

# **О жестких дисках и файловых системах**

**Режимы доступа IDE/ATA**

**Писки SCSI**

**Геометрия жесткого диска**

**РазОиение жесткого диска на разделы**

**Создание дисковых меток**

**Подготовка файловой системы к использованию**

В этой главе мы поговорим о дисках. В отличие от Windows (где диску BIOS автоматически присваивается номер) или Macintosh (где новый диск просто появляется на рабочем столе), UNIX-подобные системы требуют более глубокого знания геометрии, разделов, режимов доступа и других особенностей работы с дисками. И FreeBSD не является исключением.

Как было рассказано в главе 9, иерархическая структура файловой системы UNIX обеспечивает более высокую гибкость при работе с файлами, чем традиционные настольные операционные системы. За эту гибкость приходится платить: для добавления нового диска в систему его необходимо правильно установить физически, правильно разбить на слайсы (slices) и разделы, правильно пометить их, а затем смонтировать в нужной точке. Это требует больше шагов, чем в настольных системах, однако стандартизация оборудования упрощает этот процесс. А с помощью утилиты sysinstall процессом установки нового жесткого диска управлять совсем несложно.

## Режимы доступа IDE/ATA

Наиболее распространенный тип жесткого диска для ПК использует интерфейс IDE (Integrated Drive Electronics — устройство с интегрированной электроникой). Это название подразумевает, что микросхемы контроллера, управляющие диском, интегрированы с ним в одно устройство, не требующее отдельного адаптера (как, например, SCSI). Официально этот интерфейс называется ATA (Advanced Technology Attachment — высокотехнологичное устройство). Названия ATA и IDE зачастую взаимозаменяемы. В этой главе обсуждаются дисковые устройства IDE/ATA и SCSI. Скорее всего, вы используете IDE-диски. Они относительно недорогие и повсеместно распространены, однако им присущи определенные недостатки. Дополнительные затраты при покупке SCSI-устройств компенсируются их более надежной работой.

При выборе оборудования для системы пользователи чаще всего выбирают жесткие диски IDE (если только машина не будет использоваться как высокопроизводительный сервер). Интерфейс IDE обеспечивает достаточно высокую скорость и является стандартом, поддерживаемым всеми материнскими платами архитектуры x86. При выборе диска необходимо обратить внимание на режим доступа, поддерживаемый диском, материнской платой и системой FreeBSD.

## Режимы PIO

Режимы PIO (Programmed I/O — программируемый ввод-вывод) представляют собой исходный стандарт передачи данных интерфейса ATA. Они существуют до сих пор, поскольку режим PIO является составной частью BIOS и не требует дополнительной поддержки от операционной системы. В настоящее время почти все системы поддерживают режимы *DMA* и *Ultra DMA*. Если не используется *старое* оборудование, режим PIO не применяется.

Пять стандартных режимов передачи данных PIO и соответствующие им скорости приведены в табл. 19.1.

**Таблица 19.1** Режимы передачи данных. PIO

Режим PIO	Максимальная скорость передачи данных, Мб/с	Стандарт
-----------	---	----------

<b>Режим PIO</b>	<b>Максимальная скорость передачи данных, Мб/с</b>	<b>Стандарт</b>
2	8,3	ATA
	11.1	ATA-2
	16.7	ATA-2

**^ П Р И М Е Ч А Н И Е**

Большинство современных материнских плат используют для работы с оборудованием шину PCI. Если вам приходится иметь дело со старыми системами, где основной шиной является ISA, максимальным возможным режимом будет PIO Mode 2. Любой более скоростной режим превышает возможности самой шины.

**Режимы DMA**

Главный недостаток интерфейса с режимом PIO заключается в том, что он требует значительных ресурсов процессора для обработки потока данных. Это было одной из причин использования SCSI-дисков, так как последние используют независимый контроллер, снижая тем самым загрузку процессора. В начале 90-х появились режимы DMA. Они перечислены в табл. 19.2. Эти режимы позволяют диску взаимодействовать с оперативной памятью напрямую, минуя процессор. Естественно, это требует изменений в электронике самого диска и поддержки операционной системы. В наши дни применение этих режимов стало поистине всеобъемлющим.

