CD−Writing HOWTO

Table of Contents

Table of Contents

CD−Writing HOWTO

Winfried Trümper <winni@xpilot.org>

v2.8.8, 19 September 1999

This document explains how to write CD−ROMs under Linux.

5.Troubleshooting

6.Credits

Next Previous Contents Next Previous Contents

1. Introduction

Many people use Linux to burn CD−ROMs, because it is reliable and easy. No bluescreens while burning and no headaches about getting the right combination of hard− and software. It just works once properly set up. The CD−writing HOWTO explains the setup, how to put data on the media and gives some interesting applications kindly submitted by the readers.

1.1 Copyright, license and terms of usage

The author disclaims all warranties with regard to this document, including all implied warranties of merchantability and fitness for a certain purpose; in no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use of this document.

Short: read and use at your own risk. No money−back guarantee.

1.2 Contacting me

I get several hundred e−mails concerning the CD−Writing HOWTO each year. So please be patient with me, as I cannot always answer within hours. However, I read everything immediatly and put you on my CDR−queue. Before you ask a question, please make sure you are aware of the the newest version of this document; it is always available from http://www.guug.de/~winni/linux/.

1.3 Suggested readings

The CD−R FAQ is a general FAQ about compact−disk recordables (CD−R), CD−writers and the required software. As most CD−writers can be used to read CD−ROMs, too, you may want to read the *Linux CD−ROM HOWTO*, the *Linux SCSI HOWTO* and the *Linux Kernel HOWTO*.

1.4 Terminology ... lasers at maximum ... fire!

CD−ROM stands for *Compact Disc Read Only Memory*, a storage medium utilizing an optical laser to sense microscopic pits on a silver shimmering disk. The silver shimmering comes from an aluminized layer which is the carrier. The pits represent the bits of the information and are so petite that some billions of them fit on the disc. Thus a CD is a mass−storage medium.

The term *CD−R* is a short form of *CD−ROM recordable* and refers to a CD that doesn't have those "microscopic pits" on its surface. Thus it is empty. Instead of the silver aluminium layer a CD−R has a special film (colored) into which pits can be burned. This is done by giving the laser which normally just senses the pits a little bit more power so it burns the pits. This action can only be taken **once** on a CD−R. Although you can leave out some areas for later writing, creating a so called *multi−session CD*.

The *CD−ROM rewritable* (short: *CD−RW*) was developed to work around the limitation of CD−R media. With a CD−RW burner the laser can not even burn pits into the media, but it can also melt the media back into its original state. This is possible, because the laser does not really burn holes into the media, which would get lost in a puff of smoke. A decent analogy for the technique is an ice−hockey game: by driving over the ice, a players (laser) leave scratches in it. The pattern in the ice (media) is a recording of what happened on the ice during one round. In between the periods of the game, the Zamboni cleaning car drives over the ice and fills the scratches by melting the very top layer of the ice. (Zamboni is *the* brand name for cleaning cars in ice−hockey stadiums). This way the pattern on the ice is cleared and a new round can begin. The scientific term for (evaporating, condensing,) melting and freezing is "phase change", thus the name "phase change devices" for CD−RW−writers.

This HOWTO deals with the task of writing CD−Rs and CD−RWs. Welcome on board, captain.

1.5 Supported CD−writers

You can safely assume that most newer IDE/ATAPI− and SCSI−writers work under Linux. Newer drives are mostly MMC−compliant and therefore supported. If the SCSI−version of a particular writer works, the IDE/ATAPI−version will most likely work and vice versa. However, some people want to get a warm and fuzzy feeling by reading the exact model of their writer in some sort of compatibilty list. That is the reason why I didn't throw the following list out of the HOWTO:

Here is a comprehensive summary of drives reported to work with cdrecord:

