ASK KLAUS!

Klaus Knopper is the creator of Knoppix and co-founder of the LinuxTag expo. He currently works as a teacher, programmer, and consultant. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to:

klaus@linux-magazine.com

External USB Hard Drive

? I enjoy your column in *Linux Magazine* because I am a new Linux user and electrical engineer trying to understand how the system works.

My question is related to the Western Digital "Passport" 250GB external USB hard drive and how the mounting and synchronization of the data cache operates in Fedora 7 with the Gnome desktop. Automatic mounting of the drive takes place when the device is plugged in, and an icon is created on the desktop that can be used to access the data. However, the "eject" feature using the mouse does not properly synchronize the data cache with the physical medium during unmounting, and lost data results on a frequent basis when I disconnect the device.

The drive heads are not stopped either, and a strange whirring chirp sound occurs when unplugging the drive while the platters are still spinning.

I searched Google and found a shell script to properly unmount the drive, and it uses a tool called sdparm to sync and stop the drive heads before physically disconnecting the USB hard drive. I did not have sdparm installed by default from my Fedora 7 distribution (obtained through a *Linux Magazine* DVD) and had to search and install the tool myself using the package manager. (Zenity message boxes also had to be installed for this script.)

I have attached the script in its current working state. The script solves the problem and prevents lost data and stops the drive, as long as I remember to switch to root before calling it from the shell. Otherwise, I am down in the trenches trying to sync my drive before unplugging. The script, however, is limited to only one drive at a time, as the mount point is hard coded to a specific location in */media/WDPASSPORT*.

I would like to know how to take a shell script like this one, which manually unmounts and sync's my USB drives correctly, and associate it with the icon that is automatically created for the drive in the Gnome desktop so that when I use the mouse to "eject" the drive, it calls this script.

Alternatively, can I edit an existing script that already comes with Fedora 7 (as used by the desktop), to resolve the problem? Or is this behavior compiled in?

I believe the "eject" feature used by Gnome is only meant for flash memory drives such as USB keys, where there is no concept of flushing the data in RAM to the magnetic storage medium and stopping the heads, and hence it is a rather simple solution for unmounting that is not suitable for external portable hard drives.

Thanks in advance for considering this Linux hardware challenge.

The quite long script you sent can be reduced to (more or less) this skeleton:

```
#!/bin/bash
exec >/dev/null 2>&1       ⮐
 # No error messages
pumount $1 || umount $1
sdparm --command=sync $1
sdparm --command=stop $1
```

pumount only works if it's installed, and if the normal desktop user is a member of the group *plugdev*; otherwise, the command will fail (and the script will use umount instead). In that case, the desktop keeps the disk busy at the moment of the umount call.

Lazy umount (*umount -l*) could be called instead, which will umount the

drive whenever nobody accesses it anymore. Because a *umount* entry should be part of the usual KDE or Gnome service menu for a hard disk, the entire *umount…* command line is not really needed, but it probably does not hurt if it's there, either.

The SCSI disk control command *sdparm* is available as a Debian package, but it is not necessarily preinstalled in every distribution yet. In this example, it will send a "sync," which will write back all pending physically unwritten data back to the medium; however, in theory, the *umount* command should also have told the (virtual) SCSI driver to do this.

The most important command in the example is probably *sdparm --command=stop*, which will power off the motor and park the heads of the removable disk to avoid the noise effect you mentioned. Noise always means mechanical friction, which damages disk surface and heads in the long term.

Now, to get the script to work with the right-click context menu, edit this file in KDE:

```
/usr/share/apps/konqueror/⮐
servicemenus/media_eject.desktop
```

by changing the line that says

```
Exec=kio_mounthelper -e %u
```

to

```
Exec=/path/to/your/script %v
```

The script must be called with the device name as a parameter, which is what *%v* means.

For Gnome, I have not found a way to do this without changing the source of the mount/eject helper applets, but there may be a way to change the settings for manually created icons.

Permissions

One of the frustrations I have in Linux is folder and file permissions.

How do I give a command so that a folder and all that is underneath it (folders and files) are completely open?

I know it is

```
Chmod 777
```

but what do I write after that?

I have a folder called *church* that is under *htdocs*. At the prompt

```
dull:/srv/www/⮐
htdocs #
```

I have typed

```
chmod church/*
```

but when I checked later, I found some files were not changed.

