www.sxc.hu

Media access with mount and fstab

MOUNT UP

This month we look at tools for mounting and unmounting storage media. **BY HEIKE JURZIK**

The penguin does not use Windows-style drive letters – instead, Linux attaches media directly to the directory tree in a process known as mounting, which is performed either at boot time or using manual commands. There's a place for everything in the Linux filesystem tree; device files for devices of all kinds, such as network devices, removable media, or hard disk partitions, can typically be found below the */dev* (for "device") directory.

Name Game

IDE device names (the names of hard disks, CD-ROM, or DVD drives) start with *hd* (for "hard disk"); the letter that then follows depends on the connector and the order. Each controller provides two connectors; the first connector is called the master and the second the slave. The device names are: *hda* (master) and *hdb* (slave) for the devices attached to the first controller. Devices on the second controller are known as hdc (master) and hdd (slave).

Linux handles SATA disks, USB mass storage devices, and Zip drives as if they were SCSI devices; their device file names thus start with *sd*. SCSI CD and DVD drives are dubbed *scd* and floppy disk drives are *fd*.

Besides the letters, many devices also have numbers that reflect the logical structure of the storage medium: for example, the first primary partition on an IDE master hard disk attached to the first controller is referred to as *hda1*, the second partition is then *hda2*, etc. Logical partition numbers start at 5. The second logical partition on *hdc* thus maps to the device file */dev/hdc6*.

The system enumerates SCSI CD/DVD drives in the same way (*scd0*, *scd1*, etc.), along with floppy drives (*fd0*, *fd1*, etc.), although it is quite common to not have a floppy drive in a modern PC.

Many distributions use aliases such as */dev/cdrom* or */dev/dvd* that point to the actual device names for CD-ROM/DVD drives.

To support access to the various devices and other filesystems, you need to create a link between a device and a directory in the Linux filesystem tree: the device file is mounted to allow this to happen.

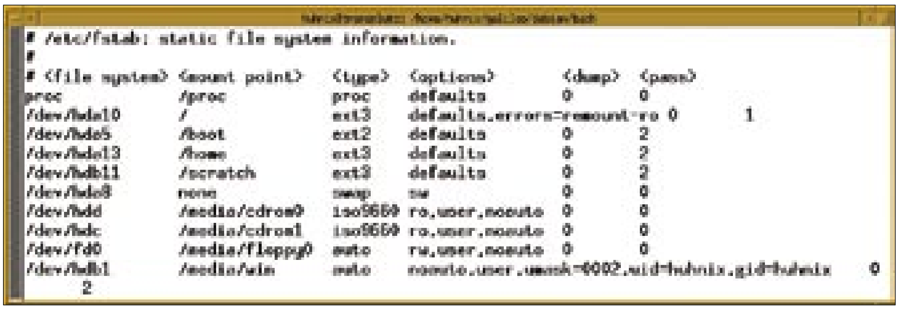
Mounting

Mounting either occurs at boot time, or manually at a later stage. Hard disk partitions are normally mounted at boot time, whereas this is a manual process for CDs, DVDs, and other removable media. Mounting is often the domain of the system administrator root – unless the privilege is specifically given to users (see the *Tabular – the "/etc/fstab" File*).

The utility for command line based mounting is titled mount; besides a number of optional parameters, you have to specify the device file and the mountpoint. If you call mount without supplying any parameters, the command tells you which media are currently mounted:

```
# mount
/dev/hda10 on / type ext3 ➥
(rw,errors=remount-ro)
/dev/hda5 on /boot ➥
type ext2 (rw)
/dev/hda13 on /home ➥
type ext3 (rw)
/dev/hdd on /media/cdrom0 ➥
type iso9660 (ro,noexec,➥
nosuid,nodev,user=huhn)
...
```

Besides the partition labels and the mountpoints, mount tells you about the filesystems for the devices, and it also lets you know the mount options that are in place. The */dev/hda10* partition has been formatted with Ext3, for example, and mounted as the root partition (at */*); the IDE CD drive contains a medium with the ISO9660 filesystem (the default filesystem for data CDs), and has been mounted under */media/cdrom0*. The listing also tells you that the hard disk partitions are readable and writable



```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/hda10      /               ext3    defaults,errors=remount-ro 0       1
/dev/hda5       /boot           ext2    defaults        0       2
/dev/hda13      /home           ext3    defaults        0       2
/dev/hdb11      /scratch        ext3    defaults        0       2
/dev/hda8       none            swap    sw              0       0
/dev/hdd        /media/cdrom0   iso9660 ro,user,noauto  0       0
/dev/hdc        /media/cdrom1   iso9660 ro,user,noauto  0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
/dev/hdb1       /media/win      auto    noauto,user,umask=0002,uid=huhnix,gid=huhnix      0
        2
```

Figure 1: The fstab file provides information on hard disk partitions as well as removable media.

(*rw* for "read-write"). The following option

```
errors=remount-ro
```

ensures that the medium will be remounted read-only in case of a system error; that is, the data will still be readable, but not have writable access.

Mounting Removable Media

Data CDs/DVDs, floppy disks and USB media are normally mounted manually, unless you have an automounter to handle the task. Linux assigns directories below */mnt* or */media* for removable media. In the command line, you need to type the device file name and the mountpoint. If you are mounting a USB mass storage device, it makes sense to check the */var/log/messages* logfile to see if the device has been correctly detected, and to discover the device filename (see the box titled "Kernel Messages for USB Sticks").