Acer:	CDRW 6206A
BTC:	BCE 621E (IDE)
Compro:	CW−7502, CW−7502B
Creative:	RW 4224E, MK 4211
Dysan:	CRW−1622
Elite:	Elite b444.41
Grundig:	CDR 100 IPW
Guillemot:	Maxi CD−R 4X/8X
HP:	SureStore 4020i, SureStore 6020i, C4324, C4325 CD−writer+ 7100, CD−writer+ 7200i, CD−writer+ 7500e, CD−writer+ 8100i, CD−writer+ 8110i,
Hi−Val:	CDD 2242, CDD−3610,
JVC:	XR−W2001, XR−W2010, XR−W2042, R−2626
Kodak:	PCD 200, PCD 225, PCD 260, PCD 600
Matsushita:	CW−7502
Memorex:	CRW−620, CRW−1622, CRW−2224
Microboards:	PlayWrite 2000, PlayWrite 4000RW, PlayWrite 4001RW
MicroNet:	MasterCD Plus 4x4, MasterCD Plus 4x6
Mitsubishi:	CDRW−226
Mitsumi:	CR−2401−TS, CR−2600 TE, CR−2801 TE, CR−4801 TE, CR−4802 TE
Nomai:	680.RW
Olympus:	CDS 615E, CDS 620E
Optima:	DisKovery 650 CD−R
OTI:	CDRW 965, CDRW 975 (Socrates 1.0)
Panasonic:	CW−7502, CW−7582
Philips:	CDD−521/10, CDD−522 CDD−2000, CDD−2600, CDD−3600, CDD−3610 Omniwriter 26, Omniwriter 26A
Plasmon:	CDR 480, CDR 4220, RF−4100, RF−4102, CDR 4400
Plextor:	CDR PX−24 CS, PX−412 C, PX−R412 C PX−R810Ti, PleXwriter 412C

Procom:	PCDR 4
Ricoh:	RO−1420C+, MP 1420C, MP 6200S, MP 6201S, MP 7040A
Samsung:	SW−204
Sanyo:	CRD−R24S
Smart and Friendly:	CD−RW226, CD−R1002, CD−R1002/PRO, CD−R1004, CD−R2004, CD−R2006 PLUS, CD−R2006 PRO, CD−R4000, CD−R4006, CD−R4012
Sony:	CDU 920S, CDU 924, CDU 926S, CDU−928E, CDU 948S, CDRX 100E, CDRX 120E
Taiyo Yuden:	EW−50
TEAC:	CD−R50S, CD−R55S, CDR−55S, CDR−56S−400
Traxdata:	CRW 2260, CDR 4120, CDR 4120 Pro, CDRW 4260,
Turtle Beach:	2040R
WPI (Wearnes):	CDRW−622, CDR−632P
YAMAHA:	CDR−100, CDR 102, CDR−200, CDR−200t, CDR−200tx CDR−400, CDR−400c, CDR−400t, CDR−400tx, CDR−400Atx CRW−2260, CRW−2260t, CRW−4250tx, CRW−4260 t, CRW−4260 tx, CRW−4261, CRW−4416 S

Table: CD−writers supported under Linux

The detailed list of models which have been reported to work or not to work under various Unix−like operating systems is available online from http://www.guug.de:8080/cgi−bin/winni/lsc.pl.

If your hardware isn't supported you can still use Linux to create an image of the CD. You may wish to do so because most burning software for DOS does not deal with RockRidge−extensions. In a second step, you can use DOS or Macintosh software to write the image to the CD−R.

1.6 Supported "features"

There are two classes of utilities for writing CD−Rs: the hardware drivers and the data−formatters. The hardware drivers support the following features:

Supported Feature	cdwrite−2.1	cdrecord−1.6	cdrdao
IDE/ATAPI	no	yes	yes
Parallel Port	no	yes	yes
CD−RW	no	yes	yes
Audio CD	yes	yes	yes
Data CD−ROM	yes	yes	partial
Multisession	partial	yes	no
TAO (Track at once)	yes	yes	yes
DAO (Disk at once)	no	no	yes

Table:

`cdwrite` is unmaintained software referenced only for completeness. Please use `cdrecord` instead, as it supports a wider range of hardware and has significantly more features. The main benefit of cdrdao is the ability to create audio CDs without two seconds of silence between the tracks (writing in disk−at−once mode).

The tools classified as "data−formatters" organize the data on the media ("put a filesystem on it").

Feature	mkisofs	mkhybrid
ISO 9660	yes	yes
RockRidge	yes	yes
El Torito	yes	yes
HFS	no	yes
Joliet	yes	yes
Multisession	yes	yes

Table:

The most obvious difference between the ISO 9660 filesystem compared to the Extended−2 filesystem is: you can't modify files once they are written in ISO 9660. Other limitations of the ISO−9660−filesystem include:

- only 8 levels of sub−directories allowed (counted from the top−level directory of the CD)
- maximum length for filenames: 32 characters
- 650 MB capacity

RockRidge is an extension to allow longer filenames and a deeper directory hierarchy for the ISO−9660 filesystem. When reading a CD−ROM with RockRidge extensions under Linux, all the known properties of files like owner, group, permissions, symbolic links appear (feels like a Unix filesystem). These extensions are not available when reading the CD−ROM under DOS or the Windows−family of operating systems.

