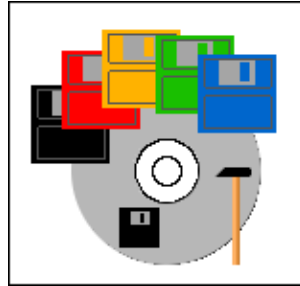


## 8cm-Multiboot-CDROM mit angepasstem Knoppix



by Tjabo Kloppenburg  
([homepage](#))



### *Abstract:*

### *About the author:*

Infiziert wurde ich 1996. Slackware 3.1 :-). Das ist eine Zeit her, aber der Virus hat mich nicht mehr losgelassen. Scripte, GAWK, Perl, PHP, Python -- das ist meine Welt.

Nach der Lektüre von Linuxfocus-Artikeln und anderen Webseiten habe ich nach einigen Versuchen schließlich eine Methode gefunden, um eine bootfähige CDROM zu erstellen, die neben diversen sauberen Bootdisketten (Viren!) auch z.B. ein angepasstes Knoppix booten kann. Als 8cm-CDROM passt die in jede Tasche, und ist dann ein praktischer Helfer in der Not...

---

## Voraussetzungen

Ich setze einfach mal ein paar Kenntnisse voraus. Man lernt ja auch nichts, wenn einfach alles da steht, und dann vielleicht auch noch ohne Fehler... :-)

Man sollte auf jeden Fall einen CD-Brenner haben, der am besten auch CD-RWs brennen kann -- das rentiert sich sehr schnell, da sich immer mal ein kleiner Fehler einschleichen kann. Einen CD-RW Rohling sollte man natürlich haben, 8cm-CDRs und ein Linuxsystem mit ein paar hundert freien MBs.

Später, beim Anpassen von Knoppix (im Grunde Debian), erleichtert eine schnelle Netzanbindung einiges, da die zu installierenden Pakete ja irgendwoher kommen müssen.

Debian-Erfahrungen sind gut und man sollte auch schon mal ein Loopdevice in der Hand gehalten haben...

## Grundlagen

Ähnlich wie beim Booten von einer Diskette oder einer Festplatte wird beim Booten von einer entsprechend präparierten CDROM vom BIOS etwas in den Speicher geladen, und dann dort gestartet. CDROMs waren ursprünglich nicht für das Booten gedacht, und so musste man sich etwas einfallen lassen.

Dazu wurde der El-Torito-Standard vereinbart, der einerseits auf der CDROM eine bestimmte Struktur

vorsieht und andererseits dem BIOS die Rolle zuweist, diese spezielle Struktur von der CDROM in den Speicher zu laden, um dort ein Diskettenlaufwerk zu simulieren. Wenn dort eine Bootdiskette ist, wird diese gebootet. Ganz normal.

So könnten wir also schon eine CDROM erstellen, die eine einzige Diskette bootet -- und das könnte z.B. auch eine 2.88MB-Diskette sein. Bzw. das Image einer solchen Diskette.

Nun, eine einzelne Diskette macht noch kein Multiboot. Hier hilft Boot-Scriptor weiter, ein Bootloader, der auf der CDROM vorliegende Diskettenimages dem BIOS unterjubeln kann. Man bootet also mit Boot-Scriptor (allerdings ohne Disk-Emulation) und kann dann per Menü zwischen vielen Bootimages wählen, die dann mit Emulation gebootet werden.

Diese Bootimages können die verschiedensten Disketten enthalten: diverse Windoze-Versionen, NT-Passwort-Changer, oder auch eine Knoppix-Bootdisk. Dazu später mehr.

## Grundgerüst

Wir müssen irgendwo ein Verzeichnis haben, in dem wir die Verzeichnis-Struktur der CDROM aufbauen können. Ich gehe einfach einmal davon aus, dass wir in "/data" viel Platz haben. Dort legen wir für das Projekt ein Verzeichnis "mboot/" an und darin z.B. "toolcd/" für den späteren CDROM-Inhalt. Ein zusätzliches Verzeichnis "mboot/archive/" kann weitere Dateien aufnehmen, wie z.B. die verwendeten Virens Scanner als Originalarchiv.

## Bootmenü mit Boot-Scriptor

Besorge Dir Boot-Scriptor, und auch die INI-Dateien von dem Server. Die Dokumentation des Programms befindet sich übrigens nur online auf dem Server, nicht im Archiv.

Boot-Scriptor braucht ein eigenes Verzeichnis "bscript/" auf der CDROM, darin den Loader, ggf. eine Grafik, und vor allem eine Datei "bscript.ini".

