

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1. В.ДВ.10.02 Операционная система FreeBSD

**Направление подготовки (специальность) 10.03.01 Информационная безопасность**

**Профиль подготовки (специализация) Безопасность автоматизированных систем**

**Форма обучения очная**

## СОДЕРЖАНИЕ

<b>1. Конспект лекций.....</b>	<b>3</b>
<b>1.1 Лекция№1-4 Основные понятия ОС FreeBSD.....</b>	<b>3</b>
<b>1.2 Лекция№5-8 Состав ОС FreeBSD.....</b>	<b>6</b>
<b>1.3 Лекция№9-12 Команды работы с файлами.....</b>	<b>15</b>
<b>1.4 Лекция№13-16 Команды работы с каталогами.....</b>	<b>17</b>
<b>2. Методические указания по проведению практических занятий.....</b>	<b>23</b>
<b>2.1 Практическое занятие № ПЗ-1-4 Знакомство с интерфейсом операционной системы FreeBSD.....</b>	<b>23</b>
<b>2.2 Практическое занятие№ПЗ-5-8 Основные приёмы работы с FreeBSD.....</b>	<b>24</b>
<b>2.3 Практическое занятие № ПЗ-9-12 Монтирование и размонтиrovание файловых систем.....</b>	<b>25</b>
<b>2.4 Практическое занятие № ПЗ-13-16 Процессы ОС FreeBSD.....</b>	<b>27</b>
<b>2.5 Практическое занятие № ПЗ-17-20 Дополнительная справочная информация.....</b>	<b>30</b>
<b>2.6 Практическое занятие № ПЗ-21-24 Настройка карт сетевых интерфейсов.....</b>	<b>32</b>
<b>2.7 Практическое занятие № ПЗ-25-28 OPENSSH.....</b>	<b>35</b>
<b>2.8 Практическое занятие № ПЗ-29-32 Учётные записи пользователей.....</b>	<b>37.</b>

# **1. КОНСПЕКТ ЛЕКЦИЙ**

## **1. 1 Лекция № 1-4 (8 часов).**

**Тема:** «Основные понятия ОС FreeBSD.(Интерактивная форма - 4ч)»

### **1.1.1 Вопросы лекции:**

1. История разработки FreeBSD, для чего нужна FreeBSD
2. FreeBSD в роли настольной ОС
3. FreeBSD в роли серверной ОС

### **1.1.2 Краткое содержание вопросов:**

1. FreeBSD — свободная операционная система семейства Unix, потомок AT&T Unix по линии BSD, созданной в университете Беркли. FreeBSD работает на PC-совместимых системах семейства Intel x86 (IA-32) (включая Microsoft Xbox), а также на DEC Alpha, SunUltraSPARC, IA-64, AMD64, PowerPC, NEC PC-98, ARM. Готовится поддержка архитектуры MIPS.

FreeBSD разрабатывается как целостная операционная система. Исходный код ядра, драйверов устройств и базовых пользовательских программ (т. н. *userland*), таких как командные оболочки и т. п., содержится в одном дереве системы управления версиями (CVS до 31 мая 2008, сейчас SVN). Это отличает FreeBSD от GNU/Linux — другой свободной UNIX-подобной операционной системы — в которой ядро разрабатывается одной группой разработчиков, а набор пользовательских программ — другими (например, проект GNU), а многочисленные группы собирают это все в единое целое и выпускают в виде различных дистрибутивов GNU/Linux.

FreeBSD хорошо зарекомендовала себя как система для построения интернет- и интранет-серверов. Она предоставляет достаточно надёжные сетевые службы и эффективное управление памятью. FreeBSD широко представлена в списке веб-серверов с наибольшим временем непрерывной работы (согласно исследованию компании, Netcraft).

Разработка FreeBSD началась в 1993 году с большой скоростью, в качестве отправной точки были взяты исходные тексты 386BSD. Однако, из соображений законности использования исходных кодов 386BSD, команда разработчиков FreeBSD переработала большую часть системы ко времени выпуска FreeBSD 2.0 в январе 1995 года используя 4.4BSD-Lite. Руководство FreeBSD содержит более подробную историческую информацию о происхождении FreeBSD.

По состоянию на 26 ноября 2009 года текущий релиз FreeBSD имеет номер 8.0.

После выпуска релизов создаются дополнительные ветви разработки для их поддержки, но в них вносятся лишь самые необходимые изменения, исправляющие серьёзные ошибки или проблемы с безопасностью системы. До четвёртой версии FreeBSD у стабильной и текущей веток был один и тот же старший номер версии. Затем текущей ветви был присвоен номер 5, а у стабильной остался номер 4. В настоящее время стабильная версия имеет номер 8, а текущая — 9. Существует также ветвь 7-STABLE для поддержки тех пользователей, которые ещё не обновили FreeBSD до версии 8.

В любом деле важно находить правильные средства для достижения поставленных целей, и выбор оптимальной операционной системы (ОС) — не исключение. FreeBSD — очень гибкая и эффективная система, имеющая множество возможных вариантов применения. Но, несмотря на это она не всегда оказывается наилучшим выбором. Вот почему нужно, прежде всего, проанализировать, в каких обстоятельствах следует инсталлировать именно FreeBSD.

FreeBSD устанавливают либо на рабочей станции, либо на сервере. Ниже рассматриваются достоинства и недостатки системы в каждом из этих случаев. Кроме того, FreeBSD — член семейства операционных систем, производных от (или смоделированных на основе) UNIX. Поэтому важно понимать, как FreeBSD (и UNIX в целом) соотносится с операционными системами других типов и какое место занимает FreeBSD в семействе UNIX-подобных систем.

2. Под термином рабочая станция понимается компьютер, за которым пользователь решает свои повседневные задачи: редактирует текст, программирует, занимается Web-дизайном. Иногда термин употребляют в более узком смысле, подразумевая, что указанные выше функции выполняются на высокопроизводительных компьютерах или системах, постоянно подключенных к сети. В этом случае низкопроизводительные или изолированные системы называют настольными компьютерами. Впрочем, уровень вычислительных мощностей постоянно растет, соответственно меняется и оценка производительности. Что касается подключения к сети, то нужно учитывать огромную популярность коммутируемых сетевых соединений в системах нижнего сегмента рынка. Исходя из этого мы будем употреблять термин «рабочая станция» в отношении компьютеров конечных пользователей.

Способность компьютера играть роль рабочей станции зависит от двух основных факторов.

- Удобный пользовательский интерфейс. Современные пользователи привыкли к наличию удобных средств работы с компьютером. Ключевое из них — пользовательский графический интерфейс. Во FreeBSD графический интерфейс представлен средой X WindowSystem (сокращенно — X-среда). Ее ядро довольно примитивно в сравнении с другими аналогичными интерфейсами, поэтому к нему добавлен ряд надстроек, таких как диспетчер окон (управляет рамками окон) и набор элементов управления (упрощают программисту задачу отображения меню, диалоговых окон и т.д.). Подобный модульный подход делает X-среду очень гибкой с точки зрения программирования интерфейса, но есть и обратная сторона медали: возникающая несогласованность между приложениями часто сбивает пользователей с толку.

- Доступность прикладных программ. На рабочей станции выполняются пользовательские приложения: текстовые редакторы, программы обработки электронных таблиц, почтовые клиенты, компиляторы, графические редакторы и даже игры. Если операционная система не может предложить пользователю необходимый набор прикладных программ, то она не подходит для установки на рабочей станции. Список приложений, доступных во FreeBSD, велик, хотя и не настолько, как хотелось бы. В частности, популярный пакет MicrosoftOffice не перенесен во FreeBSD. Существуют его альтернативы, такие как пакет OpenOffice.org, но они не обеспечивают 100-процентную совместимость файлов. Если говорить в целом, то в одних категориях (например, игры) ощущается нехватка приложений для FreeBSD, а в других их более чем достаточно.

FreeBSD вполне подходит на роль операционной системы для рабочей станции, хотя в конкретной ситуации выбор зависит от специфических потребностей пользователя.

К достоинствам FreeBSD относятся низкая стоимость, способность функционировать на старом оборудовании, стабильность, гибкий пользовательский интерфейс и близкое сходство с другими разновидностями UNIX. К недостаткам можно отнести менее согласованный, чем в Windows или Mac OS, пользовательский интерфейс, несколько непривычный для пользователей вышеуказанных систем. Что касается фактора доступности прикладных программ, то все зависит от обстоятельств. Если пользователь вынужден работать с приложениями вроде MicrosoftOffice, которые недоступны во FreeBSD, придется обратиться к другой операционной системе. Если же пользователь не привязан к конкретным форматам файлов, то можно сделать выбор в пользу FreeBSD.

Стоит подметить, что если на рабочей станции требуется установить UNIX-подобную операционную систему и при этом предполагается запускать MicrosoftOffice или другой прикладной пакет, недоступный во FreeBSD, то в качестве компромиссного варианта подойдет Mac OS X. Эта система создана на основе микроядра Mach и целого ряда компонентов FreeBSD. Конечно, Mac OS X — это не FreeBSD, но благодаря многочисленным «заимствованиям» из FreeBSD и других разновидностей UNIX она гораздо больше знакома администраторам UNIX, чем Windows-системы или более ранние версии Mac OS. В Mac OS X поддерживаются многие прикладные пакеты, включая MicrosoftOffice.

Если предполагается, что в системе будет работать большое количество пользователей, то нужно уделить особое внимание вопросам конфигурирования графических оболочек. Эти оболочки значительно усовершенствовались с середины 90-х и по-прежнему регулярно обновляются.

3. Серверы — это компьютеры с сетевыми платами, обрабатывающие запросы других компьютеров. Именно серверы обеспечивают наиболее эффективное функционирование сетей. Ниже рассмотрены примеры серверов.

- Файловый сервер позволяет другим компьютерам читать и записывать файлы серверной файловой системы. Эти серверы можно использовать для распространения свободно доступных файлов (например, компонентов FreeBSD), а также в качестве централизованного хранилища файлов локальной сети.
- Web-сервер. В определенном смысле это разновидность файлового сервера. Обычно служит для доставки файлов Web-клиенту, называемому браузером.
- Почтовый сервер — это компьютер, обменивающийся почтовыми сообщениями с другими серверами либо с почтовыми клиентами конечных пользователей. В Internet обмен почтой происходит в основном между серверами, тогда как в локальной сети достаточно одного почтового сервера.
- Сервер дистанционной регистрации. FreeBSD можно сконфигурировать на прием регистрационных запросов от других пользователей. В этом случае работа в системе ведется в дистанционном режиме, т.е. стирается грань между рабочей станцией и сервером (система реализует функции сервера, но используется в основном как рабочая станция). Иногда на сервере предусматривается возможность дистанционной регистрации администратора.
- Вспомогательные серверы выполняют служебные функции. Например, сервер DHCP (DynamicHostConfigurationProtocol — протокол динамического конфигурирования компьютеров) позволяет централизованно задавать параметры сетевой конфигурации

клиентских компьютеров. Сервер шрифтов предоставляет рабочим станциям информацию о шрифтах.

FreeBSD является прекрасным кандидатом на роль серверной ОС. Не в последнюю очередь это связано с тем, что многие популярные серверные программы изначально разрабатывались в среде UNIX и потому прекрасно работают во FreeBSD. В сочетании с недорогими аппаратными компонентами платформы x86 система на базе FreeBSD является весьма экономичным решением для большинства современных серверных программ.

Сам по себе термин «сервер» может обозначать как компьютер, так и программу, реализующую серверные функции. Это означает, что один компьютер способен быть сервером сразу нескольких типов, например почтовым и Web-сервером. В большинстве случаев смысл термина «сервер» ясен из контекста.

