BSD/OS в 1991. В июне 2000 сайт сменил систему снова, теперь на FreeBSD. *M-Net* может быть доступна через протоколы telnet и SSH и предоставляет доступ к полному набору программного обеспечения FreeBSD. Однако доступ к сети ограничен для членов и спонсоров, которые поддерживают систему, которая работает как неприбыльная организация. *M-Net* предоставляет также услуги электронной доски объявлений (BBS) и интерактивного чата.

15.8. Как зовут этого маленького симпатичного красного парня?

У него нет определённого имени, он называется просто "даемон BSD". Если вам непременно нужно имя, называйте его "beastie". Заметьте, что "beastie" произносится как "BSD".

Больше о даемоне BSD можно узнать из его домашней страницы.

15.9. Могу ли я использовать изображение даемона BSD?

Вполне может быть. Правами на даемона BSD обладает Marshall Kirk McKusick. Для выяснения подробностей относительно правил его использования обратитесь к странице автора Statement on the Use of the BSD Daemon Figure.

В общем, использовать изображение можно в высокохудожественном стиле и в личных целях, если даются соответствующие отсылки. Перед использованием знака в коммерческих целях обратитесь за разрешением к Kirk McKusick <mckusick@FreeBSD.org>. Дополнительная информация находится на домашней странице Даемона BSD.

15.10. Не найдётся ли у вас изображений даемона BSD, которые можно использовать?

В каталоге /usr/share/examples/BSD_daemon/ есть рисунки в форматах eps и Xfig.

15.11. При просмотре списков рассылки я встретил сокращение или другой термин, который мне не понятен. Где я должен посмотреть, что он значит?

Обратитесь к Глоссарию FreeBSD.

15.12. Почему я должен беспокоиться о цвете велосипедных навесов (bikeshed)?

На самом деле, очень краткий ответ на этот вопрос заключается в том, что вы этого делать не должны. Если давать более подробный ответ, то ваше умение делать навесы не должно означать, что вы должны препятствовать другим делать их просто потому, что вам не нравится цвет, в который они собираются их окрашивать. Эта метафора означает, что вам не нужно обсуждать каждую мелочь просто потому, что вы знаете о ней достаточно много. Некоторые люди отмечают, что объём шума, генерируемый при появлении некоторого изменения, находится в обратной зависимости от сложности самого изменения.

Более пространный и полный ответ заключается в том, что после очень долгого обсуждения того, должна ли утилита sleep(1) обрабатывать дробное число, заданное в качестве второго аргумента, Poul-Henning Kamp <phk@FreeBSD.org> опубликовал большое сообщение, озаглавленное Велосипедный навес (любого цвета) на зелёной траве…. Соответствующие части этого сообщения цитируются ниже.

Poul-Henning Kamp <phk@FreeBSD.org> on freebsd-hackers, October 2, 1999 "Что это за история с этим навесом для велосипеда?", кто-то из вас спрашивал меня.

Это долгая история, точнее, это старая история, но на самом деле она коротка. В начале 1960-х годов Сирил Норткот Паркинсон (C. Northcote Parkinson) написал книгу "Законы Паркинсона", которая содержит много интересных взглядов на процесс управления.

[немного выдержек из краткого содержания книги]

В конкретном примере велосипедный навес сопоставляется с другим важным объектом - атомной электростанцией. Я полагаю, что это иллюстрирует древность книги.

Паркинсон показывает, что вы можете прийти на совещание руководителей и получить добро на строительство многомиллионной или даже многомиллиардной атомной электростанции, но если вы хотите построить навес для велосипеда, то погрязнете в бесконечных обсуждениях.

Паркинсон объясняет это тем, что атомная станция настолько большой, дорогой и сложный

объект, что люди не могут его осознать и вместо того, чтобы попробовать это сделать, они полагаются на то, что кто-то уже проверил все мелочи до того, как всё зашло так далеко. В своей книге Ричард П. Фейнманн (Richard P. Feynmann) даёт несколько интересных и очень поучительных примеров, связанных с Лос Аламос.

Велосипедный навес - это противоположный случай. Любой может построить навес за один уикэнд, и у него ещё останется время посмотреть футбол по телевизору. Так что не важно, насколько хорошо вы готовились к обсуждению, насколько убедительны будут ваши аргументы, кто-нибудь воспользуется шансом показать, что он не зря ест свой хлеб, что он обращает внимание, что он *здесь*.

