4. Подготовка дисков

Содержание:

- Общие сведения о блочных устройствах
- Разработка схемы разбиения диска
- Использование fdisk для создания разделов
- Создание файловых систем
- Монтирование

4.a. Общие сведения о блочных устройствах

Блочные устройства

Мы достаточно подробно рассмотрим аспекты работы с дисками в Gentoo Linux и Linux вообще, включая файловые системы, разделы и блочные устройства. Позже, когда вы уже освоитесь с дисками и файловыми системами, мы проведем вас через процесс настройки разделов и файловых систем для установки Gentoo Linux.

Для начала, давайте познакомимся с *блочными устройствами*. Вероятно, наиболее известно блочное устройство, соответствующее в системе Linux первому IDE-диску, а именно `/dev/hda`. А если в вашей системе используются диски SCSI или SATA, то первым жестким диском будет `/dev/sda`.

Подобные блочные устройства представляют собой абстрактный интерфейс к диску. Прикладные программы могут использовать их для обращения к дискам, не беспокоясь, к какому типу те принадлежат: IDE, SCSI или какому-то еще. Программы могут просто обращаться к накопителю, как к набору смежных 512-байтных блоков с произвольным доступом.

Разделы

Хотя теоретически для размещения системы Linux диск можно использовать целиком, так почти никогда не делают на практике. Вместо этого большое физическое блочное устройство разбивают на меньшие блочные устройства, более удобные для обращения. В архитектуре x86 они называются *разделами*.

Существуют разделы трех типов: *первичные* (primary), *расширенные* (extended) и *логические* (logical).

Первичный раздел — это раздел, информация о котором хранится в MBR (Master Boot Record — главной загрузочной записи). Так как MBR очень мала (512 байт), можно определять всего четыре первичных раздела (например, от `/dev/hda1` до `/dev/hda4`).

Расширенный раздел — это особый первичный раздел (имеется в виду, что расширенный раздел должен быть одним из четырех возможных первичных разделов), в котором содержатся другие разделы. Таких разделов изначально не существовало,

Для печати

View all

Обновлено 1 января 2006

Исходный документ
обновлен 7 июля 2008

Аннотация: Чтобы установить Gentoo, нужно создать подходящие дисковые разделы. В этой главе описывается, как разбить диск для будущего использования.

Sven Vermeulen
автор

Roy Marples
автор

Daniel Robbins
автор

Chris Houser
автор

Jerry Alexandratos
автор

Seemant Kulleen
разработчик Gentoo x86

Tavis Ormandy
разработчик Gentoo Alpha

Jason Huebel
разработчик Gentoo AMD64

Guy Martin
разработчик Gentoo HPPA

Pieter Van den Abeele
разработчик Gentoo PPC

Joe Kallar
разработчик Gentoo

но их введение помогло расширить существующую схему разметки без потери совместимости, как только четырех разделов перестало хватать.

Логический раздел — это раздел, входящий в расширенный раздел. Его определение находится не в MBR, а внутри расширенного раздела.

Дополнительные возможности

На установочном компакт-диске для x86 предусмотрена поддержка EVMS и LVM2. Использование EVMS и LVM2 повышает гибкость разбиения диска. В ходе установки мы останавливаемся на «обычных» разделах, но стоит запомнить, что EVMS и LVM2 тоже поддерживаются.

4.b. Разработка схемы разбиения диска

Схема разбиения по умолчанию

Если вам не интересно заниматься разработкой схемы для своей системы, можно воспользоваться схемой, используемой в этой книге:

Раздел	Файловая система	Размер	Описание
/dev/hda1	ext2	32МБ	загрузочный раздел
/dev/hda2	(swap)	512МБ	раздел подкачки
/dev/hda3	ext3	оставшаяся часть диска	корневой раздел

Если вам интересно узнать, какого размера должны быть разделы, и сколько их вам вообще может потребоваться, читайте дальше. В противном случае приступайте к созданию разделов, описанному в главе использование fdisk для создания разделов.

