

## ЛЕКЦИЯ 13. РАЗМЕТКА ДИСКА И МОНТИРОВАНИЕ В LINUX

## 13.1. НАЗНАЧЕНИЕ КАТАЛОГОВ В LINUX

Операционная система Linux имеет четкую структуру расположения директорий и файлов. Поэтому многие люди, начинающие изучать линукс, сталкиваются с трудностью понимания нужности каждой отдельной директории.

В этой лекции приведен список основных директорий файловой системы Linux и их краткое описание. Итак, начнем.

**/** Корневой раздел.

**/bin** В этом каталоге хранятся основные команды операционной системы, такие как команды оболочки и команды файловой системы (ls, cp и т.д.).

**/boot** Здесь хранятся образы ядер Linux и менеджер загрузки: grub, lilo и т.д.

**/dev** Директория содержит файлы устройств, подключенные к операционной системе.

В Linux вообще всё рассматривается, как файл, даже различные устройства, такие как принтеры, жёсткие диски, сканеры и т.д. Для получения доступа к определённому устройству, необходимо чтобы существовал специальный файл. Аналогично устроено большинство UNIX-подобных операционных систем.

**/etc** В этой директории находится основная часть конфигурационных файлов самой операционной системы (настройки сети, список пользователей, групп и т.д.) и различных программ (Apache, Samba и т. д.).

**/home** В этом месте должна размещаться вся личная информация пользователей системы.

Почему должна? Потому что владелец компьютера может хранить свою личную информацию в любом понравившемся ему месте. В целях безопасности самой операционной системы и пользовательских данных хранить последние строго рекомендуется в директории /home. Для получения более высокой надежности операционной системы и сохранности пользовательской информации директорию /home рекомендуется выделить в отдельный раздел жесткого диска.

**/home/username** Это Домашняя папка пользователя username. В ней хранится пользовательская часть конфигурационных файлов (настройки пользовательских программ) и вся личная информация пользователя.

Linux является многопользовательской системой и поэтому каждый пользователь имеет уникальный каталог для персональных файлов. Этот каталог называется Домашней папкой пользователя. Домашняя папка суперпользователя root располагается в корне файловой системы.

Такое разделение пользовательских директорий от файлов операционной системы упрощает резервирование данных и повышает надежность самой операционной системы.

**/lost+found** В lost+found скидываются файлы, на которых не было ссылок ни в одной директории, хотя их inod не были помечены как свободные. Допустим при удалении файла из директории произошёл сбой (выключилось питание), то в системе останется потерянный inod, который вроде как и указывает на корректный файл, но этот

файл не содержится ни в одной директории. В ext2 и других нежурналируемых файловых системах задача восстановления таких файлов ложится на fsck. Он находит inod, на которые нет ссылок, и создаёт на них ссылки в lost+found. После этого пользователь сможет просмотреть файлы и при необходимости переместить им куда надо, вернув нужные имена.

В ext3 и в других журналируемых файловых системах fsck просматривает журнал и видя, что операция завершилась не полностью, откатывает ее. Поэтому потерянных inod в журналируемых файловых системах находится меньше.

**/lib** Эта директория предназначена для хранения системных библиотек, необходимые для работы программ из директорий /bin и /sbin и операционной системы в целом.

**/media** Директория, используемая для автоматического монтирования различных устройств USB-накопителей, CD-ROM и т.д. При вставке в привод CD-ROM диска он автоматически подключается во вложенный каталог этой директории.

**/mnt** Обычно эта директория используется для ручного подключения устройств. В эту директорию командой mount «руками» монтируются различные USB-накопители, CD-ROM и т.д.

**/opt** Директория, в которой обычно размещаются установленные программы, имеющие большой дисковый размер или вспомогательные пакеты (например /opt/openoffice.org).

**/proc** /proc — это директория, к которой примонтирована виртуальная файловая система procfs. Различная информация, которую ядро может сообщить пользователям, находится в "файлах" каталога /proc. Например, в файле /proc/modules находится список загруженных модулей ядра. А в файле /proc/cpuinfo — информация о процессоре компьютера.

