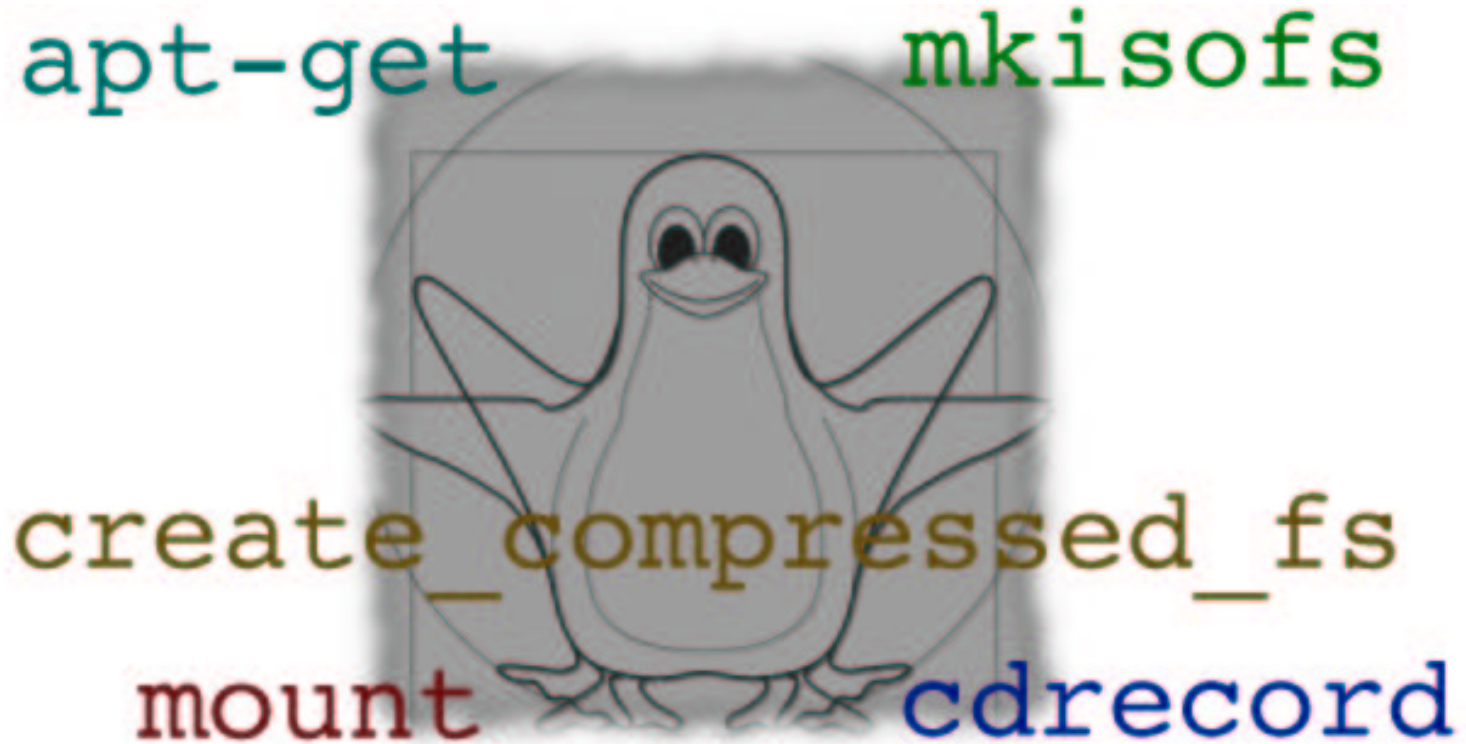


Knoppix-Remastering



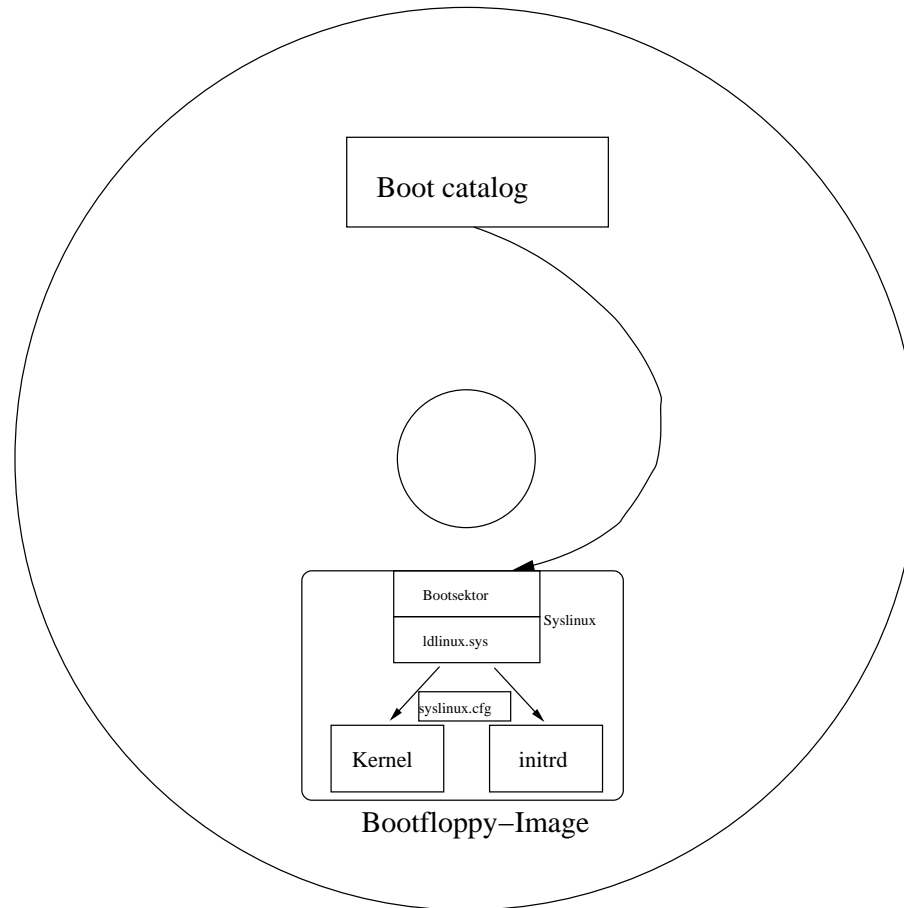
Übersicht

- Wie bootet Knoppix?
- Anpassung durch Autostart:
Optionale Datei `KNOPPIX/knoppix.sh` auf der CD
- Permanente Änderung der Bootoptionen:
Datei `syslinux.cfg` auf dem Bootfloppy-Image
- Remastering des komprimierten Dateisystems `KNOPPIX/KNOPPIX` auf der CD
- Bonus: Anpassung fürs Booten vom USB-Stick

