Глава 6

Администрирование файловой системы Linux

ЦЕЛИ СЕРТИФИКАЦИИ

6.01 Управление хранилищами и разделами
6.02 Форматы файловой системы
6.03 Основные файловые системы и каталоги Linux
6.04 Логическое управление томами (LVM)
6.05 Управление файловой системой
6.06 Автомонтирование
✓ Двухминутный просмотр темы
Самостоятельный тест Q & A

Установка Linux проста, по крайней мере, для тех, кто серьезно относится к сертификации Red Hat.

Однако большинство администраторов должны поддерживать существующие системы. К критическим навыкам, связанным с файловыми системами, относятся добавление новых разделов, создание логических томов, установка файловых систем и т. д. Во многих случаях вы захотите убедиться, что эти файловые системы смонтированы автоматически во время процесса загрузки, и для этого требуется подробное знание **/etc/fstab**.

Некоторые файловые системы, такие как те, к которым не часто обращаются, должны монтироваться только на временной основе; это задача автомонтировщика.

Внутри экзамена

Некоторые цели RHCSA, перечисленные в этой главе, перекрываются и могут быть охвачены несколькими разделами. Цели все связаны каким-то образом с управлением файловой системой и должны рассматриваться в целом в этой главе.

Управление разделами

Как и в реальном мире, результаты имеют значение. Неважно, используете ли вы **fdisk** или **parted** для создания раздела в стиле **MBR**. Однако вы должны знать, что реализация **fdisk в Red Hat** не поддерживает разделы **GPT**, тогда как **gdisk и parted** делают это. Убедитесь, что соответствующие разделы соответствуют требованиям экзамена.
Текущие цели RHCSA включают следующие связанные требования:

- Добавление новых разделов, логических томов и свопинг не разрушая систему
- Список, создание, удаление разделов на дисках MBR и GPT

Логические тома

Разделы и диски являются компонентами логических томов. Соответствующие цели RHCSA описывают некоторые необходимые навыки. Например, следующая цель предполагает, что вам нужно знать процесс, начинающийся с физических томов:

- Создавать и удалять физические тома, назначать физические тома для групп томов, создавать и удалять логические тома

Разумеется, логический том не выполняет весь свой потенциал, если вы не можете увеличить его размер, как это предлагает следующая цель:

- Расширение существующих логических томов

Управление файловой системой

Разделы и логические тома должны быть отформатированы, прежде чем они будут готовы хранить файлы. С этой целью вам необходимо знать, как выполнить следующие задачи RHCSA:

- Создавайте, монтируйте, размонтируйте и используйте файловые системы **vfat, ext4 и xfs**
- Подключите и отключите сетевые файловые системы **CIFS и NFS**
- Настройка систем для монтирования файловых систем при загрузке с помощью универсального уникального идентификатора (UUID) или метки

ЦЕЛЬ СЕРТИФИКАЦИИ 6.01

Управление хранилищами и разделами

Хотя в процессе установки проще создавать разделы, логические тома и массивы **RAID**, не каждый администратор имеет эту привилегию. Хотя этот раздел посвящен управлению регулярными разделами, методы, описанные в этом разделе, также используются для создания компонентов на основе раздела как логических томов, так и массивов **RAID**. После настройки раздел, логический том и массив RAID могут в каждом случае ссылаться как на тома. В Linux для администраторов, которым необходимо создавать и управлять разделами, все еще преобладают три инструмента: **fdisk, gdisk и parted**. Хотя эти инструменты в основном применяются к локальным жестким дискам, их также можно использовать для других носителей, таких как диски, подключенные по сети.

Текущее состояние системы

Прежде чем использовать утилиты **fdisk, gdisk или parted** для создания или изменения раздела, проверьте текущее доступное свободное пространство вместе с файлами в настоящее время. Следующие команды облегчают работу: **df и fdisk -l.** В следующем примере на рисунке 6-1 показано, как команда **df** отображает общее, используемое и доступное свободное пространство для всех файловых систем, находящихся в данный момент.

!!!!!!
Термины *filesystem and file system* взаимозаменяемы. Оба они используются в официальной документации Linux.
!!!!!!!

Обратите внимание на цифры в столбце **«1K-blocks».** В этом случае (за исключением временных файловых систем, **tmpfs и devtmpfs**) они составляют примерно **11,5 ГБ** выделенного пространства. Если жесткий диск больше, нераспределенное пространство может использоваться для другого раздела. Разделы могут быть в сочетании с другими, чтобы настроить дополнительное пространство в логических томах и массивах RAID, и это может быть полезно, когда вам нужно расширить пространство, доступное для соответствующих файловых систем, таких как **/home, /tmp и /var**.

РИСУНОК 6-1
Использование дискового пространства, отображаемое командой df

```
[root@server1 ~]# df
Filesystem                     1K-blocks    Used Available Use% Mounted on
/dev/mapper/rhel_server1-root  10229760 3387916   6841844  34% /
devtmpfs                         499652       0    499652   0% /dev
tmpfs                            508936      92    508844   1% /dev/shm
tmpfs                            508936    7120    501816   2% /run
tmpfs                            508936       0    508936   0% /sys/fs/cgroup
/dev/mapper/rhel_server1-home   1020588   70580    950008   7% /home
/dev/vda1                        508588  121244    387344  24% /boot
```

Чтобы напечатать размеры разделов в более читаемом формате, вы можете использовать команду **df -h**, показанную на рисунке 6-2.

РИСУНОК 6-2
Вывод команды df в удобочитаемом формате

```
[root@server1 ~]# df -h
Filesystem                     Size  Used Avail Use% Mounted on
/dev/mapper/rhel_server1-root  9.8G  3.3G  6.6G  34% /
devtmpfs                       488M     0  488M   0% /dev
tmpfs                          498M  148K  497M   1% /dev/shm
tmpfs                          498M  7.0M  491M   2% /run
tmpfs                          498M     0  498M   0% /sys/fs/cgroup
/dev/mapper/rhel_server1-home  997M   74M  924M   8% /home
/dev/vda1                      497M  119M  379M  24% /boot
[root@server1 ~]#
```

Вторая команда, **mount**, перечисляет параметры формата и монтирования для каждой файловой системы. На рисунке рассмотрим раздел, представленный устройством **/dev/mapper/rhel_server1-home**.

Обратите внимание, как он монтируется в каталоге **/home** с типом файла **xfs**. Он разделяет домашние каталоги обычных пользователей в выделенном разделе:

[root@server1 ~]# mount | grep home
**/dev/mapper/rhel_server1-home on /home type xfs **
(rw,relatime,seclabel,attr2,inode64,noquota)

Если вывод команды **mount** смущает вас, рассмотрите команду **findmnt**, которая печатает все смонтированные файловые системы в древовидном формате, как показано на рисунке 6-3.

В выводе обратите внимание на наличие «специальных файловых систем», таких как **proc и sysfs**, которые мы рассмотрим далее в этой главе.

РИСУНОК 6-3. Вывод команды findmnt

```
[root@server1 ~]# findmnt
TARGET                                SOURCE     FSTYPE     OPTIONS
/                                     /dev/mapper/rhel_server1-root
                                                 xfs        rw,relatime,seclabel,attr2,inode64,n
├─/proc                               proc       proc       rw,nosuid,nodev,noexec,relatime
│ ├─/proc/sys/fs/binfmt_misc          systemd-1  autofs     rw,relatime,fd=33,pgrp=1,timeout=300
│ └─/proc/fs/nfsd                     sunrpc     nfsd       rw,relatime
├─/sys                                sysfs      sysfs      rw,nosuid,nodev,noexec,relatime,secl
│ ├─/sys/kernel/security              securityfs securityf  rw,nosuid,nodev,noexec,relatime
│ ├─/sys/fs/cgroup                    tmpfs      tmpfs      rw,nosuid,nodev,noexec,seclabel,mode
│ │ ├─/sys/fs/cgroup/systemd          cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,xatt
│ │ ├─/sys/fs/cgroup/cpuset           cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,cpus
│ │ ├─/sys/fs/cgroup/cpu,cpuacct      cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,cpua
│ │ ├─/sys/fs/cgroup/memory           cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,memo
│ │ ├─/sys/fs/cgroup/devices          cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,devi
│ │ ├─/sys/fs/cgroup/freezer          cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,free
│ │ ├─/sys/fs/cgroup/net_cls          cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,net_
│ │ ├─/sys/fs/cgroup/blkio            cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,blki
│ │ ├─/sys/fs/cgroup/perf_event       cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,perf
│ │ └─/sys/fs/cgroup/hugetlb          cgroup     cgroup     rw,nosuid,nodev,noexec,relatime,huge
│ ├─/sys/fs/pstore                    pstore     pstore     rw,nosuid,nodev,noexec,relatime
│ ├─/sys/kernel/config                configfs   configfs   rw,relatime
│ ├─/sys/fs/selinux                   selinuxfs  selinuxfs  rw,relatime
│ ├─/sys/kernel/debug                 debugfs    debugfs    rw,relatime
│ └─/sys/fs/fuse/connections          fusectl    fusectl    rw,relatime
├─/dev                                devtmpfs   devtmpfs   rw,nosuid,seclabel,size=499652k,nr_i
│ ├─/dev/shm                          tmpfs      tmpfs      rw,nosuid,nodev,seclabel
│ ├─/dev/pts                          devpts     devpts     rw,nosuid,noexec,relatime,seclabel,g
│ ├─/dev/hugepages                    hugetlbfs  hugetlbfs  rw,relatime,seclabel
│ └─/dev/mqueue                       mqueue     mqueue     rw,relatime,seclabel
├─/run                                tmpfs      tmpfs      rw,nosuid,nodev,seclabel,mode=755
│ └─/run/user/1000/gvfs               gvfsd-fuse fuse.gvfs  rw,nosuid,nodev,relatime,user_id=100
├─/var/lib/nfs/rpc_pipefs             sunrpc     rpc_pipef  rw,relatime
```

Утилита fdisk

Утилита **fdisk** является общей для многих операционных систем. Mac OS имеет полнофункциональную версию **fdisk**. В старых версиях Microsoft Windows имеется упрощенная версия fdisk.

Хотя реализация **fdisk в Linux** включает в себя множество различных команд, вам нужно знать только те немногие, что обсуждаются здесь.

fdisk работает с разделами, созданными с использованием традиционной схемы разделения основной загрузочной записи (**MBR**). В более новых системах, которые запускают прошивку **UEFI**, а не традиционную **BIOS**, вы можете увидеть другой стандарт разделения: таблицу разделов **GUID (GPT).** Поддержка **fdisk GPT** считается экспериментальной. Предпочтительными инструментами для управления разделами **GPT** являются **gdisk и parted.**

Запустить fdisk: Справка и многое другое

Следующий экран выводит команды, которые показывают, как запустить **fdisk**, как получить справку и как выйти из программы. Драйвер **/dev/vda** связан с первым виртуальным диском на виртуальной машине на основе **KVM**. Поскольку другие системы могут быть сконфигурированы с различными файлами жестких дисков, вам может потребоваться проверить вывод команд **df и fdisk -l** для подсказок.

Когда вы запустите **fdisk**, введите **m**, чтобы перечислить основные команды **fdisk**:

 # fdisk /dev/vda
Welcome to fdisk (util-linux 2.23.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): m
Command action
a toggle a bootable flag
b edit bsd disklabel
c toggle the dos compatibility flag
d delete a partition
g create a new empty GPT partition table
G create an IRIX (SGI) partition table
l list known partition types
m print this menu
n add a new partition
o create a new empty DOS partition table
p print the partition table
q quit without saving changes
s create a new empty Sun disklabel
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)
Command (m for help): q
#

Большое количество команд связано с **fdisk** - и больше, если вы запустите команду **x**, чтобы получить дополнительную функциональность **fdisk**.

Использование fdisk: новый диск без разделов

После установки в Linux нового диска этот диск обычно не настроен на разделы. Утилиту **fdisk** можно использовать для настройки разделов на физических или виртуальных дисках, подключенных к системе. Например, базовая виртуальная система для этой книги включает в себя три диска:
/dev/vda, /dev/vdb и **/dev/vdc**.

!!!!!
SATA, PATA и SAS SCSI-диски представлены файлами устройств, такими как /dev/sda, /dev/sdb и т. д.
!!!!!!

Если недавно добавленный диск не был использован программой установки RHEL (или какой-либо другой программой управления дисками), она вернет следующее сообщение при первом открытии **fdisk**:

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xcb0a51f1.

Другими словами, даже если вы не создадите раздел после его открытия в **fdisk**, он автоматически напишет метку диска **DOS** на диск, если вы сохраните изменения.

Если вам нужен более четырех разделов на новом физическом диске, настройте первые три раздела как первичные разделы, а затем настройте четвертый раздел как расширенный раздел. Этот расширенный раздел обычно должен быть достаточно

большим, чтобы заполнить остальную часть диска; все логические разделы должны вписываться в это пространство.

Использование fdisk: в двух словах

В командной строке **fdisk** запустите команду **print (p)**, чтобы просмотреть таблицу разделов. Это позволяет просматривать текущие записи в таблице разделов. Предполагая свободное пространство, вы можете создать новый **new (n)** раздел.

Как правило, разделы являются либо первичными (p), либо логическими **(l)**. Если он еще не существует, вы также можете создать расширенный раздел **(e)**, чтобы содержать логические разделы. Помните, что в диске, отформатированном по схеме **MBR**, вы можете иметь до четырех основных разделов, которые соответствуют номерам с **1 по 4**. Один из основных разделов может быть настроен как расширенный раздел. Остальные разделы являются логическими разделами, пронумерованными 5 и выше. С расширенным разделом вы можете **создать максимум 12** логических разделов на диске.

Если доступно свободное пространство, **fdisk** обычно запускает новый раздел в первом доступном секторе или цилиндре. Фактический размер раздела зависит от геометрии диска.

Использование fdisk: создание раздела

Следующий пример вывода экрана показывает шаги, используемые для создания **(n)** первого раздела, сделать его **загрузочным (a),** а затем, наконец, **записать (w)** информацию раздела на диск. (Обратите внимание, что хотя вы можете указать раздел размером 500 МБ, геометрия диска может не допускать такого точного размера.)

 # fdisk /dev/vdb
Command (m for help): n
Command action
p primary partition (0 primary, 0 extended, 4 free)
e extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-2097151, default 2097151): +500M
Partition 1 of type Linux and of size 500 MiB is set
Command (m for help): a
Selected partition 1
Command (m for help): p
Disk /dev/vdb: 1073 MB, 1073741824 bytes, 2097152 sectors
...
Device Boot Start End Blocks Id System
/dev/vdb1 * 2048 1026047 512000 83 Linux
Command (m for help):

Обратите внимание, как количество блоков соответствует двоичному представлению 500 МБ. Повторите команды для создания любых других разделов, которые могут вам понадобиться.

Когда разделы добавляются или изменяются, вам обычно не нужно перезагружаться, чтобы заставить Linux читать новую таблицу разделов, если другой раздел на этом диске не был отформатирован и не смонтирован. Если это так, попытка записи таблицы разделов с помощью команды **w** временно не выполняется со следующим сообщением:

WARNING: Re-reading the partition table failed with error 16:
Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)

Если вы запустите **partprobe /dev/vdb**, ядро прочитает новую таблицу разделов, и вы сможете использовать вновь созданный раздел.

Использование fdisk: Разные типы разделов

Одна особенность, представляющая особый интерес, основана на команде **t** для изменения идентификатора системы разделов. Если вам нужно место для логических томов, массивов **RAID** или даже пространства подкачки, эта команда важна. После **нажатия t** вам предлагается ввести номер раздела (если настроено более одного). Затем вы можете перечислить доступные типы разделов с помощью **команды L**, как показано здесь. (Если на диске имеется только один раздел, он выбирается автоматически.)

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): L

Список доступных идентификаторов разделов, как показано на **рисунке 6-4,** впечатляет. Обратите внимание, как это не ограничивается разделами Linux. Однако, поскольку эта книга охватывает **Linux**, в **таблице 6-1** перечислены связанные типы разделов.

Если вы не вносите изменения, **введите идентификатор 83**. Вы вернетесь в командную строку **fdisk**.

Использование fdisk: удаление раздела

В следующем примере удаляется только настроенный раздел. Экран выбора образца сначала запускает **fdisk**. Затем вы можете распечатать **(p)** текущую таблицу разделов, **удалить (d) раздел по номеру (в этом случае 1), записать (w)** изменения на диск и выйти из **программы (q).** Излишне говорить, что не выполняйте это действие на любом разделе, где вам нужны данные.

Предполагая, что на этом диске только один раздел, он автоматически выбирается после запуска команды **d.**

РИСУНОК 6-4 Типы разделов Linux в fdisk

```
 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           27  Hidden NTFS Win 82  Linux swap / So c1  DRDOS/sec (FAT-
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extended  c7  Syrinx
 5  Extended        41  PPC PReP Boot   86  NTFS volume set da  Non-FS data
 6  FAT16           42  SFS             87  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            a0  IBM Thinkpad hi eb  BeOS fs
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         ee  GPT
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD         ef  EFI (FAT-12/16/
10  OPUS            55  EZ-Drive        a7  NeXTSTEP        f0  Linux/PA-RISC b
11  Hidden FAT12    56  Golden Bow      a8  Darwin UFS      f1  SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD          f4  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       ab  Darwin boot     f2  DOS secondary
16  Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      fb  VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 75  PC/IX           be  Solaris boot    ff  BBT
1e  Hidden W95 FAT1 80  Old Minix
Hex code (type L to list all codes): 
```

ТАБЛИЦА 6-1 Типы разделов Linux в fdisk

Идентификатор раздела	Описание
5	Расширенный раздел; а не тип раздела Linux, такие разделы являются обязательным условием для логических разделов. См. Также 85.
82	Linux swap.
83	Linux; подходит для всех стандартных форматов разделов Linux.
85	Расширенный раздел Linux; не признанные другими операционными системами.
88	Таблица разделов открытого текста Linux; редко используемый.
8e	Linux LVM для разделов, используемых в качестве физических томов.
FD	Linux RAID; для разделов, используемых в качестве компонентов массива RAID.