**/root** Это домашний каталог или Домашняя папка суперпользователя. Эта директория фактически ничем не отличается от директории обычного пользователя. Данная директория по умолчанию всегда расположена в корневом каталоге файловой системы. Это было сделано для предотвращения проблем с доступом к директории /home. В случае когда к каталогу /home по какой-то причине невозможно получить доступ, в системе всегда есть возможность залогиниться под учетной записью суперпользователя, для решения возникших проблем с доступом.

**/sbin** Основные системные программы для администрирования и настройки системы (iptables, ifconfig и т.д.).

**/srv** Параметры, которые специфичны для окружения системы. Чаще всего данная директория пуста.

**/sys** /sys — это директория, к которой примонтирована виртуальная файловая система sysfs, которая добавляет в пространство пользователя информацию ядра Linux о присутствующих в системе устройствах и драйверах. В версии ядра ниже 2.6 не использовалась.

**/sys/block** /sys/block содержит директории всех блочных устройств, присутствующих в данный момент в системе.

**/sys/bus** В этой директории находится список шин, определенных в ядре Linux (eisa, pci и т.д.).

**/sys/class** Каталог содержит список группированных устройств по классам (printer, scsi-devices и т.д.).

**/tmp** Временное хранилище данных. Аналог папки в ОС Windows — C:/Windows/Temp. Все пользователи имеют права чтения и записи в этом каталоге.

**/usr** В этом каталоге хранятся все установленные пакеты программ, документация, исходный код ядра и система X Window. Все пользователи кроме суперпользователя root имеют доступ только для чтения. Может быть смонтирована по сети и может быть общей для нескольких машин.

**/usr/bin** Директория дополнительных программ для всех учетных записей.

**/usr/games** Каталог для размещения доступных компьютерных игр в системе.

**/usr/include** Заголовочные файлы C++.

**/usr/lib** Системные библиотеки для программ, расположенных в директории /usr.

**/usr/local** По стандарту /usr должен быть общим для нескольких компьютеров и смонтирован по сети, а /usr/local должен содержать установленные пакеты программы только на локальной машине (к примеру, /usr - бюджет семьи, а /usr/local - личный кошелек каждого). Но чаще всего директория /usr/local используется для установки программ, которые не предназначены для конкретного дистрибутива (к примеру для пакетного дистрибутива Ubuntu в /usr находятся "родные" установленные пакеты, а /usr/local находятся собранные пакеты из исходников).

**/usr/sbin** Дополнительные системные программы.

**/usr/share** Общие данные установленных программ.

**/usr/share/icons** В каталоге находятся все иконки системы.

**/usr/share/doc** Директория, в которой обычно находится справочная документация по установленным программам.

**/usr/src** Исходные кода (например, здесь располагаются исходные кода ядра).

**/var** Это каталог для часто меняющихся данных. Здесь находятся журналы операционной системы, системные log-файлы, cache-файлы и т. д.

**/var/cache** В этом месте хранятся все кэши различных программ.

**/var/games** В этом месте расположены файлы с игровыми достижениями.

**/var/lib** Постоянные данные, изменяемые программами в процессе работы (например, базы данных, метаданные пакетного менеджера и др.).

**/var/lock** Здесь лежат lock-файлы, указывающие на занятость некоторого ресурса.

**/var/log** В этом месте хранятся все log-файлы.

**/var/spool** Задачи, ожидающие обработки (например, очереди печати, непрочитанные или не отправленные письма, задачи cron и т. д.).

**/var/www** В этом месте размещаются Web-страницы для сервера Apache.

**Раздел <swap>** Раздел виртуальной памяти для увеличения скорости обмена с оперативной памятью.

## 13.2. РАЗМЕТКА ДИСКА LINUX

Для того, что бы установить Linux Mint с нуля, прежде всего, понадобится провести разметку жёсткого диска, а для этого, необходимо определиться, что мы имеем и куда будем устанавливать. В принципе, у нас может быть два варианта установки, а именно: установка на компьютер, на котором кроме самой *Mint* больше ничего не будет и на компьютер, на котором уже имеется какой-либо раздел или разделы, занятые другой операционной системой (*Windows*) или просто разделом, предназначенном для хранения нужных и важных файлов.