El Torito can be used to produce bootable CD−ROMs. In order to use this feature, the BIOS of your PC must support it. Roughly speaking, the first 1.44 (or 2.88 if supported) Mbytes of the CD−ROM contains a floppy−disk image supplied by you. This image is treated like a floppy by the BIOS and booted from. (As a consequence, while booting from this virtual floppy, your original drive A: (/dev/fd0) may not be accessible.)

HFS lets a Macintosh read the CD−ROM as if it were an HFS volume (the native filesystem for MacOS).

Joliet brings long filenames (among other things) to newer variants of Windows (95, 98, NT). However, the author knows of no tool that allows long filenames under plain DOS or Windows 3.11.

Section 2.8 lists the availability of the mentioned software.

1.7 Mailing Lists

If you want to join the development team (with the intention to actively *help* them), send e−mail to cdwrite−request@other.debian.org and put the word `subscribe` in body of the message.

Next Previous ContentsNextPreviousContents

2. Setup the Linux−system for writing CD−ROMs

This section applies to all types of CD−writers, including devices for SCSI, IDE/ATAPI and the parallel port. The later two types of devices require compatibility drivers, which make them appear as if they were real SCSI devices. On the one side such a unifying strategy is easy ("everything is SCSI"), because on the application level you can share your knowledge with other users regardless of their kind of CD−writer. On the other side, you have to reconfigure applications like audio CD players or the mount utility to reflect the change of the driver name. For example, if you accessed your ATAPI CD−writer through the device file /dev/hdc before, you will have to access it through /dev/scd0 after activating the SCSI compatibility drivers.

Once you succeed setting up your hardware and the rest of your Linux−system, the command `cdrecord −scanbus` shows you a list of devices on your SCSI busses. The goal of this section is to guide you in setting up your Linux−system, so that you finally end up seeing something like:

```
shell> cdrecord −scanbus
Cdrecord release 1.7a1 Copyright (C) 1995−1998 Jörg Schilling
scsibus0:
          0) 'Quantum ' 'XP34300         ' 'F76D' Disk
          1) 'SEAGATE ' 'ST11200N        ' '8334' Disk
          2) *
          3) 'TOSHIBA ' 'MK537FB/        ' '6258' Disk
          4) 'WANGTEK ' '5150ES SCSI 36  ' 'ESB6' Removable Tape
          5) 'EXABYTE ' 'EXB−8500−85QUE  ' '0428' Removable Tape
          6) 'TOSHIBA ' 'XM−3401TASUNSLCD' '3593' Removable CD−ROM
          7) *
scsibus1:
        100) 'Quantum ' 'XP31070W        ' 'L912' Disk
        101) *
        102) *
        103) 'TEAC    ' 'CD−R55S         ' '1.0H' Removable CD−ROM
        104) 'MATSHITA' 'CD−R   CW−7502  ' '4.02' Removable CD−ROM
        105) *
        106) 'YAMAHA  ' 'CDR400t         ' '1.0d' Removable CD−ROM
        107) *
```

Listing: Detecting devices on your SCSI bus

The example was provided by Jörg Schilling and shows a total of four CD−writers. Please note that −scanbus also reports other devices, e.g. regular CD−ROMs and hard disk drives. The last column gives the SCSI description of the device, from which you cannot clearly distinguish ordinary CD−ROM drives from those with burning capability. But the product identification (middle column) often has hints about the feature in form of a R, −R or −RW.

2.1 Set up the Linux kernel

The Linux kernel can be equipped with drivers for various features. You can compile the drivers into the kernel image statically or you can compile them as a module for on−demand loading. The last method is preferred for drivers not essential for bringing your Linux−system into life, because your kernel will be smaller and faster then. However, some drivers are essential for the system to come up and you shouldn't compile them as a module. Example: if your system lives on an IDE hard disk, you must have the driver for IDE hard disks in the kernel −− not as a module.

There are three different types of CD−writers: SCSI, IDE/ATAPI and external writers that work through the parallel port. The table shows how to configure the Linux kernel for those hardware types. The first column of the table is the section of the kernel configuration menu, where you can find the setting. The second column is the description of the feature (taken from the kernel configuration menu, too). The third column gives the name of the resulting module. The columns named SCSI, IDE and PP contain the necessary options for the associated hardware (PP = parallel port).