 # fdisk /dev/vdb
Command (m for help): p
Disk /dev/vdb: 1073 MB, 1073741824 bytes, 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x2e3c116d
Device Boot Start End Blocks Id System
/dev/vdb1 2048 1026047 512000 83 Linux
Command (m for help): d
Selected partition 1
Partition 1 is deleted

Это последний шанс передумать, прежде чем удалять текущий раздел. Чтобы избежать записи изменений, выйдите из **fdisk** с помощью команды **q**. Если вы довольны изменениями, которые вы сделали и хотите сделать их постоянными, перейдите к команде **w:**

Command (m for help): w

Если не появится вышеупомянутое сообщение об ошибке 16, вот и все. Теперь у вас должен быть пустой жесткий диск.

!!!!!!!
Если вы удалите раздел, таблица разделов на диске будет изменена, чтобы отразить изменение, но фактические данные в разделе не удаляются. Это означает, что если вы заново создаете раздел с использованием одного и того же макета (такие же начальные / конечные сектора), ваши данные все равно будут там. Стоит попробовать эту процедуру, если вы случайно удалите раздел по ошибке.
!!!!!!!

Использование fdisk: создание раздела swop

Теперь, когда вы знаете, как создавать разделы с помощью **fdisk**, требуется один дополнительный шаг для настройки этого раздела для пространства подкачки(**swop)**. После того, как у вас есть раздел подкачки требуемого размера, запустите команду **t**, чтобы выбрать раздел, а затем запустите команду **l**, чтобы показать типы идентификаторов разделов, указанные на рисунке 6-4.

В этом случае в следующем приглашении введите 82 для раздела подкачки Linux:

Hex code (type L to list codes): 82

Например, вы можете выполнить следующую последовательность команд для настройки нового раздела подкачки на втором жестком диске. Детали того, что вы видите, зависят от разделов, которые вы, возможно, создали. Это будет пространство подкачки 900 Мбайт на первом основном разделе **(/dev/vdb1**).

Command (m for help): n
Command action
e extended
p primary partition (1-4)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151, default 2048): 2048
Last sector, +sectors or +size{K,M,G} (2048-2097151, default 2097151): +900M
Partition 1 of type Linux and of size 900 MiB is set
Command (m for help): p
Disk /dev/vdb: 1073 MB, 1073741824 bytes, 2097152 sectors
...
Device Boot Start End Blocks Id System
/dev/vdb1 2048 1845247 921600 83 Linux
Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 82
Changed system type of partition 'Linux' to 'Linux swap / Solaris'
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

Утилита **fdisk** фактически не записывает изменения на диск до тех пор, пока вы не запустите команду **write (w).** Кроме того, вы можете отменить эти изменения с помощью команды **quit (q).** Если у вас нет сообщения об ошибке 16, описанного ранее, изменения записываются на диск. Как описано ниже в этой главе, требуется дополнительная работа для настройки RHEL для использования этого вновь созданного раздела подкачки.

Утилита gdisk

Как мы проиллюстрировали в предыдущем разделе, схема разбиения **MBR** поддерживает максимум 15 разделов для данных (3 первичных и 12 логических), плюс расширенный раздел, который является просто «контейнером» для логических разделов. Напротив, схема разбиения на GPT может содержать до 128 разделов.

Другим ограничением MBR является размер диска. В схеме MBR используются 32-битные логические адреса, которые поддерживают дисковые накопители до 2 ТБ. С другой стороны, формат GPT основан на 64-битных адресах, которые поддерживают диски объемом до 8 миллионов терабайт!

Хотя вы можете использовать **fdisk** и выбрать команду **g** для переключения в формат таблицы разделов **GPT, утилита gdisk предпочтительнее для разделов GPT**.

Если вы знакомы с **fdisk, gdisk** тоже должен выглядеть знакомым. Когда вы запускаете **gdisk** на диске с таблицей разделов **MBR**, вы увидите следующее предупреждение:

[root@server1 ~]# gdisk /dev/vdb
...
****
Found invalid GPT and valid MBR; converting MBR to GPT format.
THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by typing 'q' if
you don't want to convert your MBR partitions to GPT format!
****

Как предложено в сообщении, **введите q** для выхода; в противном случае вы можете потерять любые данные на своем диске. После запуска **gdisk** работает аналогично **fdisk**. Введите знак вопроса **(?)**, Чтобы получить список команд:

Command (? for help): ?
b back up GPT data to a file
c change a partition's name
d delete a partition
i show detailed information on a partition
l list known partition types
n add a new partition
o create a new empty GUID partition table (GPT)
q quit without saving changes
r recovery and transformation options (experts only)
s sort partitions
t change a partition's type code
v verify disk
w write table to disk and exit
x extra functionality (experts only)
? print this menu
Command (? for help):

Следующий выход на экране показывает шаги, используемые для создания нового раздела **500 МБ (n)** на дисководе **/dev/vdc**:

[root@server1 ~]# gdisk /dev/vdc
GPT fdisk (gdisk) version 0.8.6
Partition table scan:
MBR: not present
BSD: not present
APM: not present
GPT: not present
Creating new GPT entries
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-2097118, default = 2048) or {+-}size{KMGTP}: 2048
Last sector (2048-2097118, default = 2097118) or {+-}size{KMGTP}: +500M
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'
Command (? for help): w

Как и в случае с **fdisk**, инструмент **gdisk** не записывает изменения на диск до тех пор, пока вы не наберете команду **write (w).** В любой момент вы можете выйти из утилиты, не сохраняя никаких изменений с помощью команды **quit (q).**

Утилита parted

В его различных формах утилиты **parted** становится все более популярной. Это отличный инструмент, разработанный **Free Software Foundation**. Как и в случае с **fdisk**, вы можете использовать его для создания, проверки и уничтожения разделов, но он может сделать больше. Вы также можете использовать его для изменения размера и копирования разделов, а также файловых систем, содержащихся в нем. Это основа для множества инструментов управления разделом на основе **GUI, включая GParted и QtParted**. Для получения дополнительной информации см. **www.gnu.org/software/parted.**

!!!!!!
В некотором смысле, утилита parted может быть более рискованной. Например, один из авторов этой книги случайно выполнил команду mklabel из приглашения (parted) на существующую систему RHEL. Он удалил все существующие разделы. Изменения были написаны сразу же, когда расстались все еще. К счастью, была резервная копия этой виртуальной системы, и ее можно было восстановить без особых проблем.
!!!!!!

Во время обсуждения **parted** мы перейдем от раздела к разделу, считая, что **parted** все еще открыт со следующей подсказкой:

(parted)

Использование parted: запуск, получение справки и выход из системы

В следующем экране отображаются команды, которые показывают, как запустить утилиту **parted**, как получить справку и как выйти из программы. В этом случае диск

/dev/vdb связан со вторым виртуальным диском на виртуальной машине. У вашего компьютера может быть другой жесткий диск; вы можете проверить вывод команд **df** и **fdisk -l** для подсказок.

Как вы можете видеть на **рисунке 6-5**, когда выполняется разблокировка, он открывает собственное приглашение командной строки. Введите справку для списка доступных команд.

Интерфейс **parted** имеется множество команд. По сравнению с **fdisk и gdisk, parted** может сделать больше в некотором роде, как вы увидите в следующих разделах.

Использование parted: в двух словах

В командной строке команды **parted** начните работу с команды **print**, чтобы отобразить текущую таблицу разделов. Если имеется достаточное свободное пространство, вы можете сделать

РИСУНОК 6-5 Параметры команды parted

```
(parted) help
  align-check TYPE N                        check partition N for TYPE(min|opt)
        alignment
  help [COMMAND]                            print general help, or help on
        COMMAND
  mklabel,mktable LABEL-TYPE                create a new disklabel (partition
        table)
  mkpart PART-TYPE [FS-TYPE] START END      make a partition
  name NUMBER NAME                          name partition NUMBER as NAME
  print [devices|free|list,all|NUMBER]      display the partition table,
        available devices, free space, all found partitions, or a particular
        partition
  quit                                      exit program
  rescue START END                          rescue a lost partition near START
        and END
  rm NUMBER                                 delete partition NUMBER
  select DEVICE                             choose the device to edit
  disk_set FLAG STATE                       change the FLAG on selected device
  disk_toggle [FLAG]                        toggle the state of FLAG on selected
        device
  set NUMBER FLAG STATE                     change the FLAG on partition NUMBER
  toggle [NUMBER [FLAG]]                    toggle the state of FLAG on partition
        NUMBER
  unit UNIT                                 set the default unit to UNIT
  version                                   display the version number and
        copyright information of GNU Parted
(parted)
```

новый (**mkpart**) раздел или даже сделать и форматировать файловую систему (**mkpartfs**). Для получения дополнительной информации о параметрах разделенных команд используйте команду **help**; например, следующая команда предоставляет дополнительную информацию о **mkpart**:

```
(parted) help mkpart
  mkpart PART-TYPE [FS-TYPE] START END     make a partition

        PART-TYPE is one of: primary, logical, extended
        FS-TYPE is one of: btrfs, nilfs2, ext4, ext3, ext2, fat32, fat16, hfsx,
        hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs, reiserfs,
        hp-ufs, sun-ufs, xfs, apfs2, apfs1, asfs, amufs5, amufs4, amufs3,
        amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
        affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
        START and END are disk locations, such as 4GB or 10%.  Negative values
        count from the end of the disk.  For example, -1s specifies exactly the
        last sector.

        'mkpart' makes a partition without creating a new file system on the
        partition.  FS-TYPE may be specified to set an appropriate partition
        ID.
```

Если это слишком много информации, просто запустите команду. Вам будет предложено предоставить необходимую информацию.

Использование parted: новый ПК (или жесткий диск) без разделов

Первым шагом с любым действительно новым жестким диском является создание таблицы разделов. Например, после добавления нового жесткого диска в вашу виртуальную систему RHEL практически любая команда, которую вы запускаете в разделе, приводит к следующему сообщению:

Error: /dev/vdb: unrecognised disk label

Прежде чем вы сможете сделать что-нибудь еще с этим диском, вам нужно создать ярлык. Как показано в списке доступных команд, вы можете сделать это с помощью команды **mklabel**. Если вы наберете **msdos**, будет использоваться схема разделов в стиле **MBR**. Чтобы использовать формат **GTP**, введите **gpt** в командной строке:

(parted) mklabel
New disk label type? Msdos

Теперь вы можете создать новый раздел, разделенный командой **mkpart**. Естественно, если вы выбрали схему раздела **MBR**, вам нужно указать тип раздела:

(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? xfs
Start? 1MB
End? 500MB

Для **parted** мы использовали 1 МБ для начала раздела в секторе 2048. Хотя мы могли бы использовать «0MB» в качестве отправной точки, это вызвало бы предупреждение, потому что раздел не был бы правильно выровнен на границе 1 МБ для лучшей производительности. Теперь просмотрите результаты командой **print**:

(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start End Size Type File system Flags
1 1049kB 500MB 499MB primary

Если это первый раздел, который вы создали, столбец типа файловой системы будет пустым. Однако, для целей этой главы, еще не выходите из **parted**.

Разделяемые GUI-инструменты (**GParted и QtParted**) поддерживают форматирование в более широком разнообразии форматов файловой системы, даже если они просто «**front ends**» для **parted**. Они могут быть доступны из сторонних репозиториев, таких как те, которые описаны в главе 7.

Использование parted: удаление раздела

Легко удалить раздел в **parted**. Все, что вам нужно сделать из приглашения **(parted),** это использовать команду **rm** для удаления целевого раздела по номеру. Конечно, перед удалением любого раздела вы должны сделать следующее:

- **Сохраните необходимые данные из этого раздела.**

- **Размонтируйте раздел.**
- **Убедитесь, что он не настроен в /etc/fstab, поэтому Linux не пытается установить его при следующей загрузке.**
- **После запуска *parted* запустите команду *print*, чтобы определить раздел, который вы хотите удалить, а также его идентификационный номер.**

Например, чтобы удалить раздел **/dev/vdb10** из приглашения **(parted),** выполните следующую команду:

(parted) **rm 10**

Использование parted: создание раздела свопинга(swop)

Теперь давайте повторим процесс создания раздела подкачки. При необходимости удалите ранее созданный раздел, чтобы освободить место. Запустите новый раздел 1MB после окончания предыдущего раздела. Вы можете использовать те же команды, что и прежде, просто замените тип файловой системы **linux-swap** следующим образом:
p
(parted) *mkpart*
Partition type? primary/extended? *primary*
File system type? [ext2]? *linux-swap*
Start? 5*01MB*
End? *1000MB*
Now review the result with the print command:
(parted) *print*
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start End Size Type File system Flags
1 1049kB 500MB 499MB primary
2 501MB 1000MB 499MB primary

Теперь выходите из **parted**. Чтобы использовать эти разделы, вам нужно запустить такие команды, как **mkswap, swapon и mkfs.xfs**. Мы рассмотрим эти команды позже в этой главе.

(parted) *quit*
 # mkswap /dev/vdb2
 # swapon /dev/vdb2

Теперь вы можете отформатировать новый стандартный раздел **Linux** с помощью следующей команды:

 # mkfs.xfs /dev/vdb1

Выплните команду **print**. Столбец Flags для существующих разделов должен быть пустым. Теперь вы установите этот флаг с помощью команды **set**. Из приведенных здесь команд флаги устанавливают для использования первого раздела второго диска в качестве раздела **LVM**:

(parted) **set**
Partition number? **1**

Flag to Invert? **lvm**
New state? [on]/off **on**

Сейчас посмотрите результат выполнив команду **print:**

(parted) **print**
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number	Start	End	Size	Type	File system	Flags
1	1049kB	500MB	499MB	primary	xfs	**lvm**
2	501MB	1000MB	499MB	primary	linux-swap(v1)	

Вы можете использовать аналогичные шаги для настройки раздела или компонента **массива RAID**. Это также флаг; просто замените **lvm** на **raid** в ответ на только что показанный вопрос **Flag to Invert**. Если вы следуете вместе с системой RHEL 7, сначала подтвердите результат. Выйдите из раздела и выполните следующие команды:

parted /dev/vdb print
fdisk -l /dev/sdb

Вы увидите флаг **lvm**, как показано ранее из команды **parted**; вы увидите следующее подтверждение на выходе команды **fdisk**:

Device	Boot	Start	End	Blocks	Id	System
/dev/vdb1		2048	976895	487424	8e	Linux LVM

Если вы настроили базовую виртуальную систему, описанную в главе 2, это отличная возможность настроить разделы как компоненты томов **LVM**. Теперь, когда у вас есть инструменты, неважно, используете ли вы для этого **fdisk, gdisk или parted**. Вы можете использовать все свободное пространство. Просто не забудьте создать раздел на более чем одном жестком диске для этой цели, чтобы помочь проиллюстрировать мощность логических томов.

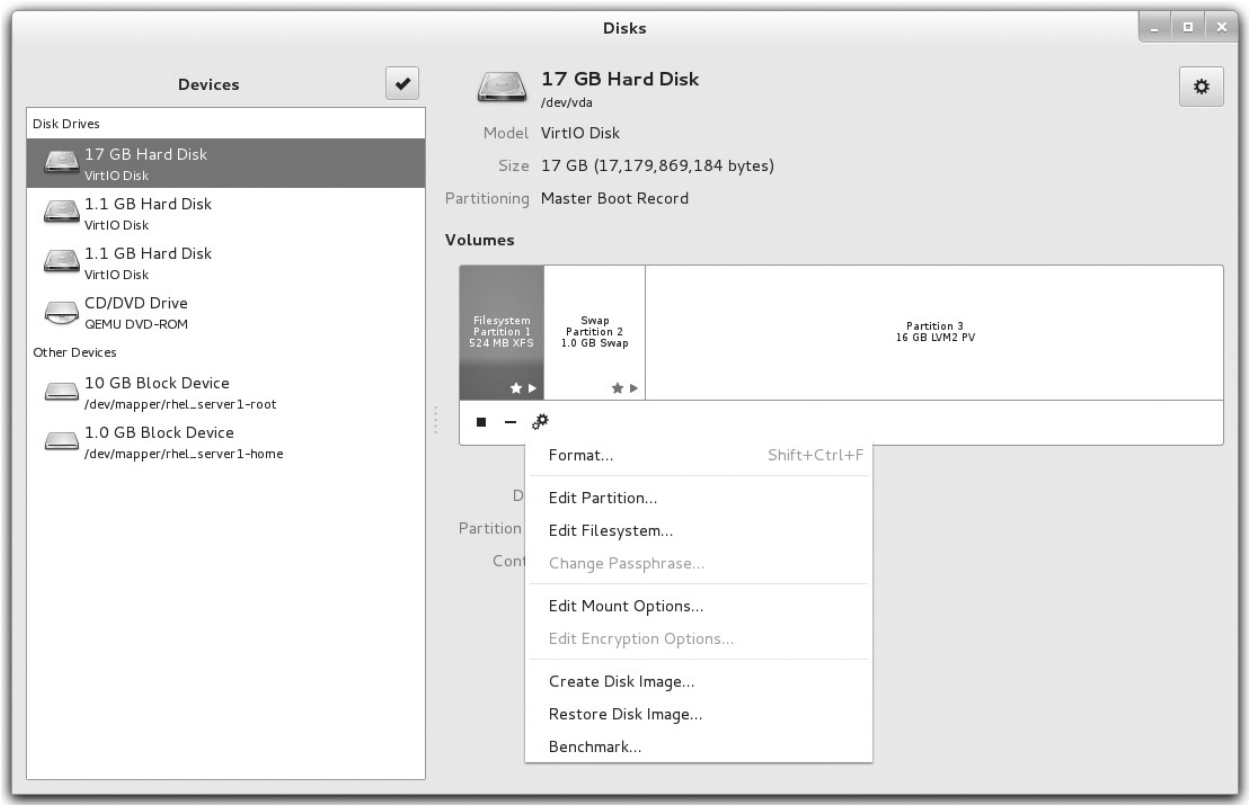
Графические параметры

Как было предложено ранее, для дисковых разделов доступны превосходные графические интерфейсы. Параметры **GParted** и **QtParted** основаны на **parted** и предназначены для рабочих сред **GNOME** и **KDE**, соответственно. Поскольку они недоступны в Red Hat Сеть, они не поддерживаются Red Hat и поэтому не будут доступны для любых экзаменов Red Hat.

