



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

## Screen Resolution



I'm using the Knoppix distribution and very much like it. Because I'm a beginning Linux user, though, I am having some problems. The screen resolution in my portable computer is 1280x800, but this mode is not available on my system. I've heard there is a solution for this, but I am having difficulties finding it. Please give me some advice. Video card drivers are present. My video adaptor is Intel Mobile 915GM; the default modes, which are found by Knoppix (hotplug, I think), are 1024x768 and 800x600.



Knoppix tries to run only *safe* resolutions and frequencies on your graphics adapter and monitor by default, if the hardware is not able

# ASK KLAUS!



to report the *right* configuration options. This is usually 1024x768 in 75Hz. If your notebook has a native 1280x800 resolution, and it is not DDC-capable (i.e., Knoppix cannot read this information from the graphics adapter and display), you only get the 1024x768 resolution.

To change this, boot with the following options, entered at the boot prompt:

```
knoppix screen=1280x800 ↵
vga=normal
```

The *vga = normal* stays in text mode until the X Server is started. It's probably not necessary if your notebook handles the framebuffer mode 1024x768 well using resolution scaling.

You may have to add horizontal and/or vertical frequencies according to your computer's technical specs in the manual:

```
knoppix screen=1280x800 ↵
hsync=90 vsync=75
```

Some other live distros use the maximum resolution and let the graphics card handle out-of-range conditions, thus giving better initial settings for high resolution monitors without needing boot options. But if you have fried an old VGA monitor once like I did (including white smoke, noise, and stench, not a pretty sight) using too high resolutions and frequencies, you tend to be more cautious not to ruin old hardware; that's why Knoppix is very conservative about resolutions and frequencies unless you either have a monitor that reports its preferred settings properly, or you specify boot options that do the trick.

If you use the "persistent Knoppix image" feature, the automatically created */etc/X11/xorg.conf* file will be reused on the next boot, so you don't have to enter the boot options again.

To manually configure your X Server, you can use

```
dpkg-reconfigure xserver-xorg
```

in Debian.

## Migrating Settings



I have been a subscriber to your magazine from the beginning, but I have the following problem. I had a web server, with SQL, email, etc. on a Pentium II machine. Now I have updated my hardware to Pentium 4 with dual core and a mother-

### Listing 1: Mount Flags

```
01 mkdir /mysystem
02 mount -o dev,suid /dev/hda2 /
mysystem
03 mount -o dev,suid /dev/hda5 /
mysystem/home
04 mount -o dev,suid /dev/hda6 /
mysystem/var
05 mount --bind /proc /mysystem/
proc
06 mount --bind /sys /mysystem/
sys
07 mount --bind /tmp /mysystem/
tmp
08 # If you are using udev, you
will also need this:
09 mount --bind /dev /mysystem/
dev
```

board with an integrated RAID controller.

My question is, how can I migrate all my data and configurations to my new machine? Bear in mind that my current kernel is 2.4 and now I will have to migrate to 2.6.16. testing, which is the only kernel that recognizes my RAID controller.



Moving the operating system, applications, and data to a new machine is a common task, yet

it is not always easy to find the best strategy to do this.

There are basically two strategies. The quick solution is to try to move all data, including the operating system, libraries, and everything else, and then just exchange the kernel to match the new hardware.

This strategy is probably the fastest way, but it is only possible if the hardware architecture does not change in a way that your applications (compiled for a specific processor) won't run anymore. When moving from a Pentium II, in your example, to a Pentium 4, this should not be a big problem. The only thing you need to do is configure and compile the new kernel to match your hardware.

When moving from Kernel 2.4 to 2.6, you will have to update at least your modutils (modprobe, insmod, rmmod) because the module handling has changed in 2.6 in a way that modutils from 2.4 simply won't work anymore. Of course, you could as well use a static kernel with everything you need com-

piled in, instead of loading modules on demand.

I would recommend you at least statically compile in everything that's needed for mounting the root filesystem, so you won't need an initial ramdisk. Marking the controller and the needed file system(s) with *y* in *make menuconfig* of the kernel can make testing, upgrading, and debugging your system much easier, because it keeps initrd/initramfs and module-related problems out of the critical boot phase.

In case something goes wrong, you can use a Kernel 2.6-based live CD to find out what you have forgotten in order to get the copied system up and running (see below).