Сколько и каких именно?

Количество разделов очень сильно зависит от назначения системы. Например, если у вас много пользователей, вам, скорее всего, захочется отделить `/home` для повышения безопасности и упрощения резервного копирования. Если вы устанавливаете Gentoo в роли почтового сервера, следует отделить `/var`, т.к. вся почта хранится там. Затем, правильно выбрав файловую систему, вы добьетесь максимальной производительности. Для игровых серверов потребуется отдельный раздел `/opt`, так как большинство программ для их работы устанавливается туда. Причины выделения те же, что и для `/home`: безопасность и резервное копирование. Определенно не помешает побольше места для `/usr`: не только потому, что там хранится большинство приложений, а еще из-за того, что лишь дерево Portage, не считая размещенных в нем архивов с исходными кодами, занимает около 500 МБ.

Как видите, все зависит от ваших целей. Наличие отдельных разделов или томов имеет следующие плюсы:

- для каждого раздела или тома можно выбрать наиболее подходящую файловую систему
- свободное место во всей системе не кончится «вдруг» из-за того, что одна-единственная сбойная программа постоянно записывает файлы в раздел или том
- необходимая проверка файловых систем будет занимать меньше времени, т.к. проверка разных разделов может выполняться параллельно (еще больший выигрыш времени дает использование нескольких физических дисков)
- можно повысить безопасность системы, монтируя часть разделов в режиме read-only (только для чтения), nosuid (игнорируется бит setuid), noexec (игнорируется бит исполнения) и т.д.

Однако, у создания множества разделов есть один большой минус: при неправильной настройке можно получить систему, в которой много свободного места на одном

SPARC

John P. Davis
редактор

Pierre-Henri Jondot
редактор

Eric Stockbridge
редактор

Rajiv Manglani
редактор

Jungmin Seo
редактор

Stoyan Zhekov
редактор

Jared Hudson
редактор

Colin Morey
редактор

Jorge Paulo
редактор

Carl Anderson
редактор

Jon Portnoy
редактор

Zack Gilburd
редактор

Jack Morgan
редактор

Benny Chuang
редактор

Erwin
редактор

Joshua Kinard
редактор

Tobias Scherbaum
редактор

Xavier Neys
редактор

Grant Goodyear
рецензент

Gerald J. Normandin Jr.
рецензент

Donnie Berkholz
рецензент

Ken Nowack
рецензент

Lars Weiler

разделе, и совершенно нет на другом. Кроме того, на дисках SCSI и SATA возможно создание не более 15 разделов.

Для примера мы покажем разбиение диска объемом 20ГБ, используемого в демонстрационном ноутбуке (с веб-сервером, почтовым сервером, средой Gnome и т.д.):

Листинг 2.1: Пример файловой системы

```
$ df -h
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/hda5     ext3    509M  132M  351M  28% /
/dev/hda2     ext3    5.0G  3.0G  1.8G  63% /home
/dev/hda7     ext3    7.9G  6.2G  1.3G  83% /usr
/dev/hda8     ext3   1011M  483M  477M  51% /opt
/dev/hda9     ext3    2.0G  607M  1.3G  32% /var
/dev/hda1     ext2     51M   17M   31M  36% /boot
/dev/hda6     swap    516M   12M  504M   2% <not mounted>
(свободное место для будущего использования: 2 ГБ)
```

Раздел `/usr`, как видим, почти заполнен (использовано 83%), но когда все программы установлены, `/usr` растет не слишком быстро. Хотя отведение нескольких гигабайт дискового пространства для `/var` может показаться расточительством, помните, что этот раздел по умолчанию используется Portage для компиляции пакетов. Если вы захотите удержать `/var` в рамках более разумного размера, например, 1ГБ, вам потребуется изменить переменную `PORTAGE_TMPDIR` в `/etc/make.conf`, чтобы она указывала на раздел, где достаточно свободного места для компиляции чрезвычайно больших пакетов, таких как OpenOffice.