Один графический параметр, доступный для RHEL 7, известен просто как **Disk Utility**, доступный из пакета **gnome-disk-utility**. После установки соответствующих пакетов вы можете открыть **Disk Utility** из командной строки с помощью команды **gnome-disks**.

Экран **Disk Utility**, показанный на **рисунке 6-6**, изображает базовую виртуальную машину, созданную в главе 2; в нем перечислены виртуальный жесткий диск, устройство **/dev/vda**, а также его корневой и домашний разделы, два дополнительных диска и DVD-привод.

Риссунок 6-6 The Disk Utility



Функциональность включает следующие параметры, доступные для клика в меню настроек **(“gear” icon)**:

- **Format**(Формат) На диске задаются форматы разделов **MBR или GPT**. На разделе, форматирует раздел для нескольких форматов файловой системы.
- **Edit partition**(Изменить раздел) Устанавливает тип раздела, например, **Linux swap** или **Linux LVM**.
- **Edit Filesystem(**Редактировать файловую систему) Устанавливает метку файловой системы; **labels**(метки) обычно использовались на RHEL 5.
- **Edit Mount Option**(Изменить параметр монтирования) Настраивает параметры монтирования файловой системы, такие как точки монтирования и тип файловой системы.
- **Create Disk Image(**Создать образ диска) Создает образ файла с содержимым диска или раздела.
- **Restore Disk Image**(Восстановление образа диска) Восстанавливает содержимое диска с образа диска.
- **Benchmark** Позволяет измерять производительность чтения и записи.
- **Unmount the Filesystem**(Демонтировать файловой системы) Отключает файловую систему. (Эта опция отображается как значок «Стоп»).
- **Delete Partition**(Удалить раздел) Удаляет раздел. (Эта опция отображается как значок «**минус**».)
- **Create Partition**(Создать раздел) Создает новый раздел. (Этот параметр отображается как значок «**плюс**».)

Не все эти опции представлены на рисунке 6-6; например, параметр «Создать раздел» не отображается, если вы не выбрали «свободную» область целевого жесткого диска. Кроме того, вы можете заметить, что функциональность Disk Utility выходит за рамки простого разбиения диска.

УПРАЖНЕНИЕ 6-1

В этом упражнении вы будете работать с утилитами **fdisk** и **parted**. Предполагается, что у вас есть новый пустой диск, например, диск на виртуальной машине. Для этого упражнения **fdisk** и **parted** будут использоваться на дисках **/dev/vdb** и **/dev/vdc** соответственно. Не стесняйтесь подставить соответственно. Сохраните результаты этой работы. Вы будете использовать его для упражнений, которые последуют далее в этой главе.

1. Запустите команду **fdisk -l /dev/vdb**, чтобы просмотреть текущее состояние диска **/dev/vdb** (если ваш первый диск имеет другое имя устройства, замените его правильным именем, например **/dev/sdb**).
2. Откройте диск **/dev/vdb** с помощью команды **fdisk /dev/vdb**.
3. Запустите команду **p** для отображения всех ранее настроенных разделов.
4. Создайте новый раздел с помощью команды **n**. Если доступны первичные разделы, создайте их с помощью команды **p**. Если представлены параметры первичных номеров разделов, выберите первый доступный.
5. Если для указания первого сектора нового раздела представлен запрос, похожий на следующий, укажите другое. Сначала попробуйте указать **сектор 1,** чтобы увидеть ответ. Затем попробуйте сектор где-то после значения по умолчанию. Для целей этого примера укажите здесь 10 000:

   Первый сектор (1845248-2097151, по умолчанию 1845248): 10000

6. Если для указания последнего сектора нового раздела представлен запрос, похожий на следующий, введите номер где-то посередине указанного диапазона. Для целей этого примера укажите здесь **1 950 000**:

   Последний сектор, + секторы или + размер {K, M, G} (1845248-2097151, по умолчанию 2097151):
   1950000
7. Выполните команду p еще раз, чтобы просмотреть результат. Запустите команду **w**, чтобы записать результат на диск.
8. Просмотрите результат на диске **/dev/vdb** с помощью команды **parted /dev/vdb print**.
9. Откройте другой доступный свободный диск **(/dev/vdc)** с помощью команды **parted /dev/vdc**.
10. В приглашении (**parted**) запустите команду **print**, чтобы просмотреть текущий статус разделов. Если вы видите сообщение об ошибке «непризнанная метка диска», запустите команду **mklabel msdos** и запустите команду **print** еще раз.
11. Создайте новый раздел с помощью команды **mkpart**. Следуйте инструкциям. Не имеет значения, является ли раздел первичным или логическим (пока не используйте расширенный раздел). Введите **xfs** в качестве типа файловой системы; запустите раздел на **100M (100MB)** и закончите его на **600M (600MB).** Запустите команду **print**, чтобы подтвердить новый раздел, и укажите номер раздела.
12. Запустите команду **quit** для выхода из раздела.
13. Запустите команду **fdisk -l /dev/vdc**, чтобы просмотреть результат.
14. Выйдите из **fdisk** с помощью команды **q**.

ЦЕЛЬ СЕРТИФИКАЦИИ 6.02

Форматы файловой системы

Количество типов файловой системы может превышать количество операционных систем. Хотя **RHEL** может работать со многими из этих форматов, по умолчанию используется **XFS**. Хотя многие пользователи используют другие файловые системы, такие как **Btrfs, Red Hat** может не поддерживать их.

Linux поддерживает множество разнообразных файловых систем. За исключением некоторых старых файловых систем, таких как **ext2**, большинство файловых систем включают в себя такие функции, **как транзакции на основе журналов, поддержку большого объема памяти, задержку распределения и сложные алгоритмы для оптимизации производительности чтения и записи**. В следующих разделах мы разделяем файловые системы на две широкие категории: «**стандартное форматирование**» и **ведение журнала**. Хотя это упрощение, достаточно классифицировать файловую систему, важную для **Linux**.

Файловые системы, описанные в этой книге, представляют собой лишь подмножество тех, которые могут быть настроены в системе RHEL. Ядро Linux позволяет настроить больше.

Стандартные форматирование файловых систем

Linux - это клон Unix. Файловые системы **Linux** были разработаны для имитации функциональных возможностей файловых систем **Unix**, доступных в то время. В первых версиях операционных систем Linux использовалась Расширенная файловая система (**ext**). В двадцатом веке **Red Hat** отформатировала свои разделы во второй расширенной файловой системе (ext2). Начиная с RHEL 5, Red Hat переместилась в третью расширенную файловую систему (**ext3**). **Для RHEL 6 Red Hat** перешла к четвертой расширенной файловой системе (**ext4**). **Оба ext3 и ext4 - это журналирование файловых систем.**

Размер существующих файловых систем увеличил важность ведения журнала, поскольку такие файловые системы более устойчивы к сбою. Так что в общем случае невозвращающиеся файловые системы, описанные в **таблице 6-2**, являются устаревшими файловыми системами. Конечно, файловые системы, такие как **ISO 9660 и swap**, все еще широко используются.

ТАБЛИЦА 6-2

Тип файловой системы	Описание
ext	Первая файловая система Linux, используемая только на ранних версиях операционной системы.
ext2 (**Второй расширенный**)	Основа для **ext3**, файловая система по умолчанию для **RHEL 5**. Файловая система **ext3** по существу **ext2** с журналированием.
swap	Файловая система подкачки Linux связана с выделенными разделами подкачки. Вероятно, вы создали хотя бы один раздел подкачки, когда вы установили RHEL.
MS-DOS и **VFAT**	Эти файловые системы позволяют вам читать файловые системы в формате **MS-DOS. MS-DOS** позволяет вам читать разделы до **Windows 95** или обычные разделы **Windows** в пределах коротких имен файлов. **VFAT** позволяет вам читать разделы **Windows 9x/NT/2000/XP/Vista/7**, отформатированные в файловой системе **FAT16 или FAT32.**
ISO 9660	Стандартная файловая система для CD-ROM. Он также известен как файловая система **High Sierra** или **HSFS** на других **Unix**-системах.

proc и sys	Две виртуальные файловые системы Linux. Виртуальный означает, что файловая система не занимает реального дискового пространства. Вместо этого файлы создаются по мере необходимости. Используется для предоставления информации о конфигурации ядра и состоянии устройства.
devpts	Реализация **Linux** для поддержки **OpenIX Unix98 PTY.**
TMPFS	Файловая система хранится в памяти. Используется на RHEL 7 для раздела **/run**.

Журнальные файловые системы

Журнальные файловые системы имеют два основных преимущества. Во-первых, они быстрее проверяют Linux во время процесса загрузки. Во-вторых, если происходит сбой, файловая система журналирования имеет лог (также известный как журнал), который может быть использован для восстановления метаданных для файлов соответствующего раздела.

Для **RHEL 7** файловая система по умолчанию - это **XFS**, очень масштабируемая файловая система на основе журнала.

Однако это не единственные доступные параметры файловой системы журнала. Мы перечисляем несколько вариантов, которые обычно используются для **RHEL** в **таблице 6-3**. Из этого списка **Red Hat** официально поддерживает только **ext3, ext4 и XFS**. На момент написания статьи **Btrfs** считается «технологическим предварительным просмотром» и не полностью поддерживается Red Hat.

Переход **Red Hat на XFS** является свидетельством его использования в качестве серверной операционной системы. Например, тома, отформатированные для **XFS**, теоретически могут достигать **8 экзабайт (EB).** Это серьезное увеличение по сравнению с максимальным объемом **ext4: 16 терабайт (ТБ**).

XFS поддерживает большое количество параллельных операций, гарантирует пространство для файлов, гарантирует более быструю проверку и многое другое. Поскольку **XFS** является частью ядра **Linux с 2004 года, это проверенная технология.**

Команды формата файловой системы

Несколько команд могут помочь вам создать файловую систему Linux. Все они основаны на команде **mkfs**, которая работает как интерфейс для конкретных команд, таких как **mkfs.ext3, mkfs.ext4 и mkfs.xfs.**

ТАБЛИЦА 6-3. Некоторые журналируемые файловые системы

Тип файловой системы	Описание
ext3	Файловая система по умолчанию для RHEL 5.
ext4	Файловая система по умолчанию для RHEL 6.
XFS	Разработанная **Silicon Graphics** как файловая система ведения журнала, она поддерживает очень большие файлы и функции, такие как индексирование индексов **B** и динамическое размещение **inodes**.
JFS	Зарегистрированная файловая система **IBM**, обычно используемая на корпоративных серверах **IBM**.
Btrfs	Файловая система **B-tree** была разработана для обеспечения набора функций, сопоставимых **с Oracle ZFS**.

	Он предлагает некоторые дополнительные функции, такие как моментальные снимки, пулы хранения и сжатие.
NTFS	Текущая файловая система **Microsoft Windows**.

Если вы хотите переформатировать существующий раздел, логический том или массив RAID, примите следующие меры предосторожности:

- **Резервное копирование любых существующих данных в разделе.**
- **Размонтируйте раздел.**

Существует два способа форматирования тома. (Как отмечалось ранее в этой главе, том представляет собой общее имя, которое может описывать раздел, RAID-массив или логический том.) Например, если вы только что создали раздел на **/dev/sdb5**, вы можете отформатировать это в файловую систему **XFS**, используя одну из следующих команд:

 # mkfs -t xfs /dev/sdb5
 # mkfs.xfs /dev/sdb5

Вы можете форматировать разделы, логические тома и массивы RAID в другие файловые системы. Варианты, доступные в RHEL 7, включают следующее:

- **mkfs.cramfs** создает сжатую файловую систему ПЗУ(ROM).
- **mkfs.ext2** форматирует том файловой системе **ext2**.
- **mkfs.ext3** форматирует том в файловую систему **ext3 RHEL 5** по умолчанию.
- **mkfs.ext4** форматирует том для файловой системы **ext4 RHEL 6** по умолчанию.
- **mkfs.fat (или mkfs.vfat, mkfs.msdos, mkdosfs)** форматирует раздел в **Microsoft**-совместимую файловую систему **FAT**; он не создает загрузочные файловые системы. (Все эти команды одинаковы, поскольку они являются символическими ссылками на **mkfs.fat**.)
- **mkfs.xfs** форматирует том в файловую систему **XFS RHEL 7** по умолчанию.
- **mkswap** устанавливает область подкачки **Linux**.

Эти команды предполагают, что вы настроили соответствующий раздел в первую очередь; например, прежде чем команда **mkswap** может быть правильно применена к разделу, для этого раздела должен быть настроен тип идентификатора раздела подкачки **Linux**. Если вы создали массив **RAID** или логический том, как описано далее в этой главе, применяются аналогичные правила.

Том Swap

Хотя **Linux** может использовать файлы подкачки, пространство подкачки обычно настраивается в правильно отформатированных разделах или логических томах. Чтобы увидеть текущее настроечное пространство подкачки, запустите команду **cat /proc/swaps**.

Как было предложено в предыдущем разделе, объемы обмена форматируются командой **mkswap**. Но этого недостаточно. Во-первых, тома **swap** должны быть активированы командой **swapon**. Если новый том подкачки распознан, вы увидите его как в файле **/proc/swaps**, так и в выводе в верхнюю команду. Во-вторых, вам нужно будет настроить новый том подкачки в файле **/etc/fstab**, как описано далее в этой главе.

Команды проверки файловой системы

Команда **fsck** анализирует указанную файловую систему и выполняет ремонт по мере необходимости. Предположим, например, что у вас возникают проблемы с файлами в каталоге **/var**, которые, монтируются на **/dev/sda7**. Если вы хотите запустить **fsck**,

сначала отключите эту файловую систему. В некоторых случаях вам может потребоваться перейти в режим восстановления(**rescue**), прежде чем вы сможете отключить файловую систему. Чтобы размонтировать, проанализировать, а затем перемонтировать файловую систему, указанную в этом разделе, выполните следующие команды:

 # umount /var
 # fsck -t xfs /dev/sda7
 # mount /dev/sda7 /var

Команда **fsck** также выполняет функцию «**front end**», в зависимости от формата файловой системы. Например, если вы форматируете файловую **систему ext2, ext3 или ext4, fsck** сам по себе автоматически вызывает команду **e2fsck**. Фактически, файлы **fsck.ext2, fsck.ext3, fsck.ext4 и e2fsck** - это разные имена для одной и той же команды! Они имеют одинаковый номер **inode**.

Вы можете подтвердить это, применив команду **ls -i** ко всем четырем файлам, которые являются частью
/sbin.

УПРАЖНЕНИЕ 6-2

Форматировать, проверять и монтировать различные файловые системы.

В этом упражнении вы будете работать с файловым форматом и проверять команды **mkfs** и **fsck**, а затем просматривать результаты с помощью команды **mount**. Это упражнение предполагает, что вы выполнили упражнение 6-1 или, по крайней мере, размонтировали разделы **Linux** без данных.

1. Просмотрите текущее состояние разделов на дисках, описанных в упражнении 6-1, с помощью команд **parted /dev/vdb print** и **fdisk -l /dev/vdc**.
2. Отформатируйте раздел, созданный первым диском командой **mkfs.ext2 /dev/vdb1**. Просмотрите текущее состояние тома с помощью команды **dumpe2fs -h /dev/vdb1|grep features**. Какие функции вы видите на выходе? Сохраняйте выход временно. Один из способов сделать это - открыть новую консоль командной строки. Проверьте систему командой **fsck.ext2 /dev/vdb1**.
3. Смонтируйте вновь отформатированный раздел с помощью команды **mount /dev/vdb1/mnt**. Просмотрите вывод с помощью команды **mount** самостоятельно. Если файл **mount** и **format** сработал, вы увидите следующий вывод:
   /dev/vdb1 on /mnt type ext2 (rw,relatime,seclabel)
4. Отключите форматированный раздел командой **umount /mnt**.
5. Запустите команду **mkfs.ext4 /dev/vdb1** и запустите команду **dumpe2fs** с предыдущего шага. В чем разница между выходом и выходом, когда раздел был отформатирован в файловой системе **ext2**?
6. Повторите шаги 3 и 4. В чем разница в выходе команды **mount**?
7. Теперь на другом разделе, созданном в упражнении 6-1, примените команду **mkfs.xfs /dev/vdc1**.
8. Установите новый форматированный раздел в каталог **/mnt**, а затем запустите команду **mount** самостоятельно. Можете ли вы подтвердить файловую систему раздела **/dev/vdc1**?

ЦЕЛЬ СЕРТИФИКАЦИИ 6.03

Основные файловые системы и каталоги Linux

Все в **Linux** можно свести к файлу. Разделы связаны с узлами устройств файловой системы, такими как **/dev/sda1**. Компоненты оборудования связаны с файлами узлов, такими как **/dev/cdrom**. Обнаруженные устройства документируются как файлы в каталоге **/sys**. Стандарт иерархии файловой системы **Filesystem Hierarchy Standard (FHS)** является официальным способом организации файлов в каталогах **Unix и Linux**. Как и в других разделах, это введение дает только самый общий обзор **FHS**. Более подробную информацию можно получить на официальной домашней странице FHS по адресу **http://refspecs.linuxfoundation.org/fhs.shtml**.