Преимуществом использования FreeBSD в качестве серверной ОС является надежность системы. Несмотря на то что компьютеры, работающие на базе процессоров x86, весьма разнородны и не всегда столь же надежны, как традиционные серверные компьютеры, выпускаемые, к примеру, компаниями Sun и SiliconGraphics, FreeBSD вполне стабильна. Нередко серверы FreeBSD работают без сбоев месяцами. Если происходит перезагрузка, то это чаще делается для обновления аппаратных, а не программных компонентов. Перезагружать зависшую или нестабильно работающую систему приходится еще реже. Поскольку от сервера зависит работоспособность множества компьютеров, высокая надежность — очень важное качество операционной системы.

Серверным компьютерам требуется эффективное и надежное сетевое программное обеспечение. Сюда входят как отдельные серверные пакеты, так и компоненты ядра операционной системы, например драйверы сетевых плат и стек TCP/IP, отвечающий за выполнение сетевых функций. Во FreeBSD стек TCP/IP зарекомендовал себя с наилучшей стороны. В нем реализованы средства защиты, позволяющие фильтровать пакеты, в том числе на основании IP-адресов отправителя и получателя. Средства фильтрации постоянно совершенствуются в современных операционных системах и жизненно необходимы серверам, часто атакуемым хакерами. Помимо средств защиты самой FreeBSD многие серверные программы обладают собственными механизмами защиты.

## **1. 2 Лекция № 5-8 (8 часа).**

**Тема:** «Состав ОС FreeBSD (Интерактивная форма - 4ч)»

### **1.2.1 Вопросы лекции:**

1. FreeBSD и неродственные неродственные операционные системы
2. FreeBSD и другие разновидности UNIX
3. Взаимодействие с другими компьютерами по сети

### **1.2.2 Краткое содержание вопросов:**

1. Чтобы решить, какая операционная система является в вашем случае наиболее предпочтительной, следует разбить возможные варианты на две группы: UNIX-подобные

системы и остальные. FreeBSD относится к первой группе. Прежде всего нужно определить, имеет ли смысл устанавливать одну из разновидностей UNIX. Если ответ положителен, можно переходить к оценке преимуществ различных систем этой группы.

На сегодняшний день спектр существующих операционных систем чрезвычайно велик. Мы остановимся на наиболее популярных системах, функционирующих на платформах потребительского сегмента рынка (процессоры x86 и PowerPC).

· DOS. Дисковая операционная система (DiskOperatingSystem, DOS) была первой системой такого рода для компьютеров IBM PC и в различных формах «дожила» до сегодняшних дней. Все эти формы довольно примитивны в сравнении с современными операционными системами. Их основные достоинства — небольшой занимаемый размер (как на диске, так и в оперативной памяти) и способность выполнять приложения в режиме реального времени, что требует точной синхронизации.

· Windows 9x/Me. Это семейство Windows-систем больше не разрабатывается. В его основе лежит ядро DOS, в которое интегрированы средства графического интерфейса и вытесняющей многозадачности. Системы Windows 9x/Me достаточно ненадежны и имеют много других недостатков. Тем не менее они все еще широко распространены, в немалой степени из-за доступности популярных пользовательских приложений, работающих только в среде Windows, таких как MicrosoftOffice.

· Windows NT /2000/XP. Изначально Windows NT представляла собой полностью переработанную реализацию среды Windows. В последующих версиях системы улучшались средства пользовательского интерфейса и повышалась совместимость с популярными программами. В 2002 г. компания Microsoft позиционировала Windows 2000 как серверную и корпоративную ОС, а Windows XP — как систему для домашнего применения. Обе они основаны на Windows NT и гораздо надежнее семейства Windows 9x/Me, основанного на DOS. Считается, что FreeBSD надежнее, чем Windows NT/2000/XP, хотя у такого мнения, естественно, немало противников. Windows 2000 или XP будет хорошим выбором в том случае, если требуется запускать программы, работающие только в среде Windows.

· WindowsVista. Абсолютное перерождение линейки Windows. Vista представляет собой ОС нового поколения. По принципам работы и взаимодействия с оборудованием она превосходит все остальные ОС. Единственным камнем преткновения могут быть приличные системные требования. По надёжности Vista не уступает другим современным ОС.

· OS/2. Несмотря на то что компания IBM не выпускала новых версий OS/2 с 1996 г. (Warp 4), эта система остается вполне жизнеспособной. По надежности она сопоставима с Windows 2000/XP, вот только способна выполнять лишь старые 16-разрядные приложения Windows, DOS-приложения и программы, написанные специально для OS/2. В среду OS/2 было перенесено множество традиционного программного обеспечения UNIX. В 2001 г. лицензию на OS/2 приобрела компания SerenitySystems, которая выпустила обновленную версию под названием eComStation.

· BeOS. Система BeOS разрабатывалась «с нуля» с учетом последних достижений в области операционных систем, но так и не стала популярной. Круг ее пользователей продолжает сокращаться, хотя у системы остаются верные поклонники. BeOS отличается стабильностью и производительностью, однако число доступных программ невелико. Как и OS/2, BeOS имеет преимущества в той среде, где вычислительная инфраструктура давно

налажена. В остальных случаях использование системы неоправданно. В 2001 г. компанию Be, Inc., разработчика BeOS, приобрела компания Palm.

· Mac OS. Mac OS долгое время являлась единственным серьезным конкурентом Windows на рынке настольных систем, но оба семейства операционных систем работают на разном оборудовании: в случае Windows это процессоры x86, а в случае Mac OS — процессоры 680x0 и (с недавнего времени) PowerPC. Mac OS славится своим пользовательским интерфейсом и имеет преданную армию поклонников. Последние версии Mac OS, называемые Mac OS X, основаны на ядре UNIX с немалыми заимствованиями из FreeBSD, так что в определенном смысле MAC OS X можно считать разновидностью UNIX. Отличия, впрочем, остаются довольно существенными, особенно в плане пользовательского интерфейса, поэтому система и была отнесена к рассматриваемой категории. Mac OS X может оказаться хорошей альтернативой в том случае, если требуется система с надежным ядром UNIX и возможностью выполнять популярные пользовательские приложения (при условии, что они перенесены в Mac OS X). FreeBSD ориентирована в основном на платформу x86, поэтому недоступна для оборудования, на котором традиционно работает Mac OS X (в настоящее время ведутся разработки по переносу FreeBSD на платформу PowerPC).

Сравнивать FreeBSD с перечисленными операционными системами сложно, так как они довольно неоднородны. Остановимся на наиболее серьезных соперниках — Windows (преимущественно Windows 2000 и XP, хотя Windows 9x/Me тоже нужно учитывать из-за огромного количества инсталлированных систем) и Mac OS.

В сравнении с Windows достоинства FreeBSD проявляются в основном в серверной части. Некоторые из наиболее популярных серверов, в частности почтовый сервер sendmail и Web-сервер Apache, ориентированы на платформу UNIX и отлично работают во FreeBSD. В последние годы эти программы привлекали меньше внимания со стороны хакеров, чем их Windows-аналоги. Например, в 2001 г. Web-серверы Windows подверглись массированному нападению вирусов-червей типа CodeRed и Nimda. Впрочем, нельзя поручиться, что ситуация не изменится в будущем.

Популярность MicrosoftOffice делает FreeBSD менее предпочтительным выбором, чем Windows и Mac OS, когда речь заходит об офисных применениях, например о редактировании текста (по крайней мере, если необходимо обмениваться файлами с другими пользователями MicrosoftOffice). Альтернативные пакеты наподобие OpenOffice.org позволяют читать и записывать файлы в формате MicrosoftOffice, но могут возникать проблемы с форматированием. Если же совместимость с MicrosoftOffice не требуется, то шансы уравниваются. Как правило, предпочтение отдается той операционной системе, которая лучше поддерживает применяемое программное обеспечение.

Очевидными плюсами FreeBSD являются гибкость и сетевые возможности X-среды. Но если пользователи имеют опыт работы с Windows или Mac OS, интерфейс FreeBSD покажется им непривычным и не всегда удобным.

2. Если вы окончательно решили отдать предпочтение UNIX, а не Windows, Mac OS или другой системе, то встает следующий вопрос какую разновидность UNIX выбрать? FreeBSD — один из нескольких доступных вариантов, и нужно понимать, чем эта система отличается от других аналогичных систем. Во многих случаях любая UNIX-система вполне подходит для решения поставленных задач, но бывают ситуации, когда FreeBSD лучше (или хуже) справляется с конкретной задачей.

Как правило, в различных UNIX-системах используется одно и то же программное обеспечение. Например, во всех системах работают популярные серверы sendmail и Apache, а также X WindowSystem. Ниже перечислено то, что действительно отличает UNIX-системы друг от друга.

- Ядро. Ядро — это программный компонент, составляющий основу операционной системы. Оно служит интерфейсом между пользовательскими программами и аппаратными устройствами компьютера, управляет файлами и процессами и выполняет ряд других низкоуровневых задач. Таким образом, ядро обеспечивает стабильность операционной системы и определяет ее характеристики, например перечень поддерживаемых аппаратных устройств (некоторые устройства обслуживаются внешними драйверами).
- Аппаратная поддержка. UNIX-системы работают не только на платформе x86. Производительность процессоров x86 ставит их на нижнюю ступень, по крайней мере в семействе процессоров универсального назначения. Некоторые UNIX-системы ориентированы на суперкомпьютеры, и обычные операционные системы, такие как FreeBSD, не могут с ними конкурировать. Поддержка конкретных аппаратных устройств (модемов, сетевых и звуковых плат и т.д.) зависит от ядра и вспомогательных программ.
- Стратегия распространения. Некоторые разновидности UNIX распространяются как коммерческие продукты, но в основном они представляют собой системы с открытым исходным кодом. Это означает, что любой желающий может получить исходный код системы, модифицировать его и распространить дальше. Системы с открытым исходным кодом значительно дешевле в эксплуатации, чем их коммерческие аналоги. Впрочем, в 1990-е годы коммерческие разновидности UNIX упали в цене, а некоторые из них даже стали бесплатными для определенного круга применений.
- Стандартный набор и конфигурация программного обеспечения. UNIX существует уже около трех десятилетий, и за это время появилось множество версий популярных утилит, конфигурируемых самыми разными способами. В каждой UNIX-системе свой набор стандартных программ. В одних системах используются конфигурационные сценарии BSD-типа, в других — сценарии System V. Наиболее распространенным почтовым сервером является программа sendmail, но в некоторых системах работают другие серверы. Подобные различия определяют уникальность каждой операционной системы семейства UNIX. Если сравнивать UNIX и, скажем, Windows, то эти различия несущественны, но в то же время их более чем достаточно для того, чтобы при переходе из одной UNIX-системы в другую приходилось покупать справочное руководство.

Ядро FreeBSD является производным от открытой реализации исходного ядра AT&T UNIX. Изначально система предназначалась для процессоров семейства Intel x86, но впоследствии стала поддерживать родственные семейства процессоров компаний AMD, VIA (Cyrix), Transmeta и других (большей частью не существующих сегодня). Ведутся работы по переносу системы на другие процессоры, например Alpha, но они еще далеки от завершения.

Ядро FreeBSD отличается монолитностью, т.е. имеет единую логическую структуру. Вспомогательные компоненты, например драйверы файловой системы, помещаются в так называемое пространство ядра и потому могут контролировать аппаратные компоненты и структуры данных ядра. В некоторых разновидностях UNIX

применяется иная модель, называемая микроядром. В такой схеме ядро имеет меньший размер и взаимодействует с внешними компонентами, реализующими традиционные функции ядра, в частности доступ к файловой системе. У каждого подхода есть свои сторонники.

