

# Remastering Knoppix

Instant-Linux im Eigenbau

Andreas Granig  
<andreas.granig@gmx.net>

25. April 2003

---

# Wofür Instant-Linux?

## Linux ohne Installation

- als Demo-, Rescue-, Test-Distribution
- auf plattenlosen Terminals

---

# Warum im Eigenbau?

- Neugierde
- Technisches Interesse
- Anpassen auf spezielle Bedürfnisse
  - Größe (BC-CD, 8cm-CD, DVD)
  - Einsatzgebiet (Firewall, Musik-Terminal, Rescue-System, etc.)

---

# Warum Knoppix?

- vorhandenes Grundgerüst
- automatische Hardware-Erkennung
- vorkonfiguriertes komprimiertes Dateisystem

---

## Vorhandene angepaßte Distributionen

- Knoppix Lite (185MB)
- Model\_K (50MB)
- Damn Small Linux (50MB)

---

# Aufbau von Knoppix

El Torito Image mit

- Boot-Image
- Boot-Catalog
- KNOPPIX-Image
- optionalen Infos und Tools

## El Torito Image

```
1  $ tree /cdrom/
2  /cdrom
3  |-- ...
4  |-- index.html
5  '-- KNOPPIX
6     |-- KNOPPIX
7     |-- boot.cat
8     |-- boot.img
9     '-- ...
10 6 directories, 49 files
```

# Boot-Image

```
1  $ tree bootimg/
2  |-- boot.msg
3  |-- f2
4  |-- german.kbd
5  |-- ldlinux.sys
6  |-- logo.16
7  |-- miniroot.gz
8  |-- syslinux.cfg
9  '-- vmlinuz
10
11 0 directories, 8 files
```

---

# Boot-Catalog

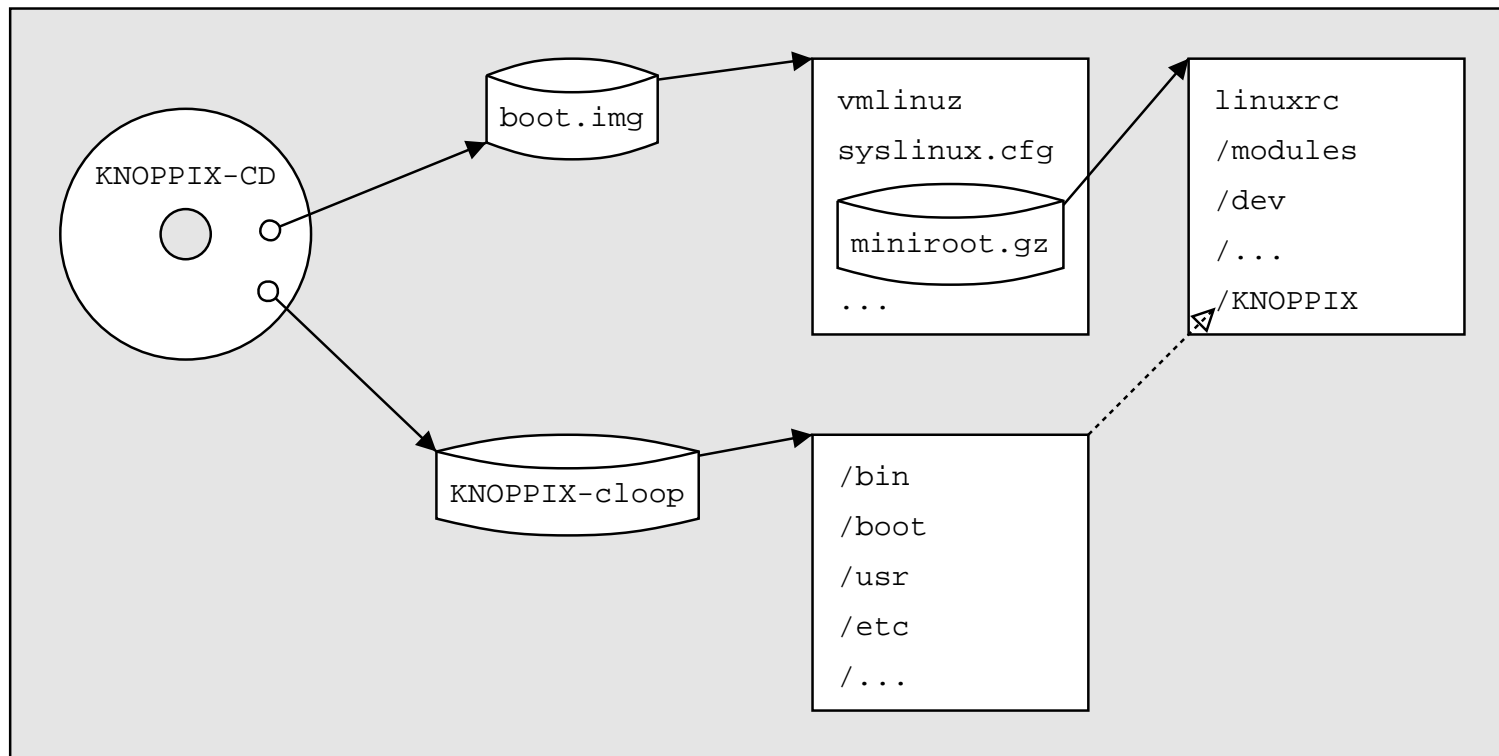
- Liste der bootbaren Images auf der CD
- 2048 Bytes groß