Отдельные файловые системы Linux

Несколько основных каталогов связаны со всеми современными операционными системами **Unix/Linux**. В этих каталогах хранятся файлы, драйверы, ядра, журналы, программы, утилиты и т. д. Способ, которым эти компоненты организованы на носителе, известен как файловая система. **FHS** упрощает распространение дистрибутивов **Linux** на общую структуру каталогов.

Каждый **FHS** начинается с корневого каталога верхнего уровня, также известного по его символу, единственной косой чертой **(/)**. Все остальные каталоги, показанные в **таблице 6-4**, являются подкаталогами корневого каталога. Если они не монтируются отдельно, вы также можете найти их файлы на том же разделе, что и корневой каталог. Вы не можете видеть некоторые из каталогов, указанных в таблице, если связанные пакеты не были установлены. Не все представленные каталоги официально являются частью **FHS**. Что еще более важно, не все перечисленные каталоги могут или должны монтироваться отдельно.

Смонтированые каталоги часто называются томами, которые могут охватывать несколько разделов при использовании с **Logical Volume Manager (LVM).** Хотя корневой каталог **(/)** является каталогом верхнего уровня в **FHS**, домашний каталог пользователя **root (/root)** является всего лишь подкаталогом.

ТАБЛИЦА 6-4 Базовые иерархии файловой системы Стандартные каталоги

Каталог	Описание
/	Корневой каталог, каталог верхнего уровня в **FHS**. Все остальные каталоги являются подкаталогами **root**, которые всегда устанавливаются на некоторый том.
/bin **/bin**	Основные утилиты командной строки. Не следует устанавливать отдельно; В противном случае при использовании аварийного(**rescue**) диска было бы сложно получить эти утилиты. На **RHEL 7** это символическая ссылка на **/usr/bin**.
/boot	Включает файлы запуска **Linux**, включая ядро **Linux**. По умолчанию 500 МБ обычно достаточны для типичного модульного ядра и дополнительных ядер, которые вы могли бы установить во время экзамена **RHCE** или **RHCSA**.
/dev	Аппаратные и программные драйверы устройств для всего: от флоппи-дисководов до терминалов. Не монтируйте этот каталог на отдельный том.
/etc	Большинство основных файлов конфигурации. Не монтируйте этот каталог на отдельный том.
/home	Домашние каталоги для почти каждого пользователя.
/lib	Библиотеки программ для ядра и различные утилиты командной строки. Не монтируйте этот каталог на отдельный том. На RHEL 7 это символическая ссылка на **/usr/lib**.

Каталог	Описание
/lib64	То же, что и **/lib**, но включает в себя 64-битные библиотеки. На RHEL 7 это символическая ссылка на **/usr/lib64**.
/media	Точка монтирования для съемных носителей, включая **DVD**-диски и **USB**-накопители.
/misc	Стандартная точка монтирования для локальных каталогов, установленных через автомонтировщик.
/mnt	Точка монтирования для временно смонтированных файловых систем.
/net	Стандартная точка монтирования для сетевых каталогов, установленных через автомонтировщик.
/opt	Общее место для **сторонних файлов приложений**.
/proc	Информация о листинге виртуальной файловой системы для текущих процессов, связанных с ядром, включая назначения устройств, такие как порты **IRQ**, адреса ввода-вывода и каналы **DMA**, а также параметры конфигурации ядра, такие как переадресация IP-адресов. В качестве виртуальной файловой системы Linux автоматически настраивает ее **как отдельную файловую систему в ОЗУ.**
/root	Домашний каталог пользователя **root**. Не монтируйте этот каталог на отдельный том.
/run	Файловая система **tmpfs** для файлов, которые не должны сохраняться после перезагрузки. В **RHEL 7** эта файловая система заменяет **/var/run**, которая является символической ссылкой для **/run**.
/sbin	Команды администрирования системы. Не монтируйте этот каталог отдельно. На **RHEL 7** это символическая ссылка на **/usr/bin**.
/smb	Стандартная точка подключения для удаленных общих сетевых каталогов **Microsoft**, установленных через автомонтировщик.
/srv	Обычно используется различными сетевыми серверами для дистрибутивов **не-Red Hat**.
/sys	Подобно файловой системе **/proc**. Используется для предоставления информации об устройствах, драйверах и некоторых функциях ядра.
/tmp	Временные файлы. По умолчанию Red Hat Enterprise Linux периодически удаляет все файлы в этом каталоге.
/usr	Программы и данные только для чтения. Включает множество команд администрирования системы, утилит и библиотек.
/var	Переменные данные, включая файлы журналов и принтера.

!!!!!!!!!!
В Linux выражение файловая система имеет разные значения. Он может относиться к FHS или к формату, например ext3. Точка монтирования файловой системы, такая как /var, представляет каталог, в котором может быть установлена файловая система.
!!!!!!!!!!

Каталоги, которые могут монтироваться отдельно

Если пространство позволяет, несколько каталогов, перечисленных в **таблице 6-4**, являются отличными кандидатами для монтажа отдельно. Как обсуждалось в главе 1, типично монтировать каталоги, такие как **/, /boot**, **/home**, **/opt**, **/tmp** и **/var** на отдельных томах. Иногда имеет смысл монтировать поддиректории нижнего уровня на отдельных томах, например **/var/ftp для FTP-сервера** или **/var/www для веб-сервера.**

Но сначала несколько каталогов всегда должны поддерживаться как часть файловой системы корневого каталога верхнего уровня. Эти каталоги включают **/dev, /etc и /root**. Файлы в этих каталогах необходимы для бесперебойной работы Linux в качестве операционной системы. Хотя то же самое может быть определенно для каталога **/boot**, это особый случай. Хранилище ядра Linux, начальный RAM-диск и файлы загрузчика в этом каталоге могут помочь защитить ядро операционной системы, когда возникают разные проблемы.

Файлы в каталогах **/proc** и **/sys** заполняются только во время процесса загрузки и исчезают, когда система выключается, и поэтому они хранятся в специальной виртуальной файловой системе в памяти.

Некоторые каталоги, перечисленные в **таблице 6-4**, предназначены для использования только в качестве точек монтирования. Другими словами, они обычно должны быть пустыми. Если вы храните файлы в этих каталогах, они не будут доступны, если на них будет установлен сетевой ресурс. Типичные точки подключения к сети включают каталоги **/media, /mnt, /net и /smb.**

ЦЕЛЬ СЕРТИФИКАЦИИ 6.04

Логическое управление томами (LVM)

Логическое управление томами (**LVM**, также известный **как Logical Volume Manager**) создает слой абстракции между физическими устройствами, такими как диски и разделы, и томами, которые отформатированы в файловой системе.

LVM может упростить управление дисками. В качестве примера предположим, что файловая система **/home** настроена на собственный логический том. Если дополнительное пространство доступно в группе томов, связанной с **/home**, вы можете легко изменить размер файловой системы. Если пространство не доступно, вы можете сделать больше места, добавив новый физический диск и выделив его объем памяти группе томов. В **LVM** группы томов подобны пулам хранения и объединяют емкость нескольких устройств хранения. **Логические тома находятся в группах томов и могут охватывать несколько физических дисков.**

!!!!!!!!!!
LVM - это важный инструмент для управления пространством, доступным для разных томов.
!!!!!!!!!!!!

Определения в LVM

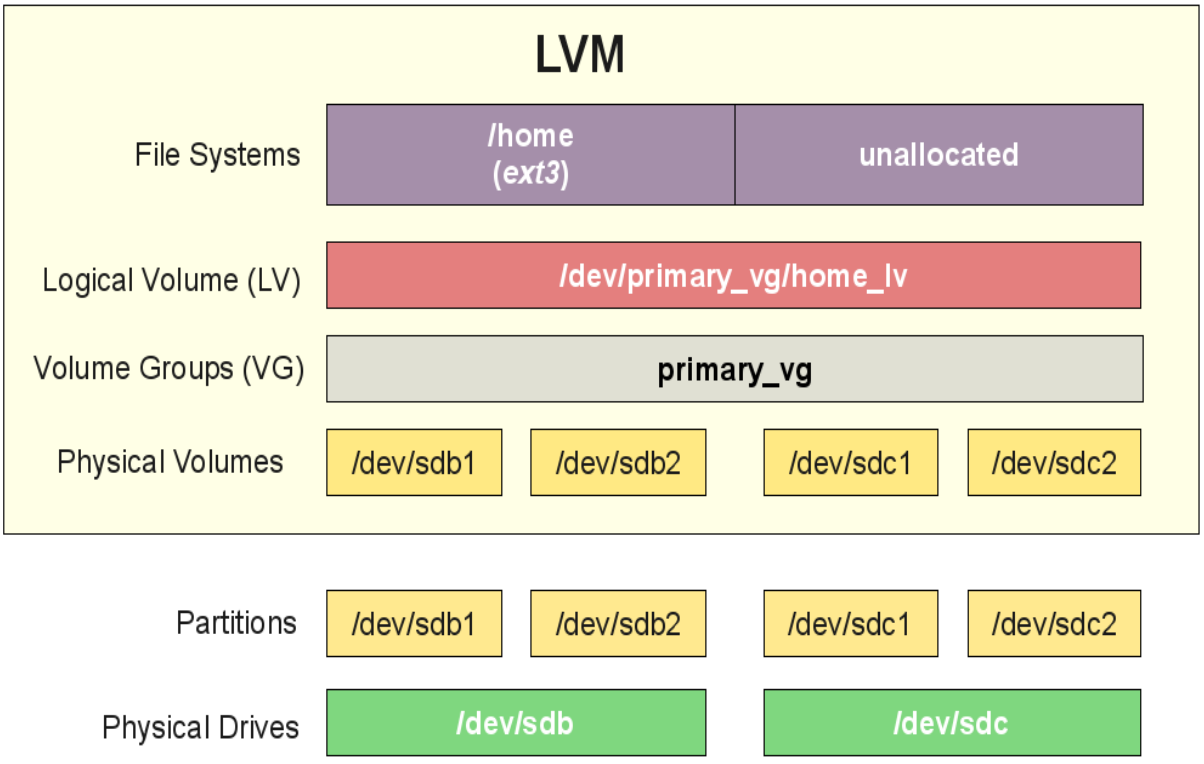
Чтобы работать с **LVM**, вам нужно понять, как используются разделы, настроенные для этой цели. Во-первых, с помощью утилиты **fdisk, gdisk** или **parted** вам необходимо создать разделы, настроенные как тип раздела **LVM**. Вы также можете использовать все дисковое устройство.

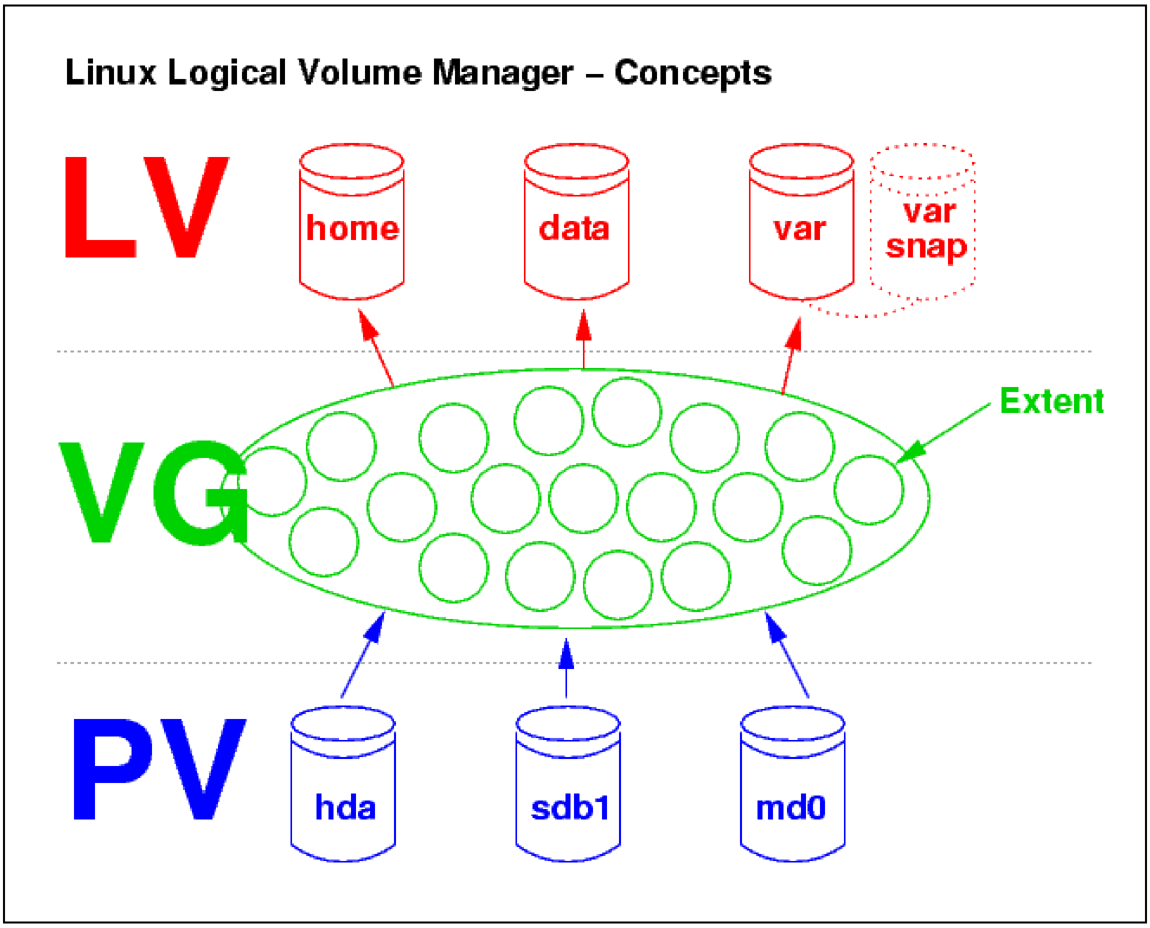
Как только эти разделы или дисковые устройства доступны, их необходимо настроить как физические тома (**PV *physical volumes***). Этот процесс инициализирует диск или раздел для использования **LVM**. Затем вы создаете группы томов (**VG *volume groups***) из одного или нескольких физических томов. Группы томов организуют физическое хранилище в наборе управляемых дисковых фрагментов, известных как физические элементы(диапазон) (**PE *physical extent***). С помощью правильных команд вы можете организовать **PE** в логические тома (LVs ***logical volume***). Логические тома сделаны из логических элементов(диапазонов, областей) (LE ***logical extent***), которые сопоставляются

с лежащим в основе **PE**. Затем вы можете форматировать и монтировать **LV**. Для тех, кто новичок в **LVM**, может быть важно разбить каждое определение:

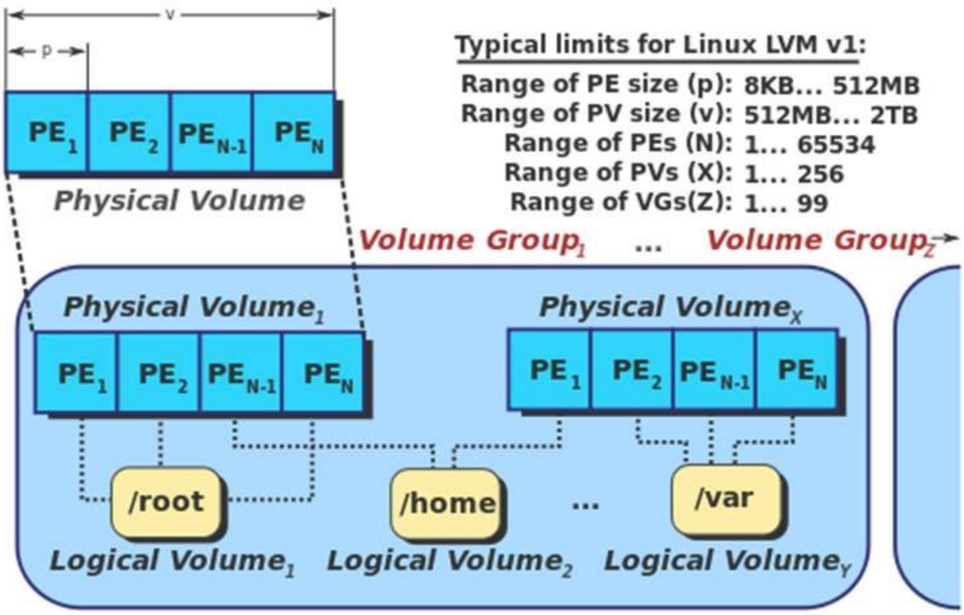
- **Физический том (PV Physical volume) PV** - это раздел или диск, инициализированный для используемый **LVM**.
- **Физический диапазон (PE physical extent) PE** - это небольшой однородный сегмент дискового пространства. **PV** разбиваются на **PE**.
- **Группа томов (VG volume groups) VG** - это пул хранения, состоящий из одного или нескольких **PV**.
- **Логический диапазон(область) (LE)** Каждый **PE** связан с **LE**, и эти **PE** могут быть объединены в логический том.
- **Логический том (LV) LV** является частью **VG** и состоит из **LE**. **LV** может быть отформатирован с файловой системой, а затем смонтирован в каталоге по вашему выбору.