4.c. Использование fdisk для создания разделов

В следующих разделах описывается, как разбить диск в соответствии с примерной схемой, описанной ранее, а именно:

Раздел	Описание
/dev/hda1	загрузочный раздел
/dev/hda2	раздел подкачки
/dev/hda3	корневой раздел

Эту схему вы можете изменять по своему усмотрению.

Просмотр текущей схемы разбиения диска

`fdisk` — это популярная и очень мощная утилита для создания разделов на ваших дисках. Запустите `fdisk`, указав свой диск в качестве параметра (в примере мы используем `/dev/hda`):

Листинг 3.1: Запуск fdisk

```
# fdisk /dev/hda
```

После запуска `fdisk` выдаст такое приглашение:

Листинг 3.2: Приглашение fdisk

```
Command (m for help):
```

Нажмите `p`, чтобы вывести текущую схему разбиения диска:

Листинг 3.3: Пример схемы разделов диска

участник

Антон Битков
переводчик

Андрей Бородай
переводчик

Василий Голубев
переводчик

Алексей Глазунов
переводчик

Игорь Короть
переводчик

Сергей Кулешов
переводчик, редактор перевода

Игорь Наум
переводчик

Антон Филимонов
переводчик

Азамат Хакимов
переводчик, редактор перевода

Эльдар Цраев
переводчик

Analyzer
переводчик

Алексей Чумаков
ведущий переводчик, редактор перевода

Donate to support our development efforts.
Make a Donation

Support OSL

VR Hosted

```
Command (m for help): p

Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders
Units = cylinders of 15120 * 512 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/hda1              1        14    105808+  83  Linux
/dev/hda2             15        49    264600   82  Linux swap
/dev/hda3             50        70    158760   83  Linux
/dev/hda4             71      2184  15981840    5  Extended
/dev/hda5             71       209   1050808+  83  Linux
/dev/hda6            210       348   1050808+  83  Linux
/dev/hda7            349       626   2101648+  83  Linux
/dev/hda8            627       904   2101648+  83  Linux
/dev/hda9            905      2184   9676768+  83  Linux

Command (m for help):
```

В данном случае на диске есть семь разделов Linux (для которых в графе «System» указано «Linux») и один раздел подкачки (в списке показан как «Linux swap»).

Удаление всех разделов

Сначала удалим с диска все существующие разделы. Для удаления раздела вводите `d`. Например для удаления существующего `/dev/hda1`:

Листинг 3.4: Удаление раздела

```
Command (m for help): d
Partition number (1-4): 1
```

Удаление раздела будет запланировано. Он больше не будет отображаться при вводе `p`, но фактически не будет удалятся до тех пор, пока вы не сохраните внесенные изменения. Если вы ошиблись и хотите прервать разбиение без сохранения изменений, немедленно введите `q` и нажмите ENTER; тогда ваш раздел сохранится в неизменном виде.

Теперь, если вы действительно собираетесь удалить в своей системе все разделы, повторяйте ввод `p` для показа оставшихся разделов, затем `d` и номера удаляемого раздела до тех пор, пока разделы не кончатся. В итоге вы получите пустую таблицу разделов:

Листинг 3.5: Пустая таблица разделов

```
Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot    Start       End    Blocks   Id  System

Command (m for help):
```

Теперь, когда мы очистили таблицу разделов, хранящуюся в оперативной памяти, настало время создавать разделы. Мы воспользуемся примерной схемой разбиения, описанной выше. Естественно, не следуйте этим инструкциям буквально, если только вам не нужна таблица разделов, идентичная нашей!

Создание загрузочного раздела

Сначала создадим маленький загрузочный раздел. Введите `n` для создания нового раздела, затем `p`, чтобы выбрать первичный раздел, и `1` для указания первого

Tek Alchemy

SevenL.net

Global Netoptex Inc.