FreeBSD — это операционная система с открытым исходным кодом. Большинство ее компонентов распространяется на условиях лицензии BSD. Эта лицензия разрешает модификацию и последующее распространение исходного кода на тех условиях, которые выберет разработчик. Теоретически это означает, что кто угодно может модифицировать код системы и продать его как коммерческий продукт. Другой популярной лицензией для систем с открытым исходным кодом является GPL (GNU General Public License). Она требует, чтобы любые модификации распространялись с соблюдением исходных требований GPL. Такую лицензию имеет ядро Linux, и отсюда происходят многие различия между FreeBSD и Linux. Во FreeBSD используются программные компоненты, имеющие лицензию BSD, а в Linux больше GPL-компонентов. Это не влияет напрямую на работу операционной системы, но многие утилиты существуют в разных вариантах в зависимости от типа лицензии.

Некоторые разновидности UNIX патентованы. Например, системы Solaris компании Sun и AIX компании IBM являются коммерческими системами. Иногда их можно использовать бесплатно, как в случае Solaris для платформы x86, но чаще всего за лицензию коммерческой версии UNIX приходится платить. Различия между открытыми и коммерческими системами лежат в основном в концептуальной и экономической плоскостях. С функциональной точки зрения FreeBSD во многом эквивалентна своим коммерческим аналогам.

Коммерческие и открытые системы часто совместимы. Программы, имеющие открытый исходный код, можно запускать в коммерческой операционной системе, и наоборот: коммерческие программы можно запускать в открытой ОС. Тем не менее большинство программ FreeBSD имеет открытую лицензию.

Помимо FreeBSD широко распространены следующие разновидности UNIX.

- NetBSD. Это одна из трех операционных систем, производных от 386/BSD. Две другие — FreeBSD и OpenBSD. NetBSD отличается от FreeBSD тем, что перенесена на большее число процессоров. Принципы их лицензирования и общая структура весьма схожи, но FreeBSD остается более популярной, так как активнее рекламируется и чуть проще в инсталляции и применении.

- OpenBSD. Эта система славится своей безопасностью. Ее разработчики потратили немало усилий на то, чтобы создать систему, которую не так просто взломать. Как и NetBSD, OpenBSD схожа с FreeBSD по конфигурации, принципам лицензирования и доступному программному обеспечению.

- Linux. Linux можно считать отдельным семейством операционных систем с открытым исходным кодом. С технической точки зрения Linux — это лишь ядро; в дистрибутив Linux входят ядро и пакеты программ, составляющие операционную систему. По общим принципам конфигурирования и составу программных средств FreeBSD больше отличается от Linux, чем от NetBSD и OpenBSD. Стоит обратить внимание на то, что пользователи, не знакомые с UNIX, иногда считают FreeBSD дистрибутивом Linux. Это не так. FreeBSD не использует ядро Linux и отличается множеством конфигурационных нюансов. FreeBSD следует считать производной от

дистрибутива BSD (Berkeley Software Distribution), тогда как Linux является свободно распространяемой, полностью переработанной реализацией UNIX, гораздо слабее связанной с ранними версиями BSD и AT&T UNIX.

- Solaris. Это коммерческая разновидность UNIX, выпущенная компанией Sun и используемая в основном на ее серверных платформах. Серверы Sun, работающие на процессорах SPARC, являются конкурентами высокопроизводительных систем на базе процессоров x86 старших моделей. Операционная система Solaris может работать и на платформе x86, однако набор ее драйверов меньше, чем у FreeBSD. Система Solaris удобна для разработчиков, которые пишут программы для компьютеров Sun. В остальных случаях более предпочтительным выбором для персональных компьютеров являются FreeBSD и Linux.

- AIX. Это коммерческая разновидность UNIX, выпущенная компанией IBM для своего специализированного оборудования. Подобно системам компании Sun, AIX нацелена на верхний сегмент рынка серверов, поэтому мало конкурирует с FreeBSD.

- IRIX. Большинство рабочих станций Silicon Graphics работает под управлением IRIX — коммерческой разновидности UNIX. Ранее основным достоинством этих систем была оптимизированная обработка высококачественной графики, с их помощью создавались спецэффекты для кинофильмов. Сегодня в этот сегмент рынка активно вторгаются недорогие компьютеры на базе процессоров x86, но позиции Silicon Graphics и IRIX все еще сильны.

В целом основными конкурентами FreeBSD являются другие BSD-системы и Linux. FreeBSD популярнее своих «родственников» на платформе x86, поэтому активнее поддерживается сторонними разработчиками. Эта система разрабатывалась с акцентом на простоту инсталляции и администрирования; здесь она опережает NetBSD и OpenBSD. Достоинством OpenBSD является улучшенная безопасность, а NetBSD поддерживает не только процессоры x86. Что касается Linux, то у нее больше отличий в плане аппаратной и программной поддержки и ее лицензирование основывается на несколько иных принципах. Для новичков, которым требуется UNIX-система универсального назначения с открытым исходным кодом, FreeBSD и Linux подойдут в одинаковой степени. Но Linux развивается гораздо хаотичнее, и различия между ее многочисленными дистрибутивами часто сбивают с толку неопытных пользователей.

FreeBSD вряд ли можно считать конкурентом коммерческих разновидностей UNIX, ориентированных на компьютеры, которые превосходят по производительности системы на базе процессоров x86. Если в организации используются высокопроизводительные серверы Sun, IBM или SGI в сочетании с недорогими персональными компьютерами, то на последних можно установить FreeBSD, поскольку процедуры администрирования FreeBSD и коммерческих разновидностей UNIX во многом схожи. Иногда системы x86 старших моделей напрямую конкурируют с системами Sun, IBM и SGI младших моделей. Достоинством первых является низкая цена, но для них сложнее заключать контракты на сервисное обслуживание.

## Программная среда

До этого мы рассматривали, какое место занимает FreeBSD в ряду других операционных систем. Теперь нужно выяснить, как она взаимодействует с другими операционными системами. Такое взаимодействие может происходить как между компьютерами сети, так и в пределах одного компьютера. Полезно также узнать кое-что

об истории FreeBSD, чтобы понять главные принципы ее разработки, ее слабости и недостатки.

## История разработки FreeBSD

История FreeBSD, как и UNIX вообще, довольно запутанна. Она начинается в 1969 г. с появления первой (довольно примитивной) версии UNIX, разработанной компанией AT&T. Далее свой вклад в систему вносили энтузиасты Калифорнийского университета в Беркли и многочисленные добровольцы, существующий код неоднократно менялся, пополнялся набор системных утилит. Исходная версия AT&T UNIX породила несколько вариантов (или ветвей, как их еще называют). Хорошие идеи, появлявшиеся в одной ветви, часто воплощались в других ветвях. В истории UNIX были периоды, когда поставщики систем работали над устранением различий между ветвями. Тем не менее сегодня UNIX — это не одна операционная система, а целое семейство.

Стоит заметить, что с технической точки зрения название UNIX относится к официальным версиям именно этой системы. Иногда официальные версии обозначают словом UNIX (все прописные), а другие однотипные системы называют Unix (с заглавной буквы). Но такое различие почти незаметно, поэтому в данной книге словом UNIX обозначаются официальные версии либо все семейство в целом, а когда нужно подчеркнуть, что речь идет также о других подобных операционных системах, например Linux, употребляется выражение «UNIX-подобные системы».

Итак, в 1969 г. Кен Томпсон (Ken Thompson) и Деннис Риччи (Dennis Ritchie) написали первую версию UNIX. Очередной крупной вехой на пути к появлению FreeBSD стала установка четвертой версии UNIX в Калифорнийском университете в Беркли в 1974 году. В течение следующих нескольких лет преподаватели, научные сотрудники и студенты университета знакомились с операционной системой, писали программы для нее и даже модифицировали саму систему. В 1977 г. все наработки были включены в единый пакет программ, названный BSD (Berkeley Software Distribution).

В 1978 г. был выпущен пакет 2BSD, содержащий улучшенный и расширенный набор утилит исходного дистрибутива. Последующие разработки привели к появлению пакетов 2.11BSD, 3BSD, 4BSD. На очереди было название 5BSD, однако компания AT&T возражала против этого, т.к. боялась потенциальной пуганицы с System V — своим вариантом UNIX. В результате следующие версии стали называться 4.1BSD, 4.2BSD и т.д. В это время и появился TCP/IP — базовый стек сетевых протоколов, лежащих в основе современной сети Internet.

Упомянутые BSD-пакеты представляли собой дополнения к ОС UNIX компании AT&T. Это не были полноценные операционные системы. Кроме того, лицензионные требования постоянно ужесточались. Один из вариантов пакета 4.3BSD, известный как 4.3BSD-Tahoe, был доступен только при условии приобретения исходной лицензии AT&T, которая определяла правила распространения ОС UNIX в семидесятые и восьмидесятые годы. Но в 1989 г. ситуация изменилась: разработчики из Беркли выпустили пакет Networking Release 1, который допускал свободное распространение кода независимо от того, имел ли получатель лицензию на исходный код AT&T. Это был первый и очень важный шаг на пути к современным системам с открытым исходным кодом. Без изменений в правилах лицензирования, впервые воплощенных в Networking Release 1, не было бы современной FreeBSD.

В 1991 г. появился пакет NetworkingRelease 2. Несмотря на сходство названий, изменения оказались разительными. Это была практически полноценная операционная система; не хватало лишь шести файлов с исходными кодами ядра AT&T UNIX. В течение шести месяцев программист Билл Джолиц (BillJolitz) переписал их, назвав полученный пакет 386/BSD. Эта система является непосредственной предшественницей NetBSD и FreeBSD (OpenBSD отделилась от ветви NetBSD в середине 90-х). Она предназначалась для работы на процессорах Intel x86, точнее, 80386 — самой последней на то время модели. Как уже отмечалось выше, FreeBSD по-прежнему тесно связана с платформой x86, хотя и предпринимаются попытки ее переноса на другие процессоры.

С 1993 г. (дата рождения системы как отдельного проекта) FreeBSD целенаправленно развивается независимо от других систем. Она, конечно же, заимствует что-то из других проектов с открытым исходным кодом; иногда заимствуют ее код. Старший номер версии увеличивается в среднем каждые несколько лет. В начале 2002 г. Была выпущена версия 4.5, а версия 5.0 появилась в конце того же года.

С самого начала разработки BSD-систем возникла потребность в сосуществовании UNIX с другими операционными системами на одном компьютере. Например, на PDP 11/45, первом UNIX-компьютере в Беркли, ОС UNIX выполнялась лишь треть рабочего времени. Остальное время отводилось другой операционной системе (RSTS). Но для компьютера, работающего в режиме разделения времени, это чрезвычайно неуклюжий подход, поэтому на большинстве UNIX-компьютеров в 1970 — 1980 гг. выполнялась одна операционная система. С появлением 386/BSD вопрос сосуществования различных операционных систем стал гораздо острее. Многие программисты-любители имели всего один компьютер и часто сталкивались с необходимостью писать программы для более популярной DOS (позднее — Windows). Такое положение дел сохранилось и сегодня. Если FreeBSD устанавливается на выделенный компьютер (сервер или специализированную рабочую станцию), то задача упрощается, так как использовать другие операционные системы не предполагается. Если же планируется время от времени запускать программы в другой ОС, то нужно разобраться, как FreeBSD работает с жесткими дисками и как на персональных компьютерах загружается операционная система.

Одна из ключевых концепций мультисистемных компьютеров — разделы. Если представить жесткий диск как шкаф для бумаг, то раздел — это один из его ящиков, часть большого документохранилища, содержащая взаимосвязанные файлы. Разделы создаются при подготовке диска к использованию. Это называется разбивкой на разделы. После того как раздел создан, его не так-то легко модифицировать. Следовательно, важно продумать такую структуру жесткого диска, которая в обозримом будущем отвечала бы потребностям пользователя.