Первый вариант, когда весь жёсткий диск компьютера будет полностью выделен под *Mint*, особых сложностей при установке, вызвать не должен. Во время инсталляции, инсталлятор спросит Вас о том, использовать весь диск или применить другие варианты? Выбираете в данном случае, использование всего диска и дальше по ходу, автоматическую разметку. Все, больше от Вас ничего особенного не потребуется, и вся процедура пройдет самостоятельно.

Второй вариант, не то, что бы сложный, но займет несколько больше различных действий. Предположим, что у Вас на компьютере уже имеется установленная *Windows* и Вы не хотите пока с ней расставаться. Или же, другой вариант, у Вас нулевой компьютер и Вы решили установить и *Windows* и *Mint*. В обоих случаях нам нужно будет провести определённые действия с жёстким диском. Единственное, что нужно знать, если Вы хотите установить сразу две системы, так это то, что *Windows* нужно устанавливать первым, а не наоборот, проще будет, хотя и наоборот тоже можно, только потом придется дополнительно напрягаться, что бы всё получилось так, как если бы Вы устанавливали *Windows* первым. Поэтому, лучше сразу выбрать правильную последовательность.

Что бы выполнить необходимую разметку и форматирование диска используют специальные программы, предназначенные для этих целей. На сегодняшний день, таких программ существует предостаточное количество, на любой вкус, как платные, так и бесплатные.

Одной из таких бесплатных утилит является *GParted Live CD (Gnome Partition Editor)*. Скачивать нужно только стабильные версии (*stable*). После закачки, записываем программу на CD-диск и получаем загрузочный диск с *GParted*. Вставив такой диск в лоток CD/DVD-привода и перезагрузив компьютер, мы получаем возможность производить различные действия с нашим жёстким диском компьютера при помощи интерфейса утилиты *GParted*. Кроме *GParted*, достаточно популярными в этом плане могут оказаться и *Acronis disk director*, а также, *Paragon* (Домашний эксперт).

Обратите внимание на то, что название дисков и разделов в *Linux*-среде, несколько отличается от обозначений в среде *Windows*. Если в *Windows* это: *C*; *D*; *E* и т.д., то

в *Linux* это: *sda1*; *sda2*; *sda3* и т.д. Следует знать также, что устанавливая *Mint*, создать необходимые разделы можно и в процессе инсталляции при помощи самого инсталлятора.

Теперь, непосредственно о разбивке и форматировании. Когда мы устанавливаем *Windows*, то просто создаем раздел или выбираем весь диск целиком, на который он будет установлен и форматируем его в *NTFS*.

Вот классическая схема для создания необходимых разделов при установке *Linux*:

1. *swap* (файл подкачки)
2. / (корневой)
3. */boot* (загрузочный)
4. */var*
5. */usr*
6. */tmp*
7. */home*

Но все это касается общих понятий, а для обычного домашнего компьютера, вполне достаточно воспользоваться следующей схемой:

1. *swap* (файл подкачки)
2. / (корневой)
3. */home*

Кроме этого, как утверждают многие, раздел "*swap*" на домашнем компьютере необязателен. Во всяком случае, если Вы его создадите, то хуже от этого не будет, это уж точно.

Еще несколько слов о разделе */home*. Он предназначен для хранения различной информации пользователя. Скаченные фильмы, фотографии, какие-нибудь текстовые файлы и пр., все это храниться на этом разделе.