!!!!! Из другого источника:

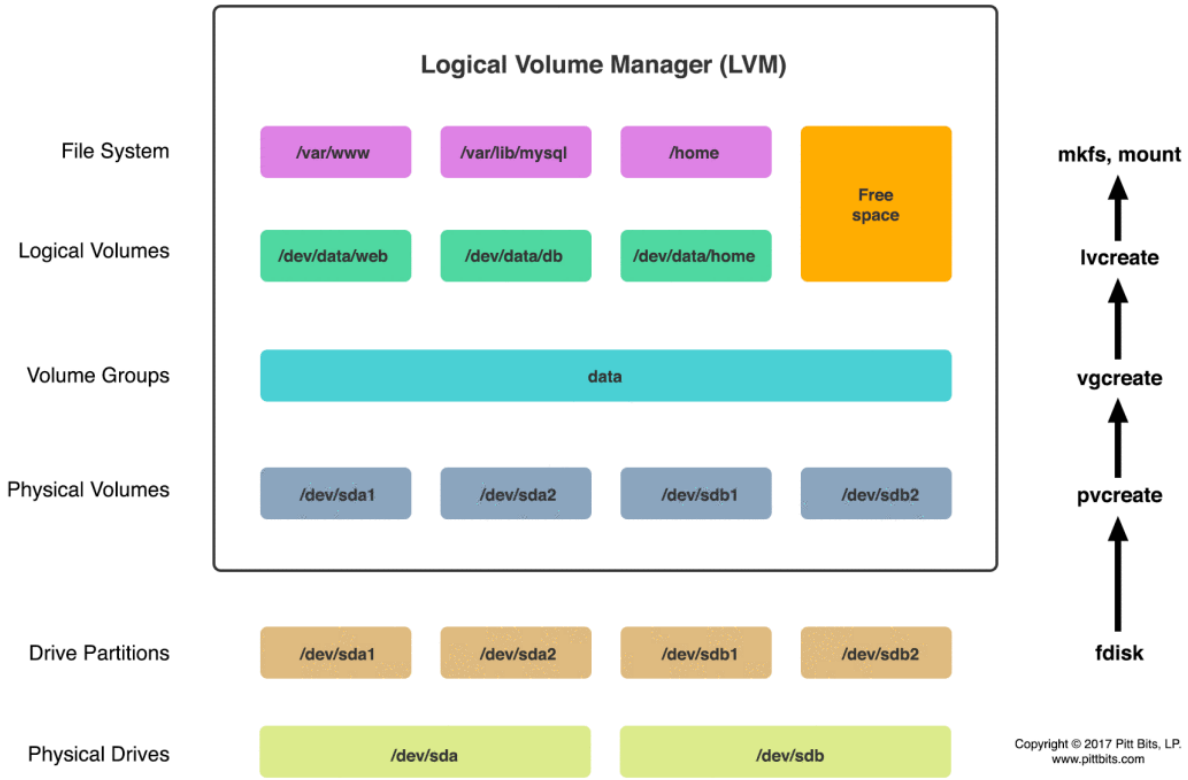




Структура Volume Group



!!



!!

Вы увидите это в следующих разделах. По сути, для создания системы **LV** вам необходимо создать новый **PV** с помощью команды **pvcreate**, назначить пространство из одного или нескольких **PV** для **VG** с помощью команды, такой как **vgcreate**, и выделить пространство из некоторой части доступных **VGs к LV** с такой командой, как **lvcreate**.

Чтобы добавить пространство к существующему логическому тому, вам нужно добавить свободное пространство из существующего **VG** с такой командой, как **lvextend**. Если у вас нет какого-либо существующего пространства **VG**, вам нужно добавить его с не назначенным пространством **PV** с помощью команды, такой как **vgextend**. Если все ваши **PV** будут задействованы, вам может потребоваться создать новый **PV** с не назначенного раздела или жесткого диска с помощью команды **pvcreate**.

Создание физического тома

Первый шаг - начать с физического раздела или жесткого диска. Основываясь на обсуждении ранее в этой главе, вы должны иметь возможность настраивать разделы в соответствии с идентификатором **LVM Linux**. Затем, чтобы настроить новый **PV** на правильно настроенном разделе, например **/dev/sda1**, примените команду **pvcreate** к этому разделу:

 # pvcreate /dev/sda1

Если в настройках **PV** имеется более одного раздела, все связанные файлы устройств могут быть перечислены в одной команде:

 # pvcreate /dev/sda1/ dev/sda2/ dev/sdb1 /dev/sdb2

Создание группы томов

Из одного или нескольких **PV** вы можете создать группу томов (**VG**). В следующей команде замените имя по вашему выбору для ***volumegroup***:

 # vgcreate *volumegroup* /dev/sda1 /dev/sda2

Вы можете включить дополнительные **PV** в любой **VG**. Предположим, что существуют существующие **PV**, основанные на **/dev/sdb1** и **/dev/sdb2**, вы можете добавить в ***volumegroup*** **VG** следующей командой:

 # vgextend volumegroup /dev/sdb1 /dev/sdb2

Создание логического тома

Однако нового **VG** недостаточно, так как вы не можете форматировать или монтировать файловую систему на нем. Поэтому для этого вам необходимо создать логический том (**LV**). Следующая команда создает **LV**. Вы можете добавить столько кусков дискового пространства в **PE**, сколько вам нужно.

 # lvcreate -l number_of_PEs *volumegroup* -n logvol

Этой командой создаётся устройство с именем **/dev/volumegroup/logvol**. Вы можете отформатировать это устройство, как если бы это был обычный раздел диска, а затем смонтировать этот новый логический том в каталоге.

Но это не полезно, если вы не знаете, сколько места связано с каждым **PE**. Вы можете использовать команду **vgdisplay** для отображения размера **PE** или указать размер с параметром **-s** команды **vgcreate** при инициализации группы томов. Кроме того, вы можете использовать ключ **-L** для установки размера в **MB, GB** или другой единице измерения. Например, следующая команда создает **LV с именем flex 200MB**:

 # lvcreate -L 200M *volumegroup* -n flex

Использовать логический том

Но это не последний шаг. Вы не можете получить полный доступ, если логический том не будет отформатирован и не смонтирован при перезагрузке системы. Этот процесс описан ниже в этой главе в обсуждении файла конфигурации **/etc/fstab**.

Больше команд LVM

Доступен широкий спектр команд **LVM**, связанных с **PV, LV и VG.** Обычно это **pv*, lv* и vg*** в каталоге **/usr/sbin**. Команды физического тома перечислены в **Таблице 6-6**.

Поскольку вы назначаете **PV для VG и LV**, вам могут потребоваться команды для управления и настройки их. **Таблица 6-6** содержит обзор большинства связанных групп групп томов.

Поскольку вы назначаете **PV для VG**, а затем подразделяете **VG на LV**, вам могут потребоваться команды для управления ими и их настройки. **Таблица 6-7** содержит обзор связанных команд **LVM.**

ТАБЛИЦА 6-5 Команды управления физическими томами

Команда физического тома	Описание
pvchange	Изменяет атрибуты **PV**: команда **pvchange –x n /dev/sda10** отключает распределение **PE** из раздела **/dev/sda10**.
pvck	Проверяет согласованность меqqqтаданных физического тома.
pvcreate	Инициализирует диск или раздел как **PV**; раздел должен быть помечен типом файла **LVM**.
pvdisplay	Отображает текущие настроенные **PV**.
pvmove	Перемещает **PE в VG** из указанного **PV** в свободные места на других **PV**; обязательное условие для отключения **PV**. Один пример: **pvmove /dev/sda10**.
pvremove	Удаляет данный **PV** из списка распознанных томов: например, **pvremove /dev/sda10**.
pvresize	Изменяет количество разделов, выделенных для **PV**. Если вы развернули раздел **/dev/sda10**, pvresize **/dev/sda10** использует дополнительное пространство. Кроме того, **pvresize --setphysicalvolumesize 100M /dev/sda10** уменьшает количество **PV**, взятых из этого раздела в указанное пространство.
pvs	Перечисляет настроенные **PV** и связанные **VG**, если таковые назначены.
pvscan	Сканирует диски на физические тома.

ТАБЛИЦА 6-6 Команды групп томов

Команда группы томов	Описание
vgcfgbackup **vgcfgrestore**	Используется для резервного копирования и восстановления метаданных, связанных с **LVM**; по умолчанию файлы резервных копий находятся в каталоге **/etc/lvm**.
vgchange	Подобно **pvchange**, эта команда позволяет вам изменять параметры конфигурации **VG**. Например, **vgchange -a y** включает все локальные **VG.**
vgck	Проверяет согласованность метаданных группы томов.
vgconvert	Поддерживает преобразования из систем **LVM1 в LVM2**: **vgconvert -M2 VolGroup00** преобразует **VolGroup00** в формат метаданных **LVM2.**
vgcreate	Создает **VG** из одного или нескольких настроенных **PV**: например, **vgcreate vgroup00 /dev/sda10 /dev/sda11** создает **vgroup00 из PV**, как определено в **/dev/sda10**, и **/dev/sda11**.
vgdisplay	Отображает характеристики настроенных в данный момент **VG**.
vgexport **vgimport**	Используется для экспорта и импорта неиспользуемых **VG** из доступных; Команда **vgexport -a** экспортирует все неактивные **VG**.
vgextend	Если вы создали новый **PV**, **vgextend vgroup00 /dev/sda11** добавляет пространство из **/dev/sda11** в **vgroup00**.
vgmerge	Если у вас есть неиспользуемая **VG vgroup01**, вы можете объединить ее с **vgroup00** с помощью следующей команды: **vgmerge vgroup00 vgroup01**.
vgmknodes	Запустите эту команду, если у вас возникли проблемы с файлами устройств **VG**.

Vgreduce	Команда **vgreduce vgroup00 /dev/sda11** удаляет **PV /dev/sda11 PV из vgroup00**, предполагая, что **/dev/sda11** не используется.
Vgremove	Команда **vgremove vgroup00** удаляет **vgroup00,** предполагая, что ему не назначены **LV**.
vgrename	Позволяет переименовывать **LV**.
vgs	Отображает основную информацию о настроенных **VG**.
vgscan	Сканирует все устройства на **VG**.
vgsplit	Разбивает группу томов.

ТАБЛИЦА 6-7 Команды логического тома

Команда логического тома	Описание
lvchange	Подобно **pvchange**, эта команда изменяет атрибуты **LV**: например, команда **lvchange -a n vgroup00/lvol00** отключает использование **LV** с меткой **lvol00**.
lvconvert	Преобразует логический том между различными типами, такими как линейный, зеркальный или снимок.
lvcreate	Создает новый **LV** в существующем **VG**. Например, **lvcreate -l 200 volume01 -n lvol01** создает **lvol01**, используя **200** экстентов(**extents)** в **VG** с именем **volume01**.
lvdisplay	Отображает настроенные в данный момент **LV**.
lvextend	Добавляет пространство к **LV:** команда **lvextend -L 4G /dev/volume01 /lvol01** расширяет **lvol01 до 4 ГБ**, предполагая, что пространство доступно.
lvreduce	Уменьшает размер **LV**; если в уменьшенной области есть данные, они будут потеряны.
lvremove	Удаляет активный **LV:** команда **lvremove volume01/lvol01** удаляет **LV lvol01** из **VG volume01**.
lvrename	Переименовывает **LV**.
lvresize	Изменяет размер **LV;** можно сделать с помощью **-L** для размера. Например, **lvresize -L + 4GB volume01 /lvol01** добавляет 4 ГБ к размеру **lvol01**.
lvs	Перечисляет все настроенные **LV**.
lvscan	Сканирует все **LV**.

РИСУНОК 6-7 Конфигурация группы томов (VG)

```
[root@server1 ~]# vgdisplay
  --- Volume group ---
  VG Name               rhel_server1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               14.53 GiB
  PE Size               4.00 MiB
  Total PE              3720
  Alloc PE / Size       2750 / 10.74 GiB
  Free  PE / Size       970 / 3.79 GiB
  VG UUID               oYuR2x-uaUH-AZsZ-McNz-92Jh-qfYk-Ma0FDT

[root@server1 ~]#
```

Вот пример, как это работает. Попробуйте команду **vgscan**. Вы можете проверить настроенные группы томов (**VG**) с помощью команды **vgdisplay**. Например, **рисунок 6-7** иллюстрирует конфигурацию **VG rhel_server1**.

Хотя установлено несколько команд **lvm*,** активны только четыре из них: **lvm**, **lvmconf, lvmdiskscan и lvmdump**. Другие команды **lvm*** устарели или еще не реализованы. Команда **lvm** перемещается в приглашение **lvm>**. Это довольно интересно, так как команда help в этом приглашении предоставляет почти полный список доступных команд **LVM**.

Команда **lvmconf** может изменить настройки по умолчанию в соответствующем файле конфигурации, **/etc/lvm/lvm.conf**. Команда **lvmdiskscan** сканирует все доступные диски на наличие физических томов, настроенных **LVM**. Наконец, команда **lvmdump** устанавливает отчет о конфигурации в домашнем каталоге пользователя **root (/root).**

Удалить логический том

Удаление существующего **LV** является простым, с командой **lvremove**. Это предполагает, что любые каталоги, ранее смонтированные на **LV**, были размонтированы. На этом этапе основные шаги просты:

1. Сохраните любые данные в каталогах, которые смонтированы на **LV**.
2. Размонтируйте файловую систему, связанную с **LV**. В качестве примера вы можете использовать команду, подобную следующей:
   # umount /dev/vg_01/lv_01
3. Примените команду lvremove к LV с помощью такой команды:
   # lvremove /dev/vg_01/lv_01
4. Теперь у вас должны быть свободные **LE** из этого **LV** для использования в других **LV**.

Изменить размер логических томов

Если вам нужно увеличить размер существующего **LV**, вы можете добавить к нему пространство от вновь созданного **PV**. Все, что для этого нужно, - это правильное использование команд **vgextend и lvextend**. Например, чтобы добавить **PE** в **VG**, связанную с каталогом **/home**, смонтированным на **LV**, выполните следующие основные шаги:

1. Сделайте резервную копию любых данных, существующих в каталоге **/home**. (Это стандартная мера предосторожности, которая не требуется, если все идет хорошо. Вы можете даже пропустить этот шаг на экзаменах **Red Hat**. Но действительно ли вы хотите рисковать пользовательскими данными на практике?)
2. Расширьте **VG**, чтобы включить новые разделы, настроенные на соответствующий тип. Например, чтобы добавить **/dev/sdd1** в **VG_00 VG**, выполните следующую команду:
   # vgextend vg_00 /dev/sdd1
3. Убедитесь, что новые разделы включены в **VG** с помощью следующей команды:
   # vgdisplay vg_00
4. Теперь вы можете расширить пространство, выделенное для текущего **LV**. Например, чтобы расширить **LV до 2000 МБ**, выполните следующую команду:
   # lvextend -L 2000M /dev/vg_00/lv_00
5. Команда **lvextend** может увеличить пространство, выделенное для **LV**, в КБ, МБ, ГБ или даже ТБ. Например, вы можете указать **2 ГБ LV** с помощью следующей команды:
   # lvextend -L 2G /dev/vg_00/lv_00

Если вы предпочитаете указывать дополнительное пространство для добавления, а не общее пространство, вы можете использовать синтаксис в следующем примере, который добавляет 1 ГБ к логическому тому:
 # lvextend -L + 1G /dev/vg_00/lv_00

6. Измените размер отформатированного тома с помощью команды **xfs_growfs** (или с помощью **resize2fs**, если это файловая система **ext2/ext3/ext4**). Если вы используете весь расширенный **LV**, команда проста:
 # xfs_growfs /dev/vg_00/lv_00
7. В качестве альтернативы вы можете переформатировать **LV**, используя команды, описанные ранее, чтобы файловая система могла в полной мере воспользоваться новым пространством, а затем восстановить данные из резервной копии. (Если вы уже успешно изменили размер **LV**, не переформатируйте его. Это не нужно и уничтожит существующие данные!)
 # mkfs.xfs -f /dev/vg_00/lv_00
8. В любом случае вы бы завершили процесс, проверив новый размер файловой системы с помощью команды **df:**
 # df –h

ЦЕЛЬ СЕРТИФИКАЦИИ 6.05

Управление файловой системой

Прежде чем вы сможете получить доступ к файлам в каталоге, этот каталог должен быть смонтирован в разделе, отформатированном для некоторой читаемой файловой системы. Linux обычно автоматизирует этот процесс, используя

Файл конфигурации **/etc/fstab**. Когда Linux проходит процесс загрузки, каталоги, указанные в **/etc/fstab**, монтируются на настроенных томах с помощью команды mount. Конечно, вы можете запустить эту команду с любыми или всеми соответствующими параметрами, так что это отличное место для начала этого раздела.

В оставшейся части этого раздела рассматриваются параметры для **/etc/fstab**. Хотя он начинается со значения по умолчанию с использованием базовой конфигурации для стандартной виртуальной машины, он включает параметры для настройки этого файла для локальной, удаленной и съемной файловых систем.

Файл /etc/fstab

Чтобы просмотреть содержимое файла **/etc/fstab**, выполните команду **cat /etc/fstab.** Из примера, показанного на **рисунке 6-8**, различные файловые системы настраиваются в каждой строке.

В RHEL 7 по умолчанию используется **UUID** для монтирования файловых систем **не-LVM**. Как вы увидите в следующем разделе, **UUID** могут представлять **раздел**, **логический том или массив RAID**. Во всех случаях тома должны быть отформатированы в файловой системе, указанной в каждой строке, и смонтированы в каталоге, указанном во втором столбце. Преимущество **UUID** и устройств логических томов заключается в том, что они уникальны, в то время как имена устройств, такие как **/dev/sdb2**, могут измениться после перезагрузки в зависимости от порядка инициализации дисков.

Но в некоторой степени **UUID** не имеют значения. Как показано **на рисунке 6-8**, шесть полей связаны с каждой файловой системой, как показано слева направо в **таблице 6-8**. Вы можете проверить как разделы фактически монтируются в файле **/etc/mtab**, как показано на **рисунке 6-9**.