Sect.	Description	Module	SCSI	IDE	PP
BLOCK	Enhanced IDE/MFM/RLL...			Y	
BLOCK	IDE/ATAPI CDROM	ide−cd		M	
BLOCK	SCSI hostadaptor emulation	ide−scsi		M	
BLOCK	Loopback device	loop	M	M	M
PARIDE	Parallel port IDE device	paride			Y/M
PARIDE	Parallel port ATAPI CD−ROMs				M
PARIDE	Parallel port generic ATAPI				M
PARIDE	(select a low−level driver)				Y
SCSI	SCSI support	scsi_mod	Y/M	Y/M	
SCSI	SCSI CD−ROM support	sr_mod	Y/M	Y/M	
SCSI	Enable vendor−specific		Y	Y	
SCSI	SCSI generic support	sg	Y/M	Y/M	
SCSI	(select a low−level driver)		Y		
FS	ISO 9660 CDROM filesystem	iso9660	Y	Y	Y
FS	Microsoft Joliet cdrom...	joliet	M/Y	M/Y	M/Y

Table: driver selection for different writer types

Y stands for yes and means you should put the beast into the kernel. M stands for module and means you should or must compile this feature as a module. Y/M gives you the option to decide between either (order indicates choices with less potential problems). Empty settings don't need to be modified and not changing them increases the chance that the resulting kernel will work for you (if it did before...). Especially in environments where SCSI and ATAPI devices are mixed, you better build most things as modules.

Compiling loopback device is optional. It allows you to test the image before writing it to the media. If you want to be able to read CD−ROMs, you need support for the ISO 9660 filesystem. This driver automatically includes RockRidge Extensions. The Microsoft Joliet CD−ROM extensions have to be explicitely added to the ISO 9660 filesystem. In any case, you need a low−level driver for your hardware. Low−level refers to the driver, which interacts directly with the hardware. For SCSI and the parallel port, there are a lot of low−level drivers available.

To have the modules loaded, you should either list them in a configuration file like /etc/modules or run the daemons kerneld or kmod, which both load the required modules automatically for you when the kernel sees a needs for them. To make your life with modules easier, you can add the following lines to /etc/conf.modules:

```
alias   scd0 sr_mod                  # load sr_mod upon access of scd0
alias   scsi_hostadaptor ide−scsi    # SCSI host adaptor emulation
options ide−cd ignore=hdb            # if /dev/hdb is your CD−writer
```

Some settings for using IDE/ATAPI writers

The aliases provide alternate names for the same module and are not essential. The options provide a way to make options for module loading permanent, i.e. after you have sucessfully used them with modprobe/insmod. To learn about different options for different modules, read on.

```
Description                     command
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
Show active drivers             cat /proc/devices
List active modules             cat /proc/modules
List available modules          modprobe −l
Remove a module                 modprobe −r
Load a module manually          modprobe
Load a module upon startup      echo "list it in /etc/modules"
                                (please consult the docs of your distr.)
Load modules on demand          echo "have kmod or kerneld running"
```

Table: commands to deal with kernel modules and drivers

If you need a module not available on your system, you must recompile and reinstall the kernel. Please read the Kernel HOWTO for more information on this. The most recent information about handling of kernel modules can be read in the Module HOWTO. If you succeed in building or finding the required modules, either add their names to /etc/modules or let an automatic facility (kmod, kerneld) load them for you. Consult the documentation of your Linux−distribution if you are unsure.

If you are on your own, then try to install a package called "Linux Kernel source", which was shipped with your distribution. Afterwards issue the following commands:

```
cd /usr/src/linux
make menuconfig  # follow instructions and menu
make dep
make zImage      # or "make bzImage"
# insert a floppy−disk before proceding
```

```
dd if=arch/i386/boot/zImage of=/dev/fd0
make modules
make modules_install
```

Table: making a bootable floppy disk

This will produce a bootable floppy disk. If you are really clueless about this whole stuff, you better read other documentation like the Kernel HOWTO or get help from a friend before you completely mess up your system. Remember the kernel is the heart of the system.