В Дании это называется “оставить отпечаток своего пальца”. Это всё касается личной гордости и престижа, это возможность указать куда-то и сказать: “Вот! Это сделал я”. Это сильно выражено в политиках, но присутствует во многих людях, которые получают возможность сделать это. Просто вспомните об отпечатках ног во влажном цементе.

Chapter 16. Юмор от FreeBSD

16.1. Насколько FreeBSD горяча?

В. Кто-нибудь делал замеры температуры при работе FreeBSD? Я знаю, что Linux® греется меньше, чем DOS, но никогда не видел упоминания FreeBSD. Наверное, эта система - горячая штучка.

О. Нет, но мы сделали различные вкусовые тесты у добровольцев с завязанными глазами, которые до этого приняли по 250 микрограмм LSD-25. 35% участников заявили, что FreeBSD имеет вкус апельсина, тогда как вкус Linux® был похож на фиолетовый туман. Ни одна из групп не отметила значительной разницы в температуре. Мы уже собирались опубликовать полные результаты этого опроса, когда обнаружили, что слишком много добровольцев покинули помещение во время тестов, что несколько смазало результаты. Думаем, что большинство добровольцев работают сейчас в Apple над их новым GUI "чеши и нюхай". Это ведь старый добрый бизнес!

Если серьёзно, то FreeBSD использует инструкцию HLT (halt), когда система простаивает, что уменьшает потребление энергии и, в свою очередь, выделение тепла. Вдобавок, если у вас настроен ACPI (усовершенствованный интерфейс управления конфигурацией и питанием), то FreeBSD может переводить процессор в режим пониженного энергопотребления.

16.2. Кто там скребётся в микросхемах памяти??

В. Делает ли FreeBSD что-нибудь "эдакое" при компиляции ядра, что вызывает поскрипывание микросхем памяти? При компиляции (и в короткий промежуток времени после обнаружения дисковода при старте системы) от, видимо, микросхем памяти исходит странный царапающий звук.

О. Да! Вы встретите частое упоминание "даемонов" в документации по BSD, но не все знают, что речь идёт о настоящих нематериальных существах, которые теперь завладели и вашим компьютером. Царапающий звук, издаваемый микросхемами памяти - это на самом деле высокочастотное перешёптывание между даемонами, когда они решают, как лучше справиться с различными задачами по администрированию системы.

Если шум достиг ваших ушей, команда DOS `fdisk /mbr` их спугнёт, но не удивляйтесь, если они отреагируют соответствующим образом и попытаются вас остановить. Фактически, если во время выполнения этой команды вы услышите сатанинский голос Билла Гейтса из встроенного динамика, бегите и даже не оглядывайтесь! Избавленные от противостояния с даемонами BSD, близнецы-демоны DOS и Windows® часто могут захватить полный контроль над вашей машиной, чтобы навлечь вечное проклятие на вашу душу. Теперь, когда вы это знаете, если бы у вас был выбор, думаем, что вы бы предпочли привыкнуть к царапающему звуку, не так ли?

16.3. Сколько требуется разработчиков FreeBSD, чтобы сменить электрическую лампочку?

Необходимо иметь ровно одну тысячу сто шестьдесят девять разработчиков:

Двадцать три сообщат в -CURRENT о том, что не горит свет;

Четыре начнут утверждать, что это проблема конфигурации и такие сообщения нужно посылать в -questions;

Трое оформят PR по этому поводу, причём одно их них будет направлено в doc и будет содержать только строчку "здесь темно";

Один закоммитит неоттестированную лампочку, что сломает построение системы, а затем через пять минут вернёт всё назад;

Восемь поругаются с авторами PR по поводу включения патчей в PR;

Пять сообщат о том, что не проходит компиляция системы;

Тридцать один человек ответит, что у них всё работает и наверное, те обновились в неподходящее время;

Один пошлёт патч для новой лампочки в -hackers;

Один пожалуется, что у него имелись патчики ещё три года назад, но когда он послал их в -CURRENT, они были проигнорированы и он имел неудачный опыт работы с системой PR; кроме того предлагаемая лампочка не имеет отражателя.

Тридцать семь начнут кричать, что лампочки не относятся к базовой системе, что коммиттеры не имеют права делать такие вещи без опроса общественности и ЧТО ВООБЩЕ -CORE ДЕЛАЕТ ПО ЭТОМУ ПОВОДУ?