What am I doing wrong?

chmod works with options that tell the command what to do. If your intention is to make the directory *church*, which is a subdirectory of */srv/htdocs*, and all files under it readable, writable, and executable for everyone on the system, the command is:

```
chmod -R ugo=rwX ⮐
/srv/www/htdocs/church
```

Be careful with the spelling. Capitals and spaces have a meaning.

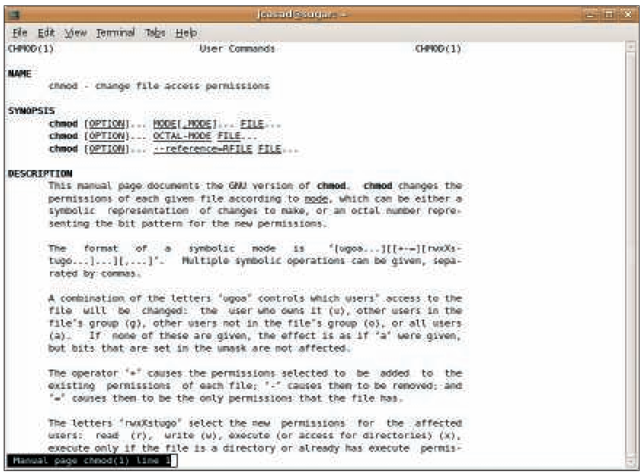
Table 1 shows a description of the options used in this command. Type *man chmod* at the command line or consult a standard Linux reference for a complete list of *chmod* options.

If you want to see what's happening while it's happening, use option *-v* right after *-R* in the preceding command.

The abbreviation *777*, which you mentioned, is a bitmask that means the same thing as *ugo=rwx* (which would also force execute permissions for files that are not supposed to have them).

Table 1: Options in chmod -R ugo=rwX

-R	(capital R) means "recursive" (i.e., not only the directory, but also everything inside).
ugo	Change the settings for: user (file creator), group (every file/dir has one), all others (who are neither user or group)
=	Set only these permissions; remove all others.
rwX	Set read and write permission and set the "executable" flag for directories, which is necessary in order to access their content.



```
CHMOD(1)                     User Commands                    CHMOD(1)

NAME
       chmod - change file access permissions

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This manual page documents the GNU version of chmod.  chmod changes the
       permissions of each given file according to mode, which can be either a
       symbolic  representation  of changes to make, or an octal number repre-
       senting the bit pattern for the new permissions.

       The   format   of   a   symbolic   mode   is   '[ugoa...][[+-=][rwxXs-
       tugo...]...][,...]'.   Multiple symbolic operations can be given, sepa-
       rated by commas.

       A combination of the letters 'ugoa' controls which users' access to the
       file  will  be  changed:  the  user who owns it (u), other users in the
       file's group (g), other users not in the file's group (o), or all users
       (a).   If  none of these are given, the effect is as if 'a' were given,
       but bits that are set in the umask are not affected.

       The operator '+' causes the permissions selected to  be  added  to  the
       existing  permissions  of each file; '-' causes them to be removed; and
       '=' causes them to be the only permissions that the file has.

       The letters 'rwxXstugo' select the new  permissions  for  the  affected
       users:  read  (r),  write (w), execute (or access for directories) (x),
       execute only if the file is a directory or already has execute  permis-
 Manual page chmod(1) line 1
```

Figure 1: Type man chmod to call up the chmod manpage.

The preceding command should do what you intended; however, with file permissions like this, everyone working on the computer will be able to read, change the content, and delete files. In case your web server uses potentially unsecure scripting extensions, the web server itself can even change the content of files now, because it has full write access. Be aware of the security implications, and make sure you always have a recent backup.

Partitioning

I just wrote you these few lines because I have a couple of questions. I am very new to Linux. I had installed a new hard drive to my laptop because the old one was bad. I used the recovery disk, and it put back the original Windows XP operating system on the hard drive.

I would like to partition my hard drive so I can install a Linux SUSE system as well. How can I partition my NTFS hard drive?

Also, how can I differentiate between which operating system I want to use at the beginning of the booting process?

Caution: There is no such thing as an "NTFS disk." A hard disk is independent of the operating system and can be partitioned into smaller parts that can be used by various operating systems with their individual filesystems.