<b>Режим DMA</b>	<b>Максимальная скорость передачи данных, Мб/с</b>	
Single Word Mode 0	2,1	ATA
Single Word Mode 1	4,2	ATA
Single Word Mode 2	8,3	ATA
Multiword Mode 0	4,2	ATA
Multiword Mode 1	13,3	ATA-2
Multiword Mode 2	16,7	ATA-2

В период расцвета простых режимов DMA вы, возможно, даже не слышали о них. Это связано с тем, что скорость передачи данных сравнима с режимами PIO. Хотя DMA-диски меньше использовали ресурсы процессора, это преимущество практически не проявлялось из-за плохой поддержки DMA в таких операционных системах, как Windows 95. А режим PIO уже был встроенным, поэтому производители операционных систем не спешили отказываться от него.

**Режим Ultra DMA**

Ситуация коренным образом изменилась, когда появились режимы Ultra DMA. Эти улучшенные режимы DMA теперь являются общим стандартом для индустрии. А по соотношению цена/производительность они даже обгоняют интерфейс SCSI, оставаясь при этом дешевыми.

Режимы Ultra DMA значительно повысили скорость передачи данных по сравнению со стандартными DMA. Это стало возможно благодаря тому, что в них для не-

передачи данных используется передний и задний фронт сигнала. Это значит, что при той же частоте скорость передачи данных удваивается до 33 Мб/с. Скорости в режимах Ultra DMA показаны в табл. 19.3. Рост связан также с переходом на новый стандарт кабеля IDE: 80 контактов вместо 40.

Таблица 19.3 Режимы передачи данных Ultra DMA

<b>Режим Ultra DMA</b>	<b>Максимальная скорость передачи данных, Мб/с</b>	<b>Стандарт</b>
0	16,7	ATA/ATAPI-1
1	25,0	ATA/ATAPI-4
2	33,3	ATA/ATAPI-4
3	44,4	ATA/ATAPI-5
4	66,7	ATA/ATAPI-5
5	100,0	ATA/ATAPI-6

Любое оборудование, выпущенное после 1998 года, поддерживает режим UDMA, поэтому обратите внимание на следующие четыре компонента вашего ПК:

- Жесткий диск, поддерживающий режим Ultra DMA;
- 80-контактный кабель IDE;
- Материнская плата (или контроллер IDE), поддерживающая режим Ultra DMA;
- Поддержка режима Ultra DMA в операционной системе (или BIOS).

Если один из элементов отсутствует, система все равно сможет работать, однако передача данных будет происходить с меньшей скоростью, скорее всего, в каком-либо из режимов PIO, автоматически поддерживаемом BIOS. С материнскими платами и контроллерами, не поддерживающими UDMA, возможны определенные проблемы, такие как неустойчивость или "зависание" системы. Их можно решить или обновив BIOS (позволив контроллеру поддерживать UDMA), или запретив режим UDMA в самом диске, с помощью утилит, поставляемых его производителем.

FreeBSD полностью поддерживает режимы Ultra DMA. Проверить, все ли нужные для поддержки элементы присутствуют в системе, позволяет поиск устройств во время загрузки. Просмотреть сообщения можно, набрав команду dmesg:

```
ad0: 6194MB <HITACHI_DK239A-65> [13424/15/63] at ata0-maat<rwuaing
ODMA33
```

Если система не определила возможность использования UDMA и жесткий диск новее, чем остальные компоненты компьютера, следует найти утилиты управления диском (поставляемые производителем), загрузить с дискеты систему MS-DOS и запретить использование UDMA. В этом случае скорость передачи данных будет ниже, однако система не будет сбоить и зависать.

## Диски SCSI

Если вам не требуются IDE-диски или вы собираетесь подключить к системе внешние диски, альтернативой может стать интерфейс SCSI. На первый взгляд режимы SCSI (приведенные в табл. 19.4) кажутся не менее запутанными, чем режимы IDE.

Стандарт	Режим передачи данных	Скорость передачи данных, Мб/с	Максимальная длина кабеля, м	Число устройств на шине	Число контактов соединительного кабеля, шт
SCSI-1	SCSI	5	6	8	50
SCSI-2	Fast SCSI	10	3	8	50
	Wide SCSI	10	6	16	68
	Fast Wide SCSI	20	3	16	68
SCSI-3	Ultra SCSI	20	1,5-3	4-8	50
	Wide Ultra SCSI	40	1.5-3	4-16	68
	Ultra2 SCSI	40	12	8	50
	Wide Ultra2 SCSI	80	12	16	68
	Ultra3 SCSI	160	12	16	68
	Ultra 160 SCSI	160	12	16	68
	Ultra320 SCSI	320	12	16	68

Несмотря на сложные правила именования, интерфейс SCSI ведет себя достаточно предсказуемо и корректно. Каждый последующий стандарт в два раза быстрее предыдущего. Используется лишь два типа разъемов соединительных кабелей. Это значит, что даже самый новый SCSI-контроллер сумеет распознать SCSI-устройство, соответствующее более раннему стандарту, и сможет с ним работать.

