

# SelfLinux-0.12.3



## cdrecord



Autor: Johnny Graber ([linux@jgraber.ch](mailto:linux@jgraber.ch))  
Formatierung: Axel Gross ([axelgross@web.de](mailto:axelgross@web.de))  
Lizenz: GFDL

CDs kann man unter Linux ganz praktisch über die Kommandozeile brennen. Wie dies geht und was man für Vorbereitungen treffen muss ist Inhalt dieses Textes.

## Inhaltsverzeichnis

### 1 Vorbereitungen

- 1.1 Einbau des Brenners
- 1.2 SCSI-Emulation
  - 1.2.1 Kernelmodul ide-scsi
  - 1.2.2 Neuen Kernel erzeugen
  - 1.2.3 Anpassen der lilo.conf
  - 1.2.4 kleine Tests

### 2 ISO-Image

- 2.1 erstellen
- 2.2 testen

### 3 CD brennen

- 3.1 ISO-Image auf CD brennen
- 3.2 On-The-Fly CD kopieren
- 3.3 Audio-CDs
- 3.4 CD-RW

### 4 Frontends

# 1 Vorbereitungen

## 1.1 Einbau des Brenners

Bevor man mit dem Brennen loslegen kann, muss natürlich ein CD-Brenner vorhanden sein. Im Rahmen von SelfLinux gehen wir aber nicht auf den Einbau ein. Dazu verweisen wir auf die mit dem Gerät mitgelieferte Erklärung.

## 1.2 SCSI-Emulation

Bei den meisten Heimanwendern steht heutzutage ein (E)IDE-Brenner herum. Diese Brenner können nicht direkt verwendet werden, sondern brauchen die SCSI-Emulation.

### 1.2.1 Kernelmodul `ide-scsi`

Diese besteht hauptsächlich im laden des Kernelmodules `ide-scsi` während dem Bootvorgang. Distributionen wie Mandrake, SuSE und RedHat haben dieses Kernelmodul schon vorbereitet und man braucht keinen neuen Kernel zu kompilieren. Dies kann man einfach mit

```
root@linux ~/ # modprobe ide-scsi
```

überprüfen. Mit diesem Befehl, der als root ausgeführt werden muss, wird das Modul getestet und geladen. Wenn dies ohne Fehler gemacht wird, können wir mit der `lilo.conf` weiterfahren. Andernfalls muss man zuerst einen neuen Kernel kompilieren.

### 1.2.2 Neuen Kernel erzeugen

Wie man einen Kernel erstellt, wurde im Rahmen von SelfLinux ja bereits [hier](#) erklärt. Daher beschränkt sich die Erklärung hier nur auf die Besonderheiten im Zusammenhang mit dem CD-Brennen.

Zum Anpassen des Kernels verwenden wir wiederum `make menuconfig`. Zuerst vergewissern wir uns, dass der Kernel Module verwendet. Dazu wechselt man zu "**Loadable modul support**" und aktiviert wenn nötig die Option und kehrt zum Hauptmenü zurück.

Unter **ATA/IDE/MFM/RLL support** geht man nach Aktivierung der Option weiter nach unten. "**Enhanced IDE/MFM/RLL disc/cdrom/ tape/floppy support**" wird auf **Y** gestellt und "**SCSI emulation support**" wird entweder als **M** oder **Y** geladen. Durch **M** wird ein Modul erstellt, das man vor der Verwendung laden muss.

Danach geht es wieder zurück ins Hauptmenü. Bei "**SCSI support**" wählt man die gleichnamige Option mit **Y**, ebenso die Optionen "**SCSI CD-ROM support**" sowie "**SCSI generic support**". Falls man sich nicht ganz sicher ist, kann man auch **M** verwenden.

Als letzte Optionen wählt man unter "**Filesystems**" im Hauptmenü die Optionen "**ISO 9660 CDROM filesystemsupport**" als **M** und "**Microsoft Joliet CDROM extensions**" mit **Y**.

Beim verlassen von `make menuconfig` wird nachgefragt, ob man seine Änderungen sichern will. Dies muss mit "**Yes**" bestätigt werden, da ansonsten die Einstellungen verloren gehen.

Der Kernel wird danach ganz gewöhnlich übersetzt und installiert.

### 1.2.3 Anpassen der lilo.conf

Als nächstes ergänzt man die `lilo.conf`. Bei der Zeile `append=""` trägt man das Argument `hdX=ide-scsi` ein. Dabei muss das `X` durch einen dieser Werte ergänzt werden:

- a CD-Brenner wurde am Port 0 als Master angeschlossen
- b CD-Brenner wurde am Port 0 als Slave angeschlossen
- c CD-Brenner wurde am Port 1 als Master angeschlossen
- d CD-Brenner wurde am Port 1 als Slave angeschlossen

So sähe die `lilo.conf` aus, wenn ihr Brenner am Port 1 als Master angeschlossen wäre:

Auszug der Datei lilo.conf
<pre>image=/boot/vmlinuz-2.4.18-6mdk label=Mandrake_8.2 root=/dev/hda8 initrd=/boot/initrd-2.4.18-6mdk.img  append="hdc=ide-scsi"  vga=788 read-only</pre>

Nach dem Editieren nicht vergessen, `lilo` als root aufzurufen. Beim nächsten Start sollte der Brenner richtig eingerichtet werden.