# KNOPPIX-Image

Komprimiertes Dateisystem, enthält schlußendliche Distribution

```
_____ $ ls compressed/ _____  
1 bin  cdrom  etc  home  lib  none  proc  sbin  usr  vmlinuz  
2 boot  dev  floppy  initrd  mnt  opt  root  tmp  var
```

# Gesamtübersicht



---

# Modifizierung der Distribution

1. KNOPPIX-Image dekomprimieren
2. Distribution anpassen
3. KNOPPIX-Image komprimieren
4. Boot-Image anpassen
5. ISO-Image erstellen

## KNOPPIX-Image dekomprimieren

**Möglichkeit 1:** Über `cloop` Kernel-Modul das Image mounten und in ein temp. Verzeichnis kopieren.

*Achtung: Über `cloop` immer von CD mounten, sonst Deadlocks!*

**Möglichkeit 2:** KNOPPIX von CD booten und `/KNOPPIX/*` in ein gemountetes temp. Verzeichnis kopieren.

## Die cloop-Methode

1. cloop runterladen und kompilieren

```
tar xvfz cloop_0.68-2.tar.gz && cd cloop-0.68 &&  
make KERNEL_DIR=/pfad/zu/kernel
```

2. cloop installieren

```
mkdir /lib/modules/misc && cp cloop.o /lib/modules/misc/ &&  
depmod -a && mknod /dev/cloop b 240 0
```

3. KNOPPIX-Image mounten und kopieren

```
mount /cdrom && insmod cloop file=/cdrom/KNOPPIX/KNOPPIX &&  
mount -o ro /dev/cloop /mnt/myknoppix/compressed &&  
cp -Rp /mnt/myknoppix/compressed/* /mnt/myknoppix/master
```

## Die KNOPPIX-Boot-Methode

1. KNOPPIX von CD booten

2. Zielpartition mounten und Distribution kopieren

```
mount /dev/hdXY /mnt/hdXY &&
```

```
cp -Rp /KNOPPIX/* /mnt/hdXY/mnt/myknoppix/master
```