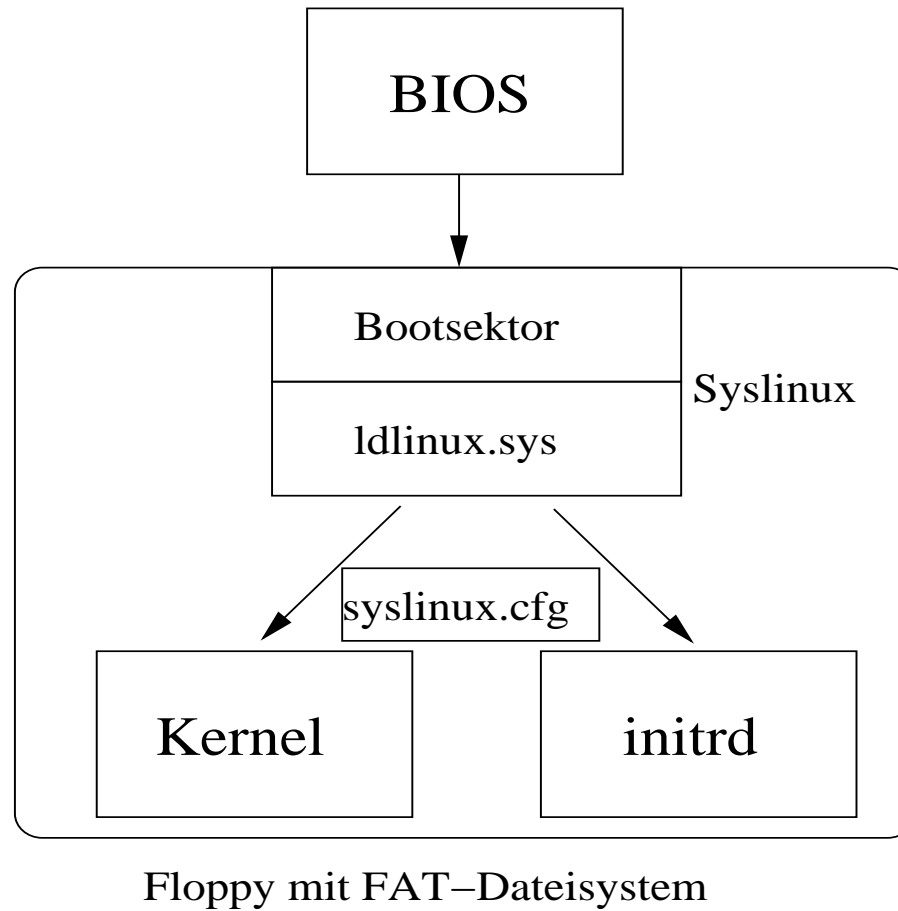
Booten von CD

- El-Torito-Standard (muss vom BIOS unterstützt sein)
- Bootfähiges Floppy-Image (1.44 MB oder 2.88 MB) auf der CD
- Alternative: „no emulation“-Modus, direktes Lesen des **iso9660**-Dateisystems durch den Bootloader
- Bootloader: **LILO** oder **Syslinux** (Floppy-Image), **Isolinux** (no emulation)
- Knoppix verwendet ein 1.44 MB Floppy-Image mit **FAT**-Dateisystem und **Syslinux** (Datei **KNOPPIX/boot.img**)

Booten von CD (2)