Цепочка устройств SCSI может содержать не больше устройств, чем задано стандартом, обычно 8 или 16. Вспомните, что структура интерфейса IDE/ATA позволяет работать лишь с четырьмя устройствами (два контроллера по два устройства). Интересно, что порядок устройств в цепочке не имеет значения. Здесь нет таких понятий, как ведущее или ведомое устройство. Существует только контроллер, или хост-адаптер (он сам является SCSI-устройством и получает свой идентификатор), и остальные устройства.

SCSI имеет немало преимуществ также и при использовании внешних дисков. Интерфейс IDE/ATA не обеспечивает удобного способа подключения внешнего диска к внутренней шине. В некоторых случаях можно воспользоваться параллельным (принтерным) портом, например, для носителей Iomega ZIP, но это чрезвычайно медленный интерфейс, неприменимый для большинства задач. SCSI был единственным интерфейсом для внешних дисков, пока не появился стандарт FireWire (IEEE 1394).

Наибольшей проблемой применения SCSI-устройств является... цена. SCSI-диски обычно дороже аналогичных IDE-дисков в 1,5–2 раза. К этому следует добавить цену SCSI-адаптера, который не поставляется с материнской платой (только некоторые материнские платы сейчас поставляются со встроенным SCSI-адаптером, хотя, нужно отметить, число их растет). Их также нельзя назвать дешевыми.

Следует отметить, что со SCSI связано множество вопросов, выходящих за рамки этой книги: терминаторы, SCSI BIOS, настройка идентификаторов устройств (посредством джамперов). Представляет интерес и то, что происходит, когда в одной цепи присутствуют устройства разных стандартов. Более подробные сведения о SCSI-устройствах можно найти, например, на Web-сайте PC Guide по адресу <http://www.pcguid.com>.

"Си Ф . . . . . ••••• WVi&i&ST •

## Геометрия жесткого диска

Как администратор, вы будете сталкиваться с этими понятиями при каждом запуске утилиты `fdisk` (или программы с более дружелюбным интерфейсом). Одним из преимуществ FreeBSD является способность эффективно работать даже на старом оборудовании, непригодном для использования, скажем, с современными версиями Windows. Материнской платы с процессором Pentium 166 МГц вполне достаточно для поддержки FreeBSD. Но здесь есть ловушка: попытайтесь подключить новый 70-гигабайтный жесткий IDE-диск — и система обнаружит не более 8 Гб. Вот здесь-то и вступает в игру геометрия жесткого диска. Чтобы использовать весь объем диска, нужно предпринять определенные действия.

В те времена, когда физический размер диска делали настолько большим, насколько позволяла цена, несмотря на будущие сложности с масштабируемостью, емкость жестких дисков IDE вычислялась как функция от их физической геометрии. Четыре параметра — число головок, цилиндров, секторов и байт в секторе — полностью описывали диск и его емкость (см. рис. 19.1), то есть объем данных, которые он мог хранить. Жесткие диски и теперь имеют эту внутреннюю геометрию, однако сейчас она не так важна.

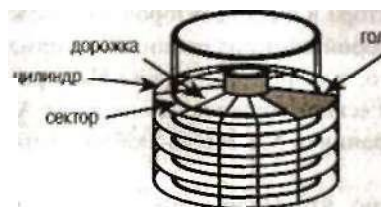


Рисунок 19.1  
Геометрия жесткого диска

Диск состоит из нескольких пластин, вращающихся вокруг общей оси. Данные сохраняются с обеих сторон пластины и читаются одной или несколькими головками с каждой стороны. Таким образом, диск с четырьмя пластинами и с одной головкой на каждую сторону имеет в сумме 8 головок. Каждая пластина разделена на концентрические кольца — дорожки. Дорожки с одинаковым номером на всех поверхностях образуют цилиндр. Дорожки разделены на секторы (обычно на 64, из них используется 63). Каждый сектор хранит определенное число байтов (обычно, 512).

