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

Ubuntu

Ubuntu 6.06 is the best Linux distro I have tried, but I do have one perplexing problem. When I use Kaffeine to play unencrypted DVDs recorded from my Phillips DVD recorder (hooked up to my television), I get a message saying that the audio device is busy and that the DVD cannot play. Unencrypted DVDs that are made from my camcorder, recorded from my computer, play fine, however. I can play unencrypted DVDs that are recorded from the television through my computer with no problem, too. Everything I record through the computer is in Windows XP, and everything is able to play with no problem in Windows.

Am I missing a codec, or do you think that the Phillips DVD recorder is just not compatible with Linux?

It could be that the error message is wrong, though I wonder why this message suggests that the audio device is busy. Kaffeine uses libxine and all codecs of xine that are either available as built-in or as installed plugins, including optional proprietary libraries/codecs.

If it is really an audio problem, try to turn off any running sound server and switch off sound server capabilities in KDE and Kaffeine, kill artsd (or Gnome's sound daemon) and retry.

If it is a codec problem, check or compare the erroneous DVDs with *lsdvd*, which you can find in Debian and other distributions. If this does not produce any obvious results or differences, you may have to further investigate the recorded VOB files on the mounted DVD manually, and check for some kind of added "play protection" like CSS.

Another possibility could be that the recorder does not really create video-DVDs, but rather records DivX files or similar. In this case, you will need the appropriate plugin to play these file formats; not all of them are free of a license fee, even under GNU/Linux.

You can check for files like this by simply mounting the DVD and listing its contents. Video DVDs usually contain a directory called *video_ts* and a directory called *audio_ts*. If *video_ts* contains a lot of VOB files, it looks like a video DVD, but the VOB files may still be encrypted or CSS-scrambled. This makes it illegal to play the DVDs without a valid CSS player license in most countries, regardless of the operating system that is being used. It's not really a technical problem, though.

However, a DVD recorder that is unable to create valid unencrypted DVD recordings that can be played with legal software players is subject to a warranty replacement, at least in my opinion. I want to be able to play a self-recorded DVD on any operating system without worrying about copy protection or content scrambling, licenses, or patent fees.

Another thing you could do is retry with a different player, like xine (on which Kaffeine is based), mplayer, or vlc. If one of them can play the DVD (without proprietary plugins), it may just be a software bug in Kaffeine, which will be removed in one of the next versions. In that case, don't forget to send a bug report.

32-bit or 64-bit

I am using Debian Linux Sarge and intend to shift to Debian Linux Etch when it comes in December 2006. I am also going to buy a new PC. I plan to use encrypting on my hard disk and am currently using an AMD Athlon XP 1800+ 1.53 GZ-type processor with 256 MB of RAM.

Do I need to get a faster processor to use encryption? Should I use a 32-bit or 64-bit processor? Are there any problems with using a 64-bit processor with Debian Linux? What is the difference between a 32-bit and 64-bit processor?

Can I use a twin processor with Debian?

Also, I am an ordinary end user (video, music, TV, etc.), and I find my system is a bit slow when doing video editing. Are there any configuration settings that will speed up my system?

Your current setup should be sufficient to do on-the-fly encryption of a hard-disk partition with, for example, loop-aes. There are different approaches for file-based or partition-based encryption. The new kernel 2.6.19 also contains ecryptfs as an alternative to partition-based encryption.

I'm also a happy user of loop-aes, which encrypts an entire partition (or a partition image that can be used as a virtual hard drive, like in Knoppix), blockwise with a well-known algorithm that is, as of now, known to be hard to break. A good Howto for loop-aes is on the loop-aes Sourceforge page at *http://loop-aes.sourceforge.net/loop-AES.README*.

Whether encrypted file access is fast enough depends on your desired use for the filesystem. When doing lots of disk I/O, you may be better off with a fast processor. The loop-aes kernel package contains some processor-specific accelerated code for various architectures, including aes-amd64.S. For example, if you plan to do realtime video recording to hard disk with encryption, you will want to have a fast processor, in addition to a fast hard disk and hard-disk controller (like SATA). However, is it really necessary to realtime-record videos to an encrypted partition?

