mount (mount file systems)

This command is used to mount file systems, e.g. disk partitions, in the directory tree. Let me briefly explain what mounting is. If you're familiar with Microsoft Windows, you know that every disk partition has its own drive letter in that OS (such as C:\ or D:\) and thus its own directory tree. Linux is much more flexible, allowing you to freely choose a directory for each partition. Choosing a directory under which a partition is accessible is called mounting.

In Linux all partitions are accessible through the main directory tree. For example, you could mount one partition as /my/games and another one as /my/movies. Whenever you enter one of these directories using →*cd*, you change to a different partition without even noticing it. You can mount your partitions wherever you want. However, there is a convention to mount user partitions in /mnt or /media, depending on your Linux distribution.

To see which partitions are mounted on your system, run *mount* without arguments:

mount

Here is some example output from my system:

sysfs on /sys type sysfs (rw,nosuid,nodev,...)
proc on /proc type proc (rw,nosuid,nodev,...)
/dev/sda6 on /home type ext4 (rw,relatime,...)
/dev/sdb1 on /media/pendrive type vfat (...)

The virtual file systems sysfs and proc, which provide information on the system's state and configuration, are mounted as /sys and /proc respectively. They have system-internal purposes and won't concern us any further. The partition sda6 (“SCSI disk A, partition 6”) is mounted as /home, and sdb1 (“SCSI disk B, partition 1”) is mounted as /media/pendrive. Mount options such as rw (“read/write access”) are given in parentheses.

You can mount a partition like so:

sudo mount DEVICE DIR

This will mount DEVICE to DIR. In Linux, every partition is represented by a virtual device. Only root can mount devices by default, that's why you have to use →*sudo*. Here is an example of how I would mount a data partition on my system:

sudo mkdir /media/data
sudo mount /dev/sda3 /media/data

First, I created the directory in /media/data as a mount point for the partition. Then I mounted /dev/sda3 to that directory. Since sda3 runs a Linux file system, I did not have to specify the file system type. This is necessary in rare cases, though, so here's how it's done:

sudo mount -t vfat -o rw /dev/sdb1 /media/pendrive

In this example I mounted a USB flash drive partition to /media/pendrive. I specified vfat as the file system type. Moreover, I provided the option rw to mount the file system with read/write access. Available mount options may differ across file system types (see →*man mount*).

Linux lets you mount ISO files, like this:

sudo mount -o loop image.iso /media/somedir

The option *loop* tells Linux to set up a virtual “loop device” that makes the contents of image.iso accessible. This is a kind of trick that Linux uses to mount ISO images without any special software.

There is an important file called /etc/fstab ("file system table") that lets you specify partitions along with their mount points for future use. Referring to one of the examples above, I could add the following line to my /etc/fstab to make mounting sda3 more convenient in the future:

/dev/sda3 /media/data ext4 user

This means that /dev/sda3 of type ext4 will be mounted as /media/data. Now, whenever I want to mount the partition, I just enter *mount /dev/sda3* or *mount /media/data* and fstab will fill in the missing details for me. I've specified the option *user*, a very useful option that you should remember. It allows regular users to mount the partition, making the use of *sudo* unnecessary. Use →*umount* to unmount a partition.