



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 want to test the latest Knoppix Live CD, but I'm having problems with the screen resolution.

I have an nVidia graphics card and an Acer screen (resolution 1440x900). When I boot the Live CD, the best resolution I get is 800x600. I have booted with *knoppix fb1440x900* and *knoppix screen = 1440x900* and *knoppix fb1440x900*, but so far I am without success. Also, I have tried *xrandr --output VGA --mode 1440x900* without any luck. Do you have any suggestions?



Newer X.org versions use the *randr* extension, which lets you change screen resolution without restarting the X server, having multi-

ASK KLAUS!



ple screens on different monitors and setting up two or more desktops beside each other.

This extension is automatically active with no additional configuration. In fact, the previous *Modes* configurations in */etc/X11/xorg.conf* is completely ignored. The X server automatically detects monitor resolutions and configures what it thinks are the best resolution settings.

Utilities such as *xrandr* let you change display settings on the fly.

That's the good part.

The bad part is that if anything goes wrong in this automatic setup (for example, incorrect detection of supported screen resolutions), the problem might be more difficult to fix unless you just disable *randr*, set up the old-style mode-lines, and restart the X server.

I'll start with (x)randr:

It could be that the X server detects a maximum resolution that is too small for your graphics card/display.

To find out, enter

```
xrandr -q
```

for all detected resolutions. The output will be similar to Listing 1.

In this example, 1400x1050 is detected as the highest possible resolution and has been chosen as the current value.

If your desired resolution of 1440x900 is not listed, *xrandr* will refuse to set it. If 800x600 is listed as the highest possible mode, X might not reserve enough memory to support higher resolutions. To fix this problem, add the line

```
Virtual 2048 2048
```

to the *Display* subsection of */etc/X11/xorg.conf*. This line will let the system switch to a higher resolution if a new monitor is connected. For some Intel graphics chipsets, virtual screen sizes of modes higher than 2048x2048 will disable DRI, and *Compiz* won't run.

Because you have an nVidia card, the problem might just be that the current *xorg nv* driver cannot handle the 1440x900 resolution, compared with the better supported ATI and Intel card drivers. nVidia still has a restrictive policy about specifications and API descrip-

Listing 1: xrandr -q Output

```
01 Screen 0: minimum 320 x 200, current 1400 x 1050, maximum 1400 x 1200
02 VGA-0 disconnected (normal left inverted right x axis y axis)
03 LVDS connected 1400x1050+0+0 (normal left inverted right x axis y axis) 0mm x 0mm
04 1400x1050 60.8*+
05 1280x1024 59.9
06 1280x800 60.0
07 1280x768 60.0
08 1024x768 60.0 59.9
09 800x600 60.3 59.9
10 640x480 59.9 59.4
11 S-video disconnected (normal left inverted right x axis y axis)
```

tions for their cards, which makes it hard to develop an open source driver that supports all features. Therefore, you might be better off using a proprietary nVidia Linux driver that you can download and install from the nVidia web page.

For Knoppix, if installing the nVidia driver, start in text mode with

```
knoppix 2
```

and run the installation from there. If you have booted from USB flash memory with the persistent Knoppix image feature, your changes will be written to flash memory and will not be lost when restarting the computer. After completing the installation, type

```
init 5
```

to switch to the graphical desktop. The `xrandr` command should now report all resolutions supported by the new driver, and you can switch to native resolution with

```
xrandr --output VGA --mode 1440x900
```

as you tried before. Instead of `--mode`, you might have to use the `-s` flag if the `nv` driver only supports `randr` 1.1 instead of 1.2.

An alternative, if you can get ahold of a modeline for your desired resolution, is to add a new mode via `xrandr`:

```
xrandr --newmode "1440x900" 106.5 1440 1520 1672 1904 900 901 904 932 -HSync +VSync
xrandr --addmode VGA 1440x900
xrandr --output VGA --mode 1440x900
```

Now compare the modeline given in the first `xrandr` command with the specification for your graphics card and monitor and change it accordingly.

The standard `nv` X.org module might or might not accept the new modeline.

Eee PC Keyboard Combinations



How can I turn on/off the card reader, WiFi, webcam on my Eee PC? The keyboard combinations don't seem to work.