Каждый диск сопровождается информацией о его параметрах. Подсчитать размер диска можно их простым перемножением. Диск, который использовался в примере работы утилиты `fdisk` в главе 9, содержит следующую информацию:

```
# fdisk /dev/adl
***** Working on device /dev/adl
parameters extracted from in-core disklabel are:
cylinders=1247 heads=255 sectors/track=63 (16065 blks/cyl)
media sector size is 512
```

Умножение этих чисел (кроме значения блок/цилиндр) дает следующий результат:

```
1247 cylinders x 255 heads x 63 sectors/track x 512 bytes/sector =
10.2GB
```

Минутку! 255 головок? Чтобы это значило? Прежде всего, число должно быть четным. Однако, в любом случае, это означает 64 пластины — количество, физически невозможное для диска высотой в сантиметр. Здесь мы столкнулись с виртуальной геометрией диска, которая стала возможной благодаря двум замечательным решени-

ям, найденным индустрией жестких дисков. Они позволили обойти ограничение физических размеров, вызванное исходной разработкой интерфейса BIOS. Итак, рассмотрим эти решения (режимы LBA и Extended INT13).

## Режим LBA и предел 528 Мб

Во времена флоппи-дисков никто не мог даже предположить, что кому-нибудь понадобится жесткий диск такого гигантского объема, как 528 Мб. Поэтому это число стало тем ограничением, которое накладывало количество бит, используемое BIOS ПК, на описание геометрических размеров диска. Значения приведены в табл. 19.5.

<i>Параметр</i>	<i>Количество бит</i>	<i>Максимальное значение</i>
Цилиндры	10	$2^{10}-1 = 1023$
Головки	8	$2^8-1 = 255$
Сектор/Дорожка	6	$2^6-1 = 63$

Таким образом, диск мог иметь до 255 головок и не более 1023 цилиндров. Поскольку количество байтов в секторе и число секторов на дорожку — величины почти фиксированные (площадь, на которой записана единица информации, определяется надежностью считывания). Кроме того, в устройстве можно было разместить не так много пластин и головок. Производители вскоре сошлись на стандарте ATA с 16 физическими головками, т.е. объем диска был ограничен 528 Мб. В любом случае, BIOS не мог адресовать более 1023 цилиндров.

К 1993 году стало ясно, что запросы дискового пространства вышли за допустимые пределы. Весь доступный объем могла занять одна-единственная видеопластинка! Поэтому производители BIOS разработали схему под названием LBA (Logical Block Addressing — логическая адресация). Благодаря LBA стало возможно расширение адресации BIOS путем пересчета цилиндров с номерами выше 1024 на "виртуальные головки", используя, таким образом, преимущество в допустимом количестве головок (до 255). Например, диск с 1852 цилиндрами и лишь 16 физическими головками можно рассматривать как имеющий 463 цилиндра и 64 головки. Размер диска при этом не изменяется, но значения цилиндров и головок находятся в пределах, допустимых BIOS. Все стало хорошо... но не надолго.

## Режимы Extended INT13 и предел 8.4 Гб

Такая схема представляла собой лишь временное решение, так как и режим LBA должен был переполниться, как только число цилиндров и головок достигнет максимальных значений. Максимально допустимый объем поднялся до 8,4 Гб. Однако прошло всего четыре года, и жесткие диски достигли и этого предела.

Решением стал интерфейс BIOS, названный Extended INT13 (Расширенное использование прерывания 13), который используется современным оборудованием и широко распространен в мире Windows (где программное обеспечение взаимодействует с диском через BIOS, что не требуется во FreeBSD). Этот новый интерфейс абстрагирует адресацию геометрии диска как 16-байтовый пакет дисковой адресации (Disk Addressing Packet), эффективно устраняя какие-либо ограничения размера (по крайней мере, на ближайшее время). Если ваше оборудование не очень старое и

поддерживает режим Extended INT13, вам не потребуется знание геометрии диска. Если воспользоваться какой-либо утилитой, отображающей параметры диска размером более 8,4 Гб, она выведет такие параметры, как 16383 цилиндра, 16 головок и 63 сектора. Подобная конфигурация, сообщает операционной системе, что нет смысла пытаться вычислить размер диска по его геометрии. Помните, что система FreeBSD сама заботится обо всем, что связано с "геометрией". По крайней мере, до тех пор, пока диски IDE не достигнут 137 Гб и не будет исчерпано максимально возможное в стандарте ATA число цилиндров — 65536... (В настоящий момент и этот барьер преодолен. — Прим ред. )

## Практические вопросы

Если вы устанавливаете FreeBSD на старом оборудовании, как, например, материнской плате эпохи LBA, то можете столкнуться с какими-либо из описанных выше ограничений. Вы не потеряете данных, но заметите такие проблемы, как, например, объем диска 8,4 Гб вместо 10 Гб. Теперь вы знаете, с чем это связано.