Две сотни напишут о цвете велосипедного навеса;

Трое скажут, что этот патч не соответствует style(9)

Семнадцать возразят, что предлагаемая новая лампа подпадает под лицензию GPL;

Пятьсот восемьдесят шесть раздуют флейм по поводу сравнения лицензий GPL, BSD, MIT, NPL и личных мнений о неизвестных основателей FSF;

Семеро пошлют различные части этих обсуждений в -chat и -advocacy;

Один закоммитит предлагаемую лампу, хотя она светит хуже, чем старая;

Двое откатят эти изменения с ужасной руганью в журнале коммитта о том, что лучше FreeBSD будет сидеть в темноте, чем с тусклой лампой.

Сорок шесть громко воспротивятся этому изменению и потребуют объяснений от -core;

Одиннадцать попросят уменьшить размер лампочки, чтобы она подошла к их Тамагочи на случай, если мы когда-нибудь соберёмся переносить FreeBSD на эту платформу;

Семьдесят три заявят о SNR в -hackers и -chat и в знак протеста отпишутся;

Тринадцать пошлют письма "unsubscribe", "How do I unsubscribe?" или "Please remove me from the list" с обычной подписью;

Один закоммитит работающую лампочку в то время, как все будут слишком заняты руганью, чтобы это заметить;

Тридцать один человек напишет, что новая лампочка будет светить на 0.364% ярче, если её откомпилировать с помощью TenDRA (хотя при этом она приобретёт форму куба) и что FreeBSD должна перейти на компилятор TenDRA, а не на GCC;

Один заметит, что у лампочки отсутствует цоколь;

Девять (включая авторов PR) спросят "что такое MFC?";

Спустя две недели после смены лампочки пятьдесят семь человек сообщат о том, что света всё равно нет.

Nik Clayton <nik@FreeBSD.org> добавил:

Я сильно смеялся над всем этим.

И тогда я подумал, "Постойте-ка, найдётся ли кто-нибудь, чтобы задокументировать это?"

И на меня снизошло озарение :-)

Thomas Abthorpe <*tabthorpe@FreeBSD.org*> говорит: "Нет, *настоящие* хакеры FreeBSD не боятся темноты!"

16.4. Куда направляются данные, записываемые в /dev/null?

Они отправляются в специальную сточную трубу для данных в CPU, где преобразуются в тепло, выдуваемое через охлаждающие вентиляторы. Вот почему охлаждение ЦП становится все более важным; так как люди используют все более быстрые процессоры, они все менее заботятся о данных, все большее их количество оканчивает свой путь в /dev/null, перегревая ЦП. Если вы удалите /dev/null (что соответственно отключит трубу данных в ЦП), то ваш процессор может охладиться, но система начнет переполняться излишними данными и начнет работать с ошибками. Если у вас быстрое сетевое подключение, вы можете охладить CPU, читая данные из /dev/random и посылая их куда-нибудь; однако вы рискуете перегреть ваше сетевое соединение и / или разозлить вашего провайдера, так как большинство данных преобразуется в тепло на его оборудовании, но, как правило, у него хорошее охлаждение, так что если вы не перестараетесь, все должно быть в порядке.

Пол Робинсон (Paul Robinson) добавляет:

Есть и другие методы. Как знает каждый хороший системный администратор, частью хорошей практики является посылка данных на экран интересным образом, чтобы феи, которые образуют картинку, были счастливы. Экранные феи (часто неправильно называемые "пикселами") различаются по цвету головных уборов, которые они носят (красные, зеленые или синие), и прячутся или появляются (показывая, таким образом, цвет своих шляп), когда получают немного пищи. Видеоадаптеры преобразуют данные в еду для фей, а затем посылают ее феям - чем дороже адаптер, тем лучше еда, тем лучше ведут себя феи. Они также нуждаются в постоянной стимуляции - вот зачем нужны хранители экранов.

Продолжив наше предположение, вы можете просто выдавать случайные данные на консоль, таким образом позволяя феям их потреблять. Это вовсе приводит к прекращению выделения тепла, феи постоянно счастливы, а данные быстро исчезают, даже если на вашем экране все выглядит несколько хаотично.

Как бывший администратор крупного провайдера, который имел много проблем при попытке поддерживать постоянную температуру в серверной комнате, я выступаю против того, чтобы люди посылали ненужные им данные в сеть. Волшебников, которые выполняют коммутацию пакетов и маршрутизацию, это также затрудняет.

16.5. Мой коллега проводит слишком много времени за компьютером.

Как я могу отвадить его от этого?