Kernels 2.6.28 and up feature special hardware support for the Asus Eee PC and other systems

through the virtual `/sys` directory. By writing to the files shown in Table 1, you can control parts of the computer. To get the current setting, use the `cat` command:

```
cat /sys/devices/platform/eeepc/camera
1
```

In this example, the value returned as `1` indicates that the webcam is powered up and therefore available to programs (provided the `uvcvideo` driver is installed), so you could use

```
mplayer tv://
```

to display a webcam video.

To change the state of a device, use `echo` (as root, because of write access):

```
echo 1 > /sys/class/rfkill/rfkill0/state
```

This command will activate the integrated WiFi card and also turn on the blue LED. The `udev` daemon should now automatically load the WiFi driver (`ath5k`), and Network Manager will start scanning for networks. This technique is not Eee PC-specific; it might also work for other notebooks with newer Atheros WiFi chipsets.



Table 1: Special Hardware Files

File	Controls	Value
<code>/sys/devices/platform/eeepc/camera</code>	Webcam	1=on 0=off
<code>/sys/devices/platform/eeepc/cardr</code>	SD card reader	1=on 0=off
<code>/sys/class/rfkill/rfkill0/state</code>	Wireless LAN	1=on 0=off

The reason the keyboard combinations don't work with some distributions is their default `/etc/acpi/` configuration, which is sometimes set very conservatively, so you might want to add these combinations by monitoring ACPI events during a key press and putting matching scripts for hardware activation in `/etc/acpi`.

USB Flash Drive



I have been experimenting with installing and booting Linux from flash drives. For this purpose, I bought myself a cheap 8GB USB stick on which is written `ZLXV`. The device reports to `/var/log/` as:

```
scsi 3:0:0:0: Direct-Access 2
CTON AP192 1.01 PQ: 0 ANSI: 4
sd 3:0:0:0: [sdb] 15925248 2
512-byte hardware sectors 2
(8154 MB)
```

I was able to repartition this drive (ext2 fs) for Linux using `gparted`, and I successfully installed (and booted) Ubuntu (Intrepid) from it. But then I had some problems (not related to the drive). I attempted to install an older version of Mandrake Linux. During installation, the partitioner program failed, and thereafter, I have not been able to access the flash drive at all. I include the full log below (Listing 2); the kernel drivers cannot read the partition block on the drive, so it must have been corrupted.

If the kernel drivers are unable to pick up `/dev/sdb`, no other program (such as `gparted`) is able to access the drive either. So now my question is: Is there a way to restore a "raw" partition table such that the kernel drivers can pick up `/dev/sdb` and enable the use of `gparted`? I have tried using `dd` without success.



The syslog messages in Listing 2 do not look good. The USB drive might not be able to handle repartitioning (i.e., changes of the first sector). This might not be a problem with the storage itself. Possibly, the integrated controller is not able to handle different partition tables other than the default. Another (equally unlikely) problem might be the connection between the flash stick and the computer that causes transmission errors, or it could be a USB speed set too high. But at this ca-