Special hints up to kernel version 2.2.9

Owners of ATAPI CD−writers beware: if your kernel detects your writer as an ATAPI device during system startup, you have no chance of getting it to work as a CD−writer under Linux. Once the IDE/ATAPI driver grabs the device, the SCSI driver (on the top of the host adaptor emulation) can no longer do so. Please recompile a kernel with the CD−ROM stuff as a module like it is shown in the table above.

With 2.2 and above, you can control the greediness of the ide−cd driver with the parameter "ignore=". If you enable this, the driver won't grab the specified device, so it will be available when the SCSI host adaptor emulation is activated. Example: use "modprobe ide−cd ignore=hdb", if your ATAPI writer is hdb and you want the ide−scsi driver to treat hdb as a scsi device [usually sr0]. There is no way to specify this on the kernel commandline (like in 2.0).

Up to kernel version 2.2.9, don't enable CONFIG_SCSI_MULTI_LUN ('probe for multiple luns') and ide−scsi support at the same time, there is an ide−scsi bug that prevents this.

Some users reported conflicts with pre−compiled binaries and the 2.2−release. This is a problem with the Linux kernel. Possible solutions:

- It may go away if you recompile cdrecord so that it adopts to the actual values in linux/include/scsi/sg.h. These values must be the values of the runing kernel due to the dumb sg user interface.
- It may be caused by a loadable sg driver which cannot get enough memory that is suitable for DMA (Kernel releases up to 2.2.5).

Special hints with kernel version 2.2.10 and later

It was reported to me that kernels from 2.2.10 onwards provide you with both the option from the 2.0.x−series and the option from the earlier 2.2.x−series. The author recommends the strategy of telling the kernel which device should be managed by which driver, like it is done in 2.0.37. The following describes that method:

Special hints for the 2.0.x−series

Kernels prior to 2.0.31 have several bugs and can't read the last file on a CD−ROM. Please upgrade to 2.0.37 or higher. For this version, the feature which used to be named "SCSI emulation" has been more accurately renamed "SCSI host adaptor emulation.

With kernels 2.0.37 and above, you can select which driver (IDE or SCSI) to use for which ATAPI device. To be more precise, you can add the feature "SCSI hostadaptor emulation" to your kernel and activate the emulation for specific devices on the kernel command line (while booting). For example, if your CD−writer has the device file name /dev/hdb, then add the statement `hdb=ide−scsi`. The following listings show how to make this setting permanent through the bootmanagers LILO and chos.

```
image=/boot/zImage−2.0.37
  label=Linux
  read−only
  append="hdb=ide−scsi"
```

Listing: Example configuration for lilo (/etc/lilo.conf)

```
linux "Linux 2.0.37" {
        image=/boot/zImage−2.0.37
        cmdline= root=/dev/hda5 readonly hdb=ide−scsi
}
```

Listing: Example configuration for chos (/etc/chos.conf)

2.2 Hardware and device files

After you have installed all necessary drivers, boot the new kernel. It should have all the features you need for the following sections.

[Users of devfs need not bother with the following steps of manually setting up device files. If you use devfs, everything will automagically appear in the right place.]

Go to the `/dev` directory and check for *loopback devices*. It's not critical if you don't have these devices, but it's convenient if you do (see 3.5). If you already have them, the `ls` command should show `loop0−loop7`:

```
shell> cd /dev
shell> ls loop*
loop0 loop1 loop2 loop3 loop4 loop5 loop6 loop7
```

Listing: device files for loopback

If you don't have those device files, then create them by using the `/dev/MAKEDEV` script:

```
shell> cd /dev/
shell> modprobe loop   # load the module called "loop"
shell> ./MAKEDEV loop
```

Listing: commands to create loopback devices

The last command only succeeds if you have the `loop` module in your kernel. What you read so far about handling of loop device files can be applied to all other device files (sg*, pg*, hd*, ...) and is not explicitely mentioned again in this document.

In case your Linux distribution does not contain the script /dev/MAKEDEV, then create the devices manually with this loop:

```
for i in 0 1 2 3 4 5 6 7
do
        mknod /dev/loop$i c 7 $i
done
```

IDE/ATAPI CD−writers

ATAPI is an extension for IDE devices which is found in all newer IDE devices. The extension allows the SCSI protocol to be used on the IDE bus. Together with the kernel module ide−scsi, which turns your IDE−controller partly into a SCSI−controller, you can treat your IDE/ATAPI devices as if they were SCSI. So just load this module and proceed as if you had a SCSI writer.