Создают этот раздел с той целью, что если вдруг произойдет внезапный сбой системы, то при последующей установке системы или ее восстановлении, этот раздел с Вашими важными личными файлами, остался не тронутым и был бы легко доступен после переустановки или восстановления. Это достаточно предусмотрительно, но не является единственным вариантом разметки. Вот еще более простая схема:

1. *swap* (файл подкачки)
2. / (корневой)

Но в этом случае, нужно уделить особое внимание своевременному архивированию системы на случай ее сбоя и возможного восстановления. Каким из вариантов воспользоваться, выбирать Вам, но чисто для дома, первый вариант не актуален. Второй вариант, это классика, третий не стандартный.

Конкретный пример применения третьего варианта, может быть использован, когда для архивирования и восстановления системы, Вы используете сторонние утилиты, одной из которых является бесплатная программа *Clonezilla*. Очень хорошая программа, хотя и имеет весьма примитивный интерфейс, тем не менее, в отличие от своих платных собратьев, таких как *Paragon Backup and Recovery* или *Acronis True Image Home* и пр. достаточно корректно работает с системой *Linux*.

Особенность этой утилиты в том, что независимо от того, на сколько и на какие разделы Вы разметите свой диск, будь-то первый, второй или третий варианты, она всё-равно, все разделы для *Linux* определяет в виде одного.

Допустим, у Вас установлена *Windows (sda1)* и *Linux* по схеме номер один(*sda2*, *sda3*, *sda4*, *sda5*, *sda6*, *sda7*, *sda8*), в общей сложности восемь разделов. При работе

с *Clonezilla*, она покажет вам всего два: *Windows (NTFS) sda1* и *Linux (ext4) sda2*. В *sda2* будут входить все разделы, которые Вы создавали для установки *Linux*, кроме *swap*, т.к. *swap* архивировать не нужно.

Ну, вот такая утилита, по другому она не "видит", но с другой стороны, это удобно, все разделы в виде одного, заархивировал и восстановил, все просто и удобно. К чему столько слов? Третий вариант разметки, как раз подходит для случая, если Вы вдруг решите пользоваться *Clonezilla*, как сторонней программой для выполнения архивирования и восстановления системы.

Ну и в заключении, давайте возьмём и проведём разметку диска на каком-нибудь примере. Допустим мы имеем новый компьютер с 4.0 ГБ оперативной памяти и 500 ГБ объёмом жёсткого диска, а также загрузочный CD/DVD с *Linux Min*, загрузочный CD/DVD с *Windows* и загрузочный с *GParted Live CD*. Устанавливать будем *Windows* и *Mint* одновременно. Вставляем *GParted* в лоток и перезагружаем компьютер, после запуска программы начинаем разметку.

#### Создаём раздел для *Windows*:

- Размер 50 ГБ (взято примерно), система *NTFS*.

Создаём разделы для *Linux* по второй схеме (в качестве примера):

- Раздел *swap*, размер 8.0 ГБ (две оперативных памяти компьютера  $4 \times 2 = 8$ )
- Раздел / (корневой) 25-30 ГБ, система *ext4* (обычно вполне хватает 7-10 ГБ, но с нашим диском в 500 ГБ не будем жадничать).
- Раздел */home* (всё оставшиеся свободное место), система *ext4*.

Таким образом у нас получится что-то вроде:

```
/dev/sda1 NTFS Windows 50.0 ГБ
/dev/sda2 swap      8.00 ГБ
/dev/sda3  /        ext4   30.0 ГБ
/dev/sda4 home     ext4   412.0 ГБ
```

Все, после этого можно в правильной очередности установить системы, сначала *Windows*, потом *Linux*. Если применяете третий вариант разметки, то оставшееся свободное место выделяете под раздел / (корневой).

### 13.3. МОНТИРОВАНИЕ РАЗДЕЛОВ И ДИСКОВ В LINUX

Рано или поздно пользователи *Linux* сталкиваются с таким понятием как монтирование разделов и дисков в *Linux*. Многие, особенно бывшие пользователи *Windows*, испытывают трудности с монтированием разделов, хотя если задуматься, то всё очень просто и логично.

Ниже приводятся описание процесса монтирования в UNIX системах и разбор наиболее типичных случаев. И даже если вы окружены графическим интерфейсом, знание консольных команд может очень пригодиться.