Der Bootloader Syslinux



Boot-Details

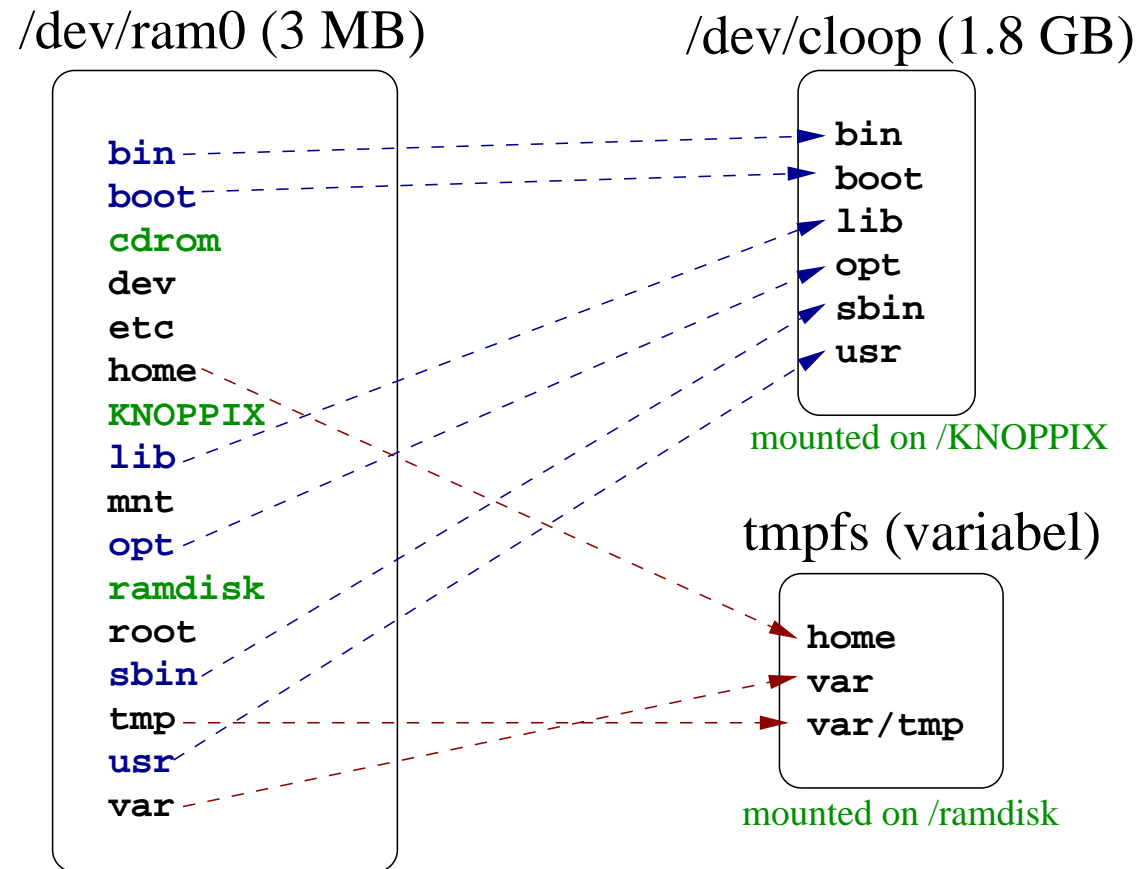
- Bootloader **Sylinux** (Floppy-Image auf CD) lädt Kernel **vmlinuz** und initiale Ramdisk **miniroot.gz**
- Initiale Ramdisk (Startskript **linuxrc**):
 - Automatische Erkennung des Bootmediums (CD) durch Testmount
 - Laden des **cloop**-Moduls, mounten des **cloop**-Dateisystems
 - Erzeugen von Symlinks von der Ramdisk auf den read-only-Bereich
 - *Kein* change root auf komprimiertes Loop-Dateisystem, Rootdateisystem verbleibt in der initialen Ramdisk
- Bereiche, die *nicht* schreibbar sein müssen, liegen per Symlink im komprimierten Loop-Dateisystem (read-only)
- Starten von **init**

Boot-Details

(2)

- Systeminitialisierung durch `knoppix-autoconfig` (Shellskript):
 - Einrichten einer zusätzlichen Ramdisk für `/home` und `/var`
 - PCI-Table-basierte Hardware-Erkennung (`hwsetup`)
 - Starten einiger Dienste (Automounter, DHCP-Client)
 - Automatische X-Serverkonfiguration
 - ggf. einbinden von gespeicherter Konfiguration (`knoppix.sh`) und persistentem Homeverzeichnis
- Default-Runlevel:
 - Starten von Rootshells (ohne Login) durch `init`
 - Starten der X-Session durch `init`
 - Herunterfahren nach Beendigung der X-Session

Knoppix-Dateisystemlayout



Anpassung durch Autostart

Schritte: CD-Inhalt auf Festplatte kopieren (komprimiertes Dateisystem bleibt unverändert), erzeugen der Datei `knoppix.sh` mit Shellkommandos, CD bootfähig remastern.

Beispiel Remote-Login: IP-Adresse für erste Netzwerkkarte auf 192.168.1.20/24 setzen, Default-Route auf 192.168.1.1 setzen, Passwort für User `knoppix` setzen, Secure Shell Server starten.

```
mkdir knx_cd  
mount /cdrom && cp -a /cdrom/* knx_cd  
vi knx_cd/KNOPPIX/knoppix.sh
```

Natürlich gibt es auch andere Editoren außer `vi` :-)