Конечно же, если вы используете жесткие диски SCSI или FireWire (IEEE 1394), то не столкнетесь с проблемами подобного рода. Эти интерфейсы были разработаны с учетом заведомо большей масштабируемости, чем IDE/ATA.

## Разбиение жесткого диска на разделы

Добавление нового диска в систему требует четырех шагов: его установки, разбиения на разделы, разметки и монтирования. В зависимости от того, используется SCSI- или IDE-диск, процедура физической установки будет разной. В этом может помочь инструкция, которая обычно сопровождает диск.

В последующих примерах мы будем предполагать, что жесткий IDE-диск емкостью 40 Гб устанавливается как ведомое (slave) устройство первого контроллера. Имя устройства, как было рассказано в главе 9, — /dev/ad1. (SCSI диск в такой же конфигурации имел бы имя /dev/dal.) Вспомните, как обычно строится имя устройства. Расширения, следующие за исходным префиксом ad1, содержат дополнительную информацию о разделах. Так происходит обращение к различным частям диска, выделенным для использования, например, разными операционными системами.

## Разделы BIOS (Slices)

Каждая операционная система позволяет разбить диск на разделы. Важно помнить, что FreeBSD имеет не один, а *два уровня разбиения*. На первом находятся разделы BIOS, адресуемые непосредственно BIOS оборудования ПК. Их может быть не более четырех, и другие операционные системы рассматривают их как "разделы" (partitions). Если разбиение диска выполнялось в системе Windows или Macintosh, каждый раздел является именно разделом BIOS.

В FreeBSD для разделов BIOS используется термин "слайсы" (slices). Об этом важно помнить. Когда же во FreeBSD речь идет о разделах, то подразумевается второй уровень разбиения, более глубокий, чем слайсы. Разделы второго уровня часто называют "разделами BSD".

## Разделы BSD

Такие подразделы используются во FreeBSD, чтобы отделить друг от друга разные части системы: /var, /usr, /home и т.д. Каждый раздел имеет имя, состоящее из одной

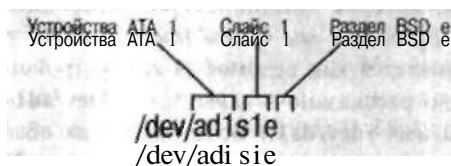
буквы. В системе может присутствовать до восьми разделов BSD, как показано в табл. 19.6. При разбиении слайса на разделы BSD (это выполняется в процессе разметки, о которой речь пойдет далее), несколько первых букв зарезервированы для специальных целей.

**Таблица 19.6** Разделы BSD и их назначение

<b>Раздел</b>	<b>Назначение</b>
<b>a</b>	Корневой раздел (/)
<b>b</b>	Раздел swap
<b>c</b>	Адресуется к слою целиком или всему диску в "опасном" режиме. ("Опасный" режим использовался в прошлом, чтобы обойти проблемы с геометрией диска. Он не используется во FreeBSD 4.4 и 5.0)
<b>e</b>	Для общего использования
<b>f</b>	Для общего использования
<b>g</b>	Для общего использования
<b>h</b>	Для общего использования

На рис. 19.2 показана структура полного имени устройства. Первая часть (ad1) задает основное имя устройства. Диски IDE/ATA имеют имена от ad0 до ad3 (четыре возможных устройства контроллера). Затем идет имя слайса, причем номер начинается с 1 (s1). И в заключение, имя раздела BSD (e).

Рисунок 19.2  
Расшифровка имени  
устройства



### Дополнительные разделы других операционных систем

Некоторые операционные системы имеют структуру разделов, схожую с разделами во FreeBSD: например, дополнительный раздел DOS (Extended DOS Partition) позволяет создать на DOS/Windows-машине подразделы (логические диски). Функционально идентичный этому метод используется и в Linux. Подобные способы разбиения диска являются довольно неуклюжими по сравнению с решением, предложенным во FreeBSD. Поэтому возможности взаимодействия различных операционных систем усложнены. Разделы внутри дополнительного раздела DOS, например, рассматриваются во FreeBSD как дополнительные слои на том же уровне, что и обычные разделы DOS, начиная со слоя 5 (т.е. после четырех стандартных разделов DOS). Для монтирования второго логического диска внутри дополнительного раздела DOS жесткого диска, установленного как устройство ad3, необходимо использовать имя /dev/ad3s6, вместо стандартного имени раздела BSD внутри слоя.