SCSI CD−writers

Please make sure that your writer is recognized by the BIOS of your computer. It makes no sense to proceed if your computer does not accept the hardware (the fact that it doesn't spit it out should not be interpreted as a sign of confirmation; a message on the screen is required).

If you plan to connect your SCSI through the parallel port (not to confuse with the IDE drives for the parallel port), you need a special active cable and a special kernel driver. Read http://www.torque.net/parport/parscsi.html to learn more about this option.

Generic SCSI devices

The device files /dev/sd* and /dev/sr* for accessing SCSI hard disks and SCSI CD−ROM drives are limited to block−oriented data transfer. This makes them so fast and well suited for data storage. Driving the laser of a CD−writer is a more complicated task and requires more than block−oriented data transfer. To keep the sd* and sr* devices clean and fast, a seperate type of SCSI device was introduced, the so called *generic SCSI devices*. As everthing can be done to SCSI hardware through the generic devices, they are not fixed to a certain purpose −− therefore the name *generic*.

As true for other devices, you should find them in the /dev directory (by convention):

```
shell> cd /dev
shell> ls sg*
sg0  sg1  sg2  sg3  sg4  sg5  sg6  sg7
```

If you don't have those device files then create them by using the `/dev/MAKEDEV` script:

```
shell> cd /dev/
shell> ./MAKEDEV sg
```

If your Linux distribution does not have the script /dev/MAKEDEV, then create the devices manually with this loop:

```
for i in 0 1 2 3 4 5 6 7
do
        mknod /dev/sg$i c 21 $i
done
```

CD−writers for the parallel port

I have no clue about this, sorry. Please read http://www.torque.net/parport/paride.html or your local file /usr/src/linux/Documentation/paride.txt.

2.3 Get the user software for burning CD−Rs

A more detailed survey of tools related to produce CD−ROMs is available from http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdb.html.

Command line utilities

One of the following packages are required to generate images of CD−Rs (only required for data CD−ROMs):

> ftp://tsx−11.mit.edu/pub/linux/packages/mkisofs/ (mkisofs)
>
> ftp://ftp.ge.ucl.ac.uk/pub/mkhfs (mkhybrid)

To write images to the CD−R, you need one of the following software packages:

> ftp://ftp.fokus.gmd.de/pub/unix/cdrecord/ (cdrecord)
>
> http://www.ping.de/sites/daneb/cdrdao.html (cdrdao)

Don't trust the man page of (old) `mkisofs` which states you need version 1.5 of `cdwrite`. Just use cdrecord and you are fine. Please note that newer versions of cdrecord ship with an enhanced version of mkisofs and some extra tools in the subdirectory misc/ (readcd, isosize) not found elsewhere.

Graphical user interfaces (optional)

Front−ends are really front−ends under Linux. That means, you still have to install the command−line utilities, but you access them in a better looking manner.

X−CD−Roast is a program package dedicated to easy CD creation under Linux. It combines command line tools like `cdrecord` and `mkisofs` into a nice graphical user interface.

> http://www.fh−muenchen.de/home/ze/rz/services/projects/xcdroast/e_overview.html

BurnIT is a JAVA front−end to cdrecord, mksiofs and cdda2wav−0.95 making it a complete package for burning CDs on the Unix platform. It is available from

> http://sunsite.auc.dk/BurnIT/

XDaodio

This is a graphical frontend for cdrdao. Its main purpose is to make 1:1−copies of audio CDs.

> http://www.lrz−muenchen.de/~MarkusTschan/

NextPreviousContentsNextPreviousContents

---

3. Burning CD−Rs

> *"If to smoke you turn I shall not cease to fiddle while you burn."* (Roman emperor Nero about burning his own classic−CDs, AD64. He misunderstood it completely and burned Rome down.)

Writing CD−ROMs consists of two steps under Linux:

- packaging the desired data (files, music or both) into files with special formats
- writing the data from the files to the CD−R with the utility `cdrecord`

This chapter describes the steps for data and audio CDs in greater detail.

3.1 Writing CD−ROMs (pure data)

Note that collecting the data to put onto a CD usually takes longer than one expects. Consider that missing files cannot be added once the CD is written and fixated. Also keep in mind that a certain amount of the free space of a CD is used for storing the information of the ISO−9660−filesystem (usually a few MB). 620 MB data will always fit on a 650 MB CD−R.

