



We invited Klaus Knopper, the legendary creator of Knoppix, to answer your questions on configuring Linux.

BY KLAUS KNOPPER

Sound Cards

? I have put several Linux distros on my computer, and I could never find one that successfully installed the sound system until I found Knoppix. I used to think my sound card simply wasn't compatible with Linux until I realized that Knoppix sets it up perfectly. I have a Gateway PC with a Creative Technologies Audigy sound card, which seems like it must be a pretty common system, but I must have an unusual model, because no other Linux installs it successfully.

The sound works great in Knoppix, however, for professional reasons, I still need access to other distros (both Suse and Debian-based systems). I'm wondering whether I can use the information in Knoppix to set up the sound on the other systems. Is it possible for me to find out how Knoppix has configured my sound card, then copy the driver and any other configuration settings to the distros on my hard drive? If so, how do I do it? If this is not possible, I would be grateful for any information you might have on

ASK KLAUS!



troubleshooting my sound configuration.



The failure to detect and set up your sound card may simply be caused by hardware problems related to interrupt routing. Knoppix has a certain sequence for probing and setting up hardware, taken from experience with various boards and card setups. For example, it seems that PCMCIA and USB/Firewire work better together with other system components if their modules are loaded first. Trying to activate them at a later time can cause interrupt problems with other components (such as sound cards) because of faulty interrupt sharing or the system simply running out of shareable interrupts.

Sometimes it works the other way. If you have a USB device with audio support (some webcams have an internal micro/sound card), this will sometimes be used as the "primary" soundcard, and all later audio devices will allocate a followup sound device (i.e., `/dev/dsp` is the sound device that was detected first, `/dev/dsp1` the second, and so on). Audio software should then be configured to use the right `/dev/dsp*` device, since the first one is maybe not what you wanted (few webcams have speakers).

The same is true if you have a cheap onboard sound card that you don't want to use and a better one as a PCI card. Regardless of the fact that you may have disabled the internal sound card in the BIOS, some (most) Linux distributions will just enable the onboard chipset as the primary sound card if it's detected before the other card.

With your Knoppix tests, it seems you were lucky and the sound card module was just loaded in the correct place, and

no interrupts with other devices are in conflict. Check `/proc/interrupts` to see the interrupt allocation in the running system.

```
cat /proc/interrupts
```

You can export this information to other distributions by just putting the information Knoppix gives you into the right config files on the distribution of your choice. So, after installation of your preferred system (ignoring a possibly non-working sound card first), you just modify `/etc/modules`, which is the file that most distributions use to get the names of the modules to load directly at startup.

You get this information by looking into the output of the `lsmod` command (whenever your sound card is working correctly), which shows you all loaded kernel modules in reverse order (kernel 2.6.x). In other words, the module that matches your first detected sound card will actually be the LAST sound card-specific module on the list.

Under Knoppix, you can use the following command to find out what Knoppix thinks is the best choice for modules to load for specific hardware components:

```
hwsetup -v -n -p
```

Together with the information from `lsmod`, you can now identify the correct module name for your sound card and

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

add this to `/etc/modules` (somewhere at the beginning) on your installed system.

You may also want to check for incorrectly added `sound-slot-*` entries in `/etc/modules.conf` or `/etc/modutils/*` – for Debian, run `update-modules` after modifications – and remove incorrect entries from these config files.

For some distributions there is probably a graphical admin tool to do this, but the description above is the generic procedure for what has to be done.

CD Drive Problems



I've tried installing Linux on my computer a couple of times. The installation CD always runs very well in the CD drive, but then when I finish installing, the CD drive becomes inaccessible. So far this has happened with both Fedora Core and Suse 9.0. Do you have any idea what could be going wrong?



This also happens with some versions of Knoppix after installing with `knoppix-installer`. The installation procedure usually does not copy every single configuration file to the hard disk – especially `/etc/fstab`, the central default for file system configuration, which is sometimes rebuilt when installing to match the specific distribution's paths to devices like CD ROMs, DVDs, USB sticks, or similar.