So, the sequence would be:

1. Copy everything to the new machine, either with a 1:1 copy of your hard disk, or remotely to one or many freshly formatted and mounted partitions on the new server. I recommend you type:

```
rsync -Hav old_root_dir [root@?
remote-machine:] new_root_dir
as root user in both cases.
```

2. Configure and compile a new kernel on the new machine, and at least update modutils. You may find it useful to (as root) *chroot* to *new\_partition* on the new machine, so you are already working "inside" your copied system, if you can't boot directly with the old kernel or the new one is not yet working right because you forgot something in the configuration. Please note the comment about *chroot* below.

If you don't like the sound of the "quick way," you may be better off with the other solution, which I would call the "right way." This approach requires installing an up-to-date operating system and all needed applications, then migrating the configuration files and data.

This is the method I would recommend to provide a clean start, get rid of obsolete programs (with potential security flaws), and reach a state-of-the-art system that's easier to keep up to date. This method requires somewhat more work, though, especially when configuration files have changed in syntax and location. Also, finding out which files need to be transferred, and which ones have been changed from the default setup, is not always easy if you did not write down a log of changes you made over the past years. (Almost nobody documents changes on their own servers well enough. I don't either, but I know people must exist somewhere who do.)

Anyway, using this strategy, you would first partition the new system and install Debian (as an example) from scratch, including all programs and services you will need. It's not always sufficient to check on your old system with

```
ps auxw
```

to see what servers are running. (*netstat -tulpe* is also a good way to see what's currently listening for connection requests.) There may be support programs or libraries (such as php) that don't show up directly in the list of processes. Use *dpkg -l* or *rpm -qa* on your old machine to see what's installed, and write down all packages that you will probably need. It's usually a long list.

After your new system is up and running (including the perfect kernel for your hardware setup), you will have to transfer and maybe rewrite your configuration files for all services. I would recommend doing this option-by-option in each config file, because, if you also switched from Apache 1.3 to the 2.x version, for instance, the syntax and location of the config files will change, and trying to just copy the configuration files is most likely a very bad idea that can introduce security holes by enabling additions you do not really want. Files such as SSL certificates, as well as authentication keys, must not be forgotten.

## Rescuing Your System

If you wish to repair or update an installed system that cannot boot directly, you can use a "rescue system." Some root server hosting providers offer this service in the form of a network-booting (diskless) *rescue boot* that allows you to mount partitions and change files on your server even if the system installed on disk is woefully defective and in a non-booting state. (You can also use Knoppix for this rescue task, eventually with a *knoppix.sh* script that starts an SSH server with your ssh key so you can perform a remote login.)

As soon as the rescue system is up, mount the hard disk partitions in the correct order and hierarchy. For Knoppix, you need the *dev,suid* mount flags (Listing 1).

Note that Listing 1 is only an example that most likely will not match your *fstab* entries:

The last *--bind* mount is useful if you want to have access to a running X Server through its socket in */tmp*, from inside the yet-to-be-handled *chroot* system.

Now, provided that the partitions have been mounted in the correct order (check *etc/fstab* in the root partition again), you should be able to

```
chroot /mysystem
```

and, until you exit this shell, you can work inside the previously installed system on disk, install a new kernel or new software package, repair your master boot record with *lilo* or *grub*, etc.

