

Монтирование файловой системы

Монтирование — это процесс, который включает в себя этапы определения файловой системы операционной системой, проверки целостности файловой системы, инициализации нужного драйвера и регистрации файловой системы в общем пространстве имён.

При помощи команды `df -h` посмотрим, какие файловые системы смонтированы у нас сейчас:

```
df -h
```

Мы видим нашу директорию `/dev`. Колонка справа — точки монтирования. То есть наша точка назначения, директория, при помощи которой мы получаем доступ к подключённой файловой системе.

Точка монтирования — `run`, в которой содержится служебная информация о работающих процессах. Файловая система здесь называется `tmpfs`. Это означает, что файлы хранятся не на жёстком диске, а в оперативной памяти и исчезают после перезагрузки.

В данном случае нас это полностью устраивает, так как работающие процессы после выключения машины тоже перестают существовать.

Заглянем в эту директорию:

```
cat /run/nginx.pid
```

Здесь мы видим идентификатор процесса `nginx`. Он доступен в этом файле для других процессов, которым может быть необходимо получить доступ к процессу `nginx`.

Далее идёт файловая система. Мы видим, что `/dev/sda2` подключён к точке монтирования `/` — это корень нашей файловой системы. Точка, с которой начинается иерархия всех директорий в нашей операционной системе.

Смонтируем файловую систему. У меня для этого есть скачанный образ диска `ubuntu`; как мы помним, `iso` — образ диска, это тоже специальная файловая система.

Первым делом создадим точку монтирования. У нас есть директория `/media`, куда обычно монтируются подключаемые устройства, создадим там папку `ubuntu_iso`:

```
sudo mkdir /media/ubuntu_iso
```

Для монтирования применяется команда `mount`. Используем мы её следующим образом: сначала указываем, что хотим примонтировать (наш `iso`-образ), затем куда мы его монтируем (точку монтирования), затем через ключ `-o` укажем тип `loop`, потому что нам нужно псевдоустройство для доступа к нашему образу.

```
sudo mount Downloads/ubuntu*.iso /media/ubuntu_iso -o loop
```

Давайте посмотрим, что получилось:

```
ls /media/ubuntu_iso
```

Так выглядит изнутри установочный диск с Linux. Можем открыть файл с хеш-суммами; чтобы убедиться, что данные на диске не побились, мы можем проверить хеш-сумму каждого каталога на диске.

```
df -h
```

В выводе команды **df** мы видим новую точку монтирования и устройство, при помощи которого мы подключили наш диск.

Команда **mount** без параметров позволяет нам просмотреть расширенную информацию о смонтированном устройстве.

Давайте посмотрим информацию о нашей точке монтирования:

```
mount | grep ubuntu_iso
```

Для того чтобы отключить образ, мы можем использовать команду **umount**:

```
sudo umount /media/ubuntu_iso  
mount | grep ubuntu_iso
```

Образ успешно отмонтировался.

Псевдоустройства

Мы знаем псевдоустройства, такие как `/dev/random` или `/dev/zero`. Чаще всего они используются программой `dd`. `dd` — очень старая программа. Она предназначена для побайтового копирования. Если команда `cp` может скопировать весь файл, то команда `dd` может, например, скопировать первые 100 байт файла.

Синтаксис команды довольно прост:

`dd if=источник of=файл назначения параметры`.

Очень важно при использовании программы `dd` не перепутать местами файлы или устройства, которые фигурируют в `if` и `of`.

С помощью этой команды можно просто вывести текстовый файл на стандартный вывод:

```
dd if=pale_fire.txt
```

Создадим образ диска. Попробуем это сделать для раздела `sda1`:

```
dd if=/dev/sda1 of=sda1.img bs=4096
```

`bs` — это размер блока, который будет читаться за один раз.

Мы можем использовать команду `dd`, чтобы скопировать всё содержимое с одного диска на другой. Для этого в `if` и `of` нужно указать соответствующие устройства.

Сразу предупрежу, что это будет не самым быстрым способом, так как в таком случае будет копироваться каждый байт. Если на диске не очень много данных — это не очень эффективно.

Далее dd можно использовать для того, чтобы безопасно удалить данные с флешки (значительна вероятность восстановления данных после удаления) или с внешнего жёсткого диска. Для этого нужно просто «заполнить» устройство нулями с использованием псевдоустройства /dev/zero:

```
dd if=/dev/zero of=/dev/DEVICE
```

Можем сгенерировать файл со случайным содержимым фиксированной величины. Это может пригодиться, например, для тестирования скорости диска или скорости соединения.

```
dd if=/dev/urandom of=randomfile bs=1M count=10  
du -sh randomfile
```

Мы видим, что у нас получился файл размером 10 Мб.

Можно создать файл и при помощи /dev/zero.

Итак, команда dd помогает нам в работе как с псевдоустройствами, так и с вполне настоящими.

Для работы с файловой системой подключённое устройство необходимо примонтировать, то есть включить в общее пространство имён файлов.