РИСУНОК 6-8 Пример /etc/fstab

```
[root@server1 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Mon Feb  2 17:41:03 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/rhel_server1-root /                       xfs     defaults        1 1
UUID=c89968bc-acc5-4d60-8deb-97542cb766c6 /boot                   xfs     defaults        1 2
/dev/mapper/rhel_server1-home /home                   xfs     defaults        1 2
UUID=9d37eaf0-2c0b-4e57-b05f-87c2e21d3a95 swap                    swap    defaults        0 0
[root@server1 ~]#
```

РИСУНОК 6-9 Пример /etc/mtab

```
[root@server1 ~]# cat /etc/mtab
rootfs / rootfs rw 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=499652k,nr_inodes=124913,mode=755 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev 0 0
devpts /dev/pts devpts rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,seclabel,nosuid,nodev,mode=755 0 0
tmpfs /sys/fs/cgroup tmpfs rw,seclabel,nosuid,nodev,noexec,mode=755 0 0
cgroup /sys/fs/cgroup/systemd cgroup rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/
lib/systemd/systemd-cgroups-agent,name=systemd 0 0
pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,nosuid,nodev,noexec,relatime,cpuset 0 0
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,nosuid,nodev,noexec,relatime,cpuacct,cpu 0 0
cgroup /sys/fs/cgroup/memory cgroup rw,nosuid,nodev,noexec,relatime,memory 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/net_cls cgroup rw,nosuid,nodev,noexec,relatime,net_cls 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,nosuid,nodev,noexec,relatime,blkio 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,nosuid,nodev,noexec,relatime,perf_event 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,nosuid,nodev,noexec,relatime,hugetlb 0 0
configfs /sys/kernel/config configfs rw,relatime 0 0
/dev/mapper/rhel_server1-root / xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs rw,relatime,fd=33,pgrp=1,timeout=300,minproto=5,maxp
roto=5,direct 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime 0 0
debugfs /sys/kernel/debug debugfs rw,relatime 0 0
mqueue /dev/mqueue mqueue rw,seclabel,relatime 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
sunrpc /proc/fs/nfsd nfsd rw,relatime 0 0
/dev/mapper/rhel_server1-home /home xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
/dev/vda1 /boot xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0
fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0
gvfsd-fuse /run/user/1000/gvfs fuse.gvfsd-fuse rw,nosuid,nodev,relatime,user_id=1000,group_id=
1000 0 0
/dev/sr0 /run/media/alex/RHEL-7.0\040Server.x86_64 iso9660 ro,nosuid,nodev,relatime,uid=1000,g
id=1000,iocharset=utf8,mode=0400,dmode=0500 0 0
[root@server1 ~]#
```

ТАБЛИЦА 6-8 Описание /etc/fstab по столбцам слева направо

Имя поля	**Описание**
Устроиство (Device)	Перечисляет устройство для установки; Вы можете заменить **UUID** или путь к устройству.
Точка монтирования (Mount Point)	Отмечает каталог, в который будет смонтирована файловая система.
Формат файловой системы (Filesystem Format)	Описывает тип файловой системы. Допустимые типы файловых систем: **xfs,**

	ext2, ext3, ext4, msdos, vfat, iso9660, nfs, smb, swap и многие другие.
Варианты монтирования (Mount Options)	Охвачено в следующем разделе.
Значение дампа (Dump Value)	**Либо 0, либо 1**. Если вы используете команду **dump** для резервного копирования файловых систем, это поле определяет, какие файловые системы должны быть выгружены.
Порядок проверки файловой системы (Filesystem Check Order)	Определяет порядок проверки файловых систем командой **fsck** в процессе загрузки. Для файловой системы корневого каталога (**/**) должно быть установлено значение **1**, а для других локальных файловых систем - значение **2**. Сменные файловые системы, например связанные с дисководами **CD/DVD,** должны быть установлены в **0**, что означает, что они не проверяются во время **Linux**. процесс загрузки.

Обратите внимание на различия, особенно использование файла устройства вместо **UUID**, и наличие виртуальных файловых систем, таких как **tmpfs** и **sysfs**, которые обсуждаются далее в этой главе.

При добавлении нового раздела вы можете просто добавить файл устройства, связанный с разделом или логическим томом, в первый столбец.

Универсальные уникальные идентификаторы в /etc/fstab

Обратите внимание на то, что в **/etc/fstab** основное внимание уделяется идентификаторам **UUID**, сокращенно универсальным уникальным идентификаторам. Каждый отформатированный том имеет **UUID**, уникальный 128-битный номер. Каждый **UUID** представляет собой раздел, логический том или массив **RAID.**

Чтобы определить **UUID** для доступных томов, выполните команду **blkid** с именем устройства в качестве аргумента. Выход выдаст UUID устройства. Например, чтобы получить UUID «корневого» логического тома в группе томов «rhel_server1», выполните следующую команду:

 # blkid /dev/rhel_server1/root/dev/rhel_server1/root: UUID="2142e97a-dbec-495c-b7d9-1369270089ff" TYPE="xfs"

В качестве альтернативы вы можете использовать команды **xfs_admin и dumpe2fs** для файловых систем **XFS** и **ext2/ext3/ext4** соответственно; например, следующая команда идентифицирует **UUID**, связанный с отмеченным **LV**:

 # xfs_admin -u /dev/rhel_server1/root

Поскольку **UUID** не ограничены **LV**, вы должны иметь возможность получить эквивалентную информацию для раздела с помощью команды, такой как следующее:
r
 # xfs_admin -u /dev/vda1

Конечно, то же самое верно, если у вас есть настроенный и отформатированный ext том с такой командой:

 # dumpe2fs /dev/mapper/rhel_server1-test | grep UUID

Команда mount

Команда **mount** может использоваться для подключения локальных и сетевых разделов к указанным каталогам. Точки крепления не фиксированы; Вы можете подключить дисковод компакт-дисков или даже общий сетевой каталог к любому пустому каталогу, если установлены соответствующие владельцы и разрешения. С ней тесно связана команда **umount** (not unmount), которая размонтирует выбранные тома из связанных каталогов.

Сначала попробуйте команду **mount** самостоятельно. Он отобразит все подключенные в настоящее время файловые системы, а также важные параметры монтирования. Например, следующий вывод предполагает, что Том **/dev/mapper/rhel_server1-root** монтируется в корневом каталоге верхнего уровня в режиме чтения-записи и форматируется в файловой системе **xfs**:

/dev/mapper/rhel_server1-root on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)

Как предлагалось ранее, команда **mount** тесно связана с файлом **/etc/fstab**. Если вы размонтировали каталог и внесли изменения в файл **/etc/fstab**, самый простой способ смонтировать все файловые системы, настроенные в данный момент в файле **/etc/fstab**, с помощью следующей команды:

 # mount -a

Однако, если файловая система уже смонтирована, эта команда не изменяет свой статус, независимо от того, что было сделано с файлом **/etc/fstab**. Но если система впоследствии будет перезагружена, параметры, настроенные в **/etc/fstab**, будут использованы автоматически.

Если вы не уверены в возможном изменении файла **/etc/fstab**, его можно проверить с помощью команды mount. Например, следующая команда перемонтирует том, связанный с каталогом / boot, в режиме только для чтения:

 # mount -o remount, ro /boot

Вы можете подтвердить результат, повторно запустив команду монтирования. Следующий вывод должен отражать результат в каталоге /boot:

/dev/vda1 on /boot type xfs (ro,relatime,seclabel,attr2,inode64,noquota)

Если вы читали эту книгу с самого начала, вы уже видели команду mount при работе со списками контроля доступа (ACL) и даже с файлами ISO, связанными с загруженными CD/DVD. Чтобы просмотреть, следующая команда перемонтирует отмеченный / home каталог с ACL:

 # mount -o remount, acl /dev /vda5 /home

А для файлов ISO следующая команда монтирует указанный файл ISO RHEL 7 в каталог / mnt:

 # mount -o loop rhel-server-7.0-x86_64-dvd.iso /mnt

Дополнительные параметры монтирования файловой системы

Многие параметры команды монтирования подходят для файла **/etc/fstab**. В этом файле чаще всего используется опция по умолчанию. Хотя это подходящая опция монтирования для большинства файловых систем **/etc/fstab**, есть и другие опции, такие как перечисленные в **Таблице 6-9**. Если вы хотите использовать несколько опций, разделите их запятыми. Не используйте пробелы между опциями. Список в **Таблице 6-9 не является исчерпывающим. Вы можете узнать больше на man-странице mount, доступной с помощью команды man mount.**

ТАБЛИЦА 6-9 Опции для команды монтирования и /etc/fstab

Варианты монтирования	Описание
async	Все операции ввода-вывода выполняются в этой файловой системе асинхронно.
atime	Обновляет время доступа к **inode** каждый раз, когда к файлу обращаются.
auto	Может быть смонтирован командой **mount -a.**
difaults	Используются параметры монтирования по умолчанию: **rw, suid, dev, exec, auto, nouser и async**.
dev	Предоставляет доступ к символьным устройствам, таким как терминалы или консоли, и блочным устройствам, таким как накопители.
exec	Позволяет запускать двоичные файлы (скомпилированные программы) в этой файловой системе.
noatime	Не обновляет время доступа к inode каждый раз, когда к файлу обращаются.
noauvto	Требуется явный монтаж. Это распространенный вариант для приводов компакт-дисков и съемных носителей.
nodev	Файлы устройств в этой файловой системе не читаются и не интерпретируются.
noexec	Двоичные файлы (скомпилированные программы) не могут быть запущены в этой файловой системе.
nosuid	Запрещает setuid и setgid разрешения для этой файловой системы.
nouser	Только root-пользователи могут монтировать указанную файловую систему.
remount	Перемонтирует монтированную файловую систему.
ro	Монтирует файловую систему только для чтения.
rw	Монтирует файловую систему как чтение / запись.
Suid	Разрешает setuid и setgid разрешения для программ в этой файловой системе.
Sync	Все операции ввода-вывода выполняются синхронно в этой файловой системе.
user	Позволяет пользователям без полномочий root монтировать эту файловую систему. По умолчанию это также устанавливает параметры noexec, nosuid и nodev.

Виртуальные файловые системы

В этом разделе описываются некоторые виртуальные файловые системы, используемые **RHEL 7** и перечисленные в **/etc/mtab**. Вот наиболее распространенные:

■ **tmpfs** Эта файловая система виртуальной памяти использует как оперативную память, так и пространство подкачки.

■ **devpts** Эта файловая система относится к псевдо-терминальным устройствам.

■ **sysfs** Эта файловая система предоставляет динамическую информацию о системных устройствах. Исследуйте связанный каталог **/sys**. Вы найдете широкий спектр информации, касающейся устройств и драйверов, подключенных к локальной системе.
■ **proc** Эта файловая система особенно полезна, потому что она предоставляет динамически настраиваемые параметры для изменения поведения ядра. Как навык RHCE, вы можете больше узнать об опциях в файловой системе **proc в главе 12.**
■ **cgroups** Эта файловая система связана с функцией группы управления ядра Linux, которая позволяет устанавливать ограничения на использование системных ресурсов для **процес**

Добавьте свои собственные файловые системы в /etc/fstab

Если вам нужно настроить специальный каталог, иногда имеет смысл установить его на отдельном томе. Разные тома для разных каталогов означают, что файлы на этом томе не могут перегружать критические каталоги, такие как **/boot**. Хотя хорошо следовать стандартному формату файла **/etc/fstab**, это дополнительное усилие. Если потребуется на экзамене Red Hat, это будет в инструкциях, которые вы видите.

Так что в большинстве случаев достаточно установить новый том в **/etc/fstab** с помощью связанный файл устройства, такой как раздел **/dev/vda6**, **UUID** или устройство **LVM**, такое как **/dev/mapper/NewVol-NewLV** или **/dev/NewVol/NewLV**. Убедитесь, что файл устройства отражает созданный вами новый том, предполагаемый каталог монтирования (например, **/special**) и примененный формат файловой системы (например, **xfs**).

Сменные носители и /etc/fstab

Как правило, съемные носители не должны монтироваться автоматически во время процесса загрузки. Это возможно в файле конфигурации **/etc/fstab** с такой опцией, как **noauto**, но, как правило, в RHEL не принято настраивать съемные носители в **/etc/fstab**.

Для чтения съемных носителей, таких как смарт-карты и **CD/DVD, RHEL** частично автоматизирует монтирование таких носителей в среде рабочего стола GNOME. Хотя подробности этого процесса не являются частью руководства по подготовке к экзамену Red Hat, этот процесс основан на файлах конфигурации в каталоге **/usr/lib/udev/rules.d.** Если RHEL обнаружит ваше оборудование, нажмите Places; в появившемся меню выберите запись для съемного носителя. Если загружено несколько параметров съемных носителей, вы можете выбрать носитель для подключения в подменю «Съемный носитель».

Если это по какой-то причине не работает, вы можете напрямую использовать команду **mount**. Например, следующая команда монтирует **CD/DVD** в дисковод:

 # mount -t iso9660 /dev/sr0 /mnt

Ключ -t указывает тип файловой системы (**iso9660**). Файл устройства **/dev/sr0** представляет первый привод **CD/DVD; /mnt** - это каталог, через который вы можете получить доступ к файлам с **CD/DVD** после монтирования. Но **/dev/sr0**? Как кто-то должен помнить это?

К счастью, Linux решает эту проблему несколькими способами. Во-первых, он устанавливает ссылки из более разумно названных файлов, таких как **/dev/cdrom**, что можно подтвердить с помощью команды **ls -l /dev/cdrom**. Во-вторых, он предоставляет команду **blkid**. Попытайся. Если подключен съемный носитель (кроме CD/DVD), вы увидите его в выводе команды, включая связанный файл устройства.

Просто помните, что важно отключить съемные носители, такие как **USB**-ключи, прежде чем извлекать их. В противном случае данные, которые, по вашему мнению, были записаны на диск, могут все еще находиться в неписанном кэше **ОЗУ**. В этом случае вы потеряете эти данные.

Принимая во внимание эти примеры того, как можно смонтировать съемный носитель, вы должны иметь лучшее представление о том, как такой носитель можно настроить в файле конфигурации **/etc/fstab**. Стандартная опция по умолчанию не подходит в большинстве случаев, потому что она монтирует систему в режиме чтения-записи (даже для DVD-дисков только для чтения), пытается монтировать автоматически во время процесса загрузки и ограничивает доступ пользователя с правами администратора. Но это можно изменить с помощью правильных опций. Например, чтобы настроить дисковод для компакт-дисков, который могут монтировать обычные пользователи, вы можете добавить следующую строку в **/etc/fstab**:

/dev/sr0 /cdrom auto ro,noauto,users 0 0

Эта строка устанавливает монтирование в режиме только для чтения, не пытается монтировать его автоматически во время процесса загрузки и поддерживает доступ для обычных пользователей.

По желанию, аналогичные параметры возможны для съемных носителей, таких как USB-ключи, но это может быть более проблематичным с несколькими USB-ключами; например, один может быть обнаружен как **/dev/sdc** один раз, а затем определяется как **/dev/sdd**, если установлен второй USB-ключ. Однако при правильной настройке каждый USB-ключ должен иметь **уникальный UUID**. Есть и другой вариант: вместо использования статического монтирования для съемных устройств, вы можете положиться на автомонтирование, как мы покажем позже в этой главе.

Сетевые файловые системы

Файл **/etc/fstab** можно использовать для автоматизации монтирования из общих каталогов. Интерес представляют две основные службы обмена: **NFS и Samba**. В этом разделе представлен только краткий обзор того, как такие общие каталоги можно настроить в файле **/etc/fstab**; Для получения дополнительной информации см. главы 15 и 16.

В целом, открытые для доступа сетевые каталоги следует считать ненадежными. Люди наступают на линии электропередач, кабели Ethernet и так далее. Если ваша система использует беспроводную сеть, это добавляет еще один уровень ненадежности. Другими словами, настройки в файле **/etc/fstab** должны учитывать это. Поэтому, если есть проблема с сетевым подключением или, возможно, проблема, такая как сбой питания на удаленном сервере **NFS**, вы должны указать в опциях монтирования, как вы хотите, чтобы клиент вел себя.

Соединение с общим каталогом **NFS** основывается на **его имени хоста или IP-адресе**, **а также на полном пути к каталогу на сервере**. Таким образом, чтобы подключиться к удаленному **NFS-серверу** на системном сервере **server1**, который совместно использует каталог **/pub**, вы можете подключить этот общий ресурс с помощью следующей команды (при условии, что каталог **/share** существует):

 # mount -t nfs server1.example.com:/pub /share

Но это монтирование не указывает никаких параметров. Вы можете попробовать следующую запись в **/etc/fstab**:

server1:/pub /share nfs rsize=65536,wsize=65536,hard,udp 0 0

Переменные **rsize и wsize** определяют максимальный размер (в байтах) данных, которые будут считываться и записываться в каждом запросе. Жесткая директива указывает, что клиент будет повторять неудавшиеся запросы **NFS** на неопределенный срок, потенциально блокируя запросы клиентов, пока сервер **NFS** не станет доступным. И

наоборот, мягкая опция приведет к сбою клиента после предопределенного количества повторных передач, но ценой риска целостности данных. **UDP** определяет соединение с использованием протокола пользовательских дейтаграмм (**UDP**). Если соединение с сервером **NFS** версии 4, замените **nfs4 на nfs** в третьем столбце. Обратите внимание, что **NFS версии 4 требует TCP**. Напротив, общие каталоги **Samba** используют другой набор параметров. Следующая строка - это, как правило, все, что нужно для одного и того же каталога и сервера:

//server1/pub /share cifs rw,username=user,password=pass, 0 0

Если вас беспокоит открытое отображение имени пользователя и пароля в файле **/etc/fstab**, который доступен для чтения всем, попробуйте следующую опцию:

//server1/pub /share cifs rw,kredentaials=/etc/secret 0 0

Затем вы можете настроить файл **/etc/secret** как доступный только для пользователя с правами администратора **root**, используя имя пользователя и пароль в следующем формате:

username=user
password=password

ЦЕЛЬ СЕРТИФИКАЦИИ 6.06

Автомантирование

При подключении к сети и переносных носителях могут возникнуть проблемы, если соединения потеряны или носитель удален. В процессе настройки сервера вы можете монтировать каталоги из нескольких удаленных систем. Вам также может потребоваться временный доступ к съемным носителям, таким как **USB-ключи**. В этом может помочь демон автомантирования, также известный как **autofs**. Он может автоматически монтировать определенную файловую систему по мере необходимости. Он может размонтировать файловую систему автоматически через определенный промежуток времени.