## Distribution anpassen

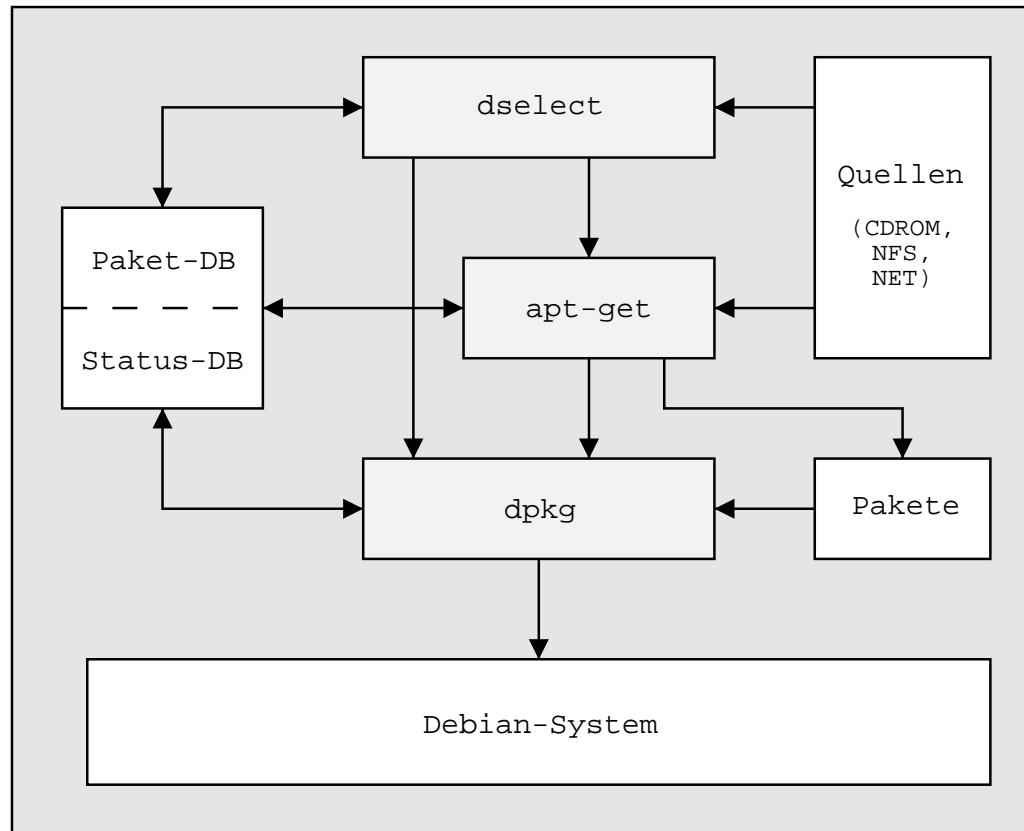
1. Ins Root-Verzeichnis der KNOPPIX-Distribution wechseln

```
chroot /mnt/myknoppix/master && mount -t proc /proc /proc
```

2. Nicht benötigte Software deinstallieren, neue installieren, Konfigurationsänderungen vornehmen:

- /etc/skel - Homeverzeichnis von knoppix
- /etc/inittab - Startup-Konfiguration
- /etc/init.d/knoppix-autoconfig - Hardware/Language-Konfiguration
- /etc/init.d/xsession - X Server Startup-Konfiguration
- /etc/X11/Xsession.d/45xsession - Desktop Startup-Konfiguration

# Paketmanagement



## dpkg

```
$ dpkg -l
```

```
1  ...  
2  ii  a2ps          4.13b-16  GNU a2ps 'Anything to PostScript' converter  
3  ii  aalib-bin      1.4p5-17  sample programs using aalib  
4  ...
```

```
$ dpkg -L foo
```

```
1  /usr/bin/foo  
2  /usr/lib/libfoo.la  
3  /etc/foo.conf
```

## dpkg (2)

```
$ dpkg -s foo
```

```
1 Package: foo
2 Status: install ok installed
3 Priority: optional
4 Section: base
5 Installed-Size: 1234
6 Maintainer: Foo Bar <foo@bar.org>
7 Version: 1.0-1
8 Depends: libc6 (>= 2.2.4-4)
9 Conffiles:
10  /etc/foo.conf c558fc15703d0566bfe74a5798830f81
11 Description:
12  Blah blah
```

## dpkg (3)

```
_____ $ dpkg -S /usr/bin/foo _____
```

```
1 foo: /usr/bin/foo
```

```
_____ $ dpkg -P foo _____
```

```
1 (Reading database ... 108922 files and directories currently installed.)
```

```
2 Removing foo ...
```

```
3 Purging configuration files for foo ...
```

## apt-get

```
apt-get remove --purge <Paket>
```

Entfernt das angegebene und alle davon abhängigen Pakete

```
apt-get install <Paket>
```

Installiert Paket

```
apt-get update
```

Aktualisiert Liste der verfügbaren Pakete von Quellen in `/etc/apt/sources.list`

```
apt-get upgrade
```

Aktualisiert installierte Pakete

## dselect

```

1 Debian 'dselect' package handling frontend.
2
3 * 0. [A]ccess      Choose the access method to use.
4   1. [U]pdate     Update list of available packages, if possible.
5   2. [S]elect     Request which packages you want on your system.
6   3. [I]ninstall  Install and upgrade wanted packages.
7   4. [C]onfig     Configure any packages that are unconfigured.
8   5. [R]emove     Remove unwanted software.
9   6. [Q]uit       Quit dselect.
10
11 ...
```

## dpkg-awk

Nicht im Standard-Knoppix enthalten.

```
$ dpkg-awk "Status: .* installed$" -- Package Installed-Size | \  
  awk '{print $2}' | grep -v "^$" | xargs -n2 echo | \  
  perl -pe 's/(\S+)\s(\S+)/$2 $1/' | sort -rg
```

... gibt alle installierten Pakete sortiert nach Größe aus.