Creating an image of the later CD−ROM

Before any storage medium (e.g. floppy disk, hard disk or CD) can be used, it must get a filesystem (DOS speak: get formatted). The filesystem is responsible for organizing and incorporating the files that should be stored on the medium.

The usual utilities for creating filesystems on hard disk partitions write an empty filesystem onto them, which is then mounted and filled with files by the user as they need it. A writable CD is only writable once so if we wrote an empty filesystem to it, it would get formatted and remain completely empty forever. This is also true for rewritable media as you cannot change arbitrary sectors; you must erase their whole content.

So what we need is a tool that creates the filesystem while copying the files to the CD. This tool is called `mkisofs`. A sample usage looks as follows:

```
        mkisofs  −r   −o cd_image   private_collection/
                      `−−−−−−−−−'   `−−−−−−−−−−−−−−−−−'
                           |               |
                write output to   take directory as input
```

The option '−r' sets the permissions of all files to be public readable on the CD and enables RockRidge−extensions. You probably want to use this option unless you really know what you're doing (hint: without '−r' the mount point gets the permissions of `private_collection`!).

`mkisofs` will try to map all filenames to the 8.3 format used by DOS to ensure the highest possible compatibility. In case of naming conflicts (different files have the same 8.3 name), numbers are used in the filenames and information about the chosen filename is printed via STDERR (usually the screen). Don't panic: Under Linux you will never see these 8.3 filenames because Linux makes use of the Rock Ridge extensions which contain the original file information (permissions, filename, etc.).

Now you may wonder why the output of `mkisofs` is not directly sent to the writer device. There are three reasons:

- `mkisofs` knows nothing about driving CD−writers.
- You may want to test the image before burning it.
- On slow machines it would not be reliable (see section 4.).

There is a method to write a CD−R in one go which will be described below.

One also could think of creating an extra partition and writing the image to that partition instead to a file. I vote against such a strategy because if you write to the wrong partition (due to a typo), you can lose your complete Linux system (read: that happened to me...). Furthermore, it is a waste of disk space because the CD−image is temporary data that can be deleted after writing the CD. However, using raw partitions saves you the time for deleting files of 650 MB size.

Test the CD−image

Linux has the ability to mount files as if they were disk partitions. This feature is useful to check that the directory layout and file access permissions of the CD image matches your wishes. Although media is very cheap today, the writing process is still time consuming, and you may at least want to save your time by doing a quick test.

To mount the file cd_image created above on the directory /cdrom, give the command

```
mount −t iso9660 −o ro,loop=/dev/loop0 cd_image /cdrom
```

Now you can inspect the files under /cdrom −− they appear exactly as they were on a real CD. To umount the CD−image, just say umount /cdrom. (Warning: On Linux kernels prior to 2.0.31 the last file on /cdrom may not be fully readable. Please use a more recent kernel like 2.0.36. The option −pad for cdrecord applies to audio CDs only and the option −pad for mkisofs requires a patch, which is as much work to apply than to upgrade to a bug−free Linux kernel.)

Note:

> Some ancient versions of mount are not able to deal with loopback devices. If you have such an old version of mount, then upgrade your Linux−system. Several people have already suggested putting information about how to get the newest mount utilities into this HOWTO. I always refuse this. If your Linux distribution ships with an ancient mount, report it as a bug. If your Linux distribution is not easily upgradable, report it as a bug.
>
> If I include all the information that is necessary to work around bugs in badly designed Linux distributions, this HOWTO would be a lot bigger and harder to read.

Write the CD−image to a CD

Not much more left to do. If you haven't already tried, it's a good time for the command

```
cdrecord −scanbus
```

This will tell you to which SCSI device your CD−writer is attached to. All other methods of guessing the information printed so nicely by cdrecord have been removed from the HOWTO (especially the somewhat dangerous naming scheme for generic SCSI devices).

Before showing you the last command, let me warn you that CD−writers want to be fed with a constant stream of data because they only have small data buffers. So the process of writing the CD image to the CD must not be interupted or a corrupt CD will result. It's easy to interrupt the data stream by deleting a very large file. Example: if you delete a previous image of 650 Mbyte size, the kernel must update information about 650,000 blocks on the harddisk (assuming you have a blocksize of 1 Kbyte for your filesystem). That takes some time and is very likely to slow down disk activity long enough for the data stream to pause for a few seconds. However, reading mail, browsing the web, or even compiling a kernel generally will not affect the writing process (on modern machines).