Now, the KDE or GNOME icons for these devices may still point to paths that KDE or GNOME think are appropriate. So, if you click on one of these icons, you may get a `directory/mount-point does not exist` message, or even a `no such device`.

To check your filesystem setup, you should first have a close look into `/etc/fstab` (as the root user, if you plan to change something).

The general syntax of a line in this configuration file is

```
device mountpoint ↗
filesystem-type ↗
filesystem-option-list ↗
backup-frequency ↗
checking-order
```

separated by spaces, where the last two columns (backup + checking) are less important to removable media and are set to 0 in this case.

The line above matches a `mount` command like:

```
mount -t ↗
filesystem-type -o ↗
filesystem-option-list ↗
device mountpoint
```

Now, you should check if the CD ROM (or other peripherals) are listed there correctly, and if their mountpoints exist as empty directories that are not used by anything else: I typically use `/mnt/devicename`.

If mountpoints are missing, you should create them accordingly. Mountpoints are the location where you want the content of a device to appear. If you think the mountpoints should be at a different place than the one listed in `/etc/fstab`, feel free to change them.

A typical entry for a CD-Rom would be:

```
/dev/hdc /mnt/cdrom iso9660 ↗
ro,users,noauto 0 0
```

The `ro` option makes the filesystem read-only (which is the case for a CD-Rom filesystem anyway), and `users` allows ANY user to mount and umount the CD (so you don't need root permissions). If you want to make sure that only the user who mounted the CD can umount it, use `user` (without `s`) instead.

So, the line means that any user can mount or umount a CD from the first master drive of the secondary IDE controller that contains a CD-Rom (iso9669/Rockridge/Joliet) file system into the `/mnt/cdrom` directory.

Make sure to have `noauto` in the option list. Otherwise, your system will try to mount a CD automatically when booting up (`mount -a`) and will give you a nasty `corrupt filesystem` error if there is no CD to mount.

If you plan to use CDs containing file systems other than iso9660, you may want to try the `auto` keyword instead of `iso9660`.

Now with this line, you are ready to mount a CD manually by typing the following command:

```
mount /mnt/cdrom
```

as a normal, unprivileged user. If you want to do this with a mouseclick, you can add a KDE icon: press the menu

mouse button on the desktop, select `New | Device Link | CD-Rom`, and in the settings, select the offered `/mnt/cdrom` mount entry. That should be it.

Note that KDE from version 3.2 and later supports an automatic setting for desktop icons for devices (use the control panel). I prefer the manual method, though, because I don't trust automatic features enough.

The following line would be the `/etc/fstab` entry to allow users to mount/umount a USB memory stick or digital camera storage read-write. Such devices are usually handled like SCSI hard disk partitions internally; check the output of `dmesg` shortly after plugging in the device if you can't guess its device name.

```
/dev/sda1 /mnt/sda1 ↗
vfat rw,users,umask=↗
000,noauto 0 0
```

The `umask = 000` option virtually replaces the non-existing DOS file permissions with Unix-like permissions that allow anyone to read and write files on the storage device. This is, of course, not advisable on multiuser systems. With multiple concurrent users, you should instead use `user` instead of `users` and no `umask` option.

Which Laptop?



I'm getting ready to buy a laptop computer. I have heard that power management, wireless, and other portable features are sometimes difficult to set up in Linux. Do you have any thoughts on which portable PCs are most compatible with Linux? What should I watch out for? Are there any particular components or technologies I should try to avoid? If I do encounter problems with configuring wireless networking, are there some common settings I should check first? I realize it is difficult to give any detailed troubleshooting advice, since I don't have a computer yet, but I would be interested in any standard troubleshooting steps for how to identify the source of a problem with the wireless configuration.



Laptops are, and have always been, *different*...

Due to physical space limitations and power management issues, laptops use chipsets that are, in the best case, strange but compatible to standard

chipsets on desktop computers. But sometimes, they are genuinely exotic, and drivers only exist for a certain version of the delivered operating system (even true for Linux, sometimes with binary-only modules that only work for a specific kernel). So, after the next major upgrade, you are likely to lose components of these laptops due to *no longer supported* issues in the next “service pack.” (I guess you know what I’m talking about.)