Установите пакет games/sl и дождитесь момента, когда коллега ошибочно введёт `sl` вместо `ls`.

Chapter 17. Сложные темы

17.1. Как можно узнать больше о внутреннем устройстве FreeBSD?

Обратитесь к Руководству по архитектуре FreeBSD.

Кроме того, большинство общих знаний о UNIX® непосредственно применимо к FreeBSD.

17.2. Как внести свой вклад в проект FreeBSD? Что можно сделать в качестве помощи?

Мы принимаем помощь в любой форме: документации, кода и даже художественной графики. Обратитесь к соответствующей статье Участие в проекте FreeBSD, в которой вы найдёте советы относительно того, как это сделать.</para>

И спасибо вам за то, что вы об этом подумали!

17.3. Что такое снапшоты и релизы?

В хранилище Git сейчас находятся 3 активно/полуактивно развивающихся ветки FreeBSD. (Более ранние ветки изменяются очень редко, именно поэтому в разработке только 3 активные ветки):

- stable/12, также известная как 12-STABLE
- stable/13, также известная как 13-STABLE
- main, также известная как *-CURRENT* и 15.0

На данный момент *-CURRENT* является находящимся в разработке деревом 14.X; ветка 13-STABLE, stable/13, отделилась от *-CURRENT* January 2021, а ветка 12-STABLE, stable/12, отделилась от *-CURRENT* December 2018.

17.4. Что делать при аварийном останове системы?

Вот типичная паника ядра:

```
Fatal trap 12: page fault while in kernel mode
fault virtual address   = 0x40
fault code              = supervisor read, page not present
instruction pointer     = 0x8:0xf014a7e5
stack pointer           = 0x10:0xf4ed6f24
frame pointer           = 0x10:0xf4ed6f28
code segment            = base 0x0, limit 0xfffff, type 0x1b
                        = DPL 0, pres 1, def32 1, gran 1
processor eflags        = interrupt enabled, resume, IOPL = 0
```

```
current process         = 80 (mount)
interrupt mask          =
trap number             = 12
panic: page fault
```

Этого сообщения не достаточно. Хотя значение указателя инструкций важно, но оно зависит от конфигурации, поскольку значение меняется для каждого конкретного файла ядра. Если это ядро GENERIC из одного из снэпшотов, то кто-то ещё может отследить функцию, вызвавшую ошибку, но в случае со специально сконфигурированным ядром только вы можете сказать нам, где случилась ошибка.

Чтобы продолжить:

1. Запишите значение указателя инструкций. Заметьте, что часть `0x8:` в этом случае не важна: нам нужна часть `0xf0xxxxxx`.
2. Когда система перезагрузится, сделайте следующее:

   ```
   % nm -n kernel.that.caused.the.panic | grep f0xxxxxx
   ```

   где `f0xxxxxx` - это значение указателя инструкций. Однако неприятность заключается в том, что вы не получите точного соответствия, так как в таблице имен ядра для точек входа в функции даны адреса на начало функций, а указатель инструкций будет указывать куда-то внутрь её тела. Если вы не получили точного соответствия, опустите последнюю цифру в значении указателя инструкций и попробуйте снова:

   ```
   % nm -n kernel.that.caused.the.panic | grep f0xxxxx
   ```

   Если и это не привело ни к каким результатам, отрежьте следующую цифру. Повторяйте, пока не получите хоть что-то. Результатом будет список функций, которые, возможно, привели к аварийному останову. Этот механизм обнаружения ошибочного места довольно неточен, но это всё же лучше, чем ничего.

Тем не менее, лучшим способом выяснить причину, вызвавшую аварийный останов, является получение аварийного дампа системы, а затем использование kgdb(1) для получения трассировки вызовов в этом дампе.

В любом случае, метод таков:

1. Убедитесь в том, что в файле конфигурации ядра имеется следующая строка:

   ```
   makeoptions     DEBUG=-g          # Build kernel with gdb(1) debug symbols
   ```

2. Перейдите в каталог /usr/src:

```
# cd /usr/src
```

3. Скомпилируйте ядро:

```
# make buildkernel KERNCONF=MYKERNEL
```

4. Дождитесь завершения компиляции.

```
# make installkernel KERNCONF=MYKERNEL
```

5. Выполните перезагрузку.

Если не указать `KERNCONF`, то будет собрано и установлено ядро GENERIC.