Есть утилиты, позволяющие менять размеры существующих разделов. Это, в частности, коммерческие программы PartitionMagic и PartitionCommander. Среди программ с открытым исходным кодом назовем FIPS и GNUMParted. Это более простые программы с ограниченными возможностями, но свою работу они выполняют. Особенно полезны они в том случае, когда FreeBSD устанавливается в системе, где уже инсталлирована DOS, Windows, OS/2, Linux или другая операционная система.

За много лет были придуманы различные схемы разбивки на разделы. В основном они разрабатывались независимо друг от друга для конкретных аппаратных платформ. На персональных компьютерах поддерживается три типа разделов. Их существование не так

уж необходимо, просто предыдущая схема была расширена, когда стало очевидно, что ее возможности исчерпаны.

Первичный раздел. Исходный тип раздела персональных компьютеров теперь называется первичным разделом. Всего таких разделов может быть четыре. Не некоторые операционные системы, такие как DOS и Windows, должны загружаться с первичного раздела, расположенного на первом физическом диске. FreeBSD тоже должна находиться в первичном разделе, но он может быть не первым. Есть операционные системы, например Linux и OS/2, которые могут загружаться с расширенного раздела. Все эти нюансы нужно учитывать, планируя инсталляцию системы.

· Расширенный раздел. В определенном смысле расширенный раздел представляет собой всего лишь особый тип первичного раздела. Он занимает одну из четырех доступных областей первичных разделов и служит для последующей разбивки на логические разделы.

· Логический раздел. Логические разделы создаются внутри расширенных. Запись для расширенного раздела в таблице разделов необходима для того, чтобы зарезервировать место под один или несколько логических разделов. Их число ограничивается емкостью жесткого диска и способом адресации разделов, применяемым в операционной системе. Например, в DOS и Windows разделы помечаются буквами (именами дисков), поэтому теоретически их может быть 26 (число букв английского алфавита). В UNIX-подобных системах число логических разделов зависит от ядра и содержимого каталога /dev. В любом случае редко какая система даже приближается к теоретическому пределу. Поскольку логические разделы находятся в расширенном разделе, они должны последовательно занимать смежные области жесткого диска.

В DOS и Windows жесткий диск часто представляет собой один большой первичный раздел. Но если на одном компьютере требуется установить FreeBSD и какую-то другую операционную систему, то придется создавать дополнительные разделы. В частности, под FreeBSD должен быть отведен один из первичных разделов. Одна из возможных конфигураций изображена на рис. 1.1. В данном случае FreeBSD и Windows занимают по одному первичному разделу, а в расширенном разделе Windows созданы два логических диска.

Рис. 1.1. Разбивка на разделы позволяет отделять операционные системы друг от друга и создавать области хранения для различных типов данных

Компьютеры на базе новых 64-разрядных процессоров Intel IA64 поддерживают дополнительную, более гибкую схему разбивки на разделы.

В самой FreeBSD ситуация намного сложнее. Прежде всего нужно понять, что в UNIX-системах разделы часто используются для обособленного хранения различных типов данных. Это позволяет монтировать каталоги с различными параметрами и защищать данные в случае переполнения или повреждения других разделов. Например, каталог /home, хранящий файлы пользователей, часто отделяют от корневого каталога (/). Во FreeBSD, как и в других разновидностях UNIX, каждый раздел монтируется к определенному каталогу. Нет логических дисков, как в DOS или Windows. По этой причине FreeBSD-системы часто требуют нескольких разделов, даже если на компьютере установлена только FreeBSD.

Традиционно во FreeBSD применяется схема разбивки на разделы, отличающаяся от стандартной схемы для персональных компьютеров. Обычно разбивка осуществляется в пределах одного первичного раздела. Его подразделы аналогичны логическим дискам расширенного раздела. Работать с подразделами может только BSD-система.

Если нужно обмениваться данными с другими операционными системами, лучше выделить для этого отдельный раздел. К сожалению, в разных операционных системах для хранения данных применяются разные файловые системы. Файловая система, занимающая целый раздел, ведет учет файлов, следит за их расположением и управляет служебной информацией, такой как дата создания файла.

Как правило, для перекрестного обмена данными подходит файловая система FAT(FileAllocationTable) из DOS, поскольку ее поддерживают все распространенные операционные системы персональных компьютеров.

Лишь немногие системы, не относящиеся к семейству UNIX, поддерживают FFS (FastFileSystem), собственную файловую систему FreeBSD.

Вообще говоря, в разных версиях UNIX применяется несколько иная реализация FFS, так что, даже если планируется установить на одном компьютере вместе с FreeBSD еще одну разновидность UNIX, может потребоваться использовать FAT в качестве общей среды обмена файлами.

### **1. 3 Лекция № 9-12 (8 часов).**

**Тема:** «Команды работы с файлами»

#### **1.3.1 Вопросы лекции:**

1. Аппаратная среда
2. Типы и производительность процессоров
3. Требования к оперативной памяти

#### **1.3.2 Краткое содержание вопросов**

1. Прежде чем инсталлировать FreeBSD, нужно узнать аппаратные требования этой системы. Платформа x86 чрезвычайно разнообразна, поэтому в компьютерах часто присутствуют компоненты, не поддерживаемые во FreeBSD. Как правило, это не мешает инсталлировать систему, но ее возможности будут ограничены. Если в процессе инсталляции системы будет обнаружено неподдерживаемое устройство, то следует заменить его, прежде чем продолжать инсталляцию. Это позволит избежать последующего переконфигурирования системы.

2. Сердцем» компьютера является центральный процессор. Как уже говорилось выше, FreeBSD изначально предназначалась для работы на процессорах Intel x86, по крайней мере на 32-разрядных процессорах, начиная с 80386. В 2002 г. FreeBSD поддерживала процессоры Intel 80386, 80486, Pentium, PentiumPro, PentiumII, Pentium III, Pentium 4 и серию Celeron со всеми разновидностями. Помимо этого FreeBSD работает на аналогичных процессорах других компаний, таких как AMD, VIA, Transmeta, Cyrix, IDT и NexGen (последние три обанкротились, но их разработки внедрены в процессоры AMD и VIA). Некоторые компании, например Evergreen, продают комбинированные системы и

позволяют «разгонять» старые материнские платы до ранее недоступных скоростей. FreeBSD работает с такими процессорами, но могут возникать трудности с выбором материнских плат.

Существуют проекты по переносу FreeBSD на другие платформы, в том числе Intel IA64, AMD x86-64 (ответ компании AMD на архитектуру IA64), Alpha компании Compaq, PowerPC (PPC) компании Motorola и UltraSparc компании Sun. В настоящее время архитектура x86 остается наиболее популярной, но появились признаки того, что со временем она будет вытеснена 64-разрядными платформами. Компания Intel продвигает процессоры платформы IA64 под названием Itanium. Компания AMD готовит конкурирующую платформу x86-64 (K8, или Hammer). Эти процессоры, без сомнения, будут играть очень большую роль в недалеком будущем. К счастью, функционирование FreeBSD не зависит от используемой платформы, хотя тип процессора нужно учитывать при выборе скомпилированных программ. При инсталляции FreeBSD на альтернативных платформах может меняться порядок разбивки на разделы и установки системы. Впрочем, в этой курсовой работе мы сосредоточимся на платформе x86 как наиболее популярной.

Что касается скорости работы процессора, то для FreeBSD здесь нет ограничений. Система работает даже на самых низкоскоростных процессорах 80386, хотя такое применение в наши дни мало кого заинтересует. Систему на базе процессора 80386 можно использовать разве что в качестве выделенного брандмауэра для домашней сети или DHCP-сервера небольшой сети. Для обработки графики и выполнения современных ресурсоемких программ производительности такого процессора недостаточно. Подбирай комплектацию нового компьютера, старайтесь устанавливать самые новые процессоры компаний Intel, AMD, VIA или Transmeta. Если планируется запускать программы, предъявляющие повышенные требования к производительности, выбирайте более новые модели процессоров.

Цена на процессоры резко снижается при переходе от самой последней модели к предыдущей и гораздо меньше — при последующем снижении скорости процессора. Таким образом, оптимальным выбором с точки зрения соотношения цена/производительность часто являются процессоры промежуточного звена. Планка «промежуточного» уровня повышается чуть ли не ежемесячно, так что конкретные рекомендации по поводу выбора оптимального процессора давать сложно.

Как и в отношении центрального процессора, требования к оперативной памяти, предъявляемые во FreeBSD, умерены по современным меркам. В официальной документации говорится, что для инсталляции системы требуется минимум 5 Мбайт ОЗУ, а для работы самой системы требуется минимум 4 Мбайт. Сегодня это кажется смехотворным. Такие объемы памяти достаточны лишь для самых тривиальных приложений. Современные компьютеры оснащаются минимум 128 Мбайт ОЗУ. Этого более чем достаточно для инсталляции FreeBSD, загрузки графической среды и запуска большинства серверов (хватит даже 64 Мбайт). Но в определенных ситуациях требуется дополнительная память.

· Многопользовательская среда. Если в системе одновременно работает множество пользователей (например, по сети), то для обслуживания пользовательских приложений и графических оболочек нужна дополнительная память.

· Крупные серверы. Для большинства серверов достаточно 64 Мбайт ОЗУ, но требования к памяти повышаются по мере роста вычислительной нагрузки. Если на одном компьютере функционируют файловый сервер, Web-сервер, почтовый сервер и все они

обслуживают сотни или тысячи пользователей, то минимально допустимый объем оперативной памяти будет гораздо больше, чем 128 Мбайт. Конкретный показатель зависит от предполагаемого уровня загруженности.

· Ресурсоемкие программы. Некоторые программы, например, анимационные или моделирующие пакеты, требуют больших объемов памяти, даже если с ними работает один пользователь. Конкретные значения указаны в документации к программе. Как правило, с повышением требований к памяти возрастают и требования к скорости центрального процессора, но так бывает не всегда.

## **1.4 Лекция № 13-16 (8 часов).**

**Тема:** «Команды работы с каталогами»

### **1.4.1 Вопросы лекции:**

1. Жёсткие диски. Диски SCSI и EIDE
2. Общие требования к объёму дискового пространства
3. Графическое и сетевое оборудование

### **1.4.2 Краткое содержание вопросов:**

1. При инсталляции операционной системы нужно учитывать объем дискового пространства. Выше уже описывались принципы разбивки на разделы, применяемые во FreeBSD. Но прежде чем приступить к инсталляции, следует выбрать тип жесткого диска и оценить общий объем дискового пространства, отводимый под операционную систему.

На многих современных персональных компьютерах установлены жесткие диски EIDE (Enhanced Integrated Drive Electronics). Материнские платы обязательно содержат контроллеры EIDE, поэтому подключение дисков не представляет проблемы. Но существует и альтернативный тип жестких дисков: SCSI (Small Computer System Interface). Очень немногие материнские платы оснащены соответствующими разъемными соединениями, поэтому для работы со SCSI дисками нужно установить SCSI-адаптер (обычно это PCI-плата). FreeBSD поддерживает оба интерфейса — SCSI и EIDE. У каждого из них свои достоинства и недостатки.

Основное преимущество стандарта EIDE — стоимость. Поскольку соответствующие контроллеры встроены в материнские платы, их не приходится покупать, как в случае SCSI (контроллер можно добавить для повышения производительности или при наличии в системе более четырех жестких дисков). Жесткие диски EIDE тоже недороги в сравнении с аналогичными SCSI-дисками. Общая экономия составляет от 10 до 50 долларов, иногда даже больше. Все контроллеры EIDE поддерживают единый минимальный стандарт низкоуровневого форматирования жестких дисков, по этому диски работают на любых контроллерах, по крайней мере в минимальной низкоуровневой конфигурации. Но чтобы обеспечить наилучшую производительность, следует использовать жесткий диск, рассчитанный на имеющийся контроллер.