Если ядро *Linux* опознало ваше устройство-носитель данных, то оно должно предоставить какой-то внешний интерфейс пользователю для работы с устройством. Этим интерфейсом является создание файлов-устройств в каталоге */dev*

*Пример:* устройствам, подключённым к IDE, будут соответствовать файлы-устройства */dev/hda*, */dev/hdb* и так далее.

Устройствам типа SCSI, а также близкие им по духу SATA-устройства и USB-флешки, будут иметь файлы-устройства **/dev/sda**, **/dev/sdb** и тому подобное.

Если на диске есть разделы, то цифра в имени файла-устройства будет соответствовать номеру раздела.

*Пример:* если на USB-флешке есть два раздела, то первый будет называться **/dev/sda1**, а второй **/dev/sda2**

Монтирование разделов = объяснение системе, как добраться до ваших данных и сделать их доступными для использования. Системе нужно объяснить три простые вещи:

1. какая файловая система на разделе;
2. какой файл-устройство вам нужно;
3. куда его подключить для просмотра = точка монтирования;

Каталог, в котором вы будете просматривать содержимое ваших разделов, называется точкой монтирования (mount point). Поэтому нужно объяснить системе - командой или через графический интерфейс - что вы хотите смонтировать, куда и что за файловая система на этом разделе. В переводе на язык UNIX, это звучит так:

**mount -t vfat /dev/hda3 /mnt/storage**

Если нужно часто монтировать одни и те же разделы, занесите их в */etc/fstab* - это общесистемный конфиг, в котором указаны все необходимые разделы для монтирования. Редактировать этот файл может не абы кто, а только root. Так что разживитесь рутовыми полномочиями и редактируйте */etc/fstab*

Файл */etc/fstab* состоит из колонок, разделителями между которыми являются символы табуляции (клавиша TAB). Вот пример содержимого файла */etc/fstab*:

**# /etc/fstab: static file system information.**

**#**

**#**

**/dev/hda2 / reiserfs notail,noatime 0 1**

**/dev/hdb /mnt/cdrom iso9660 ro,user,noauto 0 0**

**/dev/sda1 /mnt/flash vfat iocharset=koi8-r,codepage=866,rw,user,auto 0 0**

Формат колонок такой:

- **Монтируемое устройство** (файл-устройство, он же раздел, который вы хотите примонтировать)
- **Точка монтирования** (в какую директорию в файловой системе монтировать)
- **Тип монтируемой файловой системы**
- **Опции монтирования** (кодировка языка, необходимость монтирования при старте)
- **Dump** (обычно 0)
- **Pass** (обычно 0)

Важно, чтобы вы вставляли Tab после каждой колонки при редактировании */etc/fstab*

Ниже приводятся уточнения для каждого случая файловой системы, но *идея монтирования разделов в Linux одна и та же.*

```
mount -t vfat /dev/hda1 /home/vasya/tempdir
```

прикрутить тип FAT что прикрутить куда это прикрутить

Пишем буквально то, чего хотим: от имени суперпользователя, прикрутить логическую структуру с файловой системой типа FAT, находящуюся в *файле* **/dev/hda1**, к каталогу **/home/vasya/tempdir**

После всех операций раздел следует размонтировать, чтобы данные на него записались из памяти - точно так же, как при работе с флешками в Windows. Для этого следует дать команду:

```
# umount /mnt/XX
```

или

```
$ sudo umount /mnt/XX
```

где **/mnt/XX** - точка монтирования. Если ни одно приложение не работает с разделом, то раздел размонтируется, отключится и данные на него полностью запишутся. После этого устройство можно безопасно извлечь.