Bytemark

первичного раздела. На вопрос о первом цилиндре нажмите ввод. На вопрос о последнем цилиндре введите +32M, чтобы создать раздел размером 32МБ:

Листинг 3.6: Создание загрузочного раздела

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-3876, default 1): (нажмите ввод)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-3876, default 3876): +32M
```

Теперь, введя p, вы должны увидеть следующий текст:

Листинг 3.7: Созданный загрузочный раздел

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/hda1          1        14    105808+  83  Linux
```

Нам необходимо сделать этот раздел загружаемым. Введите a для включения признака загрузки с раздела, затем нажмите 1. Если снова ввести p, можно увидеть, что в столбце «boot» появился знак *.

Создание раздела подкачки

Теперь создадим раздел подкачки. Для этого введите n (создание нового раздела), затем p для выбора первичного раздела. Потом нажмите 2, чтобы создать второй первичный раздел, в нашем случае /dev/hda2. На вопрос о первом цилиндре просто нажмите ввод. На вопрос о последнем — ответьте +512M, чтобы создать раздел размером 512МБ. Сделав это, введите t для указания типа раздела, 2, для выбора только что созданного раздела, и 82, чтобы установить тип раздела «Linux Swap». При нажатии p по завершении этих действий, таблица разделов должна выглядеть примерно так:

Листинг 3.8: Список разделов после создания раздела подкачки

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/hda1 *        1        14    105808+  83  Linux
/dev/hda2         15        81    506520   82  Linux swap
```

Создание корневого раздела

Теперь создадим корневой раздел. Для этого введите n (создание нового раздела), затем p (первичный раздел). После этого нажмите 3 для создания третьего первичного раздела, в нашем случае /dev/hda3. На вопрос о первом цилиндре нажмите ввод. На вопрос о последнем — также нажмите ввод, чтобы раздел занял все оставшееся

свободное место на диске. По завершении этих шагов, при вводе `p` должна выводиться подобная таблица разделов:

Листинг 3.9: Список разделов после создания корневого раздела

```
Command (m for help): p

Disk /dev/hda: 30.0 GB, 30005821440 bytes
240 heads, 63 sectors/track, 3876 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/hda1 *        1        14    105808+  83  Linux
/dev/hda2         15        81    506520   82  Linux swap
/dev/hda3         82      3876  28690200   83  Linux
```

Сохранение схемы разбиения

Для сохранения схемы разбиения и выхода из `fdisk`, введите `w`.

Листинг 3.10: Сохранение и выход из fdisk