Основным недостатком EIDE является то, что к одному шлейфу (кабель, подключающийся с другой стороны к материнской плате или контроллеру EIDE) можно подключить лишь два EIDE-устройства. В большинстве материнских плат всего два

шлейфа, что обеспечивает подключение четырех EIDE-устройств. Это вызывает очень большие неудобства, если в системе есть дисководы CD-ROM, съемные диски, ленточные накопители и т.д. Обойти ограничение можно, добавив еще один контроллер, но ему требуются отдельный PCI-слот и линия запроса прерывания, а это тоже ограниченные ресурсы. Кроме того, стандарт EIDE разрешает одновременный доступ лишь к одному устройству на шлейфе, поэтому в системах с большим числом пользователей EIDE-устройства становятся узким местом.

Раньше контроллеры EIDE поддерживали лишь режимы доступа PIO(ProgrammedInput/Output), которые требуют участия процессора в операциях обмена данными. Это делало стандарт EIDE малопригодным в случае многозадачной операционной системы, такой как FreeBSD. Современные контроллеры поддерживают более оптимальные режимы DMA (DirectMemoryAccess), при которых контроллер может записывать данные непосредственно в память. Эта поддержка должна также обеспечиваться драйвером EIDE.

SCSI — гораздо более гибкий интерфейс в сравнении с EIDE. Он допускает подключение от 7 до 15 устройств к одному шлейфу, в зависимости от разновидности интерфейса, что делает его гораздо более приемлемым вариантом для компьютера с большим числом физических дисков. SCSI поддерживает одновременный доступ к нескольким устройствам одного шлейфа, а жесткие диски SCSI обычно работают быстрее, чем диски EIDE аналогичной емкости. Для SCSI предлагается более широкий диапазон устройств. Например, существуют сканеры SCSI, а ленточные накопители моделей верхнего уровня почти всегда имеют интерфейс SCSI. Большинство SCSI-адаптеров всегда работает в режиме DMA.

Недостатком SCSI является стоимость. SCSI-адаптер дорог сам по себе (от 50 до 100 долларов), к тому же SCSI-устройства обычно дороже своих EIDE-аналогов. В интерфейсе SCSI не предусмотрен минимальный стандарт низкоуровневого форматирования, поэтому FreeBSD должна поддерживать тот SCSI-адаптер, который установлен в системе.

В целом стоимость EIDE-дисков позволяет применять их в системах нижнего и среднего уровня. Если в компьютере установлен EIDE-контроллер, для которого FreeBSD поддерживает режим DMA, то однодисковая система обеспечит хорошую производительность. В случае крупного сервера, когда требуется улучшенная производительность работы с диском, повышенная стоимость SCSI может оказаться оправданной.

На одном компьютере разрешается использовать оба интерфейса. Например, к SCSI-контроллеру могут быть подключены высокопроизводительные жесткие диски и ленточный накопитель, а к EIDE-контроллеру — недорогие дисководы CD-ROM и Zip.

В будущем важную роль начнут играть другие дисковые интерфейсы. В частности, интерфейс IEEE-1394 (FireWire) может заменить SCSI, по крайней мере в некоторых случаях. Жесткие диски FireWire доступны уже сегодня, однако поддержка этого интерфейса во FreeBSD 5.0 весьма ограничена.

Чтобы выяснить, какие EIDE-контроллеры и SCSI-адAPTERы поддерживаются, просмотрите файл HARDWARE.TXT, входящий в комплект поставки FreeBSD. Учтите, что в нем указываются микропроцессорные наборы, а информация о конкретных моделях устройств может быть не приведена. Посмотрите названия микросхем на PCI-плате, чтобы

узнать, с чем вы имеете дело. Если на компьютере уже инсталлирована Windows, выберите элемент Система (System) в панели управления. Затем щелкните на вкладке Устройства (DeviceManager) и раскройте список Контроллеры жестких дисков (Harddiskcontrollers). В компьютере установлен EIDE-контроллер VIA (IDE — это старый стандарт, трансформировавшийся в EIDE, но в некоторых утилитах используется старая терминология). Причём, нет необходимости искать драйверы для конкретных устройств EIDE или SCSI. К примеру, если поменять один EIDE-дисковод CD-ROM на другой, тоже EIDE, то обновлять драйверы FreeBSD не понадобится, поскольку оба дисковода используют стандартизированный набор команд.

Драйверы нужны для конкретных типов устройств, таких как жесткий диск или дисковод CD-ROM. Эти драйверы включены в стандартное ядро FreeBSD, поэтому применяются практически незаметно для пользователя.

2. Важно выяснить возможности EIDE-контроллера или SCSI-адаптера системы, чтобы избежать неприятных проблем с дисками. Но независимо от типа интерфейса остается вопрос, связанный с общим объемом дискового пространства. Официально для FreeBSD требуется 100 Мбайт дисковой памяти. Но это минимально необходимый объем. Возможности такой системы крайне ограничены. Обычно требуется не менее 1 Мбайт. Если же инсталлируется много программного обеспечения, то даже 2 или 3 Гбайт не будут излишними. Дополнительное пространство может понадобиться для хранения пользовательских или серверных данных. Потребности в дисковой памяти определяются функциями, которые выполняет система.

Современные жесткие диски имеют емкость не менее 20 Гбайт, поэтому при инсталляции FreeBSD никаких трудностей не возникает. Если же FreeBSD инсталлируется в дополнение к уже установленной системе, то есть два варианта.

- Повторная разбивка диска на разделы. Можно воспользоваться специальными утилитами, например, коммерческой программой PartitionMagic или свободно распространяемой программой FIPS, для сокращения размеров существующего раздела и выделения пространства под FreeBSD. Это эффективный и экономный подход, если на диске достаточно свободного пространства, но одновременно и не лишенный риска. Утилиты динамической разбивки на разделы иногда сбоят, вследствие чего происходит потеря данных. Прежде чем запускать такую утилиту, создайте резервную копию существующих данных.

- Добавление нового диска. Можно добавить новый жесткий диск, который будет использоваться либо исключительно под FreeBSD, либо совместно с текущей операционной системой. Такой подход безопаснее, чем изменение существующих разделов и добавление новых на одном диске, но на практике он может оказаться неосуществимым из-за того, что уже достигнут предел возможного числа дисковых устройств.

В любом случае при инсталляции FreeBSD нужно установить системный загрузчик. Это очень простая программа, загружающая одну или несколько операционных систем. Загрузчик FreeBSD позволяет выбрать систему, запускаемую по умолчанию при включении питания. Эту программу можно применять в сочетании со сторонними загрузчиками, такими как BootMagic компании PowerQuest, SystemCommander компании V-Сомили или LILO операционной системы Linux.

3 Чтобы на экране появилось изображение, FreeBSD должна поддерживать имеющуюся плату видеоадаптера. Что касается монитора, то он очень редко вызывает проблемы и не требует наличия специальных драйверов.

Все современные видеоплаты поддерживают определенные базовые режимы, в которых может работать и FreeBSD. Это, в частности, текстовый режим и режим с низким разрешением VGA (640x480). Иногда поддерживаются стандартные графические режимы с более высоким разрешением. Текстовый режим можно применять, например, на сервере, но пользователи рабочих станций предпочитают графическую среду. VGA-режимы редко удовлетворяют современных пользователей, поэтому для серьезной системы необходима более мощная графическая плата, поддерживаемая FreeBSD (точнее, X-средой).

Графические средства, хоть и включены в стандартный комплект FreeBSD, в действительности являются частью независимого проекта: XFree86. Пакет XFree86 используется и другими UNIX-системами с открытым исходным кодом, в частности NetBSD, OpenBSD и Linux. Он также доступен в коммерческих разновидностях UNIX (в них обычно имеются еще и собственные X-серверы) и даже в системах других типов, включая Windows, Mac OS и OS/2.

Во FreeBSD подсистема XFree86 взаимодействует большей частью непосредственно с аппаратными устройствами, поэтому она должна поддерживать имеющуюся видеоплату. Если средства поддержки реализованы не в полном объеме, производительность графической подсистемы существенно снизится. Это может привести к тому, что FreeBSD обеспечит меньшее разрешение, чем в Windows, или меньшую частоту обновления экрана. Все современные графические микропроцессорные наборы располагают средствами ускорения основных графических операций, таких как перемещение объектов по экрану. Если же их драйверы несовершенны, эти средства могут не поддерживаться, вследствие чего пострадает производительность графической подсистемы.

В зависимости от версии FreeBSD в систему может быть включен старый набор драйверов XFree86, поэтому в этом случае нужно обновить пакет XFree86 или хотя бы драйвер видеоплаты.

Как и в случае устройств EIDE и SCSI, список поддерживаемых видеоплат формируется на основе микропроцессорных наборов, хотя изготовители видеоплат не всегда публикуют эту информацию. Следовательно, чтобы собрать необходимую информацию, нужно посмотреть надписи на микросхемах видеоплаты или, если на компьютере установлена Windows, просмотреть информацию о видеоадаптере в панели управления. Некоторые изготовители популярных видеоплат выпускают собственные наборы микросхем, поэтому найти драйверы для них несложно.

Если вы собираетесь приобрести видеоплату для установки в компьютер с ОС FreeBSD, остерегайтесь самых последних моделей. Разработчики XFree86 часто отстают на несколько месяцев, поэтому последние модели могут поддерживаться плохо или вообще не поддерживаться.

В процессе инсталляции система пытается обнаружить и сконфигурировать видеоплату. Если эта процедура выполняется с ошибками или возникает необходимость изменить конфигурацию платы, то нужно отредактировать файлы конфигурации XFree86.

Многие компьютеры, работающие под управлением FreeBSD, играют роль серверов или по крайней мере сетевых рабочих станций. В настоящее время наиболее распространенным сетевым стандартом является Ethernet. Существует много версий этого стандарта, отличающихся двумя ключевыми особенностями.

· Скорость передачи данных. Самая низкая скорость в сетях Ethernet— 10 Мбит/с. Самая высокая из доступных на сегодняшний день — 1000 Мбит/с (1 Гбит/с). Сейчас наиболее распространены стомегабитные сети, хотя гигабитное технологии завоевывают все большую популярность.

Среда передачи данных. В сетях Ethernet данные передаются по кабелям различных видов. Раньше использовались две разновидности коаксиального кабеля: толстый и тонкий. Оба напоминают провода, используемые в кабельном телевидении, но отличаются по толщине. Такие сети имеют шинную топологию, в которой сетевой кабель последовательно соединяет устройства друг с другом. Коаксиальные кабели сейчас используются редко. Их вытеснила витая пара, напоминающая телефонный провод, но с более широкими разъемными соединениями. В таких сетях имеется центральное устройство, называемое концентратором или коммутатором, которое соединяет все остальные устройства по топологии «звезда». В последнее время все активнее используются оптоволоконные кабели в качестве альтернативы витой паре в гигабитных сетях Ethernet. Такие сети, как правило, тоже имеют звездообразную топологию.

Большинство современных сетей создается на основе стомегабитной витой пары, хотя иногда применяются старые десятимегабитные кабели (витая пара или коаксиал). Если нужна более высокая скорость работы, следует перейти на гигабитные технологии.