Чтобы посмотреть, что можно смонтировать, и увидеть информацию о разделах, можно воспользоваться командой (запуск от **root**):

```
# fdisk -l
```

Увидите что-то вроде:

```
Disk /dev/hda: 60.0 GB, 60011642880 bytes
255 heads, 63 sectors/track, 7296 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Device Boot Start End Blocks Id System
```

```
/dev/hda1 1 36 289138+ 82 Linux swap / Solaris
/dev/hda2 * 37 7296 58315950 83 Linux
```

### 13.3.1. Предварительные действия перед монтированием

Прежде, чем монтировать разделы, нужно создать точки монтирования. Точки монтирования следует создавать в каталогах **/mnt** или **/media**. **Пример:** Для того, чтобы создать точку монтирования с названием **/mnt/win**, в консоли даём команду:

```
$ sudo mkdir /mnt/win
```

или

```
# mkdir /mnt/win
```

и даём каталогу права доступа для обычных пользователей:

```
$ sudo chmod 777 /mnt/win
```



или

```
# chmod 777 /mnt/win
```

Это позволит считывать и записывать данные не только суперпользователю-руту, но и обычному пользователю. Даже если в системе вы - единственный пользователь. Свою точку монтирования в директории /mnt нужно создать для *каждого* раздела. Не бойтесь, если вы не знаете точно, как теперь называются ваши виндовые диски C:\ или D:\ и монтируйте разделы с указанием типа файловой системы fat или ntfs - если ошибётесь, раздел не примонтируется и вы увидите ошибку вроде

```
mount: wrong fs type, bad option, bad superblock on /dev/sda2,  
missing codepage or other error  
In some cases useful info is found in syslog - try  
dmesg | tail or so
```

Не боимся - просто нам команда mount сказала, что примонтировать у неё не получилось.

### 13.3.2. Как примонтировать раздел с файловой системой FAT в Linux

Самое главное здесь - кодировка. Microsoft изобрела свою собственную кодировку CP866 для MS DOS, CP1251 для Windows и только в последнее время дошла до UTF8.

#### Однократное монтирование FAT

Временное, однократное монтирование, например, флешки, выглядит в командной строке так:

```
$ sudo mount -t vfat /dev/XX /mnt/YY -o iocharset=utf8,codepage=866,rw
```

или

```
# mount -t vfat /dev/XX /mnt/YY -o iocharset=utf8,codepage=866
```

*Пример:* монтирование типичной флешки выглядит так

```
$ sudo mount -t vfat /dev/sda1 /mnt/flash -o iocharset=utf8,codepage=866,rw
```

То есть монтируем раздел /dev/sda1 типа vfat в точку монтирования /mnt/flash

#### Постоянное монтирование FAT

Если нужно постоянно обращаться к этим разделам, правим /etc/fstab добавляя строку:

```
/dev/XX /mnt/YY vfat iocharset=utf8,codepage=866,user,rw 0 0
```

*Пример:* та же типичная флешка в /etc/fstab смотрится так

```
/dev/sda1 /mnt/flash vfat iocharset=utf8,codepage=866,user,rw 0 0
```

То есть монтируем раздел */dev/sda1* типа *vfat* в точку монтирования */mnt/flash*. После этого раздел FAT в Linux будет доступен. Отмонтирование по команде `umount /mnt/flash`. Как правильно говорят в комментариях, **iocharset** совпадает с текущей локалью - например, `iocharset=utf8`.

### 13.3.3. Как примонтировать раздел с файловой системой NTFS

Монтировать и записывать на NTFS в Linux возможно благодаря усилиям разработчиков, создавших драйвер **NTFS-3g**. Этот драйвер можно установить командой:

```
aptitude install ntfs-3g
```

или поискать пакет **ntfs-3g** в графической оболочке установки программ вашего любимого дистрибутива.