For just storing personal data (like mailboxes, personal settings, and web form passwords) with encryption, even an old P1 processor with 300 MHz is more than sufficient. Encrypting the entire hard disk is probably not required, but home directories (on a separate partition), */etc*, and perhaps swapspace are a good idea for encryption.

The difference between a 32-bit and 64-bit processor is, well, the 64-bit processor can compute more bits at once and has an extended set of instructions that don't exist in the 32-bit version (in a nutshell). Most of the 64-bit processors run 32-bit programs fine, but a mixture of bitness can and will cause problems (see below). Since newer 64-bit machines usually have newer and faster boards, the throughput to and from hard disk is usually faster (not necessarily related to 64-bit), which would be good for your desired video editing.

Problems with changing processor architectures are sometimes little vs. big-endian (the sequence in which the bits are stored) issues, and 32-bit software running in a 64-bit environment.

You *can* run 32-bit programs with a 64-bit processor (Knoppix 32 bit will run fine on 64-bit processors), however, the system libraries that are linked and used with each program must match the bit and endianness of that program, otherwise you will experience various types of crashes. If you want to continue using "old" programs, you should probably set up a 32-bit chroot environment in which they can run without accessing any 64-bit parts.

However, this doesn't affect encryption (well, it shouldn't). At least data created with 32-bit loop-aes will still be accessible from a 64-bit-compiled loop-aes kernel module. The same is true for PGP/GNUPG. I cannot guarantee this with other proprietary or architecture-specific encryption software, though.

Debian GNU/Linux is available native for 64-bit processors, especially the AMD64 port. Installation CDs and instructions can be found at *http://www.debian.org/ports/amd64/*. You may need different Debian installation CDs for different 64-bit processors, since instruction sets from different vendors are also slightly different.

If you plan to use a multi-core processor, which is well supported in Linux, or a multiprocessor board, be sure your kernel is compiled for multiple CPU cores. As a last resort, kernel recompilation for SMP is not too difficult.

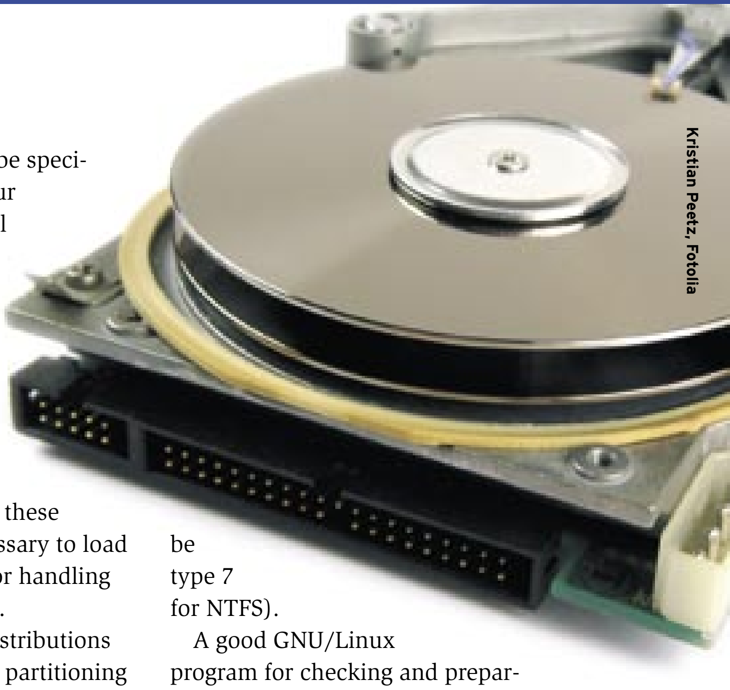
Extended Partition

I realize this is not strictly a Linux question, but you have some experts there that might be able to help. I have a laptop that came with XP. Until I have a proper working Linux on it, I want to keep XP to verify that hardware works, etc. My hard drive has eight partitions on it at the moment. One XP, one XP recover, an HP utilities partition, and some that I don't know. The bottom line is that to install Linux, I need to create an extended partition and do all the other Linux partitions inside of that.