Anpassung durch Autostart

(2)

```
# knoppix.sh -- wird beim Knoppix-Boot ausgefuehrt
# Netz konfigurieren
killall pump ; sleep 1
ifconfig eth0 192.168.1.20 netmask 255.255.255.0
route add default gw 192.168.1.1
# Passwort setzen
echo knoppix:knoppix | chpasswd
# Symlinks fuer sshd anlegen (Runlevel 2 und 5)
(cd /etc/rc2.d ; ln -s ../init.d/ssh S20ssh)
(cd /etc/rc5.d ; ln -s ../init.d/ssh S20ssh)
CD bootfähig remastern:
mkisofs -r -J -b KNOPPIX/boot.img -c KNOPPIX/boot.cat \
-o knx_autossh.iso knx_cd
```

Permanente Änderung der Bootoptionen

Schritte: CD-Inhalt auf Festplatte kopieren (komprimiertes Dateisystem bleibt unverändert), **loop**-mounten des Bootfloppy-Image, ändern der Datei **syslinux.cfg** auf dem Image, CD bootfähig remastern.

Beispiel: Standard-Desktop auf **XFCE** umstellen, Desktop-Auflösung 1280x1024 setzen, permanentes Homeverzeichnis vom USB-Stick einbinden.

```
mkdir knx_cd knx_floppy
mount /cdrom && cp -a /cdrom/* knx_cd
mount -o loop knx_cd/KNOPPIX/boot.img knx_floppy
vi knx_floppy/syslinux.cfg
umount knx_floppy
```

Permanente Änderung der Bootoptionen

(2)

```
DEFAULT vmlinuz
APPEND ramdisk_size=100000 desktop=xfce screen=1280x1024 \
home=scan init=/etc/init lang=de ...
TIMEOUT 300
KBDMAP german.kbd
PROMPT 1
DISPLAY boot.msg
F1 boot.msg
F2 f2
...
CD bootfähig remastern:
mkisofs -r -J -b KNOPPIX/boot.img -c KNOPPIX/boot.cat \
-o knoppixfce.iso knx_cd
```

Remastering des komprimierten Dateisystems (**cloop**-Image)

- Inhalt des komprimierten Dateisystems auf Festplatte kopieren
- in die Kopie mittels **chroot** hineinwechseln
- Änderungen vornehmen:
 - Pakete (de)installieren
 - Konfigurationsdateien ändern
 - Startskripte anpassen. . .
- neues **cloop**-Image erzeugen
- restlichen CD-Inhalt auf Festplatte kopieren
- CD bootfähig remastern

Remastering des komprimierten Dateisystems

(2)

Vorsicht: Es wird *viel* Platz auf der Festplatte benötigt. Außerdem braucht die Erzeugung des **cloop**-Image viel Arbeitsspeicher (ggf. vorher zusätzlichen Swap-Bereich anlegen).

Der Inhalt des **cloop**-Image wird direkt von der gebooteten Knoppix-CD auf die Festplatte kopiert (Beispielpartition `/dev/hda8`):

```
mount /dev/hda8 /mnt/hda8
mkdir /mnt/hda8/knxroot
cd /KNOPPIX ; cp -a * /mnt/hda8/knxroot
```

„Change root“-Umgebung betreten, `/proc`-Dateisystem einbinden:

```
chroot /mnt/hda8/knxroot
mount -t proc none /proc
```

Remastering des komprimierten Dateisystems

(3)

apt-Konfiguration anpassen, Pakete (de)installieren, Konfigurationsdateien ändern, aufräumen, **/proc** unmounten, Umgebung verlassen:

```
vi /etc/apt/apt.conf
apt-get update
apt-get remove --purge openoffice-de-en
apt-get remove --purge ttf-openoffice
apt-get install gkrellm mp3blaster
vi /etc/profile
vi /etc/inittab
apt-get clean ; rm -rf /tmp/* /tmp/.[^.]*
umount /proc
exit
```

Remastering des komprimierten Dateisystems

(4)

Anlegen einer Swap-Datei (optional), Erzeugung des neuen komprimierten Dateisystems:

```
dd if=/dev/zero of=/mnt/hda8/swap.img bs=1M count=512
mkswap /mnt/hda8/swap.img
swapon /mnt/hda8/swap.img
mkisofs -R -hide-rr-moved /mnt/hda8/knxroot | \
create_compressed_fs - 65536 > /mnt/hda8/KNOPPIX
swapoff /mnt/hda8/swap.img
```