Монтирование с помощью авто монтирование

Когда раздел монтируется вручную с помощью команды **mount** или с помощью **/etc/fstab**, он остается монтированным до тех пор, пока вы не отключите его или не выключите систему. Постоянство монтирование может вызвать проблемы. Например, если вы подключили **USB**-ключ, а затем физически удалили его, у Linux, возможно, не было возможности записать файл на диск. Данные будут потеряны. Та же проблема относится к защищенным цифровым картам или другим съемным дискам с возможностью горячей замены.

Другая проблема: смонтированные файловые системы **NFS** могут вызвать проблемы в случае сбоя удаленного компьютера или потери соединения. Системы могут замедляться или даже зависать, так как локальная система ищет смонтированный каталог. Вот где автомонтирование может помочь. Она полагается на демон **autofs** для монтирования настроенных каталогов по мере необходимости на временной основе. В RHEL соответствующие файлы конфигурации**: auto.master, auto.misc, auto.net и auto.smb**, все в каталоге **/etc**. Если вы используете автомонтирование, оставьте каталоги **/misc и /net** свободными. Red Hat настраивает автомонтирования в этих каталогах по умолчанию, и они не будут работать, если там будут храниться локальные файлы или

каталоги. Подразделы будут охватывать каждый из этих файлов. Вы не будете Подразделы будут охватывать каждый из этих файлов.

!!!!
Вы даже не увидите каталоги /misc и/или /net, если не настроите должным образом /etc/auto.master и не запустите демон autofs.
!!!!

Настройки автомонтирования по умолчанию настраиваются в **/etc/sysconfig/autofs**. Настройки по умолчанию включают время ожидания 300 секунд; другими словами, если в течение этого времени на автомонтировании ничего не происходит, ресурс автоматически отключается:
TIMEAUT=300
BROWSE_MODE может позволить вам искать из доступных монтирований. Следующая директива отключает ее по умолчанию:
BROWSE_MODE="no"
Доступно множество дополнительных настроек, как описано в **/etc/sysconfig/autofs.**

/etc/auto.master

Стандартный файл **/etc/auto.master** содержит серию директив с четырьмя незакомментированными строками по умолчанию. Первый ссылается на файл **/etc/auto.misc** как файл конфигурации **для /misc** каталога. Директива **/net -hosts** позволяет вам указать хост для автоматического монтирования сетевого каталога, как указано в **/etc/auto.net**.

/misc /etc/auto.misc
/net -hosts
+dir:/etc/auto.master.d
+auto.master

В любом случае эти директивы указывают на файлы конфигурации для каждого сервиса. Общие каталоги из каждого сервиса автоматически монтируются по требованию в данный каталог (**/misc и /net**).

Вы можете настроить автомонтирование в других каталогах. Один из популярных вариантов - настроить автомонтирование в каталоге **/home**. Таким образом, вы можете настроить домашние каталоги пользователей на удаленных серверах, подключенных по требованию. Пользователи получают доступ к своим домашним каталогам при входе в систему, и в соответствии с директивой **TIMEOUT** в файле **/etc/sysconfig/autofs** все подключенные каталоги автоматически отключаются через 300 секунд после выхода пользователей из системы.

 # /home /etc/auto.home

Это работает только в том случае, если каталог **/home** еще не существует в локальной системе. Поскольку экзамен Red Hat требует настройки нескольких обычных пользователей, в ваших системах должен быть каталог **/home** для обычных пользователей. В этом случае вы можете заменить другой каталог, ведя к строке, например:

/shared /etc/auto.home

Помните, что для любой системы, доступ к которой осуществляется по сети, вам необходимо быть уверенным, что брандмауэр разрешает трафик, связанный с данной службой.

/etc/auto.misc

Red Hat удобно предоставляет стандартные директивы автомонтирования в комментариях в **/etc/auto.misc** файл. Полезно проанализировать этот файл подробно. Мы используем версию этого файла по умолчанию RHEL. Первые четыре строки - это комментарии, которые мы пропускаем. Первая директива

cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

В RHEL эта директива активна по умолчанию, если вы активировали службу **autofs**. Другими словами, если у вас есть компакт-диск в приводе **/dev/cdrom**, вы можете получить доступ к его файлам через автомонтировщик с помощью команды **ls /misc/cd**, даже как обычный пользователь. Автомонтировщик обращается к нему с помощью файловой системы ISO9660. Он смонтирован только для чтения (**ro**), установленные права доступа **ID** пользователя владельца (**nosuid**) и файлы устройств в этой файловой системе не используются (**nodev**).

Ряд других директив закомментирован, готов к использованию. Конечно, вам придется удалить символ комментария **(#)** перед использованием любой из этих строк конфигурации, и вам придется соответствующим образом настроить имена и файлы устройств; например, **/dev/hda1** больше не используется в качестсве файла устройства в последних системах Linux, даже для жестких дисков **PATA**.

Как было предложено в одном из комментариев: «Следующие записи являются примерами для обмана вашего воображения». Первая из этих закомментированных строк позволяет вам настроить точку монтирования **/misc/linux** из общего каталога **NFS, /pub/linux**, из хост **ftp.example.org**:

#linux -ro,soft,intr ftp.example.org:/pub/linux

В следующей строке предполагается, что файловая система хранится в разделе **/dev/hda**1. С помощью этой директивы вы можете автоматически смонтировать файловую систему в **/misc/boot**.

#boot -fstype=ext2 :/dev/hda1

Следующие три строки относятся к дисководу гибких дисков. Не смейся; виртуальные дискеты довольно легко создавать и настраивать в большинстве систем виртуальных машин. Первая директива, установленная на тип «**auto**» файловой системы, ищет в **/etc/filesystems**, чтобы попытаться найти то, что находится на вашей дискете. Следующие две директивы предполагают, что дискета отформатирована в файловой системе **ext2**.

#floppy -fstype=auto :/dev/fd0
#floppy -fstype=ext2 :/dev/fd0
#e2floppy -fstype=ext2 :/dev/fd0

Следующая строка указывает на первый раздел на третьем диске **SCSI**. **Jaz** в начале предполагает, что это подходит для старого привода **Jaze** типа **Iomega**.

#jaz -fstype=ext2 :/dev/sdc1

Наконец, последняя команда основана на более старой системе, где автомонтирование применяется к устаревшему диску **PATA**. Конечно, файл устройства **/dev/hdd** больше не используется, поэтому замените его соответствующим образом. Но съемный в начале предполагает, что это также подходит для съемных жестких дисков.

Конечно, вам, вероятно, придется изменить формат файловой системы на что-то вроде **XFS**. Как предлагалось ранее в этой главе, команда **blkid** может помочь идентифицировать доступные файлы устройств со съемных систем, таких как USB-ключи и портативные накопители.

#removable -fstype=ext2 :/dev/hdd

В общем, вам нужно изменить эти строки для доступного оборудования.

/etc/auto.net

С помощью скрипта конфигурации **/etc/auto.net** вы можете просматривать и читать общие каталоги **NFS**. Он работает с именами хостов или IP-адресами серверов **NFS**. По умолчанию исполняемые разрешения включены для этого файла.

Предполагая, что автомонтирование активно и может подключаться к серверу **NFS** с IP-адресом 192.168.122.1, вы можете просмотреть общие каталоги NFS в этой системе с помощью следующей команды:

 # /etc/auto.net 192.168.122.1
-fstype=nfs,hard,intr,nodev,nosuid /srv/ftp \ 192.168.122.1:/srv/ftp

Эти выходные данные сообщают, что каталог **/srv/ftp** в системе 192.168.122.1 является общим для **NFS**. Основываясь на директивах в **/etc/auto.master**, вы можете получить доступ к этому общему ресурсу (при условии соответствующих настроек брандмауэра и SELinux) с помощью следующей команды:

 # ls /net/192.168.122.1/srv/ftp

/etc/auto.smb

Одна из проблем, связанных с настройкой общего каталога **Samba** или **CIFS**, заключается в том, что он работает, по крайней мере в его стандартной конфигурации, только с общими каталогами. Другими словами, если вы активируете файл **/etc/auto.smb**, он будет работать только с общими каталогами без имени пользователя или пароля.

Если вы примете эти небезопасные условия, можно настроить файл **/etc/auto.smb** так же, как файл **/etc/auto.net**. Во-первых, вам нужно добавить его в файл **/etc/auto.master** аналогичным образом со следующей директивой:

/smb /etc/auto.smb

Затем вам нужно будет специально перезапустить службу автомонтирования с помощью следующей команды:

 # systemctl restart autofs

После этого вы сможете просматривать общие каталоги с помощью следующей команды; замените имя хоста или IP-адрес, если хотите. Конечно, это не сработает, если сервер Samba не активирован в отмеченной системе **server1.example.com** и брандмауэр не настроен на разрешение доступа через соответствующие порты **TCP/IP**.

 # /etc/auto.smb server1.example.com

Активировать автомонтер

После настройки соответствующих файлов вы можете запустить, перезапустить или перезагрузить автомонтирование. Поскольку он управляется демоном **autofs**, вы можете остановить, запустить, перезапустить или перезагрузить эту службу с помощью одной из следующих команд:

 # systemctl stop autofs
 # systemctl start autofs
 # systemctl restart autofs
 # systemctl reload autofs

С помощью команды по умолчанию в файле **/etc/auto.misc** вы теперь сможете автоматически смонтировать компакт-диск в каталог **/misc/cd**, просто получив доступ к настроенному каталогу. Если у вас есть компакт-диск в приводе, должна работать следующая команда:

 # ls /misc/cd

Если вы перейдете в каталог **/misc/cd**, автомонтировщик будет игнорировать любые тайм-ауты. В противном случае **/misc/cd** автоматически отключается в соответствии с тайм-аутом, который согласно директиве **TIMEOUT в /etc/sysconfig/autofs** составляет 300 секунд.

УПРАЖНЕНИЕ 6-3

Настроить автомонтирование

В этом упражнении вы протестируете автомонтировщик. Вам понадобится хотя бы компакт-диск. В идеале у вас также должен быть USB-ключ или защищенная цифровая (SD) карта. Однако сначала необходимо **убедиться**, что демон **autofs работает**, изменить соответствующие файлы конфигурации, а затем **перезапустить autofs**. Затем вы можете проверить автомонтирование в этой лаборатории.

1. В интерфейсе командной строки выполните следующую команду, чтобы убедиться, что демон autofs запущен:
   # systemctl запускает autofs
2. Просмотрите файл конфигурации /etc/auto.master в текстовом редакторе. По умолчанию достаточно активировать параметры конфигурации в **/etc/auto.misc и /etc/auto.net.**
3. Проверьте файл конфигурации /etc/auto.misc в текстовом редакторе. Убедитесь, что он содержит следующую строку (которая уже должна быть там по умолчанию). Сохраните и выйдите из **/etc/auto.misc.**
   cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
4. Теперь перезагрузите демон **autofs**. Поскольку он уже запущен, все, что вам нужно сделать, - это убедиться, что он перечитывает связанные файлы конфигурации.
   # systemctl reload autofs
5. Сервис автомонтирования теперь активен. Вставьте **CD** или **DVD** в соответствующий привод и выполните следующую команду. В случае успеха, он должен отображать содержимое **CD** или **DVD**:
   # ls /misc/cd
6. Запустите команду **ls /misc** немедленно. Вы должны увидеть каталог **CD** в выходных данных.
7. Подождите как минимум пять минут, а затем повторите предыдущую команду. Что ты видишь?

СЦЕНАРИЙ И РЕШЕНИЕ	
Вам необходимо настроить несколько новых разделов для стандартного раздела **Linux**, для пространства подкачки и для логического тома.	Используйте утилиту **fdisk, gdisk** или **parted** для создания разделов, а затем измените их типы разделов с помощью команды **t или set.**
Вы хотите настроить монтирование во время процесса загрузки на основе **UUID**.	Определите **UUID** тома с помощью команды **blkid** и используйте этот **UUID** в файле **/etc/fstab**.
Вам необходимо отформатировать том в тип файловой системы **XFS**.	Отформатируйте целевой том с помощью команды **mkfs.xfs**.
Вы должны отформатировать том с типом файловой системы **ext2, ext3 или ext4**.	Отформатируйте целевой том с помощью команды, такой как **mkfs.ext2, mkfs.ext3 или mkfs.ext4.**
Вы хотите настроить логический том.	Используйте команду **pvcreate** для создания **PV**; используйте команду **vgcreate** для объединения **PV** в **VG**; используйте команду **lvcreate** для создания **LV**; отформатируйте этот **LV** для использования.
Вы хотите добавить новые файловые системы без разрушения других.	Используйте свободное пространство, доступное на существующих или вновь установленных жестких дисках.
Вы хотите расширить пространство, доступное для **LV**, отформатированного с помощью файловой системы **XFS**.	Используйте команду **lvextend** для увеличьте пространство, доступное для **LV**, а затем используйте команду **xfs_growfs**, чтобы соответствующим образом расширить форматированную файловую систему.
Вам необходимо настроить автоматическое монтирование на общую сетевую файловую систему.	Сконфигурируйте файловую систему либо в **/etc/fstab**, либо через автомонтировщик.

РЕЗЮМЕ СЕРТИФИКАЦИИ

Как администратор **Linux**, вы должны знать, как создавать и управлять новой файловой системой объемы. Чтобы создать новую файловую систему, вам нужно знать, как создавать, управлять и форматировать разделы, а также как настроить эти разделы для логических томов.

RHEL 7 также поддерживает настройку **логических томов**. Процесс немного сложный, так как требует настройки раздела как **PV**. Один или несколько **PV** могут быть настроен как **VG**. **Логические тома** могут затем быть настроены из желаемых частей **VG**. Связанными командами являются **pv *, vg * и lv ***; и те, и другие могут быть доступны из **lvm**> Быстро.

Linux поддерживает формат разделов, **RAID**-массивов и логических томов для широкого Разнообразие файловых систем. Хотя по умолчанию XFS, Linux поддерживает форматы и проверки связанные с регулярными и журнальными файловыми системами, связанными с Linux, Microsoft и другие операционные системы.

После настройки разделы и логические тома, зашифрованные или нет, могут быть настраивается в файле **/etc/fstab**. Эта конфигурация читается во время процесса загрузки и может также будет использоваться командой **mount**. При желании съемные файловые системы и общедоступные сетевые каталоги также можно настроить в **/etc/fstab**.

Файл **/etc/fstab** - не единственный вариант настройки монтирования. Вы можете автоматизировать этот процесс для постоянных пользователей с автомонтировщиком. Правильно настроенный, он позволяет пользователям получать доступ к общим сетевым каталогам, съемным носителям и многое другое через пути, определенные в **/etc/auto.master**.

Пару минут тренировки

Вот некоторые из ключевых моментов целей сертификации в главе 6.

Управление хранилищем и разделами

- Утилиты fdisk, gdisk и parted могут помочь вам создавать и удалять разделы.
- fdisk, gdisk и parted можно использовать для настройки разделов для логических томов и RAID-массивы.
- Диски могут использовать традиционную схему разбиения в стиле MBR, которая поддерживает основной, расширенный и логический разделы или схема GPT, которая поддерживает до 128 разделов.

Форматы файловой системы

- Инструменты Linux можно использовать для настройки и форматирования томов различныt файловые системы.
- Примеры стандартных файловых систем включают MS-DOS и ext2.
- Журналироване файловые системы, включают журналы, которые могут восстанавливать метаданные, являются более гибкими; файловая система RHEL 7 по умолчанию - **XFS**.
- RHEL 7 поддерживает множество команд форматирования файловых систем **mkfs.*** А также команды проверок формата файловой системы **fsck*.**

Основные файловые системы и каталоги Linux

- Файлы и файловые системы Linux организованы в каталоги на основе FHS. FHS (англ. Filesystem Hierarchy Standard, «стандарт иерархии файловой системы») — стандарт, унифицирующий местонахождение файлов и каталогов с общим назначением в файловой системе UNIX.)
- Некоторые каталоги Linux хорошо подходят для настройки в отдельных файловых системах.

Управление логическими томами (LVM)

- **LVM** основан на **физических томах, логических томах и группах томов**.
- Вы можете создавать и добавлять системы **LVM** с широким спектром команд, начиная с **pv *, lv * и vg *.**
- Пространство из новых разделов, настроенных как **PV**, может быть выделено для существующего тома группы с командой **vgextend**; они могут быть добавлены в **LV** с помощью **lvcreate** и Команды **lvextend**.
- Дополнительное пространство можно использовать для расширения существующей файловой системы **XFS** команда **xfs_growfs**.

Управление файловой системой

- Стандартные файловые системы монтируются, как определено в **/etc/fstab**.
- Тома файловой системы обычно идентифицируются по их **UUID**; для списка, запустите **blkid** команда.
- Команда **mount** может использовать настройки в **/etc/fstab** или смонтировать файловую систему напрямую.
- Также возможно настроить монтирование **общих сетевых каталогов из NFS и Серверы Samba в /etc/fstab**.

Автомонтирование

- С помощью автомонтирования вы можете настроить автоматическое монтирование съемных носителей и общие сетевые диски ответственный за автомонтирование демон **autofs**(необходимо сначала его установить для работы).
- Ключевыми файлами конфигурации автомонтирования являются **auto.master, auto.misc и auto.net**, в каталог **/etc**.

САМОПРОВЕРКА

Следующие вопросы помогут оценить ваше понимание материалов, представленных в этой главе. Поскольку на экзаменах Red Hat не появляется вопросов с несколькими вариантами ответов, вопросы с несколькими вариантами ответов не отображаются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Получение результатов, а не запоминание пустяков, это то, что рассчитывает на экзамены Red Hat. Может быть более одного правильного ответа на многие из этих вопросов.