Although Linux can read and write to NTFS without problems using ntfs-3g, NTFS is still not a good filesystem for hosting a Linux operating system because it simply does not support all of the features and file types that Linux needs. The steps for creating a new partition for Linux – or better, two new partitions, are:

1. Run an NTFS defragmentation on Windows. This should put all files at the beginning of the NTFS partition, so there is enough free space at the end to resize and create new partitions.
2. After defragmenting the filesystem, resize the partition holding the NTFS/Windows filesystem – reduce the partition to a size that you think is necessary. Be aware that Windows is not capable of writing to Linux filesystem partitions without additional drivers, but Linux can write to NTFS. You can do the NTFS resizing on Windows (there should be an utility in "properties") or use a hard disk partitioning program under Linux that can handle NTFS (qtparted, gparted, or ntfsresize).
3. Create at least two new partitions on the remaining free space. One should be a swap partition that is used by Linux to extend the available RAM with available hard disk space. (The size of the swap partition depends on what you plan to do; for large graphics/video editing programs, use 2GB or more for swap), and use the second (third, in total) partition for the Linux installation. You can create the Linux partitions from the installer of most Linux distributions, so you could also just go straight to step 4.
4. Run the Linux installation disk, and be careful to install to the new Linux partition, not the Windows partition.

Usually, the GNU/Linux installations will create a master boot record that lets you choose the operating system from a boot menu.

Internet

? I am hoping you can help me. I am writing this under Windows (ugh) because whichever version of Linux I try I cannot get on the Internet or receive email. I was on Linux for about a year using SUSE, Xandros, or Mandrake 9, but one day all connectivity ceased.

Both my computers are AMD. I was using a D-LINK ASDL modem router DSL-504T and an Ethernet card. The machine detects the card , but I cannot call *10.1.1.1* to configure, or the programs can't reach home to finish updates.

Both computers use Firefox and Thunderbird, the same as Windows, using the same asdl modem. As I said, all was working well and suddenly stopped. D-Link is advertised as Linux compatible. I am not a power user – just an old guy who rebels against Microsoft.

You don't need to be an expert or a rebel to use Linux, but you are right about the fact that free software is a way out of dependencies. By the way, I'm unfortunately living in a DSL-free zone, and I am waiting for bandwidth. Anyway, let's try to free your DSL modem.

There are two ways to configure the DSL modem, the easier of which is *router* mode. The default is probably *modem/pppoe* mode.

In router mode, all you need to do is send a DHCP request from the network card for a complete autoconfiguration. In modem mode, you need to configure authentication credentials on your computer, which can be done by using the program pppoeconf (or whatever tool is provided by your distribution).

To configure the modem for one or the other mode, you first need to reach it from the computer. If your modem's fixed address is *10.1.1.1*, your network card must be configured as a member of the same network in order for you to connect to the modem. Usually, the modem is set up to deliver an appropriate local IP address to the connected network card when the card sends a DHCP broadcast, but this may depend on the modem's preset configuration.

If your network card is *eth0*, you would set an IP address for the card either via a graphical configuration front end of your Linux distribution – use DHCP with *pump -i eth0* or *dhclient eth0* – or set a static address by typing (as root)

```
ifconfig eth0 10.1.1.3 ↩
netmask 255.0.0.0 ↩
broadcast 10.255.255.255
```

and add a gateway route through the modem with

```
route add default gw 10.1.1.1
```

which should, as soon as the modem is functional as a router, allow network traffic to and from the Internet.

One last thing you need, in case DHCP did not set a name server, is the line *nameserver 10.1.1.1* in the file */etc/resolv.conf*. This tells Linux to use the modem as a name server for resolving computer names (such as linux-magazine.com) into IP addresses (which is what every network connection needs).

After having configured your local network this way, you should be able to reach the modem with any browser by entering *http://10.1.1.1/* as a website address. If the modem is already set to router mode, then you'll already have immediate Internet access.

At each step, you can check to see whether the modem is reachable from the configured network card with the command:

```
ping 10.1.1.1
(Control-C to quit)
```

I think that the only thing missing in your DSL configuration was the local IP address of the network card, which can be handled with either DHCP or a static setting.

Also, be aware that in rare cases, some DSL routers/modems only accept network cards when they have been plugged in prior to switching on the modem. However, this should not affect the DSL-504T. ■

Send your Linux questions to klaus@linux-magazine.com.