#### Однократное монтирование NTFS

Временное, однократное монтирование, например, раздела с данными Windows, выглядит так:

```
$ sudo mount -t ntfs-3g /dev/XX /mnt/YY -o umask=0,nls=utf8
```

или

```
# mount -t ntfs-3g /dev/XX /mnt/YY -o umask=0,nls=utf8
```

*Пример:* монтирование раздела Windows с NTFS разделом выглядит так

```
$ sudo mount -t ntfs-3g /dev/sda1 /mnt/flash -o umask=0,nls=utf8
```

То есть монтируем раздел */dev/sda1* типа *ntfs-3g* в точку монтирования */mnt/flash*

#### Постоянное монтирование NTFS

Если нужно постоянно обращаться к этим разделам, правим */etc/fstab* добавляя строку:

```
/dev/XX /mnt/YY ntfs-3g umask=0,nls=utf8,user,auto,rw 0 0
```

*Пример: та же типичная флешка в /etc/fstab смотрится так*

```
/dev/sda1 /mnt/windows ntfs-3g umask=0,nls=utf8,exec,user,auto,rw 0 0
```

Иногда в */etc/fstab* это уже есть с настройками вида:

```
/dev/sda1 /mnt/windows ntfs-3g umask=0,nls=ru-RU.UTF-8 0 0
```

При такой записи всё нормально, кроме одного - файлы и каталоги с русскими именами не отображались, как-будто их не было вовсе. Чтобы это исправить, можно попробовать изменить на:

```
/dev/sda1 /mnt/windows ntfs-3g exec,user,auto,rw 0 0
```

То есть монтируем раздел `/dev/sda1` типа `ntfs-3g` в точку монтирования `/mnt/windows` После этого раздел NTFS в Linux будет доступен.

Отмонтирование по команде **umount** `/mnt/windows`

#### 13.3.4. Как примонтировать CD/DVD диск

Всё проще - скорее всего, это уже сделано в вашем дистрибутиве. Тем не менее, здесь это для примера.

##### Однократное монтирование CD/DVD

Временное, однократное монтирование, например, DVD-диска с данными в Linux, выглядит так:

```
$ sudo mount -t iso9660 /dev/XX /mnt/YY
```

или

```
# mount -t iso9660 /dev/XX /mnt/YY
```

*Пример:* монтирование DVD-диска с данными выглядит так

```
$ sudo mount -t iso9660 /dev/cdrom /mnt/dvd
```

То есть монтируем раздел `/dev/cdrom` типа `iso9660` в точку монтирования `/mnt/dvd`

##### Постоянное монтирование CD/DVD

Если нужно постоянно обращаться к этим разделам, правим `/etc/fstab` добавляя строку:

```
/dev/XX /mnt/YY iso9660 user,ro 0 0
```

*Пример:* оптический диск в `/etc/fstab` смотрится так

```
/dev/cdrom /mnt/cdrom iso9660 user,ro 0 0
```

То есть монтируем раздел `/dev/cdrom` типа `iso9660` в точку монтирования `/mnt/cdrom` После этого оптический диск в Linux будет доступен.

Отмонтирование по команде **umount** `/mnt/cdrom`

### 13.3.5. Как примонтировать ISO-файл или другой файл

В Linux и вообще в UNIX монтирование ISO-образа ничем для системы не отличается от монтирования CD/DVD-диска.

Нужно только передать один параметр *-o loop* чтобы сообщить системе, что монтировать мы собираемся на петлевое устройство (*loopback*).

Для монтирования ISO-образа в командной строке даём команду:

```
$ sudo mount -t iso9660 file.iso /mnt/YY -o loop
```

или

```
# mount -t iso9660 file.iso /mnt/YY -o loop
```

*Пример:* монтирование DVD-диска с данными выглядит так

```
$ sudo mount -t iso9660 file.iso /mnt/dvd -o loop
```

То есть монтируем файл *file.iso* типа *iso9660* в точку монтирования */mnt/dvd* на петлевое устройство *loop*

Точно так же можно монтировать любой CD/DVD образ, а также файл-образы сжатых\зашифрованных\экзотических файловых систем типа SquashFS. Отмонтирование по команде `umount /mnt/dvd`

### 13.3.6. Как примонтировать раздел с файловой системой

Монтирование нормальных файловых систем проблем не создаёт. Приведем пример только с **ext3**, остальные файловые системы в Linux монтируются точно так же, только вместо **ext3** подставьте буквально: **ext2**, **reiserfs**, **xfs**, **jfs** в зависимости от вашей файловой системы.