К счастью, для FreeBSD не важны ни тип кабелей, ни топология. Нужно лишь убедиться в том, что сетевая плата Ethernet поддерживается системой (в некоторых материнских платах верхнего уровня имеется встроенная поддержка Ethernet). В уже упоминавшемся выше файле HARDWARE.TXT указано, какие Ethernet-платы поддерживаются во FreeBSD. Как и в случае жестких дисков и видеоадаптеров, в файле описывается поддержка микропроцессорных наборов, установленных на плате, поэтому придется изучать надписи на микросхемах или, если есть возможность, просмотреть информацию в панели управления Windows.

FreeBSD должна поддерживать лишь сетевую плату, тогда как поддержка коммутатора или концентратора не требуется. Что касается серверного сетевого оборудования, то FreeBSD должна поддерживать конкретный тип сервера, например, файловый сервер или сервер печати.

В последнее время широкую популярность приобретают беспроводные устройства, особенно в домашних и небольших офисных системах. Многие из этих устройств поддерживаются во FreeBSD, и с логической точки зрения они функционируют примерно так же, как и обычные платы Ethernet. Информация о доступных беспроводных устройствах содержится в том же файле HARDWARE.TXT. Многие сетевые установки можно задать на этапе инсталляции системы.

Еще один тип сетевых устройств заслуживает особого внимания: модемы. Модемы служат средством подключения компьютеров к сети Internet через маршрутизаторы провайдеров. FreeBSD поддерживает все внешние модемы RS-232 и некоторые внутренние модемы. Впрочем, многие внутренние модемы часто являются программными. Для этих устройств нужен специальный драйвер, который выполняет

функции, в обычных модемах реализованные на аппаратном уровне. Такие драйверы — редкость для FreeBSD. Сколько бы ни стоили такие модемы, их лучше избегать.

Модемы, предназначенные для широкополосного доступа в Internet (кабельные и DSL-модемы), часто имеют Ethernet-интерфейсы. Для FreeBSD эти устройства выглядят как часть обычной сети Ethernet и потому не требуют специальных драйверов. Но у этого правила есть два исключения. Первое: если модем является внутренним или имеет интерфейс USB, для него нужен отдельный драйвер, которых практически нет для FreeBSD. Второе: некоторые провайдеры широкополосного доступа применяют протокол PPPoE (PPP over Ethernet) для назначения IP-адресов. Сам по себе такой протокол не нужен широкополосному модему, но он поддерживается во FreeBSD.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ**

### **2.1 Практическое занятие № 1-4 (8 часов).**

**Тема:** «Знакомство с интерфейсом операционной системы FreeBSD»

#### **2.1.1 Задание для работы:**

- 1.Знакомство с интерфейсом FreeBSD
2. Знакомство с принципами работы

#### **2.1.2 Краткое описание проводимого занятия:**

1. После входа пользователя в систему запускается программа, предоставляющая пользователю командную строку. Это так называемая оболочка (shell). Задача оболочки — дать возможность пользователю запускать на выполнение программы. Кроме того, в shell встроен язык программирования, так называемый shellscript. На этом языке можно писать командные файлы аналоги .bat файлов в MS Windows. В дальнейшем будем называть программы, написанные на этом языке просто скриптами. За время существования UNIX было написано большое количество программ оболочек. Самая первая — это BourneShell (программа называется sh). Язык программирования, встроенный в нее не похож на язык C.

Наиболее известная переменная — это PATH. Во FreeBSD есть специальная переменная, в которой хранится имя оболочки, в которой Вы сейчас работаете — SHELL. Для того, чтобы посмотреть ее содержимое можно воспользоваться программой echo, которой в качестве аргумента в командной строке следует указать переменную, перед именем которой стоит специальный символ «\$». В FreeBSD учитывается регистр букв. То есть если Вы напишите file или File — это будут два разных слова. В командной строке FreeBSD, аргументы командной строки обязательно отделяются друг от друга пробелами или табуляциями. Нельзя, как это принято в Windows, писать параметр сразу после имени программы. Итак, чтобы посмотреть содержимое переменной среды окружения SHELL, необходимо выполнить следующую команду:

```
$ echo $SHELL /bin/csh $
```

В результате, на экран терминала был выведен полный путь к файлу shell. Символ «\$» в начале строки — это приглашение командной строки. Если Вы будете вводить команды в командной строке, «\$» писать не надо. В Вашей системе приглашение командной строки может выглядеть по-другому, все зависит от настроек.

Для завершения работы в системе, можно воспользоваться одной из перечисленных ниже команд: `logout` или `exit`, комбинация клавиш «`Ctrl+D`». `Logout` — это классическая команда, которую можно использовать в любой оболочке. Будет ли работать команда `exit` или комбинация клавиш «`Ctrl+D`», зависит от реализации оболочки. После выхода из системы, сама система продолжает работать. Просто еще один пользователь покинул систему. Для выключения FreeBSD необходимо иметь соответствующие права для использования совсем другой программы.

### **2.1.3 Результаты и выводы:**

В результате выполнения практического занятия ознакомиться с ОС FreeBSD, изучить содержимое переменной среды, завершить работу пользователя.

## **2.2 Практическое занятие №5-8 (8 часов).**

**Тема:** «Работа с консолью. Работа в сети. Понятие о работе WEB-сервера и размещении и отладке WEB-приложений»

### **2.2.1 Задание для работы:**

1. Работа с консолью
2. Работа в сети
3. Выход из сети.

### **2.2.2 Краткое описание проводимого занятия:**

1. FreeBSD можно использовать разными способами. Один из них — набор команд в текстовом терминале. Используйте этот способ, и вся гибкость и мощь систем UNIX будет в ваших руках. Этот раздел рассказывает о "терминалах", "консолях" и их использовании в FreeBSD.

2. FreeBSD это многопользовательская, многопроцессорная система. Это формальное описание системы, которая может быть использована множеством разных людей, одновременно запускающих большое количество программ на одном компьютере.

Любой многопользовательской системе нужен способ отличать каждого "пользователя" от остальных. В FreeBSD (и всех UNIX подобных операционных системах), эта задача решается путем "входа" пользователя в систему перед запуском каких-либо программ. У каждого пользователя есть уникальное имя ("имя пользователя") и персональный, секретный ключ ("пароль"). Перед тем, как разрешить пользователю выполнять какие-либо программы, FreeBSD запрашивает их оба.

Сразу после загрузки FreeBSD и завершения работы стартовых скриптов, система предложит вам ввести имя пользователя:

login:

В этом примере, предположим, что ваше имя пользователя root. Введите root в этом приглашении и нажмите Enter. Далее должно появиться приглашение ввести "пароль":

login: root

Password:

Введите соответствующий имени root пароль и нажмите Enter. Пароль не виден! Не беспокойтесь об этом. Это сделано по соображениям безопасности. Если вы ввели пароль правильно, то сразу же войдете в FreeBSD и можете начать выполнять команды. Вы увидите сообщение дня (MOTD, или messageoftheday, которое хранится в текстовом файле /etc/motd) за которым последует командная строка (с символом #, \$, или %). Это означает, что вы успешно вошли в FreeBSD.

3. Как и другие операционные системы, FreeBSD требует корректного завершения работы, позволяющего сохранить необходимую системе служебную информацию, завершить в обычном порядке работающие программы, закрыть сеансы и соединения, уведомив об этом удаленных пользователей. Для завершения работы во многих UNIX-системах используется утилита shutdown:

shutdownactiontime.

action — указывает на действие, которое следует совершить, например, -r (от restart) — перезагрузить компьютер; -p (от power) — выключить питание компьютера.

time — указывает на время совершения события action, например, now заставит выполнить утилиту shutdown немедленно.

Пример. shutdown -p now.

### **2.2.3 Результаты и выводы:**

В результате выполнения практического занятия научиться работать с консолью FreeBSD, осуществлять корректный вход и выход через нее.

## **2.3 Практическое занятие № 9-12 (8 часов).**

**Тема: «Команда Mount. Команда Unmount»**

### **2.3.1 Задание для работы:**

1. Команда Mount
2. Команда Unmount

### **2.3.2 Краткое описание проводимого занятия:**

Файловая система лучше всего представима в виде дерева, с корнем в /. Каталоги, /dev, /usr и прочие - это ветви дерева, которые, в свою очередь, являются корнями для поддеревьев, также имеющих ветви (/usr/local), и т.д. Хорошей практикой является разнесение некоторых особо важных каталогов на разные файловые системы. Например, /var, содержит log/, spool/, а также всевозможные временные файлы и нередко увеличивается в размерах настолько, что занимает все свободное место на диске. Поэтому лучше смонтировать /var отдельно, чтобы избежать переполнения корневой директории/. Часто бывает так, что некоторые разделы файловой системы расположены на разных физических носителях (дисках, CDROM), виртуальных или сетевых (например, сетевая файловая система (NetworkFileSystem, NFS)). В этом случае узлы файловой иерархии будут расположены на разных файловых системах.

1. Команда mount(8) используется для монтирования файловых систем:

```
# mount устройство точка_монтирования
```

где устройство – имя монтируемого устройства; точка\_монтирования – директория, в которую будет вмонтирована файловая система устройства. Перечислим некоторые опции, которые может принимать команда mount:

-r — монтировать файловую систему в режиме "только для чтения"

-w — монтировать файловую систему в режиме "чтение-запись".

-t fstype — монтировать файловую систему как систему указанного типа, или, в случае опции -a, только файловые системы данного типа. По умолчанию, тип файловой системы - "ufs".

-v — выдавать более подробную информацию.

Для монтирования специальных файловых систем, например, файловых систем Microsoft, существуют модификации команды mount, например

mount\_msdosfs — для монтирования систем FAT16 и FAT32;

mount\_ntfs — для монтирования систем NTFS;

mount\_smbfs, mount\_nullfsидр.

Подробнее смотрите на страницах справочника man. В остальном использование этих команд абсолютно одинаково с использованием команды mount. Для некоторых общеизвестных устройств команда mount принимает упрощенный вид, например, mount /cdrom, mount /mnt/floppy. Для других устройств, их имена необходимо указывать в явном

виде, например: `/dev/ad0sN` - для жестких дисков, `/dev/da0sN` - для USB Flash-карт, `/dev/fdN` – для приводов гибких дисков и т.д. Здесь N – порядковый номер устройства текущего типа. Посмотреть наличие устройств и их точные имена можно в директории `/dev`.

2. Команда `umount(8)` принимает в качестве параметра точку монтирования какой-либо файловой системы либо имя устройства, либо опцию `-a` или `-A`. Кроме того, вы можете дополнительно указать опцию `-f` для форсированного размонтирования файловой системы, и `-v` для получения более подробной информации. Имейте ввиду, что это в общем случае опасно и потому не рекомендуется, так как тем самым вы можете нарушить работу компьютера или повредить данные на файловой системе. Опции `-a` и `-A` используются для размонтирования всех файловых систем (разве что вы укажете опцию `-t`). Разница состоит в том, что `-A` не пытается размонтировать корневую файловую систему.

### **2.3.3 Результаты и выводы:**

В результате выполнения практического занятия научиться монтировать и размонтировать систему FreeBSD с помощью команд `Mount` и `Unmount`.