Listing 2: USB Stick Problems

```

01 Mar 13 16:09:13 M2-laptop kernel: [ 488.512962] usb 2-3: not running at top speed; connect to a high speed hub
02 Mar 13 16:09:13 M2-laptop kernel: [ 488.531979] usb 2-3: configuration #1 chosen from 1 choice
03 Mar 13 16:09:13 M2-laptop kernel: [ 488.535387] scsi4 : SCSI emulation for USB Mass Storage devices
04 Mar 13 16:09:18 M2-laptop kernel: [ 493.544174] scsi 4:0:0:0: Direct-Access    CION      AP192      1.01 PQ: 0 ANSI: 4
05 Mar 13 16:09:18 M2-laptop kernel: [ 493.555151] sd 4:0:0:0: [sdb] 15925248 512-byte hardware sectors (8154 MB)
06 Mar 13 16:09:18 M2-laptop kernel: [ 493.562137] sd 4:0:0:0: [sdb] Write Protect is off
07 Mar 13 16:09:18 M2-laptop kernel: [ 493.612181] sd 4:0:0:0: [sdb] 15925248 512-byte hardware sectors (8154 MB)
08 Mar 13 16:09:18 M2-laptop kernel: [ 493.619180] sd 4:0:0:0: [sdb] Write Protect is off
09 Mar 13 16:09:49 M2-laptop kernel: [ 493.619765]   sdb:<6>usb 2-3: reset full speed USB device using ohci_hcd and address 10
10 Mar 13 16:10:19 M2-laptop kernel: [ 554.544029] usb 2-3: reset full speed USB device using ohci_hcd and address 10
11 Mar 13 16:10:50 M2-laptop kernel: [ 585.280030] usb 2-3: reset full speed USB device using ohci_hcd and address 10
12 Mar 13 16:11:00 M2-laptop kernel: [ 595.700029] usb 2-3: reset full speed USB device using ohci_hcd and address 10
13 Mar 13 16:11:11 M2-laptop kernel: [ 605.944055] usb 2-3: USB disconnect, address 10
14 Mar 13 16:11:11 M2-laptop kernel: [ 605.946247] sd 4:0:0:0: Device offlined - not ready after error recovery
15 Mar 13 16:11:11 M2-laptop kernel: [ 605.946268] sd 4:0:0:0: [sdb] Result: hostbyte=DID_ABORT driverbyte=DRIVER_OK,SUGGEST_OK
16 Mar 13 16:11:11 M2-laptop kernel: [ 605.946339] Dev sdb: unable to read RDB block 8
17 Mar 13 16:11:11 M2-laptop kernel: [ 605.946398]   unable to read partition table
18 Mar 13 16:11:11 M2-laptop kernel: [ 605.946530] sd 4:0:0:0: [sdb] Attached SCSI removable disk
19 Mar 13 16:11:11 M2-laptop kernel: [ 605.946643] sd 4:0:0:0: Attached scsi generic sg2 type 0

```

capacity, the flash drive should be USB 2.0 capable. Usually, flash drives of 16GB and higher seem to have performance problems with write operations.

In any case, if the flash drive locks up during a write procedure, resist the urge to just pull it from the computer. Many flash drives enter a yet unrecoverable mode if they are disconnected during a write. Some card readers have a file under `/sys` that allows you to shut down the reader without physically disconnecting the card:

```
echo 0 > /sys/bus/platform/
devices/eeepc/cardr
```

(Also, see the question below in “Eee PC Keyboard Combinations.”)

For USB flash, you can try `scsi-spin` or `eject` to force an already killed write attempt to really terminate,

```
scsi-spin -d /dev/sda
```

or

```
eject -f /dev/sda
```

provided the flash drive is connected as `/dev/sda`. Both commands might wait until the currently writing process is either complete, or at least until it ceases to lock the device, which takes a long time.



Just watch the output of `dmesg` for timeout or device reset errors.

Now the problem might be how to get the partition table back into its initial state after a failed partitioning attempt. If the flash drive is still shown with its correct size with `fdisk -l /dev/sda`, it could help to repartition it with `fdisk` as root (for the final `ioctl` system call that tells the kernel to physically reread the partition table) and then use one or both of the above commands to tell the kernel to complete all operations on this device and remove it from the system. Even then, flash drives might need some time to reorganize segments internally, so take a flashing/blinking light on the drive as a hint of activity, even if Linux tells you the device is completely synchronized and can be removed.

If you are not afraid of the cost, you can buy the exact same flash drive again and copy its partition table

```
dd if=/dev/sda of=sda.mbr
bs=512 count=1
```

and, after switching out the new drive for the one that has the defective partition table, use the same command with reversed in/out options:

```
dd of=/dev/sda if=sda.mbr
bs=512 count=1
```

Again, make sure you are reading/writing the right device files. Check `dmesg`

after you insert the drive for the detected device file name.

As with other USB stick operations, when you use the `eject` and `scsi-spin` commands, make sure all writing activity is finished before removing the stick.

In general, changing the original partition table might just be impossible for some flash media (because of bad hardware design). In my opinion, this is a warranty case, and you should be able to get a free replacement.

The default partitioning of flash drives, which is usually a single primary FAT16 or FAT32 partition, is suitable for booting Linux. For example, you can use

```
syslinux /dev/sda1
```

to create the boot loader on the first partition. A bootable master boot record is created with

```
ms-sys -s /dev/sda
```

but the same caveat about changing the partition table applies.

Because some USB controllers will still boot from the first partition when they find an empty MBR, you might want to try going with the default empty MBR first. ■

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