### 1.2.4 kleine Tests

Wird der Brenner richtig erkannt, wird ein entsprechender Eintrag in den Bootmeldungen auftauchen. Da die Bootmeldungen während dem Start in einer hohen Geschwindigkeit vorbeisaulen, ist dies nicht sehr komfortabel.

```
user@linux ~/ $ dmesg | grep scsi | less

Kernel command line: auto BOOT_IMAGE=Mandrake_8.2 ro root=308 quiet
devfs=mount hdc=ide-scsi
ide_setup: hdc=ide-scsi
scsi0 : SCSI host adapter emulation for IDE ATAPI devices
Attached scsi CD-ROM sr0 at scsi0, channel 0, id 0, lun 0
sr0: scsi3-mmc drive: 24x/24x writer cd/rw xa/form2 cdda tray
```

Gibt man nach dem Einloggen in die Konsole ein. `dmesg | less` zeigt einem die Meldungen an, `grep` filtert diese Ausgabe nach `scsi`.

Man kann aber auch schon `cdrecord` für diesen Test brauchen:

```
user@linux ~/ $ cdrecord --scanbus

Cdrecord 1.11a15 (i586-mandrake-linux-gnu) Copyright (C) 1995-2001 Jörg
```

```
Schilling
Linux sg driver version: 3.1.22
Using libscg version 'schily-0.5'
scsibus0:
 0,0,0      0) 'MATSHITA' 'UJDA710          ' '1.50' Removable
CD-ROM
 0,1,0      1) *
 0,2,0      2) *
```

Wir sehen, das vor dem gefundenen Brenner (MATSHITA) 0,0,0 steht. Diese Werte teilen `cdrecord` mit, welches Gerät er für das Brennen verwenden soll.

## 2 ISO-Image

Für die Bearbeitung von ISO-Images verwendet man das Tool `mkisofs` (make ISO FileSystem). Sollte `mkisofs` nicht schon installiert sein, findet man es im Paket `cdrtools` auf <ftp://ftp.berlios.de/pub/cdrecord/alpha/>

### 2.1 erstellen

Will man eine Daten-CD erstellen, macht man zuerst ein ISO-Image. Will man z.B. sein `/home/user` in ein ISO-Image mit dem Namen `meineDaten.iso` packen, genügt ein:

```
user@linux ~/ $ mkisofs -o /mnt/disk2/meineDaten.iso -R -T -V MeineDaten /home/user/
```

Der Befehl ist ein wenig lang, doch wird damit schön gezeigt, was `mkisofs` alles kann. Mit `-o` teilt man mit, das man das zu erstellende Image in eine Datei ablegen will (output). Wichtig ist, das sich die zu erstellende Datei nicht dort befindet, wo man die Dateien her kopieren will.

Mit `-R` erstellt eine CD vom Type RockRidge, da diese dem Linux-Dateisystem am nächsten kommt. Mit der Option `-T` wird zusätzliche Datei (`TRANS.TBL`) erzeugt. Darin werden die Dateinamen in einer Kurzform abgelegt, so das auch DOS mit der CD klar kommt. Unter Windows ist es üblich, der CD einen Namen zu vergeben, der im Dateimanager angezeigt wird. Dies geschieht mit `-V` und zeigt in unserem Fall "**MeineDateien**" an. Am Ende des Befehls listet man alle Verzeichnisse und Dateien auf, die auf die CD kommen sollen.

Für den täglichen Gebrauch reichen die hier erwähnten Optionen. Will man speziellere CDs brennen, empfiehlt sich wie immer der Blick in die Manualpage (`man mkisofs`).

### 2.2 testen

Wer kennt es nicht: Nach dem Brennen bemerkt man, das die wichtigste Datei in einem anderen Verzeichnis lag und man sie daher nicht hat. Unter Windows bedeutet dies ein neuer Anlauf mit einem neuen Rohling.

Linux ist aber auch hier geschickter und bietet einem eine Testmöglichkeit. Dazu wird das ISO-Image in den Verzeichnisbaum eingehängt/gemountet:

```
user@linux ~/ $ mount -t iso9660 -o loop,ro /mnt/disk2/meineDaten.iso /cdrom
```

Damit wird ein Loop-Device erzeugt und das Image `meineDaten.iso` unter `/cdrom` eingehängt. Man kann so das Image wie eine gewöhnliche CD verwenden.

Bemerkt man einen Fehler, kann man das ISO-Image löschen und ein neues Erstellen - ohne das man für diese Erkenntnis einen Rohling brauchte.