Power management and chipset issues are there for EVERY operating system. Some laptops work better with Linux

than with Windows, and some the other way round. There is unfortunately no general rule except: *Don’t Buy Anything That Just Came Out; wait for test results and compatibility reports.* Following that rule, “special one-time offers” are a no-go, since there is no way of knowing in advance what’s inside the box. (The vendors usually won’t tell you, and opening the case voids your warranty.)

There are some hardware resellers who sell laptops that are tested and approved for Linux. They can tell you which parts work well and which are problematic even if you don’t plan to buy from them. Also, you may want to check Linux + laptop specific websites like <http://www.linux-laptop.net/> or just google around for laptops with specific features and Linux.

Power management is usually not as big a problem as you may have heard. It’s more the fact that some boards have serious errors in their implementations of the ACPI standard, and Linux tries to provide more or less automatic workarounds to turn off features like incorrect interrupt routing or CPU frequency mis-

calculations. In some cases, you may succeed in making a misbehaving laptop work by enabling kernel boot options such as

```
acpi=off noapic ↗
pnpbios=off pci=bios
```

which basically reverts to the old-but-working hardware-configuration of legacy computers.

But sometimes, you will even see boards where you have to start the Linux kernel with:

```
acpi=force
```

or even

```
irqpoll
```

to get things like half-compatible disk controllers working. (They used to kill the kernel with error messages like “aieeee — interrupt lost.”)

Another important issue is graphics chipsets. Nowadays, the main vendors keep the specifications of their chipsets a big secret. (My guess is that they may just be trying to hide how they cheat in graphics benchmarks.) You can get most, even *non-existing* (by device id), “single-shot” graphics chipsets to work with XFree86 or Xorg if you fiddle around with config options long enough. Sometimes, you have to specify the wrong chipset in the config files to get the right results. It’s very annoying. So, if you are up for painful research, trial and error, and possibly burning your board at the end, go on and buy the brand-new, 16-pipes-3D-pixelboost-never-been-there-before-special-deal notebook.

For the rest of us, I’d recommend checking for a “normal” chipset that is also used in other computers, one that is known to work with the standard Linux XFree/Xorg graphics drivers. You should not rely on a vendor promising they have accelerated but binary-only plugins that work on one distribution and will “probably” work on others. These features usually fall apart at the next update, or they fail to work from the outside in spite of promises from resellers.

If you are up for gaming, you should check if the chipset of your desired laptop has either direct acceleration sup-

ported by XFree or Xorg, or if it is verified to work flawlessly in 3D acceleration with vendor-provided (you have been warned...) binary plugins.

There is a games-knoppix live CD edition containing accelerated graphics drivers especially for games, see <http://games-knoppix.unix-ag.uni-kl.de/>. You may want to test run the CD, and play some games on the laptop, before buying it.

In general, running Knoppix with or without boot options on a notebook for a test and checking if all hardware components are found is probably not a bad idea. Since most notebook displays do NOT support DDC (plug and play for monitors), you will have to boot with the following:

```
knoppix screen=1400x1050 ↗
vsync=75
```

to switch to a 1400x1050 at 75Hz resolution when X starts, since by default, Knoppix will try 1024x768 with a guessed vertical frequency (something around 72Hz, if not otherwise specified). Change the screen = resolution option to what the display really has.

Wifi cards (or “new” network cards) can also be an issue if these cards are not natively supported by the Linux kernel yet. For some cards you can simply use the supplied Windows driver disk along with the *ndiswrapper* kernel module (<http://ndiswrapper.sourceforge.net/>) to load the *.inf file. Prism/Prism2/Ori-noco, Atmel, and some other chipsets widely in use are supported by the Linux kernel as well, but some cards need additional firmware that has to be present with a card-specific name in the */usr/lib/hotplug/firmware* directory.

So, “try before you buy” is the best advice I can give. ■

“Laptops are, and have always been, different ...”



THE AUTHOR

Klaus Knopper graduated from the University of Kaiserslautern with a degree in electrical engineering. Since then he has worked as a teacher, programmer, and consultant. He is the co-founder of the LinuxTag expo and the creator of the popular Knoppix live Linux CD distribution.