To mount the device detected here, *sdd*, in an existing directory, */media/usb*, type the following:

```
mount /dev/sdd /media/usb
```

You do not need to specify the partition, as the USB stick in our example is not partitioned – you would need to specify */dev/sdd1* for a formatted stick.

Linux typically auto-detects the filesystem type for a medium. If you get an error message instead, you can give Linux a hand and explicitly specify the filesystem by supplying a value for the *-t* parameter, for example,

```
mount -t vfat /dev/sdd ➥
/media/usb
```

for an older Windows filesystem on a FAT-formatted medium. Besides *vfat* (for the DOS/Windows filesystem), the supported values are *ext2* (Extended Filesystem, Version 2), *ext3* (Extended Filesystem, Version 3), *reiserfs* (Reiser Filesystem), *iso9660* (ISO9660), *ntfs* (NT Filesystem), etc., to name just a few.

Most systems define the device names and mountpoints for CDs/DVDs and floppies (see the *Tabular – the "/etc/fstab" File* box), and this means that a command such as

```
mount /media/cdrom
```

may be all it takes to mount a CD.

Critical Mount Options

The mount program has a number of interesting parameters. The *-o ro* option makes a device "read-only"; it's counterpart is *-o rw* (for "read-write"), and also the default setting. Combinations are also supported: to remove write access for a medium mounted with read-write access, you need to supply two parameters when running the command, for example:

```
mount -o remount,ro /media/usb
```

This tells mount to *remount* the medium while at the same time disabling write access (*ro*).

The mount command has a practical parameter that allows you to mount ISO images. To mount a 1:1 copy of a disk prior to burning on CD/DVD for test purposes, enter the following:

```
mount -o loop file.iso ➥
/media/tmp
```

This tells Linux to use a loop device to access the image just like a genuine device.

Tabular – the "/etc/fstab" File

As mentioned previously, Linux mounts some filesystems directly at boot time. The */etc/fstab* file has entries for the filesystems to mount. Besides the full set of hard disk partitions, this file contains definitions for various removable media (Figure 1).

The first column has the device files, whereas the second provides the mountpoints. The other entries specify the filesystem for the media (the kernel normally auto-detects this – *auto*), and various mount options.

Besides the parameters referred to earlier on, *ro* and *rw*, you will often see entries such as *user* (the device can be mounted without root privileges), *nouser* (the opposite), *auto* (the filesystem is mounted at boot time), *noauto*, *exec* (programs on this medium are executable), or *noexec*. As Figure 1 shows, the CD-ROM, DVD, and floppy drives in our example are not mounted at boot time (*noauto*), and any user can mount these media manually (*user*). As the appropriate mountpoints (e.g., */media/cdrom0*, */media/floppy0*), are defined, a user can just type

```
mount /media/cdrom0
```

to mount a CD.

As Windows filesystems do not support file owners or groups, you can spec-

Kernel Messages for USB Sticks

```
01 Mar 31 19:05:14 transpluto kernel: Attached scsi removable disk ➥
   sdd at scsi0, channel 0, id 0, lun 3
02 Mar 31 19:05:14 transpluto usb.agent[4316]:     usb-storage: ➥
   loaded successfully
03 Mar 31 19:05:14 transpluto kernel: usbcore: registered new ➥
   driver usb-storage
04 Mar 31 19:05:14 transpluto kernel: USB Mass Storage support
   registered.
```

ify a default owner (*uid =*) and/or group (*gid =*) for directories on partitions of this type. Additionally, the *umask* entry in our example assigns default access privileges. To modify the */etc/fstab* file, become root, and open the file in your favorite text editor.

Out!

The command for unmounting filesystems is *umount*. Although Linux automatically dismounts all mounted media when you shutdown your computer, there are manual approaches, too. You can unmount removable media, such as CD-ROMs/DVDs, floppies, and USB devices, for example, using the following:

```
umount /media/usb
umount /media/cdrom0
```

While it is important to ensure that USB media and floppies are properly unmounted before you remove them, CD and DVD drives block automatically, and refuse to open the drive bay while a medium is mounted.

There is an additional safety mechanism that comes in handy: umount will not unmount a filesystem while a process is accessing the files:

```
umount: /media/cdrom0: ➥
device is busy
```

A program might be using the data on the CD in the drive, or the data might be part of the working directory used by the Shelll or a file manager, */media/cdrom0* (or one of its subdirectories). There is a trick for using the command line to determine which process is blocking the device: run the lsof program, which displays open files and directories.

Working as root, you can discover the blocking process by running lsof against the device name of the drive, as in the following:

```
# lsof /dev/hdc
COMMAND PID  USER   FD  TYPE ➥
DEVICE SIZE NODE NAME
kdeinit 4466 huhn   12r BLK  ➥
22,0 2846 /dev/hdc
```

The ps tool can give information on the process number (PID in the listing). Your best approach is to output a long list of all processes in wide display mode, pipe the output to the grep tool, and search the output for the process ID:

```
# ps auxwww | grep 4466
huhn  4466  0.0  2.4 27972 ➥
12572 ?  S  11:33  0:00 ➥
kdeinit: kio_audiocd audiocd ➥
/tmp/ksocket-huhn/klauncherz➥
9ZRha.slave-socket ➥
/tmp/ksocket-huhn/konqueror➥
VPZ1va.slave-socket
```

In this case, it looks like Konqueror is the culprit. Close the file manager window showing the CD content, or just quit Konqueror; you should then be able to unmount the CD by entering:

```
umount /media/cdrom0
```

If this doesn't help, you might have to be more assertive and use the *kill* command. ■