## **2.4 Практическое занятие № 13-16 (8 часов).**

**Тема: «Процессы ОС FreeBSD»**

### **2.4.1 Задание для работы:**

- 1.Даemons
2. Сигналы

### **2.4.2 Краткое описание проводимого занятия:**

1. FreeBSD является многозадачной операционной системой. Это означает, что одновременно может быть запущена более чем одна программа. Каждая программа, работающая в некоторый момент времени, называется процессом. Каждая команда, которую вы запускаете, порождает хотя бы один процесс. Есть несколько системных процессов, запущенных все время и поддерживающих функциональность системы. У каждого процесса есть уникальный номер, называемый process ID, или PID, и, как и у файлов, у каждого процесса есть владелец и группа. Информация о владельце и группе процесса используется для определения того, какие файлы и устройства могут быть открыты процессом с учетом прав на файлы, о которых говорилось ранее. Также у

большинства процессов есть родительский процесс. Например, при запуске команд из оболочки, оболочка является процессом, и любая запущенная команда также является процессом. Для каждого запущенного таким путем процесса оболочка будет являться родительским процессом. Исключением из этого правила 13 является специальный процесс, называемый init(8). init всегда первый процесс, его PID всегда 1. init запускается автоматически ядром во время загрузки FreeBSD. Две команды очень полезны для просмотра работающих в системе процессов, это ps(1) и top(1). Команда ps используется для получения списка запущенных процессов и может показать их PID, сколько памяти они используют, команду, которой они были запущены и т.д. Команда top показывает запущенные процессы и обновляет экран каждые несколько секунд, что позволяет наблюдать за работой компьютера в реальном времени. По умолчанию, ps показывает только принадлежащие вам процессы. Например:

```
% ps
  PID  TT  STAT      TIME COMMAND
    298  p0  Ss      0:01.10 tcsh
    7078 p0   S      2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
  37393 p0   I      0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
  48630 p0   S      2:50.89 /usr/local/lib/netscape/navigator4.77.bi
  48730 p0  IW      0:00.00 (dns helper) (navigator-linux-)
 72210 p0  R+      0:00.00 ps
    390 p1  Is      0:01.14 tcsh
  7059 p2  Is+     1:36.18 /usr/local/bin/mutt -y
  6688 p3  IWs     0:00.00 tcsh
 10735 p4  IWs     0:00.00 tcsh
 20256 p5  IWs     0:00.00 tcsh
    262 v0  IWs     0:00.00 -tcsh (tcsh)
    270 v0  IW+     0:00.00 /bin/sh /usr/X11R6/bin/startx - -bpp 16
    280 v0  IW+     0:00.00 xinit /home/nik/.xinitrc - -bpp 16
    284 v0  IW      0:00.00 /bin/sh /home/nik/.xinitrc
    285 v0   S      0:38.45 /usr/X11R6/bin/sawfish
```

Как вы можете видеть в данном примере, вывод ps(1) организован в несколько колонок. Идентификатор процесса PID уникален для каждого процесса. PID назначаются с 1 до 99999 и опять с начала, если последнее число будет превышено. Колонка TT показывает терминал (tty), на котором запущена программа. STAT показывает состояние программы. TIME это количество времени центрального процессора, использованное программой — это обычно не время, прошедшее с запуска программы, поскольку большинство программы проводят много времени в ожидании некоторого события перед тем, как занять время процессора. Наконец, COMMAND это команда, которой программа была запущена. У ps(1) есть множество различных опций, влияющих на выводимую информацию. Один из наиболее полезных наборов опций это auxww, а позволяет показать информацию о всех запущенных процессах, а не только тех, которыми вы владеете.

показывает имя пользователя, владеющего процессом, и информацию об используемой памяти. x показывает информацию о процессах-демонах и ww указывает ps(1) показать всю командную строку, вместо обрезания ее, когда она станет слишком длинной, чтобы уместиться на экран. Вывод top(1) похож на только что описанный.

Если вы запускаете редактор, им можно легко управлять, открывать в нем файлы и т.д. Вы можете делать это, поскольку редактор предоставляет такие возможности и потому, что редактор присоединен к терминалу. Некоторые программы разработаны без поддержки интерфейса пользователя, поэтому они отсоединяются от терминала при первой возможности. Например, веб-сервер целый день отвечает на запросы из сети, и ему, как правило, не требуется ваше вмешательство. Программы, передающие почту от сервера к серверу — другой пример приложений этого класса. Мы называем эти программы демонами (от англ.: daemon). Демоны (демоны) это персонажи греческой мифологии; хорошие или плохие, они были спутниками человека и, вообще говоря, выполняли полезную работу для людей. Почти как веб и почтовые серверы выполняют полезную работу сегодня. Это причина, по которой талисманом BSD долгое время является веселый демон в тапочках и с вилами. Есть соглашение, по которому имя программы, которая обычно запускается как демон, заканчивается на "d". BIND это Berkeley Internet NameDaemon (выполняемая программа называется named), программа веб сервера Apache называется httpd, демон очереди печати это lpd и так далее. Это соглашение, а не жесткое правило; например, главный почтовый демон для Sendmail называется sendmail, а не maild, как вы могли бы предположить.

2. Иногда может потребоваться взаимодействие с процессом демона. Эти можно сделать с помощью сигналов, т.е. взаимодействовать с демонами (или с любыми запущенными процессами), посыпая им сигнал. Есть множество различных сигналов — некоторые из них имеют специальное значение, другие обрабатываются приложением, реакция которого на эти сигналы должна быть описана в документации. Вы можете посыпать сигналы только тем процессам, владельцем которых являетесь. Если вы отправите сигнал какому-то другому процессу с помощью kill(1) или kill(2), доступ будет запрещен. Исключением из правил является пользователь root, который может отправлять сигналы любому процессу.

В некоторых случаях FreeBSD тоже посыпает сигналы приложениям. Если приложение плохо написано и пробует обратиться к области памяти, к которой оно не должно обращаться, FreeBSD посыпает процессу сигнал нарушение сегментации (SIGSEGV). Если приложение использует системный вызов alarm(3), чтобы получить

уведомление по истечении определенного периода времени, будет отправлен сигнал Alarm (SIGALRM) и т.д.

#### Завершение процесса

Два сигнала могут быть использованы для завершения процесса, SIGTERM и SIGKILL. SIGTERM это корректный способ завершить процесс; процесс может поймать сигнал, определить, что его хотят завершить, закрыть любые файлы, которые он мог открыть, и закончить то, что он делал в момент перед закрытием. В некоторых случаях процесс может даже игнорировать SIGTERM, если выполняет задачу, которая не может быть прервана. SIGKILL не может быть проигнорирован процессом. Этот сигнал говорит "Меня не волнует, что ты делаешь — остановись немедленно". Если вы посыпаете процессу SIGKILL, FreeBSD сразу же остановит этот процесс. Другие сигналы, которые возможно вам понадобятся, SIGHUP, SIGUSR1, и SIGUSR2. Это сигналы общего назначения, различные приложения могут по-разному реагировать на них. Предположим, что вы изменили файл конфигурации веб сервера — теперь нужно указать ему перечитать конфигурацию. Можно остановить и запустить httpd, но это приведет к кратковременной остановке сервера, кото- 16 рая может быть нежелательна. Большинство даемонов написаны так, чтобы при получении сигнала SIGHUP перечитывать файлы конфигурации. Поэтому вместо уничтожения и запуска httpd можно послать сигнал SIGHUP. Поскольку нет стандартного способа реагирования на этот сигнал, различные даемоны будут вести себя по-разному; прочтайте документацию на даемон по этому вопросу.

#### 2.4.3 Результаты и выводы:

В результате выполнения практического занятия научиться работать с процессами FreeBSD, изучить даемоны и управление ими с помощью сигналов завершения и отправки.

### 2.5 Практическое занятие № 17-20 (8 часа).

Тема: «Дополнительная справочная информация»

#### 2.5.1 Задание для работы:

1. Системный справочник (man)

#### 2.5.2 Краткое описание проводимого занятия:

1. Пожалуй, самым полным руководством по FreeBSD является системный справочник (man). Практически каждое приложение или утилита имеют соответствующую страницу (часто не одну), описывающую тот или иной аспект работы

программы, всевозможные опции и настройки. Для просмотра этих страниц существует команда man:

```
% man command
```

Здесь command - это команда, о которой вы хотите получить информацию. Например, чтобы узнать побольше о команде ls, наберите:

```
% man ls
```

Содержимое системного справочника для удобства разделено на несколько разделов:

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

В некоторых случаях (не так уж редко), страницы, относящиеся к одной и той же команде, находятся в различных разделах справочника. Например, есть команда cdmod и системный вызов chmod(). В этом случае, необходимо явно указать раздел man, в котором нужно искать соответствующую страницу:

```
% man 1 chmod
```

Эта команда выведет справку об утилите chmod. По традиции, конкретный раздел справочника указывается в скобках после команды, например, chmod(1) относится к утилите chmod, а chmod(2) - к соответствующему системному вызову.

Часто бывает так, что вы не знаете название команды, но имеете представление о том, что она должна делать. В этом случае можно попытаться найти нужную команду, по ключевым словам, встречающимся в ее описании, используя опцию -k программы man:

```
% man -k mail
```

Вы получите список команд, имеющих слово "mail" в своих описаниях. Это эквивалентно использованию команды apropos. Или, например, вы видите список файлов в каталоге /usr/bin, при этом не имея ни малейшего представления о том, какие функции выполняет каждый из них? Просто наберите:

```
% cd /usr/bin
```

```
% man -f *
```

или

```
% cd /usr/bin
```

```
% whatis *
```

что фактически одно и то же.

### **2.5.3 Результаты и выводы:**

В результате выполнения практического занятия ознакомиться со средствами краткой справочной информации в ОС FreeBSD, изучить стандартные команды для работы с системным справочником.

## **2.6 Практическое занятие №21-24 (8 часа).**

**Тема:** «Настройка карт сетевых интерфейсов»

### **2.6.1 Задание для работы:**

1. Настройка сети. Использование IPconfig
2. Разрешение имён узлов Интернета в IP-адреса

### **2.6.2 Краткое описание проводимого занятия:**

1. Как только для сетевой карты загружен подходящий драйвер, ее потребуется настроить. Для вывода информации о настройке сетевых интерфейсов системы, введите следующую команду:

```
% ifconfig
dc0:      flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
          inet 192.168.1.3 netmask 0xffffffff broadcast 192.168.1.255
          ether 00:a0:cc:da:da:da
          media: Ethernet autoselect (100baseTX <full-duplex>)
          status: active
dc1:      flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
          inet 10.0.0.1 netmask 0xffffffff broadcast 10.0.0.255
          ether 00:a0:cc:da:da:db
          media: Ethernet 10baseT/UTP
          status: no carrier
lp0:      flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0:      flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
          inet 127.0.0.1 netmask 0xff000000
tun0:     flags=8010<POINTOPOINT,MULTICAST> mtu 1500
```

В этом примере были показаны следующие устройства:

dc0: первый Ethernet интерфейс

dc1: второй Ethernet интерфейс

lp0: интерфейс параллельного порта

lo0: устройство loopback

tun0: туннельное устройство, используемое ppp

Для присвоения имени сетевой карте FreeBSD использует имя драйвера и порядковый номер, в котором карта обнаруживается при инициализации устройств. Например, sis2 это третья сетевая карта, использующая драйвер sis(4).

В этом примере, устройство dc0 включено и работает. Ключевые признаки таковы: UP означает, что карта настроена и готова.

У карты есть интернет (inet) адрес (в данном случае 192.168.1.3).

Установлена маска подсети (netmask; 0xffffffff00, то же, что и 255.255.255.0).

Широковещательный адрес (в данном случае, 192.168.1.255).

Значение MAC адреса карты (ether) 00:a0:cc:da:da:da

Выбор физической среды передачи данных в режиме автovыбора (media: Ethernet autoselect (100baseTX)).

Статус соединения (status) active, т.е. несущая обнаружена.

Для dc1, мы видим status: no carrier. Это нормально, когда Ethernet кабель не подключен к карте. Если ifconfig(8) показывает примерно следующее: dc0: flags=8843 mtu 1500 ether00:a0:cc:da:da:da это означает, что карта не была настроена. Для настройки карты вам потребуются привилегии пользователя root. Как и многое другое, сетевая карта может быть настроена во время установки системы.