В процессе выполнения команды make(1) будут построены два ядра, /usr/obj/usr/src/sys/MYKERNEL/kernel и /usr/obj/usr/src/sys/MYKERNEL/kernel.debug. kernel будет установлен как /boot/kernel/kernel, тогда как kernel.debug может быть использован в качестве источника отладочной информации для kgdb(1).

Чтобы получать аварийный дамп, отредактируйте файл /etc/rc.conf так, чтобы устройство `dumpdev` указывало на раздел подкачки или имело значение `AUTO`. В этом случае скрипты rc(8) будут вызывать команду dumpon(8) для создания аварийных дампов. Эту команду можно также запускать вручную. После аварийной остановки аварийный дамп может быть получен с помощью программы savecore(8); если значение переменной `dumpdev` было задано в /etc/rc.conf, то скрипты rc(8) запустят savecore(8) автоматически и поместят аварийный дамп в каталог /var/crash.

Аварийные дампы FreeBSD обычно имеют размер, равный объёму оперативной памяти. Поэтому убедитесь в наличии достаточного места для хранения дампа в каталоге /var/crash. Либо запустите вручную savecore(8), чтобы создать аварийный дамп в другом каталоге, где достаточно места. Размер аварийного дампа можно уменьшить, указав в конфигурации ядра `options MAXMEM=N`, где *N* - значение в Кбайт для объёма памяти, которое будет использоваться ядром. Например, для 1 Гбайт ОЗУ установите ограничение на использование памяти ядром в 128 Мбайт, так чтобы размер аварийного дампа был равен 128 Мбайт, а не 1 Гбайт.

Как только аварийный дамп получен, трассировку вызовов можно получить таким образом:

```
% kgdb /usr/obj/usr/src/sys/MYKERNEL/kernel.debug /var/crash/vmcore.0
```

```
(kgdb) backtrace
```

Заметьте, что это может дать несколько экранов полезной информации. Лучше всего использовать script(1) для перехвата всего вывода. При использовании необработанного файла ядра со всей отладочной информацией может быть найдена конкретная строка исходного текста ядра, при достижении которой случилась аварийная остановка. Для выяснения последовательности событий, приведших к аварийному останову, трассировка стека обычно читается снизу вверх. Также можно использовать kgdb(1) для вывода значений различных переменных или структур, чтобы выяснить состояние системы во время аварии.

Если есть второй компьютер, то можно настроить kgdb(1) на режим удалённой отладки, включая определение точек останова и пошаговый проход по коду ядра.

Если включена поддержка `DDB` и ядро переходит в режим отладки, можно намеренно вызвать аварийный останов и создание аварийного дампа, набрав `panic` в приглашении командной строки `ddb`. Выполнение фазы аварийного останова может снова остановиться с вызовом отладчика. В этом случае наберите `continue`, и процесс будет завершён созданием аварийного дампа.

17.5. Перестала работать функция dlsym() для исполняемых файлов ELF!

По умолчанию при работе с форматом ELF символы, определённые в исполняемом файле, не доступны динамическому загрузчику. Поэтому при вызове функции `dlsym()`, которая осуществляет поиск по дескриптору, полученному после вызова `dlopen(NULL, flags)`, желаемый результат достигнут не будет.

Чтобы осуществить поиск символов в исполняемом файле процесса с помощью функции `dlsym()`, выполните компоновку исполняемого файла с параметром `--export-dynamic` компоновщика ELF (ld(1)).

17.6. Как я могу увеличить или уменьшить адресное пространство ядра в архитектуре i386?

По умолчанию размер адресного пространства ядра для i386 равен 1 Гбайт (2 Гбайт для PAE). Для работы сервера с интенсивной сетевой нагрузкой или при использовании ZFS этого может быть недостаточно.

Чтобы увеличить доступное пространство, добавьте следующую строку в файл конфигурации ядра и пересоберите ядро:

```
options KVA_PAGES=N
```

Чтобы получить нужное значение для *N*, разделите желаемый размер адресного пространства (в мегабайтах) на четыре (для 2 Гбайт это будет 512).

Chapter 18. Наши благодарности

Этот небольшой скромный документ с ответами на часто задаваемые вопросы создавался, переписывался, редактировался, сокращался, растягивался, уродовался, потрошился, пристально разглядывался, полностью перетряхивался, обдумывался, отвергался, перестраивался, критиковался и снова укреплялся в течение последнего десятилетия силами сотен, если не тысяч, людей. Постоянно.

Мы хотим поблагодарить каждого из них и приглашаем вас присоединиться к ним, чтобы сделать этот FAQ ещё лучше.