Попытка продвинуться в обратном направлении будет выглядеть еще сложнее. Вероятно, вам не удастся смонтировать диск с разметкой FreeBSD в системе DOS/Windows, однако на Linux-машине это вполне возможно. Linux работает с разделами

(которые во FreeBSD называются слайсами) так же, как и FreeBSD: они именуются численно и последовательно, а не иерархически. Для монтирования слайса FreeBSD в Linux воспользуйтесь табл. 19.7, где приведено соответствие имен. В таблице описаны разделы ведомого устройства первого контроллера, на котором третий слайс/расширенный раздел DOS содержит файловую систему FreeBSD.

Таблица 19.7 Соответствие меток а Linux и FreeBSD

<b>Метка Linux</b>	<b>Метка FreeBSD</b>
/dev/hdb4	/dev/ad1s3a
/dev/Aidb.5	/dev/ad1s3b
/dev/hdb-б	/dev/ad1s3e
/dev/hdb7	/dev/ad1s3f

## Редактор слайсов (fdisk) в программе sysinstall

Как пользователь root запустите программу /stand/sysinstall и выберите пункт "Configure". Прокрутите экран до опции "Fdisk", отмеченной как "Slice (PC-Style partition) Editor". Выбор этой опции даст возможность получить список всех дисков, которые найдены в системе. Убедитесь, что в нем содержится ad1 (новый диск, установленный как slave-устройство первого контроллера). Если нет, значит, диск установлен неправильно. Чтобы проверить, был ли диск вообще обнаружен при загрузке системы, просмотрите файл /var/run/dmesg.boot. Если устройство не обнаружено, необходимо проверить кабель и состояние джамперов на диске. Помните, что, если на первом контроллере раньше был только один диск, его джампер наверняка был в положении "Single". Необходимо убедиться, что теперь джампер старого диска установлен в положении "Master", а нового диска — в положении "Slave".

Если диск представлен в списке, выберите его (пробелом) и программа запустит визуальный редактор fdisk (fixed disk — жесткий диск). Такая утилита в той или иной форме имеется в любой операционной системе, поскольку каждой из них необходим инструмент для управления разделами BIOS (слайсами). Программу fdisk можно использовать прямо из командной строки, однако она имеет много опций и потенциальных ловушек, поэтому лучше воспользоваться визуальным интерфейсом, который обеспечивает программа sysinstall. В таких вопросах, как разбиение и форматирование диска, лучше подстраховаться!

В меню будут отображены все существующие разделы. Если это новый, неформатированный диск, список слоев будет пустым. Здесь вы можете воспользоваться достаточно гибкими возможностями. При желании можно создать разделы BIOS для других операционных систем или FreeBSD, в стиле Linux или DOS. Но, скорее всего, потребуется выбрать опцию A — "Use Entire Disk" (Использовать весь диск). Меню программы показано на рис. 19.3. О других опциях можно пока не думать, они добавлены для повышения гибкости программы и улучшения совместимости.

Диск содержит три элемента: небольшую область, "зарезервированную" для диспетчера загрузки, который будет установлен позже, один слой FreeBSD (ad1s1) и "оставшийся" участок дискового пространства (малой емкости), не включенный в основной слой.

:

```

Disk name:      3D
DISK Geometry: 4S65 cyls/255 heads/63 sectors = 78156225 sectors <3S162№>

```

Si:e(ST)	Heme	PType	Desc	StE>ti:pe
<b>B</b>	63	62		
63	ТВ1ЭД62			

Рисунок 19.3

*Slice Editor (после выбора опции "Use Entire Disk").*

The following commands are supported <in upper or lower case>:

```

H - Use Entire Disk      G * set Drive Geometry    C = Create Slice
D * Delete Slice        Z * Toggle Size Units    S = Set Bootable
T = Change Type         U = Undo n11 Changes      U = Urite Changes

```

Use F1 or ? to get ler\* help, arrow keys to select.