Die Kopie des **cloop**-Inhalts kann jetzt gelöscht werden:

```
rm -rf /mnt/hda8/knxroot
```


Remastering des komprimierten Dateisystems

(5)

Restliche Dateien der CD auf Festplatte kopieren:

```
mkdir -p /mnt/hda8/knx_cd/KNOPPIX
cd /cdrom
cp -a $(ls | grep -v KNOPPIX) /mnt/hda8/knx_cd
cd KNOPPIX
cp -a $(ls | grep -v '^KNOPPIX$') /mnt/hda8/knx_cd/KNOPPIX
```

Neues **cloop**-Image in die CD-Struktur verschieben, CD bootfähig remastern:

```
mv /mnt/hda8/KNOPPIX /mnt/hda8/knx_cd/KNOPPIX/
mkisofs -r -J -b KNOPPIX/boot.img -c KNOPPIX/boot.cat \
-o /mnt/hda8/myknoppix.iso /mnt/hda8/knx_cd
```

Häufige Fehler/Probleme

- Vergessenes `/proc`-Dateisystem (→ Inhalt von `/proc` wird mit eingepackt, Partition läuft voll)
- Hardlinks im `cloop`-Dateisystem werden in der Kopie auf Festplatte separate Dateien (→ vergrößerter Platzbedarf)
- Remastering-Partition wurde mit Knoppix-Defaults `nodev,nosuid` gemountet (→ Gerätedateien können von (de)install-Skripten nicht benutzt werden)
- Debian testing/unstable „Pakethorror“ (→ Abhängigkeiten müssen ggf. von Hand aufgelöst werden)

Anpassung fürs Booten vom USB-Stick

Voraussetzungen: **cloop**-Dateisystem passt auf den Stick, BIOS kann von USB-ZIP/Floppy booten, Stick kann USB-ZIP/Floppy emulieren.

- Kernelmodule `usb-uhci.o`, `usb-ohci.o` und `usb-storage.o` in die initiale Ramdisk (`miniroot.gz`) integrieren
- Startskript `linuxrc` anpassen, so dass die USB-Module geladen werden
- Verzögerung (1–2 Sec) vor dem Testmounten einbauen
- USB-Speichermedien werden vom Kernel als virtuelle SCSI-Platten behandelt (→ keine zusätzlichen Gerätedateien nötig)
- **Syslinux** auf dem Stick installieren: `syslinux /dev/sda1`

Anpassung fürs Booten vom USB-Stick

(2)

```
gunzip miniroot.gz
# initiale Ramdisk loop-mounten
mount -o loop miniroot /mnt
# ggf. Platz schaffen
rm /mnt/modules/scsi/{aic7xxx,BusLogic}.o
# neue Module integrieren
cp usb-{uhci,ohci,storage}.o /mnt/modules/scsi
# linuxrc anpassen
(cd /mnt ; patch < /patches/usbboot.diff)
# mit Nullbytes auffüllen (Kompression)
dd if=/dev/zero of=/mnt/dummy ; sync
rm /mnt/dummy ; sync ; umount /mnt
gzip -9 miniroot
```

Anpassung fürs Booten vom USB-Stick

(3)

Zeile 36:

```
seagate.o t128.o tmscsim.o u14-34f.o ultrastor.o wd7000.o \  
a100u2w.o 3w-xxxx.o"  
+USB_MODULES="usb-uhci.o usb-ohci.o usb-storage.o"  
# Misc functions  
mountit(){
```

Zeile 198:

```
fi  
done  
fi  
+loadmodules SCSI $USB_MODULES  
+sleep 2  
# All interactively requested modules should be loaded now.
```

Links

Knoppix:

<http://www.knopper.net/knoppix>

Knoppix-Derivate:

<http://www.knoppix.net/docs/index.php/KnoppixCustomizations>

Syslinux, Isolinux:

<http://syslinux.zytor.com>

El-Torito-Standard:

<http://www.phoenix.com/PlatSS/products/specs.html>