Утилита Ifconfig предназначена для присвоения адресов сетевым интерфейсам и/или конфигурирования параметров сетевых интерфейсов. Адрес является эквивалентом компьютерного имени. При использовании ifconfig, возможно указание адреса по технологии CIDR (или слешь-технологии), когда после ip-адреса, через прямой слеш указывается облегченная маска (количества единиц в бинарном представлении маски), например: 192.168.0.1/16. Утилита ifconfig может быть использована со следующими параметрами:

alias (add) - устанавливает дополнительный сетевой адрес для данного интерфейса. Этот механизм особенно эффективен при смене сетевых адресов, когда необходимо функционирование как старого, так и нового адресов сети.

-alias (delete) - удаляет адрес из конфигурации сетевого интерфейса. Используется если адрес был добавлен с ошибками или более не требуется.

arp - активизирует использование протокола ARP для разрешения ip-адресов в ethernet - адреса

broadcast - позволяет установить адрес для широковещательных сообщений.

down - помечает сетевой интерфейс, как отключенный. Интерфейс с пометкой "down" не может быть использован системой для передачи пакетов.

`netmask`- позволяет задать сетевую маску, для верного определения адресов ЭВМ локальной сети.

`up`- помечает сетевой интерфейс, как "up". Команда запускает интерфейс, действуя обратно команде "ifconfigdown". Срабатывает автоматически при первой конфигурации адреса интерфейса. Если интерфейс был "сброшен" предыдущей командой "down", то сетевое оборудование будет инициализировано заново. Утилита ifconfig отображает конфигурацию всех сетевых интерфейсов, если запущена без параметров. При конфигурации интерфейса после слова ifconfig сначала указывается имя интерфейса (например, dc0), а затем через пробел набор параметров.

```
Ifconfigdc0 alias 192.168.1.3/24 broadcast 192.168.1.255 up
```

Для проверки правильности настройки сетевой карты, попробуйте выполнить ping для самого интерфейса, а затем для другой машины в локальной сети. Сначала проверьте локальный интерфейс:

```
% ping -c5 127.0.0.1
```

Затем проверьте другую машину в локальной сети:

```
% ping -c5 109.123.141.4
```

Вы можете также использовать имя машины вместо IP-адреса, если настроен, файл /etc/hosts.

И, наконец, проверьте машину вне локальной сети:

`% ping -c5 87.250.250.3.` В случае необходимости использовать шлюз по умолчанию для пересылки пакетов во внешний Интернет, можно добавить адрес шлюза в таблицу маршрутизации при помощи команды

```
route: routeadddefault 109.123.141.1, где 109.123.141.1 — IP-адрес шлюза вашей сети.
```

2. Несмотря на то, что для доставки пакетов любому узлу сети необходимо использовать IP-адрес этого узла, людям гораздо удобнее пользоваться буквенными именами узлов, типа `ya.ru` или `fas.aics.ru`. Для того, чтобы ваши программы могли самостоятельно преобразовать интернет-имя в IP-адрес, необходимо указать локальным программам, какие DNS-сервера (DomainNameSystem) можно использовать для разрешения имен в адреса. В UNIX- системах адреса DNS-серверов прописываются в файле `/etc/resolv.conf`. Этот файл используется процедурами инициализации из библиотеки resolver (3RESOLV) языка C. Процедуры разрешения имен обеспечивают доступ к системе доменных имен Интернета. Файл конфигурации содержит информацию, которая читается процедурами разрешения имен при первом их вызове процессом. Файл создается администратором, или службой DHCP и содержит список пар, состоящих из ключевых

слов и значений, дающих различного рода информацию для процедур разрешения имен. Пары ключевое слово-значение имеют вид: **ключевое\_слово значение**. Поддерживаются следующие опции конфигурации: nameserver адрес Задает IP-адрес (в формате чисел с точками) сервера имен, на который надо переправлять все DNS-запросы. В файле resolv.conf можно указывать до MAXNS (в настоящее время - трех) серверов имен, по одному в строке. Если указано несколько серверов, процедуры разрешения имен будут обращаться к серверам имен в порядке перечисления. Если в файле нет строк nameserver, процедуры разрешения имен используют сервер имен на локальной машине. domain имя задает локальное доменное имя в качестве стандартного домена. Большинство запросов имен в домене может использовать короткие имена, относительно локального домена. Если строки domain в файле конфигурации нет, домен определяется по значению переменной среды, LOCALDOMAIN, если оно задано. search список\_поиска. Задает список поиска для имен узлов. Список поиска обычно определяется по имени локального домена; по умолчанию он содержит только имя локального домена. Это можно изменить, перечислив домены, в которых нужно вести поиск, в списке\_поиска. Имена доменов в этом списке разделяются символами пробела или табуляции. При разрешении большинства запросов сервер имен пытается использовать поочередно каждый компонент пути поиска, пока не будет найдено соответствующее имя. В настоящее время список поиска может содержать не более шести доменов и не может быть длиннее 256 символов. Пример файла /etc/resolv.conf: search aics.acs.cctpu.edu.ru nameserver 109.123.141.5 nameserver 109.123.141.3

### **2.6.3 Результаты и выводы:**

В результате выполнения практического занятия научиться работать с сетью и настройками сетевого оборудования, работать с утилитой Ifconfig управляемать сетевыми адресами компьютеров.

## **2.7 Практическое занятие №25-28 (8 часа).**

**Тема: «OPENSSH»**

### **2.7.1 Задание для работы:**

- 1.Преимущества использования OpenSSH
2. SSH клиент
3. Безопасное копирование

### **2.7.2 Краткое описание проводимого занятия:**

1. OpenSSH это набор сетевых инструментов, используемых для защищенного доступа к удаленным компьютерам. Он может быть использован в качестве непосредственной замены rlogin, rsh, rcp и telnet. Кроме того, любые другие TCP/IP соединения могут быть безопасно туннелированы/перенаправлены через SSH. OpenSSH шифрует весь трафик, эффективно предотвращая кражу данных, перехват соединения и другие сетевые атаки. OpenSSH поддерживается проектом OpenBSD, он основан на SSH v1.2.12 со всеми последними исправлениями и обновлениями, совместим с протоколами SSH версий 1 и 2. OpenSSH включен в базовую систему начиная с FreeBSD 4.0.

Обычно при использовании telnet(1) или rlogin(1) данные пересылаются по сети в незашифрованной форме. Перехватчик пакетов в любой точке сети между клиентом и сервером может похитить информацию о пользователе/пароле или данные, передаваемые через соединение. Для предотвращения этого OpenSSH предлагает различные методы шифрования.

2. Утилита ssh работает подобно rlogin

```
# ssh user@195.208.176.80 или # sshuser@fas.aics.ru.
```

Как правило система безопасности выдаст предупреждение, на которое вы должны будете ввести «yes»:

```
Host key not found from the list of known hosts.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Host 'example.com' added to the list of known hosts.
```

```
user@example.com's password: *****
```

Вход продолжится так же, как если бы сессия была инициирована с использованием rlogin или telnet. SSH использует систему опознавательных ключей для проверки подлинности сервера при подключении клиента. Пользователю предлагается yes только при первом подключении. Дальнейшие попытки входа предваряются проверкой сохраненного ключа сервера. SSH клиент сообщает вам, если сохраненный ключ будет отличаться от только что полученного. Ключи серверов сохраняются в `~/.ssh/known_hosts`, или в `~/.ssh/known_hosts2` для SSH v2. По умолчанию, сервер OpenSSH настроен для приема соединений SSH v1 и SSH v2. Клиент может выбирать между этими двумя протоколами. Версия 2 безопаснее своего предшественника. Команде ssh можно указать использование определенной версии протокола, запустив ее с параметром `-1` или `-2` для версии 1 или 2 соответственно.

3. Команда scpработает подобно rcp, она копирует файл с удаленного компьютера (или на удаленный компьютер), но делает это безопасным способом. Поскольку файлы

передаются по сети через протокол SSH, для указания удаленного файла аргументы принимают форму:

user@host:

```
#scp vasya@fas.aics.ru:/usr/home/vasya/file /usr/file.
```

Параметры, передаваемые scp, похожи на параметры cp, с файлом или файлами в качестве первого аргумента и приемником копирования во втором, иными словами, указывается сначала что копировать, потом, куда копировать.

### **2.7.3 Результаты и выводы:**

В результате выполнения практического занятия научиться обращаться SSHключами и безопасным подключением к другим хостам и компьютерам.

## **2.8 Практическое занятие №29-32 (8 часа).**

**Тема:** «Учётные записи пользователей»

### **2.8.1 Задание для работы:**

1.Разновидность паролей Passwd иPW

### **2.8.2 Краткое описание проводимого занятия:**

1. Учетные записи пользователей в основном означают доступ в систему для обычных людей, и эти учетные записи отделяют пользователя и его рабочую среду, предотвращая повреждение пользователем системы или данных других пользователей, и позволяя пользователям настраивать свою рабочую среду без влияния на других пользователей. Каждая персона, получающая доступ к вашей системе, должна получить уникальную учетную запись пользователя. Это позволит вам выяснить кто что делает, предотвращая сбивание одним пользователем настроек других пользователей, чтение чужой почты и так далее. Каждый пользователь может настраивать свою собственную рабочую среду для приспособления системы под свои нужды с помощью альтернативных оболочек, редакторов, привязки клавиш и настроек языка.

**passwd** это обычный способ изменения собственного пароля пользователя, или пароля другого пользователя суперпользователем. Замечание: Для предотвращения случайного или неавторизованного изменения, перед установкой нового пароля необходимо ввести старый. Пример: Изменение пароля

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

Pw - это утилита командной строки для создания, удаления, модификации и отображения пользователей и групп. Она функционирует как внешний интерфейс к системным файлам пользователей и групп. У pw очень мощный набор параметров командной строки, что делает эту программу подходящей для использования в shell скриптах, но новым пользователям она может показаться более сложной, чем другие представленные здесь команды. После команды pw следует команда действия, например, useradd, usershow, groupshow или groupadd, а за ней, набор параметров, разделенных пробелами. Команда pwuseradd позволяет регистрировать пользователей в системе, внося всю необходимую информацию в файлы конфигурации в автоматическом режиме. pwuseradd может быть использована с различными параметрами.

Пример. Добавление пользователя с использованием pw:

```
# pw useradd -n sasha -c Alexandr_Ivanov -g teamtwo  
-d /var/home/sasha -s /bin/csh -L russian  
# pw usershow sasha
```

Команда pwgroupadd позволяет регистрировать новые группы в системе, внося всю необходимую информацию в файлы конфигурации в автоматическом режиме. pwgroupadd может быть использована с параметрами. Пример. Добавление группы с использованием

```
# pw groupadd teamtwo  
# pw groupshow teamtwo  
teamtwo:*:1100:
```

pw: Число 1100 это (GID) ID группы teamtwo.

В большинстве случаев пользователю принадлежат как его рабочий каталог, так и все файлы в этом каталоге. Так, например, если для пользователя по имени Фред (регистрационное имя fred) создан рабочий каталог /home/fred, то для того, чтобы объявить Фреда владельцем этого каталога, администратору надо выполнить следующие команды:

```
# cd /home 28  
# chownfredfred
```

Или можно перейти прямо в рабочий каталог Фреда, и изменить владельца для всех файлов этого каталога, включая сам каталог:

```
# cd /home/fred  
# chownfred ..??*
```

Чтобы назначить группу teamtwo группой владельцев для домашнего каталога Фреда, можно выполнить команды:

```
# cd /home  
# chgrp teamtwo fred.
```

### **2.8.3 Результаты и выводы:**

В результате выполнения практического занятия научиться работать с паролями и учетными записями разных типов, научиться создавать группы и учетные записи, редактировать владельцев каталога.