In dieser Datei kann man ein Bootmenü in einer Sprache ähnlich BASIC programmieren, wobei mittels "onkey" auf Tastendruck hin im Programm verzweigt werden kann. Nach Verzweigung wird dann z.B. per "memdisk" das Image einer Bootdiskette gebootet:

```
print "\ac\c0e--=> Meine MultiBoot Mini-CDROM <---"  
print "F1 Bootdisk1"  
print " w warm reboot"
```

```
MenuLoop:  
onkey f1 goto bootDisk1  
onkey w reboot warm  
goto MenuLoop
```

```
bootDisk1:  
memdisk \images\boot1.img
```

Natürlich gibt es noch eine ganze Reihe weiterer Kommandos. Die findest du in der Doku, bzw. den INI-Dateien von der Boot-Scriptor Homepage.

## Images von Bootdisketten erzeugen

Für einen ersten Test reicht es aus, eine einzelne existierende Bootdiskette mit dem Unixtool "dd" einzulesen. Dazu eignet sich z.B. eine normale Windoze – Bootdiskette, oder eine mit LILO erstellte Bootdiskette:

```
( Diskette einlegen)
# cd /data/mboot/toolcd
# mkdir images          (für die Bootdisketten-Images)
# dd if=/dev/fd0 of=images/boot1.img
```

## ISO-File erstellen und Brennen

Das Programm "mkisofs" muss so aufgerufen werden, dass der Bootloader korrekt eingebunden wird. Der Pfad zum Bootcode ("-b") ist relativ zum Basisverzeichnis der CDRW ("toolcd/"):

```
# cd /data/mboot
# mkisofs -r -o iso.01 -b bscript/loader.bin \
        -no-emul-boot -boot-load-size 4 toolcd
```

Das erzeugte ISO-File muss nun noch auf die CDRW gebrannt werden, die wir ggf. vorher löschen müssen. Dafür gibt es bei "cdrecord" zwei Modi: "blank=fast" und "blank=all". Die "fast"-Variante ist wesentlich schneller und hat bei mir bislang jedes Mal ausgereicht.

Um das ISO-File auf das korrekte Device zu brennen, muss man mit "cdrecord -scanbus" herausfinden, welches Gerät das richtige ist. Wird kein Gerät angezeigt, muss man ggf. mit "modprobe ide-scsi" erst das Modul für den Betrieb von SCSI-Geräten über IDE nachladen.

```
# cdrecord -scanbus
Cdrecord 1.10 (i686-pc-linux-gnu) Copyright (C) 1995-2001 J. Schilling
Linux sg driver version: 3.1.22
Using libscg version 'schily-0.5'
scsibus0:
  0,0,0    0) 'LG          ' 'CD-RW CED-8083B ' '1.05' Removable CD-ROM
  ...
# cdrecord -v dev=0,0,0 speed=32 iso.01
```

Wenn die CDRW nur kleinere Geschwindigkeiten verträgt, wird "cdrecord" automatisch diese kleine Geschwindigkeit verwenden.

Wenn die CD fertig gebrannt ist, sollte man einen Bootversuch wagen.

## Makefile erstellen

Der Aufruf von "mkisofs" hat viele Zeichen, und damit auch viele Chancen für Vertipper. "cdrecord" will auch korrekt aufgerufen werden – es liegt also nahe, ein Bashscript oder ein Makefile zu erstellen, das die korrekten Aufrufe übernimmt. Dabei ist es eine gute Idee, das Makefile im CDRW-Verzeichnis unterzubringen, und nach draußen zu verlinken.

Überhaupt ist es eine gute Idee, von Anfang an alles auf die CDROM mit drauf zu tun, was man für die Erstellung einer neuen CDROM benötigt. Das versetzt einen in die Lage, auch draußen "im Feld" eine neue Version zu zaubern. Und gleichzeitig ist die CDROM ihr eigenes Backup.

Das Makefile kann relativ simpel aufgebaut sein, z.B. so:

```
BASE = toolcd
DEV = 0,0,0
SPEED = 4
VERSION = 01
ISO = iso.$(BASE).$(VERSION)

blank:
    cdrecord -v dev=$(DEV) blank=fast

blankall:
    cdrecord -v dev=$(DEV) blank=all

iso:
    echo "Lösche ~ files:"
    @find $(BASE) -name "*~" -exec rm {} \;
    echo "Erzeuge ISO file:"
    @mkisofs -r -o $(ISO) -b bscript/loader.bin -no-emul-boot -boot-load-size 4 $(BASE)
    @echo
    ls -al $(ISO)

burn:
    cdrecord -v dev=$(DEV) speed=$(SPEED) $(ISO)
```

Die Einrückungen sind TAB-Zeichen.