#### Однократное монтирование

Временное, однократное монтирование, например, флешки, выглядит в командной строке так:

```
$ sudo mount -t ext3 /dev/XX /mnt/YY
```

или

```
# mount -t ext3 /dev/XX /mnt/YY
```

*Пример:* монтирование раздела с данными выглядит так

```
$ sudo mount -t ext3 /dev/sda1 /mnt/data
```

То есть монтируем раздел */dev/sda1* типа *ext3* в точку монтирования */mnt/data*

## Постоянное монтирование

Если нужно постоянно обращаться к этим разделам, правим `/etc/fstab` добавляя строку:

```
/dev/XX /mnt/YY ext3 user,auto,rw 0 0
```

*Пример: та же типичная флешка в `/etc/fstab` смотрится так*

```
/dev/sda1 /mnt/data ext3 user,auto,rw 0 0
```

То есть монтируем раздел `/dev/sda1` типа `ext3` в точку монтирования `/mnt/data`. После этого раздел EXT3 в Linux будет доступен. Отмонтирование по команде `umount /mnt/data`

## 13.3.7. Как узнать тип файловой системы на диске

Для этого есть утилита `file`, которая может определить тип файловой системы, используемый файлом. Так как в UNIX/Linux/\*BSD исповедуется принцип "всё есть файл", то раздел с файловой системой это тоже файл. Так что команда **file** выдаст нам информацию о файловой системе дисковых разделов - надо только указать программе воспринимать их как *специальные файлы*.

Используем опцию **-s** (**special-files**) для выдачи информации о типе файловой системы блочного устройства. Пример:

```
# file -s /dev/sda1
```

*или*

```
$ sudo file -s /dev/sda1
```

получаем в ответ:

```
/dev/sda1: ReiserFS V3.6 block size 4096 (mounted or unclean) num blocks 17920496 r5 hash
```

То есть на разделе `/dev/sda1` имеем файловую **ReiserFS V3.6**.

Или, например:

```
# file -s /dev/sda1
```

выдаст в ответ:

```
/dev/sda1: x86 boot sector, code offset 0x58, OEM-ID "MSWIN4.1", sectors/cluster 64, reserved sectors 126, Media descriptor 0xf8, heads 255, hidden sectors 63, sectors 284237982 (volumes > 32 MB) , FAT (32 bit), sectors/FAT 34693, reserved3 0x800000, serial number 0x287d1dfa, unlabeled
```

Это поможет в определении того, какая файловая система (и наметит на то, какие данные там лежат) на данном разделе.

### 13.3.8. Что делать, если устройство в Linux не хочет размонтироваться

Вы пытаетесь размонтировать раздел с данными (флешка, CD/DVD диск) и получаете сообщение с ошибкой - устройство занято. Например, при выполнении команды `umount /mnt/windows` получаете сообщение вида:

```
umount /mnt/windows
```

```
umount: /mnt/windows: device is busy
```

Ну и что делать, если раздел в Linux не размонтируется!? Очень просто - посмотреть, какой процесс держит раздел или файл и не даёт отмонтировать его. Далее либо закрыть это приложение, либо придушить его с помощью команды **kill**. Даём команду:

```
lsof | grep /mnt/windows
```

Вертикальная черта означает передачу вывода команды `lsof` программе `grep` и Фильтрации по слову `/mnt/windows`

После этого вы получите ответ вида:

```
$ lsof | grep /mnt/windows
```

```
lsof: WARNING: can't stat() reiserfs file system /dev/.static/dev
```

```
Output information may be incomplete.
```

```
mc 14134 beast cwd DIR 8,1 4096 1 /mnt/windows
```

Стало быть, ваш раздел держит MC (Midnight Commander), который вы, скорее всего, открыли на другом рабочем столе и просто забыли о нём. Найдите это приложение, закройте его и отмонтируйте раздел снова - теперь проблема должна быть решена.