## deborphan

Sucht u.a. Bibliotheken, die von keinem anderen Paket mehr benötigt werden.

```
# deborphan > /tmp/debor.tmp && \  
    dpkg -P `cat /tmp/debor.tmp` && rm /tmp/debor.tmp
```

## Anpassungen bereinigen

- `bash_history`, `tmp` etc. entfernen, `apt-get clean`, `updatedb`  
(`Knoppix.clean`)
- `umount /proc`
- `chroot` verlassen (`Strg+D` oder `exit`)
- `/mnt/myknoppix/master/usr/bin/create_compressed_fs` in ein `PATH`-  
Verzeichnis kopieren
- Ev. Paket-Bibliothek `var/lib/dpkg` aus Distribution verschieben.

## KNOPPIX-Image komprimieren

### Standard-Variante:

```
# mkdir -p /mnt/myknoppix/KNOPPIX
# mkisofs -R -l -V "KNOPPIX iso9660 filesystem" \
  -hide-rr-removed -v /mnt/myknoppix/master | \
  create_compressed_fs - 65536 \
  > /mnt/myknoppix/iso/KNOPPIX/KNOPPIX
```

Benötigt je nach Image-Größe bis zu 700MB RAM/Swap

---

## Image-Performance optimieren

`Knoppix.mkcompressed`:

Ordnet u.a. die Daten zum schnelleren Lesezugriff um, rekonfiguriert Netzwerk-Interfaces etc.

## Boot-Image anpassen

### Boot-Image mounten:

```
# cp /cdrom/KNOPPIX/boot.img /mnt/myknoppix/iso/KNOPPIX/ && \  
    mount -o loop /mnt/myknoppix/iso/KNOPPIX/boot.img /somewhere
```

### Anpassungen vornehmen:

z.B. Bootup-Keymap erzeugen, LILO-Hintergrund oder Boot-Optionen ändern

### Boot-Image unmounten

## Boot-Keymap erzeugen

z.B. Verwenden einer deutschen Mac-USB Tastatur:

Erzeugen durch:

```
# keytab-lilo  
/usr/share/keymaps/i386/qwertz/mac-usb-de-latin1.kmap.gz  
> /somewhere/mac_latin1.kbd
```

In /somewhere/syslinux.cfg:

```
KBDMAP mac_latin1.kbd
```

## LILO-Hintergrund ändern

logo.16, ein lss16 Format-Image

1. Erzeugen eines png-Images mit 640x400x16, z.B. logo.png
2. `# pngtopnm < logo.png > logo.pnm`
3. `# ppmtolss16 < logo.pnm > logo.16`

Durch das 640x400 Format bleibt noch Platz für eine Boot-Meldung vor dem LILO-Prompt, welche in `boot.msg` definiert werden kann.

## Boot-Optionen ändern

Notwendig u.U. bei Kernel- oder Startup-Config-Änderungen.

Beispiel-Eintrag in `/somewhere/syslinux.cfg`:

```
$ cat /somewhere/syslinux.cfg
```

```
1  ...
2  LABEL myconfig
3  KERNEL vmlinuz
4  APPEND ramdisk_size=100000 init=/etc/init lang=de apm=power-off \
5     hda=scsi hdb=scsi hdc=scsi hdd=scsi hde=scsi hdf=scsi hdg=scsi hdh=scsi \
6     vga=791 initrd=miniroot.gz BOOT_IMAGE=knoppix
7  ...
```

## ISO-Image erstellen

- **Dateien überprüfen**

KNOPPIX/KNOPPIX, KNOPPIX/boot.img, KNOPPIX/boot.cat, ev. index.html

- **Image erzeugen**

```
# mkisofs -l -r -J -V "KNOPPIX-MyKnoppix" -hide-rr-moved \  
-v -b KNOPPIX/boot.img -c KNOPPIX/boot.cat \  
-o /mnt/myknoppix/MyKnoppix.iso /mnt/myknoppix/iso
```

- **brennen und testen**

---

## **Herzlichen Dank für Ihre Aufmerksamkeit...**

. . . und viel Spaß und Erfolg mit Ihrem eigenen Live-Linux