Однако в этой фазе еще никакие изменения не записаны непосредственно на сам диск. Чтобы зафиксировать разметку диска и сохранить изменения, используется опция W. Для выхода из Slice Editor воспользуйтесь опцией Q. Программа запросит, какой тип диспетчера загрузки следует установить. Если это второй диск, добавляемый к работающей системе, выберите None (Нет). Если же вы будете загружаться с этого диска, установите FreeBSD Boot Manager (BootMgr). Это может понадобиться, например, когда диск готовится к использованию на другой машине или для замены текущей системы. В ином случае выберите "Standard" — программа выполнит команду, эквивалентную команде DOS fdisk /MBR, которая удаляет из MBR диспетчера загрузки, специфичные для операционных систем (как LILO Linux и BootMgr FreeBSD), и делает его доступным для других систем, например, Windows.

## Создание дисковых меток

Итак, диск разбит на слайсы и подготовлен для работы с FreeBSD, но еще не отформатирован — его нельзя подключить к системе. Далее необходимо воспользоваться утилитой Disklabel Editor (Редактор дисковых меток).

Находясь в программе sysinstall, выберите Label (Метка), будет запущен Disklabel Editor. Утилита вновь попросит выбрать диск из списка установленных жестких дисков. Для выбора ad1 используйте пробел. На экране появится меню программы Disklabel Editor — визуального интерфейса команды disklabel. В нем создаются и форматируются разделы BSD.

Disklabel Editor во FreeBSD не позволяет изменять размеры существующих разделов. Этого нельзя сделать, не переформатировав его с потерей всех данных, поэтому тщательно продумайте схему создания разделов до того, как сохранять изменения!

## Полная разметка раздела FreeBSD

Если вы добавили в систему новый диск, перейдите к разделу "Adding a New Disk". Этот раздел демонстрирует, как воспользоваться утилитой Disklabel Editor для создания разделов BSD и точек монтирования в полной системе FreeBSD, включая корневой раздел, /usr, /var и раздел подкачки. Это особенно полезно при инсталляции FreeBSD на новый диск, например, при замене текущего диска новым, большим по объему.

Нажмите A и выберите опцию "Auto Defaults for all!" (Автоматические настройки для всех разделов). Программа автоматически вычисляет размер разделов, которые будут пригодными для большинства систем (см. рис. 19.4). Если FreeBSD-система будет использоваться как рабочая станция или небольшой сервер с несколькими пользователями, вы вполне можете воспользоваться этими настройками по умолчанию.

Рисунок 19.4  
Утилита *Disklabel Editor*  
(после выбора опции "Auto Defaults for all!").

```

Port      hount          S Ue He*fs Part          Six* HMGA
adBifa    /                seie UFS v
BoSstb    swap            261 MB SUPP
odeale    /var            28NB UFS V
odBHf     /usr            378381B UFS V

The felto-ing co—and* arm valid here (upper or IOIT coje):
C - Create      D • Delete      И - Point pt  U - Urit%
M • Heefs Opts  T • Newsf Toggle U = Undo      Q - Finish
R = fjtto Defaults for all!

Use Fl or ? to get mere help, arrow keys to select.

```

Для сервера настройки по умолчанию наверняка не подойдут. Почтовые сообщения и log-файлы размещаются в разделе /var. Если в системе присутствует 100 пользователей, хранящих свой почтовый ящик на сервере, или Web-сервер генерирует log-файлы размером 100 Мб, раздел /var размером 20 Мб будет мгновенно исчерпан. Таким образом, для /var требуется больший объем дискового пространства. Все эти ситуации — замечательный пример того, насколько гибкой может быть работа утилиты Disklabel Editor.

Прежде всего, необходимо удалить оба раздела /var и /usr. Для этого выберите их и нажмите D. Прокрутите экран к верхней части, где поле "Free:" показывает пространство, доступное после удаления разделов. Теперь разделы нужного размера можно создать заново. (Помните, что ничего не будет записано на диск до тех пор, пока вы не нажмете клавишу W.)

Разместив курсор в строке "Partition name: adlsl", нажмите "C", чтобы создать новый раздел. На экране появится диалоговое окно, запрашивающее объем нового раздела. Отведите /var 200 Мб, их должно быть вполне достаточно для среднего сервера. (Если есть возможность, можно отвести и больше.) Вместо числа, отображенного в окне, введите "200t". Затем выберите FS для создания файловой системы (а не раздела подкачки). Программа запросит точку монтирования — введите /var. После этого раздел будет создан и добавлен в список, а пункт "Free:" изменен соответствующим образом.