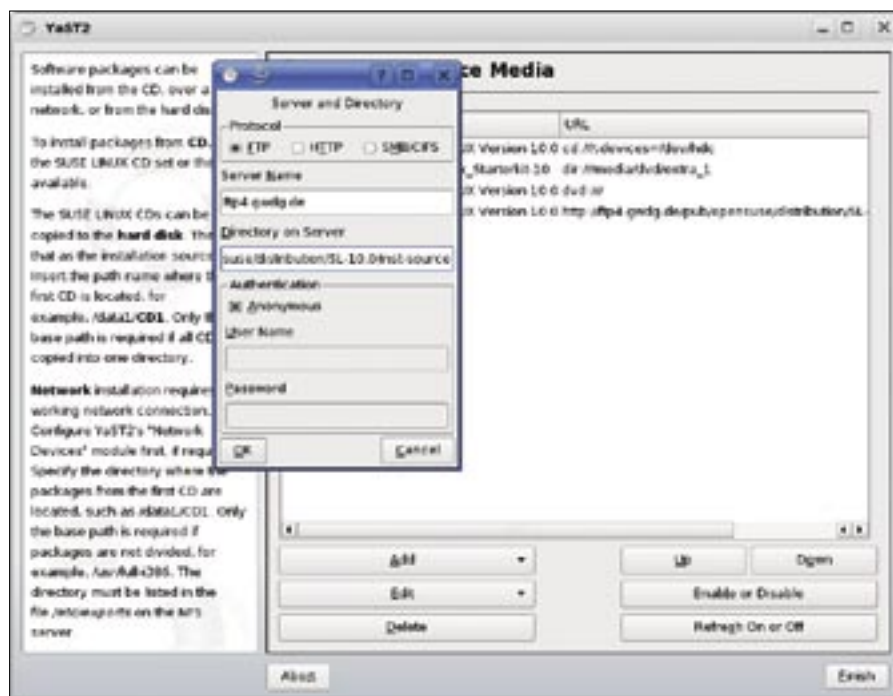


Figure 1: On a Suse system, you can use Yast to install software from an online source.

After the configuration files are done, you can copy the user data, like databases in `/var/lib/mysql`, mailboxes and homedirs (`/var/mail/*`, `/home/*`), and so forth. Make sure the eventual user accounts you create match the file owner and group properties of the copied files. You can use the `-u` and `-g` flag for *useradd* on account creation, or just change the file credentials as needed by *chown/chgrp*. Also, don't forget group memberships in `/etc/group`.

You will then have to thoroughly test each service. Most daemons have their own error logfiles, like `/var/log/apache2/error_log`, which show what's going wrong. For others, check `/var/log/syslog` or `/var/log/messages` (`/var/log/daemon.log` in some distributions).

Though this method is much slower and requires more effort than the "quick way," it provides the most flexibility and safety, and it may save a lot of work and trouble when you are doing updates.

If you break your boot configuration along the way, you can use the rescue method described in the box titled "Rescuing your System."

## Finding GCC



I'm in my second year of reading Linux Magazine. I have Debian 9.3. (10.1 won't load, because my memory is below 512 MB). How do I find files on the DVDs? I have

been looking for a GCC compiler. I have an old (Red Hat) CD that has it. I have been under the assumption that a compiler is on the Debian 9.3 DVD. Is it? If so, how can I get to it?

Most articles skip over the basic details, assuming everyone is familiar. I am not a beginner; I was probably writing code before you were born (as far back as 1962). It has been a long time since I last compiled source code, but if I can find GCC, I know how to use it.



First an expression of surprise from my side: where did you get Debian 9.3? The latest stable

Debian version as of October 2006 is still 3.1, see <http://www.debian.org/releases/>. Debian will definitely boot with less than 512 MB RAM. Even 64 MB should be sufficient, though if you want to work with a graphical desktop, 256 MB is good, and 512 MB or more is nice.

Depending on which distribution you are referring to, your question goes into the direction of how to find and install a specific software package, since most GNU/Linux distributions nowadays use a fine packaging system that allows you to install precompiled software.

If you use Debian, the names of software packages are usually `*.deb`. If you use Suse/Novell, Mandriva, or Fedora GNU/Linux, the package names are `*.rpm`. So, in order to find packages on a

mounted CD or DVD that may contain something GCC-related, the command to find those packages manually would be

```
find /media/cdrom -iname \
  *gcc*.rpm -o -iname \
  *gcc*.deb
```

which finds both, DEB and RPM packages on a DVD or CD that has been previously mounted under `/media/cdrom`.

To install a package for an rpm-based system, use

```
rpm -Uvh /path/to/package.rpm
```

and for DEB use

```
dpkg -i /path/to/package.deb
```

You may get errors because of missing dependencies. These are libraries and other parts of the software that you need to install in order for the new package to install or run properly. Usually, the error message contains a hint about which package to search for and install first.

Debian has some helper tools like *apt-get* (and GUI variants of *apt-get* like *synaptic*) that resolve and install dependencies automatically if a list of installation resources has been added to `/etc/apt/sources.list`. The details are a little too lengthy for an article, but you'll find information in the manpages.

Thinking about it, you probably have a "personal edition" of Suse/Novell GNU/Linux (not Debian), because the personal edition, as far as I'm aware, comes without a C compiler by default. The compilers should be available on the installation DVD, though, and can be added by the Yast system administration tool if you prefer to work with a GUI instead with a manual installation.

Linux Magazine actually received several letters asking about GCC after the Suse 10 DVD in Issue #62 (January 2006). A subsequent response on the Letters page described how to configure your system to download Suse packages from an online source. See <http://www.linux-magazine.com/issue/63/Letters.pdf>. ■

Send your Linux questions to  
[klaus@linux-magazine.com](mailto:klaus@linux-magazine.com).