FC5 can install, and so can Centos 4.2, but I want Mepis or Ubuntu for the wireless support the others don't offer. However, neither Mepis nor Ubuntu (6.06) seems able to work within the extended partition. Even after I manually run QParted, Mepis wants to reformat the XP partition as the swap partition. Is there anything I can do about this?

GNU/Linux works perfectly on logical partitions created within an extended partition. However, not all partitioning GUIs or installers are ready to handle complex partitioning schemes, especially if the previously created partition table is strange (i.e., contains partitions in incorrect order, has overlapping partitions, or has several extended partitions).

A hard disk usually has four base partitions that can be used either as *primary partitions* or as *extended partitions* (think of them as hard disk inside a hard disk) that can contain another partition



Kristian Peetz, Fotolia

table with *logical* partitions. This has nothing to do with logical volume management yet. In most setups, there are up to three primary partitions used, and one (of the four base partitions) as extended partition, which contains as many additional logical partitions as the user wants.

With IDE hard disks, the naming scheme is usually */dev/hda1…/dev/hda4* for the four base partitions, and */dev/hda5* and up for logical partitions on the extended partition(s) on the master hard disk attached to the first IDE controller. If you have more than one extended partition, it's not easy to determine which of the logical partitions on the extended ones becomes */dev/hda5* and up. Your pre-existing partitioning may therefore simply confuse some installers, which may be the reason why repartitioning and preparing partitions for Linux becomes difficult.

GNU/Linux usually needs at least one partition for the Linux file system (partition type 83) and one partition as swapspace (partition type 82), though it is possible to use a swapfile instead of a swap partition.

It does not matter whether these partitions are primary or logical partitions.

The used partitions have to be specified in the boot loader of your choice in order for the kernel to find and mount the / ("root") file system at system start.

If the computer's BIOS is unable to recognize a partition because of a bad or invalid partitioning scheme, and additional software is needed to access these partitions properly, it's necessary to load additional kernel modules for handling these partitions via an initrd.

Since some GNU/Linux distributions seem to work fine with your partitioning setup, this is probably not the case and it is simply a flaw in the installer or partitioner of the distributions who can't. A workaround could be preformatting the needed partitions from within a working distribution, and then just skipping the partitioning part and going directly to the filesystem setup.

It may also be that some partitions in your setup simply have wrongly set partition types, so the installer *thinks* that the first partition is desired as swap, whereas it is simply a NTFS partition with a wrongly set partition ID (should be type 7 for NTFS).

Listing 1: An fdisk Session

```
$ sudo fdisk /dev/hda

Command (m for help): p

Disk /dev/hda: 40.0 GB, 40007761920 bytes
255 heads, 63 sectors/track, 4864 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1               1         200     1606468+   b  W95 FAT32
/dev/hda2             201         800     4819500   83  Linux
/dev/hda3             801         950     1204875   82  Linux swap /
Solaris
/dev/hda4             951        4864    31439205    5  Extended
/dev/hda5             951        1200     2008093+  83  Linux
/dev/hda6            1201        2200     8032468+  83  Linux
/dev/hda7            2201        2325     1004031   83  Linux
/dev/hda8            2326        3400     8634906   83  Linux
/dev/hda9            3401        4864    11759548+  83  Linux

Command (m for help): q
```

A good GNU/Linux program for checking and preparing partition tables is *fdisk* (like the old DOS-based program).

Fdisk only changes the partition table when you issue a *write* command within, but it won't change anything (even if you accidentally hit *delete*) if you leave with Control-C or the *q* command. See the example in Listing 1 for an fdisk session.

This example shows that */dev/hda* contains three primary partitions of type FAT32, Linux, and Swap, and an extended partition that contains five more logical Linux partitions.

A workaround option, if installers fail to handle logical partitions, is to install in a virtual environment and then copy the partition data (*cp* or *dd*) to the real partition later. Renaming the */dev/partitionname* files from within the installer session is also possible option, but is more adventurous.

Make sure that */etc/fstab* and the boot loader have the correct entries for the real partitions. You are able to change this prior to booting the installation in question by mounting the partitions from a working GNU/Linux installation, or by mounting the partitions from a Live CD, chrooting into the root partition and performing the modifications.

As always, make sure that you have a working backup of all your important data before you start changing anything on your hard disk setup, especially when you are working with multiple operating system installations. ■

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