Теперь этот же процесс следует повторить для раздела "/usr", используя оставшееся дисковое пространство. В результате список разделов будет выглядеть как приведенный на рис. 19.5.

## Добавление нового диска

На новом диске можно создать до восьми разделов BSD. Не используйте опцию A, если вы хотите просто добавить дисковое пространство к работающей системе. Вместо этого выберите "Partition name: adlsl" и нажмите клавишу C, создав новый раздел. Если вы хотите создать на диске один раздел BSD большого объема (например, добавить 40 Гб к /home), примите значение по умолчанию, т.е. полный объем слайса. Можно поступить

и по-другому: указать размер одного раздела (например, 20G), а из того, что осталось, создать другой раздел.

```

Disk: odi      Partition name: ad1s1  Free: 0 blocks (0MB)
Port         noun          Size HeFS Part      Itant
odQslo      /              56GB UFS  '
adQs1b      ?*ap          геvпв UFS V
adOsle      /var

```

Рисунок 19.5

Разметка раздела.

```

The following commands are valid here (upper or lower case):
C = Create      D = Delete      tl = Mount pt.  H a ll te
H = Newfs Opt  T = Newfs Toggle  U = Undo       0 = Finish
H - finite Defaults for oil

Use F1 to get more help, row keys to select.

```

Для создания файловой системы (а не области подкачки) выберите FS и укажите точку монтирования. В данном случае это не очень важно. Этот пункт указывает лишь на то, что будет записано в файл /etc/fstab для новой файловой системы. Л это происходит лишь в случае инсталляции системы.

Таким способом можно создать необходимое число разделов, пока не будет использовано все доступное дисковое пространство. Чтобы удалить раздел, выберите его и нажмите клавишу D. Помните, что эти операции не затрагивают диск. Все изменения записываются лишь при нажатии клавиши W.

#### ^ П Р И М Е Ч А Н И Е

Если вы создадите более четырех разделов, все "зарезервированные" имена будут использованы, начиная с a и B; c будет пропущено, поскольку оно имеет специальное значение. Для разделов после d будет использована "пустая" метка X, и вы не сможете их корректно применить.

## Сохранение изменений и форматирование диска

Форматирование дисков во FreeBSD осуществляется командой newfs. Эта утилита командной строки также запускается автоматически из sysinstall. Каждая файловая система из списка, для которой в столбце Newfs установлено значение Y, будет отформатирована после принятия изменений. Включить или отключить значение можно с помощью клавиши T.

Посмотрите внимательно на список разделов, создаваемых в редакторе. Их метки содержатся в столбце Part. Независимо от того, как вы создаете разделы, имена от a до d зарезервированы для специального использования, как показано в табл. 19.6. Первый создаваемый раздел имеет метку e. Это значит, что имена новых монтируемых разделов, добавляемых в систему, будут выглядеть так: ad1s1e, ad1s1f и т.д. Имена меток понадобятся при монтировании файловых систем.

По завершении разметки диска нажмите W для сохранения изменений. Диск будет размечен, после чего запустится процесс newfs, формирующий каждый из созданных разделов. По окончании нажмите Q для выхода из программы sysinstall.

## Подготовка файловой системы к ИСПОЛЬЗОВАНИЮ

Итак, почти все готово. Остается лишь смонтировать новые файловые системы. Кстати, их автоматическое монтирование при загрузке можно настроить в файле */etc/fstab*. *Подробное обсуждение возможностей команды **mount** и содержимого файла /etc/fstab можно найти в главе 9.*

Здесь мы сталкиваемся с простейшим из возможных случаев: новые отформатированные разделы FreeBSD. Команду `mount` можно использовать наиболее прямолинейным способом.

Создайте каталог, который будет точкой монтирования, например, `/mnt/newdisk`. (Он может находиться в любом месте файловой системы.) Затем запустите команду `mount`:

```
# mount /dev/ad1e1e /mnt/newdisk
```

Для проверки того, что диск смонтирован, выполните команду `df`. То же самое можно сделать и для любого раздела, разместив точку монтирования в произвольном месте файловой системы. Для демонтирования применяется команда `umount`:

```
# umount /mnt/newdisk
```

Примите наши поздравления — вы овладели основами работы с жесткими дисками в FreeBSD!