```
Command (m for help): w
```

Теперь, создав все разделы, перейдем к созданию файловых систем.

4.d. Создание файловых систем

Введение

Разделы созданы, настало время разместить на них файловые системы. Если вам безразлично, какую файловую систему использовать, и вы вполне довольны той, что мы предлагаем в книге по умолчанию, переходите к размещению файловой системы в разделе. Если нет, читайте дальше, чтобы разузнать о существующих файловых системах...

Файловые системы?

В ядре Linux поддерживаются различные файловые системы. Мы расскажем о самых распространенных из них: ext2, ext3, ReiserFS, XFS и JFS.

ext2 — испытанная файловая система Linux, в которой отсутствуют средства журналирования метаданных. Поэтому ее рядовая проверка при запуске может быть очень долгой. Сегодня существует довольно много журналируемых файловых систем нового поколения, целостность которых проверяется очень быстро, и поэтому обычно есть смысл использовать их. Журналируемые файловые системы позволяют избежать долгих задержек при запуске, когда состояние файловой системы неустойчиво.

ext3 — это журналируемая версия ext2, где для ускорения восстановления появилось журналирование метаданных, а также другие режимы, наподобие полного или упорядоченного журналирования данных. ext3 — очень хорошая и надежная файловая система. В ней есть дополнительная возможность индексации через хэшированные сбалансированные деревья (b-tree), что в большинстве случаев обеспечивает высокую скорость. Такую индексацию можно включить, добавив к команде `mke2fs` параметр `-O dir_index`. Короче говоря, ext3 — превосходная файловая система.

ReiserFS — файловая система, построенная на B*-деревьях. У нее очень хорошая скорость, и она намного (часто в 10-15 раз) быстрее ext2 и ext3 при работе с маленькими файлами (менее 4КБ). ReiserFS также великолепно масштабируется; в ней есть средства журналирования. На дату выхода ядра 2.4.18, ReiserFS признана стабильной и пригодной как для систем общего назначения, так и для крайних случаев

типа создания больших томов, использования множества маленьких файлов, гигантских файлов или же каталогов с десятками тысяч файлов.

XFS — это файловая система с журналированием метаданных и дополнительными средствами, оптимизированными для работы с максимальным объемом данных. Ее применение рекомендуется только в Linux-системах с высококлассными дисками SCSI или дисками с оптическим подключением (fibre-channel), при наличии источников бесперебойного питания. Из-за того, что XFS выполняет очень агрессивное промежуточное кэширование в оперативной памяти, при внезапном отключении питания очень вероятна потеря небрежно спроектированными программами (не принимающими адекватные меры предосторожности при записи файлов на диск, и таких довольно много) изрядного количества данных.

JFS — высокопроизводительная журналируемая файловая система производства IBM. Она лишь недавно стала пригодной для широкого применения, и пока слишком мало данных, чтобы сказать что-то хорошее или плохое о ее общей стабильности.

Размещение файловой системы в разделе

Для создания в разделе или томе файловой системы каждого типа существуют специальные средства:

Файловая система	Команда создания
ext2	`mke2fs`
ext3	`mke2fs -j`
reiserfs	`mkreiserfs`
xfs	`mkfs.xfs`
jfs	`mkfs.jfs`

Например, чтобы у загрузочного раздела (`/dev/hda1` в наших примерах) была файловая система ext2, а у корневого раздела (`/dev/hda3` в наших примерах) — ext3, требуется выполнить:

Листинг 4.1: Создание файловых систем разделов

```
# mke2fs /dev/hda1
# mke2fs -j /dev/hda3
```

Теперь самостоятельно создайте файловые системы на своих только что созданных разделах (логических томах).

Подключение раздела подкачки

`mkswap` — команда, используемая для инициализации разделов подкачки:

Листинг 4.2: Создание идентификатора раздела подкачки

```
# mkswap /dev/hda2
```

Для подключения раздела подкачки воспользуйтесь `swapon`:

Листинг 4.3: Подключение раздела подкачки

```
# swapon /dev/hda2
```

Теперь командами, приведенными выше, создайте и подключите раздел подкачки в своей системе.

4.e. Монтирование

Теперь, когда разделы созданы, а файловые системы размещены, настало время смонтировать (подключить к системе) эти разделы. Используйте команду `mount`. Не забудьте создать для каждого раздела соответствующие каталоги монтирования. Например, смонтируем корневой и загрузочный разделы:

Листинг 5.1: Монтирование разделов

```
# mount /dev/hda3 /mnt/gentoo
# mkdir /mnt/gentoo/boot
# mount /dev/hda1 /mnt/gentoo/boot
```

Примечание: Если вы хотите разместить каталог `/tmp` в отдельном разделе, не забудьте изменить права доступа к этому каталогу после монтирования: `chmod 1777 /mnt/gentoo/tmp`. Это также относится к `/var/tmp`.

Еще нам потребуется смонтировать файловую систему proc (виртуальный интерфейс ядра) в каталог `/proc`. Но сначала надо поместить в разделы нужные файлы.

Переходите к установке установочных файлов Gentoo.

[<<] [<] [В начало] [>] [>>]

Текст этого документа распространяется на условиях лицензии Creative Commons - Attribution / Share Alike.

Copyright 2001-2008 Gentoo Foundation, Inc. Questions, Comments? Contact us.