Das Makefile kann z.B. in "/data/mboot/toolcd/scripte" liegen, und nach "/data/mboot/Makefile" verlinkt werden:

```
# cd /data/mboot
# ln -s toolcd/scripte/Makefile Makefile
```

Das Erstellen einer neuen CDROM nach einer Veränderung im CD-Verzeichnis ist dann eine Sache weniger simpler Kommandos im "mboot/" - Verzeichnis:

```
# make blank
# make iso
# make burn
```

Das ist dann schon ein Hauch von Rapid-Application-Development...

## Weitere Bootdisketten

Weitere Bootdisketten findet man einerseits in seinen eigenen Diskettenboxen und andererseits z.B. im Internet. Natürlich weiß man im Grunde nie ganz genau, was auf so einer Bootdisk aus dem Internet wirklich drauf ist. Ggf. lohnt es sich, eine Suchmaschine nach der Url einer Bootdisk-Sammlung zu befragen, um zu sehen, ob es da mal Probleme gab.

Eine Quelle, von der ich ein paar Bootdisks ausprobiert habe, ist [www.bootdisk.com](http://www.bootdisk.com). Die Disketten dort sind

englische Versionen, zumeist mit CDROM-Support. Praktisch ist auch die "drdfly"-Diskette, eine minimale Bootdiskette. Falls man mal "im Feld" eine Bootdiskette für einen Flasher braucht, kopiert man das Diskettenimage einfach mit Rawrite.exe (DOS) oder dd (Linux) auf eine Diskette. Dann noch den Flasher drauf kopieren, und schon kann man sein BIOS flashen...

Rawrite.exe findet man z.B. auf der ersten CDROM der meisten Linuxdistributionen.

## Andere sinnvolle Inhalte

Wenn man mit der CDROM eine Möglichkeit in der Hand hat, sauber zu booten, liegt der Gedanke nahe, einen Virenschoner mit auf die CDROM zu packen. Ich verwende dafür "F-Prot" von F-Secure. Von dem Programm gibt es eine Linux- sowie DOSversionen, die man frei herunterladen kann.

Dabei ist es etwas lästig, dass man für Updates der Virendefinitionen eine neue CDROM brennen muss. Allerdings soll es neuerdings 8cm-CDRWs geben, die man dann einfach neu beschreiben würde. Alternativ könnte man -- ein entsprechend ausgerüstetes Linux auf der CDROM und die entsprechende Infrastruktur vorausgesetzt -- die Updates auch übers Internet holen. Solch ein Linux -- Knoppix in einer abgespeckten Version -- kommt also nächstes auf unsere CDROM.

## Knoppix integrieren

Wer Knoppix noch nicht kennt, sollte es sich dringend anschauen: Ein von CDROM bootendes, umfangreiches Linux - Desktopsystem. Mit Hardwareerkennung. Und vielen andern tollen Dingen.

Allerdings passt Knoppix nicht auf eine 8cm-CDROM... Praktischer Weise kann man Knoppix jedoch relativ einfach an eigene Bedürfnisse anpassen, ein paar spezielle Debiankenntnisse vorausgesetzt.

Knoppix verwendet einen speziellen Trick, um 2-3 GB Software auf einer 650MB-CDROM unterzubringen: Das komplette Filesystem liegt als komprimierte Datei vor, die beim Booten über einen speziellen Treiber ("cloop.o", "compressed loop device") gemountet wird. Das findet statt in der Datei "boot.img" im Knoppixverzeichnis, bzw. dem Bereich der Knoppix - Bootcdrom, die mit einem Kommando wie

```
# mkisofs ... -b knoppix/boot.img ...
```

erstellt wurde.

Es lohnt sich absolut, die Diskette über ein Loopdevice zu mounten, und den Inhalt zu untersuchen. Unter anderem findet man dort eine Initial Ramdisk (initrd), die besagtes "cloop.o" enthält, und Code zum Laden dieses Kernelmoduls.

Beim Booten wird das in die CDROM "geimpfte" boot.img vom BIOS in ein emuliertes Diskettenlaufwerk geladen, wo die "Diskette" dann bootet. Dabei wird der Kernel von der Diskette geladen und diesem die Initial Ramdisk übergeben. Der Initprozess lädt u.a. das cloop-Modul und durchsucht die angeschlossenen Geräte nach einer Datei "/KNOPPIX/KNOPPIX". Wird ein passendes Gerät gefunden, dann wird versucht, die KNOPPIX-Datei per "cloop" ins Filesystem einzuhängen.

