

# Inhaltsverzeichnis

- Linuxrechner kopieren** ..... 3
- Vorbereitungen** ..... 3
- Nvidia- Grafikkarte** ..... 5
- Update des Nvidia- Treibers nach einem Kernel- Update** ..... 6



# Linuxrechner kopieren

## Vorbereitungen

Alte Festplatte ausbauen und mittels USB-Interface an den neuen Rechner anschliessen

Rechner mit Live-CD booten

Struktur und Partition-IDs der alten Platte überprüfen

```
fdisk -l /dev/sdb
```

und neue Platte mittels `fdisk` entsprechend partitionieren, dabei die gleichen Partition-IDs wie die der alten Platte verwenden (also z.B. `Linux(83)` und `swap(82)`)

Filesysteme einrichten

```
mkfs.ext4 /dev/sda1  
mkswap /dev/sda2
```

alte Festplatte mounten

```
mkdir /media/alt  
mount /dev/sdb1 /media/alt
```

neue Festplatte mounten

```
mkdir /media/neu  
mount /dev/sda1 /media/neu
```

Daten kopieren mit `tar`

```
( cd /media/alt; tar --one-file-system -cpf - . ) | ( cd /media/neu; tar xf  
- )
```

Natürlich kann dies auch mit „`cp -a -preserve=all`“ erfolgen (`-a = -dpR = -no-dereference -preserve=link,mode,ownership,timestamps -recursive`). (nur wie wäre der genaue Aufruf??)

So, geschafft, die Daten wären drüben

Jetzt kommt das Aufräumen

Zuerst einmal in der frisch kopierten `fstab` die Disk-UUID durch die absoluten Gerätenamen ersetzen, also z.B.

```
/dev/sda1 / ext4 usw.
```

Löschen der `Udev`-Regeln der alten Netzwerkkarte und des optischen Laufwerks sowie die alte `xorg`-

## config

```
rm /etc/udev/rules.d/70-persistent-cd.rules rm /etc/udev/rules.d/70-persistent-net.rules rm /etc/X11/xorg.conf
```

Diese werden beim nächsten Systemstart wieder automatisch neu erzeugt.

In den beiden Grub- Dateien

```
/mnt/boot/grub/menu.lst (bei Grub 1)
/mnt/boot/grub/grub.cfg
```

die Disk-UUID durch den Gerätenamen ersetzen und den Partitionseintrag bei root= anpassen

Now, here's an example of a typical GRUB entry: (hd0,1)

- The brackets are a must; all devices listed in GRUB menu must be enclosed in brackets.
- hd stands for hard disk; alternatively, fd stands for floppy disk, cd stands for CD-ROM etc.
- The first number (integer for geeks) refers to the physical hard drive number; in this case, the first drive, as they are counted from zero up. For example, hd2 refers to the third physical hard drive.
- The second number refers to the partition number of the selected hard drive; again, partitions are counted from zero up. In this case, 1 stands for the second partition.

From here, it is evident that GRUB (menu) does not discriminate between IDE or SCSI drives or primary or logical partitions. The task of deciding which hard drive or partition may boot is left to BIOS and Stage 1.

As you see, the notation is very simple.

Primary partitions are marked from 0 to 3 (hd?,0), (hd?,1), (hd?,2), (hd?,3). Logical partitions in the extended partition are counted from 4 up, regardless of the actual number of primary partitions on the hard disk, e.g. (hd1,7).

Dann proc und dev nach mnt re- mounten

```
mount -t proc none /mnt/proc
mount -o bind /dev /mnt/dev
```

```
chroot /mnt
```

```
grub install /dev/sda
```

```
mit Strg+D ''chroot'' wieder verlassen und den Rechner neu starten
```

Falls das in die Hose geht, am Grub- Prompt mit

```
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-30-generic root=/dev/sda1 mem=256M init S
```

in den Recovery-Mode von Ubuntu gehen und dort über das dortige Menu Grub noch mal installieren

Im Betrieb dann wieder die absoluten Gerätenamen in der fstab durch die Disk-UUIDs ersetzen, die man durch

```
ls -l /dev/disk/by-uuid
```

angezeigt bekommt.

per

```
uname -r
```

noch mal checken, welche Kernel- Version man hat und

Für das Asus P7P55D das Packet `linux-backports-modules-alsa-lucid-generic-pae` (halt passend zum Kernel) installieren,

## Nvidia- Grafikkarte

- 1) Download Newest Nvidia drivers from here
- 2) Open module blacklist as admin

```
gksudo gedit /etc/modprobe.d/blacklist.conf
```

Add these lines and save:

```
blacklist vga16fb
blacklist nouveau
blacklist riva16b
blacklist nvidiafb
blacklist rivatv
```

- 3) Uninstall any previously installed Nvidia drivers:

```
sudo apt-get --purge remove nvidia-*
```

- 4) Reboot your computer

5) When an error message pops up saying that Ubuntu cannot load Nvidia drivers, choose Exit to terminal (Exit to console)

- 6) Login and cd to the directory where you saved your file

- 7)Install drivers

```
sudo sh NVIDIA-Linux-x86_64-195.36.24-pkg2.run
```

- 8)Start GDM

```
sudo service gdm start
```

## Update des Nvidia- Treibers nach einem Kernel- Update

siehe <http://ubuntuforums.org/showthread.php?t=835573>

From:

<http://koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://koehlers.de/wiki/doku.php?id=pc:linuxcopy>

Last update: **2016/04/01 06:22**