## 3 CD brennen

`cdrecord` ist wie `mkisofs` bei jeder Distribution dabei. Falls nicht, findet man es auch auf [ftp://ftp.berlios.de/pub/cdrecord/](http://ftp.berlios.de/pub/cdrecord/)

### 3.1 ISO-Image auf CD brennen

Der häufigste Fall Daten-CDs zu erstellen, ist das selbst gemachte ISO-Image auf eine CD zu brennen. Dafür genügt ein einfacher Befehl:

```
user@linux ~/ $ cdrecord -v speed=8 dev=0,0,0 /mnt/disk2/meineDaten.iso
```

Mit `-v speed=8` veranlasst man den Brenner, mit **8x** zu schreiben. Man sollte daran denken, die Geschwindigkeit nicht über die Möglichkeiten des Gerätes zu setzen.

Bei `dev=` gibt man die 3 Zahlen an, die einem `cdrecord --scanbus` liefert (siehe ► "kleine Tests" zu Beginn des Textes). Am Schluss gibt man noch das ISO-Image an, das man brennen will.

Es gibt auch bei `cdrecord` noch zahlreiche Optionen:

- `-eject` CD wird nach dem Beschreiben ausgeworfen
- `-dummy` Der Schreibvorgang wird nur simuliert
- `-multi` Für Multisession-CDs immer angeben, ausser bei der letzten Session
- `-msinfo` Zeigt einem Informationen über Multisession-CDs an

### 3.2 On-The-Fly CD kopieren

Bei `cdrecord` gibt es die Möglichkeit, Daten von der Standardeingabe zu lesen. So ist es möglich, auf das Schreiben eines ISO-Images zu verzichten. Damit die Daten ins richtige Format kommen, muss man aber trotzdem `mkisofs` verwenden:

```
user@linux ~/ $ mkisofs -R -T -V MeineCD /cdrom | cdrecord -v speed=8 dev=0,0,0 -
```

Dabei wird die Ausgabe von `mkisofs` nicht in eine Datei geschrieben, sondern durch die Pipe an `cdrecord` weitergereicht. Dies wiederum weiss dank dem `-`, das es von der Standardeingabe lesen muss.

So kann man nicht nur CDs direkt kopieren, sondern auch lokale Dateien. Dafür wird `/cdrom` durch den entsprechenden Pfad ersetzt.

### 3.3 Audio-CDs

Fürs Erstellen von Audio-CDs braucht man die Option `-audio`. Damit kann man `*.wav`-Dateien auf die CD brennen. Zusätzlich braucht man noch `-pad`, das die Audio-Dateien auf ein Vielfaches von 2352 Byte setzt.

```
user@linux ~/ $ cdrecord -v speed=8 dev=0,0,0 -pad -audio lied1.wav lied2.wav
```

Die Lieder werden entsprechend ihrer Auflistung auf die CD gebrannt.

### 3.4 CD-RW

Es ist auch möglich, CD-RWs zu beschreiben. Beschrieben wird wie bisher in den Beispielen genannt. Zum Löschen der CD-RW verwendet man `blank=Methode`. Es gibt momentan folgende Methoden:

<code>help</code>	Zeigt alle Möglichen Methoden an
<code>all</code>	Löscht die gesamte CD, kann also ein wenig dauern
<code>fast</code>	Schnelles Löschen. Dabei werden nur PMA, TOC und das Pregap gelöscht.
<code>track</code>	Löscht einen Track.
<code>unreserve</code>	Gibt einen reservierten Track wieder frei.
<code>trtail</code>	Löscht das Ende eines Tracks.
<code>unclose</code>	Gibt die letzte Session wieder frei.
<code>session</code>	Löscht die letzte Session.

Ein ganze CD-RW löscht man mit diesem Befehl:

```
user@linux ~/ $ cdrecord blank=all
```



## 4 Frontends

Es gibt eine grosse Anzahl von Frontends für `cdrecord`. Wem die Kommandozeile zu kompliziert ist, sollte sich einmal eines der folgenden Programme ansehen. Da sie alle mit `cdrecord` arbeiten, haben sie die gleichen Einschränkungen, aber auch meistens die gleiche Vielfalt. In der nächsten Version von SelfLinux wird es zu einigen Frontends ausführliche Erklärungen geben.

Xcdroast	 <a href="http://www.xcdroast.org/">http://www.xcdroast.org/</a>
BurnIT	 <a href="http://sunsite.dk/BurnIT/">http://sunsite.dk/BurnIT/</a>
Eroaster	 <a href="http://sourceforge.net/projects/eroaster/">http://sourceforge.net/projects/eroaster/</a>
Gnome Toaster	 <a href="http://gnometoaster.rulez.org/">http://gnometoaster.rulez.org/</a>
K3b	 <a href="http://k3b.org/">http://k3b.org/</a>