Da es der Diskette in "boot.img" egal ist, ob es von einer echten Floppy oder einer emulierten Floppy startet, können wir es recht einfach in die "bscript.ini" einbinden und Knoppix so auch von unserer Multiboot-CDROM starten.

## Knoppix anpassen

Knoppix mit seinen 650MB ist viel zu groß für unsere handliche 8cm-CDROM. Da ich keine Lust hatte, von Hand eine komplette Knoppix auf ein vertretbares Minimum zu verkleinern, habe ich von der Arbeit anderer profitiert: Die Webseite [Knoppix Customizations](#) hält eine ganze Reihe von modifizierten Knoppix-Versionen bereit, wobei ich mich für "Model\_k" entschieden habe. "Model\_k" ist schön klein (rund 50MB), allerdings fehlt ein GUI und Support für andere Tastaturbelegungen als Englisch. "Damn Small Linux" ist sicher eine gute Alternative.

Die Anleitung, die ich hier gebe, ist eine Ultrakurz-Version der Anleitung von [Sunil Thomas Thonikuzhiyil](#), die ausführlich auf Englisch auf das Remastern von Knoppix eingeht.

Der eigentliche Witz beim Remastern von Knoppix liegt darin, dass Knoppix ein komplettes Linux mit funktionierender Paketverwaltung ist — ein schreibbares Filesystem vorausgesetzt.

Wir booten nun das von uns auserkorene Basis-Knoppix, egal ob von einer eigenen CDROM oder unserer Toolcd, gehen dann so vor:

- Mounten einer Datenpartition,
- Kopieren des unkomprimierten Filesystems in die Datenpartition,
- Wechseln in das unkomprimierte System (chroot),
- Anpassen des Systems,
- Sauberes Verlassen des Systems,
- Erstellen eines neuen komprimierten Filesystems.

Das neue komprimierte Filesystem wird dann in das CD-Verzeichnis nach "/KNOPPIX/KNOPPIX" kopiert, wo es dann beim Booten von der Knoppix – Bootdiskette "gefunden" wird.

Boote nun Knoppix und dann...:

Datenbereich mounten und Filesystem kopieren:

```
# mkdir /1
# mount -t ext2 /dev/hda<n> /1
# cp -Rp /KNOPPIX /1
```

Ins kopierte, unkomprimierte System wechseln:

```
# chroot /1/KNOPPIX
# mount -t proc /proc proc
```

Netzwerk konfigurieren:

```
# (ggf. mit ifconfig eine IP einstellen, wenn keine per DHCP gefunden wurde.)
# (ggf. Nameserver in /etc/resolv.conf einstellen. Dazu den Link löschen.)
```

Pakete (de)–installieren:

```
# apt-get install joe      (Model_k 1.2 hat keinen editor)
# ...
```

Wenn das System fertig konfiguriert ist, muss man es sauber verlassen:

```
# (ggf. den Link der /etc/resolv.conf wieder herstellen)
# umount /proc
# exit
```

Wir hatten gesehen, dass die Datei "/KNOPPIX/KNOPPIX" über "cloop" gemountet wird. Vom "loop"-Device her wissen wir, dass damit das Image eines Filesystems gemountet werden kann. Bei Knoppix wird das Image zusätzlich komprimiert --- wir müssen nun also lediglich mit "mkisofs" ein ISO-Image einer CDROM erstellen und entsprechend komprimieren. Das Tool dazu ist Teil von Knoppix.

Knoppix enthält also auch alles, was man braucht, um ein neues Knoppix zu erstellen.

```
# mkisofs -R /1/KNOPPIX | create_compressed_fs - 65536 > /1/KNOPPIX.2
```

Und nun müssen wir nur noch die neue komprimierte KNOPPIX-Datei auf unsere Mini-CDROM (erst die CD-RW) jonglieren, und ausprobieren...

## Fazit

Wie wir gesehen haben, ist es relativ einfach, selbst eine bootfähige CDROM zu erstellen. Dazu noch eine mit sinnvollem Inhalt. Wir haben dabei nicht jedes Rad neu erfunden, sondern konnten von der Arbeit anderer profitieren.

Ich denke, es ist überhaupt eine gute Idee, die eigenen Sachen so auszustatten, dass wiederum jemand anders davon profitieren kann.

Und wie gesagt, wenn die CDROM ihr eigenes Backup ist, profitieren auch wir selbst davon.

---

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Tjabo Kloppenburg "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: de --&gt; -- : Tjabo Kloppenburg (<a href="#">homepage</a>)</p>
---	---