Управление хранилищем и разделами

1. Какой параметр команды **fdisk** выводит список настроенных разделов со всех подключенных жестких дисков?

_____

2. Какая команда активирует его после создания раздела подкачки?

_____

Форматы файловой системы

3. Каково основное преимущество журналируемой файловой системы, такой как **XFS**?

_____

4. Какая команда форматирует **/dev/sdb3** в формат файловой системы Red Hat по умолчанию?

_____

Основные файловые системы и каталоги Linux

5. Какая файловая система смонтирована в каталоге, отдельном от корневого каталога верхнего уровня в установка по умолчанию RHEL 7?

_____

6. Назовите три директории чуть ниже / которые не подходят для монтирования отдельно от тома с корневым каталогом верхнего уровня.

_____

Управление логическими томами (LVM)

7. Как только вы создали новый раздел и установите для него тип управления логическими томами, что Команда добавляет это как **PV**?

_____

8. Как только вы добавите больше места в LV, какая команда развернет подчеркивающую **XFS** файловая система для заполнения нового пространства?

_____

Управление файловой системой

9. Чтобы изменить параметры монтирования для локальной файловой системы, какой файл вы бы отредактировали?

_____

10. Что бы вы добавили в файл **/etc/fstab**, чтобы настроить доступ к разделу **/dev/vda6**, смонтированному на каталог **/usr** только для чтения с другими параметрами по умолчанию? Предположим, вы не можете найти **UUID /dev/vda6**. Также примите значение дампа 1 и порядок проверки файловой системы 2.

_____

Автомонтирование

11. Если вы запустили демон **autofs** и хотите прочитать список общих каталогов **NFS** из **server1.example.com** компьютера, какую команду, связанную с автомонтировщиком, вы бы использовали?

_____

12. Назовите три файла конфигурации, связанных с установкой по умолчанию автомонтирования на RHEL 7.

ТЕСТОВЫЕ ОТВЕТЫ

Управление хранилищем и разделами

1. Параметр команды **fdisk**, в котором перечислены сконфигурированные разделы со всех подключенных жестких дисков: **fdisk -l**.
2. После создания раздела подкачки вы можете использовать **mkswap** ***имя устройства*** и **swapon** ***имя устройства*** команды для инициализации и активации тома; просто замените файл устройства, связанный с тома (например, **/dev/sda1** или **/dev/VolGroup00/LogVol03**) в ***имя устройства***.

Форматы файловой системы

3. Основным преимуществом **журналируемой** файловой системы, такой как **XFS**, является более быстрое восстановление данных.
4. Команда, которая форматирует **/dev/sdb3** в формат файловой системы **Red Hat** по умолчанию, имеет вид **mkfs.xfs /dev/sdb3**. Команда **mkfs -t xfs /dev/sdb3** также является приемлемым ответом.

Основные файловые системы и каталоги Linux

5. Файловая система **/boot** монтируется отдельно от **/.**
6. Есть много правильных ответов на этот вопрос; некоторые из каталогов не подходят для монтируется отдельно от корневого каталога **/include /dev, /etc и /root**. (Напротив, несколько каталогов по существу, показывается как заполнители для монтажа, включая **/media и /mnt**.)

Управление логическими томами (LVM)

7. После того, как вы создали новый раздел и установили для него тип файла управления логическими томами, команда, которая добавляет его как **PV - это pvcreate**. Например, если новый раздел - **/dev/sdb2,** команда **pvcreate /dev/sdb2**.
8. Как только вы добавите больше места в LV, команда, которая расширит базовую XFS Файловую систему для заполнения нового пространства - **xfs_growfs**.

Управление файловой системой

9. Чтобы изменить параметры монтирования для локальной файловой системы, отредактируйте файл **/etc/fstab**.
10. Поскольку **UUID** неизвестен, вам необходимо использовать файл устройства для тома (в данном случае **/dev/vda6**). Таким образом, строка для добавления в **/etc/fstab /dev/vda6 /usr xfs defaults, ro 1 2**

Автомонтирование

11. Если вы запустили демон **autofs** и хотите прочитать список общих каталогов **NFS** с компьютера **first.example.com**, команда, связанная с автомонтировщиком, которую вы используете для перечисления этих каталогов Это **/etc/auto.net server1.example.com**.
12. К файлам конфигурации, связанным с установкой автомонтирования по умолчанию, относятся **auto.master, auto.misc, auto.net и auto.smb,** все в каталоге **/etc**, а также **/etc/sysconfig/autofs**.

Лабораторная работа

Во время экзаменов Red Hat задания будут представлены в электронном виде. Таким образом, эта книга также представляет большинство лабораторий в электронном виде. Для получения дополнительной информации см. Раздел «Лабораторные вопросы» в конце главы 6.

Если вы выполняли упражнения во время чтения главы, лучше использовать другую виртуальную машину. Если один из них недоступен, первое, что вы должны сделать, это удалить все логические тома, группы томов и физические тома, ранее созданные в этой главе. Для этого полезны команды **lvremove, vgremove и pvremove**.

Затем удалите все разделы, созданные во время главы. Если бы вы использовали запасные жесткие диски, созданные для виртуальных машин, указанных в главе 2, они были бы в файлах устройств **/dev/vdb и /dev/vdc.**

Лаборатория 1

В этой лабораторной работе предполагается, что у вас есть новый жесткий диск (или хотя бы свободное место на текущем жестком диске, куда вы можете добавить новый раздел). Вы можете смоделировать новый жесткий диск, добавив соответствующие настройки в виртуальную машину **KVM**. В этой лабораторной работе вы создадите один новый раздел с помощью **parted**, отформатируете его в файловую систему **xfs** и сконфигурируете его в каталоге **/test1** в **/etc/fstab**, чтобы новый раздел был правильно смонтирован при следующей загрузке Linux. Вы также создадите второй новый раздел с

помощью **fdisk**, отформатируете его и сконфигурируете как дополнительное пространство **подкачки** в **/etc/fstab**, чтобы это пространство также использовалось при следующей загрузке Linux. Кроме того, используйте **UUID в /etc/fstab**.

Лаборатория 2

В этой лабораторной работе вы настроите отформатированный логический том на основе правильно настроенных разделов. Если вы используете разделы, созданные в Lab 1, не забудьте удалить или хотя бы закомментировать любые связанные настройки в файле **/etc/fstab**.

Создав **VG**, не используйте его все. Например, если вы настроили **VG** на 1000 МБ, настройте 900 МБ как **LV**. Смонтируйте полученный логический том в каталог **/test2** и подтвердите результат с помощью команд **mount и df**. Назовите **VG *volgroup1* и LV *logvol1***.

Установите **LV** в каталог **/test2** в файле **/etc/fstab**, отформатированный в файловой системе **XFS**. Используйте **UUID** для соответствующего устройства логического тома в **/etc/fstab**.

Лаборатория 3

В этой лабораторной работе вы продолжите работу, проделанную в лабораторной работе 2, расширив пространство, доступное для форматированного **LV**, ближе к емкости **VG**. Например, если вы смогли следовать рекомендациям по размеру в Lab 2, используйте соответclотвующие команды, чтобы увеличить пространство, доступное для LV, с 900 МБ до 950 МБ. Не удаляйте содержимое файловой системы во время операции изменения размера.

Только один намек: слишком легко пропустить шаги во время процесса.

Лаборатория 4

Перед началом этой лабораторной работы удалите все существующие тома, созданные на дисках **/dev/vdb** и /**dev/vdc**. Затем установите два **PV** на двух дисках (или виртуальных дисках), таких как **/dev/vdb и /dev/vdc**, соответственно, используя весь размер дисков. Настройте новый **VG** с именем «**vg01**», используя только что созданные **PV**, с размером **PE 2 МБ.**

Настройте новый **LV** с именем **«lv01»,** размер которого должен быть **800** логических экстентов. (Сколько МБ они соответствуют?) Отформатируйте **LV** в файловой системе **ext4** и установите его в каталоге **/test4** в файле **/etc/fstab**. Используйте имя устройства **LV** в **/etc/fstab**.

Лаборатория 5

В этой лабораторной работе вы настроите автомонтирование на автоматическое чтение установленного **CD/DVD**. Хотя службы **NFS** рассматриваются в главе 16, подготовительный курс Red Hat RHCSA действительно предполагает, что вам необходимо знать, как использовать автомонтирование для подключения к общему каталогу **NFS**. Таким образом, шаги, включенные в эту лабораторную работу, предназначены для того, чтобы помочь вам настроить простой общий каталог **NFS** в одной системе с подключениями, разрешенными из второй системы. Если вы настроили системы, описанные в главе 2, первая система будет **server1.example.com**, а вторая система будет **tester1.example.com**.

1. В первой системе создайте резервную копию текущих файлов конфигурации **/etc/auto.master и /etc/auto.net.**

2. Установите файлы конфигурации сервера **NFS** с помощью следующей команды:

 # yum install nfs-utils

3. Если вы работаете с базовой виртуальной системой, настроенной в главе 2, этот NFS и связанные с ним пакеты уже должны быть установлены.
4. Откройте каталог **/tmp**, добавив следующую строку в **/etc/exports**. Будьте осторожны, чтобы не включать лишние пробелы:

/tmp *(ro)

5. Активируйте службу **NFS** и установите правило для межсетевого экрана на основе зон с помощью следующих команд. (Межсетевые экраны описаны в главе 4.)

 # systemctl restart nfs-server
 # firewall-cmd --permanent --add-service=nfs --add-service=rpc-bind --add-service=mountd
 # firewall-cmd --reload

6. Запишите IP-адрес локальной системы, как показано командой **ip addr show**. Если это система server1.example.com, описанная в главе 2, она должна быть **192.168.122.50**; но другой IP-адрес тоже подойдет.
7. Перейдите к другой системе RHEL 7, такой как система **tester1.example.com**, описанная в главе 2. Следующая команда должна подтвердить хорошее соединение с удаленным сервером NFS; замените IP-адрес удаленной системы на **remote_ipaddr**.

 # showmount -e *remote_ipaddr*

8. Теперь вы можете настроить локальный автомонтер как для подключения к **CD/DVD**, так и для общего каталога **NFS**, используя методы, описанные в этой главе.

ОТВЕТЫ Лабораторной работы

Одно из предположений в этих лабораторных работах заключается в том, что там, где указан каталог, такой как **/test1**, вы создаете его перед монтированием на него файла устройства тома или включением его в файл конфигурации ключа, такой как **/etc/fstab**.

В противном случае вы можете столкнуться с неожиданными ошибками.

Лаборатория 1

1. Не должно иметь значения, создаются ли разделы в утилите **fdisk** или **parted**. Если типы разделов были настроены правильно, вы должны увидеть один раздел **Linux** и один раздел **swop Linux** в сконфигурированных жестких дисках в выводе команды **fdisk -l**.
2. Если вы не уверены, какой **UUID** использовать в **/etc/fstab**, запустите команду **blkid**. Если дано разделы были правильно отформатированы (с помощью команд **mkfs.xfs** и **mkswap**), вы см. **UUID** для новых разделов в выводе **blkid**. Вы должны быть в состоянии проверить настройка нового раздела и каталога в **/etc/fstab** с помощью команды **mount -a**. Затем Команда **mount** сама по себе должна подтвердить правильность конфигурации в **/etc/fstab**.
3. Вы должны быть в состоянии подтвердить конфигурацию нового раздела подкачки в выводе команды **cat /proc/swaps**. Вы также должны иметь возможность проверить результат swop, командой связанной с командой **top**, а также в выводе команды **free -h**.

4. Помните, что все изменения должны пережить перезагрузку. Для целей этой лабораторной работе вы можете перезагрузите эту систему, чтобы подтвердить это. Однако перезагрузки требуют времени; если у вас есть несколько задач во время экзамена, вы можете подождать до завершения как можно больше, сначала выполнить, прежде чем перезагрузить систему.

Лаборатория 2

Целью данного обсуждения является то, как вы сможете проверить результаты этой лабораторной работы. Даже если вы сконфигурировали резервные разделы в точности как описано в лабе и точно следовали инструкциям, вполне возможно что ваш логический том не будет в точности занимать 900MB. Некоторые из этих различий происходят от различий между размерами **units** (единиц), которые полагаются на основе 2 или 10. Не паникуйте; эта разница нормальная. Та же оговорка относится и к Лаборатории 3.

Имейте в виду, что логические тома основаны на правильно настроенных разделах, настроенных как PV, собраны в VG, а затем подразделены на LV. Затем LV форматируется и монтируется на соответствующий каталог; для этой лабораторной работы этот каталог - **/test2. UUID** этого отформатированного Затем том можно использовать для настройки этого **LV** в качестве монтирования в файле **/etc/fstab**.

1. Чтобы проверить разделы, подготовленные для логических томов, введите команду **fdisk -l**. Подходящее разделы должны отображаться с меткой «**Linux LVM**». Чтобы проверить конфигурацию **PV**, запустите команда **pvs**. Выходные данные должны сообщать устройства и пространство, выделенное для **PV**.
2. Чтобы проверить конфигурацию **VG**, выполните команду **vgs**. Вывод должен содержать список созданных **VG** во время лаборатории с **PV**, включая доступное пространство.
3. Чтобы проверить конфигурацию **LV**, выполните команду **lvs**. На выходе должен быть указан **LV, VG** откуда он был создан, и количество места, выделенного для этого **LV**.
4. Чтобы проверить UUID вновь отформатированного тома, выполните следующую команду:
   # blkid <device_path>
5. Если файл **/etc/fstab** настроен правильно, вы сможете запустить команду **mount -a**. Затем вы должны увидеть логический том, смонтированный в каталоге **/test2**.
6. Как и в **лаборатории 1**, все изменения должны пережить перезагрузку. В какой-то момент вы захотите перезагрузить локальный Система для проверки успеха или неудачи этой и других лабораторий.

Лаборатория 3

Основываясь на информации из лабораторной работы 2, вы уже должны знать, каков размер текущего **LV** связанная команда **df** должна подтвердить результат; команда **df -m** с выводом в МБ может в этом помочь.

Ключевыми командами в этой лабораторной работе являются **lvextend** и **xfs_growfs**. Хотя есть ряд отличных доступны параметров команды, все, что вам действительно нужно с этой командой, это файл устройства для **LV**. Как с Лабораторная работа 2, результат может быть подтвержден после соответствующей команды **mount и df**. Тем не менее, чтобы убедиться, что в процессе не было потеряно никаких данных, вы можете создать несколько тестовых файлов перед изменением размера **LV** и файловой системы.

Лаборатория 4

Есть несколько шагов, связанных с этой лабораторией:

1. Убедитесь, что все разделы и тома, созданные в предыдущих лабораториях, были размонтированы, исключено из **/etc/fstab** и удалено (с помощью команды **{lv, vg, pv}remove**).
2. Вам не нужно разбивать диски **/dev/vdb** и **/dev/vdc**. Достаточно инициализировать все диски как **PV**, используя команды **pvcreate /dev/vdb** и **pvcreate /dev/vdc.**
3. Выполните команду **vgcreate -s 2M vg01 /dev/vdb /dev/vdc**, чтобы создать **VG**.
4. Создайте **LV** с помощью команды **lvcreate -l 800 -n lv01 vg01.**
5. Отформатируйте файловую систему с помощью команды **mkfs.ext4 /dev/vg01/lv01.**
6. Добавьте правильную запись в **/etc/fstab**.
7. Создайте точку монтирования (**mkdir /test4**).

Если файл **/etc/fstab** настроен правильно, вы сможете запустить команду **mount -a**. Затем вы должны увидеть **/dev/mapper/vg01-lv01**, смонтированный в каталоге **/test4**.

Лаборатория 5

Конфигурирование автомонтирования в общем каталоге **NFS** проще, чем кажется. До убедитесь, что общий каталог **NFS** доступен с удаленного компьютера с команда **showmount -e remote_ipaddr**, где **remote_ipaddr** - это **IP**-адрес удаленной **NFS** сервер. Если это не сработает, возможно, вы пропустили шаг, описанный в лаборатории. Для получения дополнительной информации о NFS, см. Главу 16.

Конечно, есть **CD/DVD**. Если автомонтирование работает, а привод **CD/DVD** находится в соответствующем местоположение, вы должны быть в состоянии прочитать содержимое этого диска с помощью команды **ls /misc/cd**.

Это соответствует конфигурации по умолчанию для файлов **/etc/auto.master и /etc/auto.misc**.

Что касается общего каталога **NFS**, существует два подхода. Вы можете изменить следующие комментарии например директивы конфигурации **NFS**. Конечно, вам нужно как минимум изменить ftp.example.org на имя или **IP**-адрес сервера **NFS** и **/pub/linux** в **/tmp** (или любое другое имя каталога общедоступного ресурса).

linux -ro,soft,intr ftp.example.org:/pub/linux

В качестве альтернативы, вы можете просто напрямую воспользоваться скриптом **/etc/auto.net**. Например, если удаленный **NFS**-сервер находится на **IP**-адресе **192.168.122.50**, выполните следующую команду:

 # /etc/auto.net 192.168.122.50

Вы должны увидеть каталог **/tmp**, общий для доступа. Если это так, вы сможете получить к нему более прямой доступ с помощью следующей команды:

 # ls /net/192.168.122.50/tmp

Если вы действительно хотите изучить автомонтирование, попробуйте изменить вышеупомянутую директиву в файл конфигурации **/etc/auto.misc**. Предполагая, что автомонтирование уже запущено, вы можете убедиться, что **automounter** перечитывает соответствующие файлы конфигурации с помощью команды **systemctl reload autofs**.

Если вы используете ту же первую директиву в вышеупомянутой строке, вы сможете использовать автомонтирование для того, что бы получить доступ к тому же каталогу с помощью команды **ls /misc/linux**.