Please note that no writer can re−position its laser and continue at the original spot on the CD when it gets disturbed. Therefore any strong vibrations or other mechanical shocks will probably destroy the CD you are writing.

When you are mentally prepared, dress up in a black robe, multiply the SCSI−id of the CD−writer with its SCSI−revision and light as many candles, speak two verses of the ASR−FAQ (newsgroup alt.sysadmin.recovery) and finally type:

```
shell> SCSI_BUS=0   # taken from listing 1 "scsibus0:"
shell> SCSI_ID=6    # taken from listing 1 "TOSHIBA XM−3401"
shell> SCSI_LUN=0
shell> cdrecord −v speed=2 dev=$SCSI_BUS,$SCSI_ID,$SCSI_LUN \
                   −data  cd_image

# same as above, but shorter:
shell> cdrecord −v speed=2 dev=0,6,0  −data  cd_image
```

For better readability, the coordinates of the writer are stored in three environment variables with natural names: SCSI_BUS, SCSI_ID, SCSI_LUN. The option −data is not mandatory but is given to make the commandline comparable to the one used for writing audio CDs.

If you use cdrecord to overwrite a CD−RW, you must add the option "blank=..." to erase the old content. Please read the man page to learn more about the various methods to blank the CD−RW.

In times where everybody except me owns a 400 Mhz machine, people feed the output of mkisofs directly into cdrecord:

```
shell> IMG_SIZE=`mkisofs −R −q −print−size private_collection/  2>&1 \
| sed −e "s/.* = //"`
shell> echo $IMG_SIZE
shell> [ "0$IMG_SIZE" −ne 0 ] && mkisofs −r  private_collection/  \
|cdrecord  speed=2  dev=0,6,0
            tsize=${IMG_SIZE}s  −data  −
#       don't forget the s −−^         ^−− read data from STDIN
```

The first command is an empty run to determine the size of the image (you need the mkisofs from the cdrecord distribution for this to work). Maybe your writer does not need to know the size of the image to be written, so you can leave this out. The printed size must be passed as a tsize−parameter to cdrecord (it is stored in the enviroment variable IMG_SIZE). The second command is a sequence of mkisofs and cdrecord, coupled via a pipe.

3.2 Writing audio CDs

Writing audio CDs is very similar to the steps described above for data CDs. There are two main differences. One, the audio CD consists of audio tracks, which are organized as separate images. So if you want to have ten tracks on your CD, you have to make ten images. The other differnce is that the format of the images is not ISO−9660 (or whatever filesystem you prefer), but it is "16 bit stereo samples in PCM coding at 44100 samples/second (44.1 kHz)".

One utility to convert your sound files into the required format is sox. Its usage is straightforward:

```
shell> sox  killing−my−software.wav  killing−my−software.cdr
```

This command would convert the song killing−my−software from the WAV−format into the CDR audio−format. See the man page for sox for more details about formats and filename−extensions sox recognizes. Because the output of the conversion takes up much disk space, it was made a built−in feature of cdrecord for the formats WAV and AU. So as long as your soundfiles have the extensions .wav or .au (and the samplerate "stereo, 16 bit, 44.1 kHz"), you can use them as images without manual conversion.

Cdrecord writes CD images as audio tracks if the option −audio is specified. The other options are identical to those used for writing data−CDs (unless you have very special requirements). These three examples all do the same thing, but read the tracks from different sound file formats:

```
shell> cdrecord −v speed=2 dev=0,6,0  −audio  track1.cdr track2.cdr...
shell> cdrecord −v speed=2 dev=0,6,0  −audio  track1.wav track2.wav...
shell> cdrecord −v speed=2 dev=0,6,0  −audio  track1.au  track2.au...
```

One notable exception are MPEG Layer 3 files, which can be converted to the CD format with the command "mpg123 −s track1.mp3 > track1.cdr". (Warning: this creates a file with a byte order which needs to be reversed by using the option −swab in cdrecord.) Converting the other direction can be done with "8hz−mp3" for WAV−files (extract the track with cdda2wav from the audio CD and encode it into MP3 with 8hz−mp3). To create a CD−R from a whole bunch of MP3−files, you can use the following command sequence:

```
for I in *.mp3
do
    mpg123 −s $I | cdrecord −audio −pad −swab −nofix −
done
cdrecord −fix
```

Depending on the speed of your machine, you may want to slow down writing to "speed=1" (cdrecord−Option). If you use "speed=4", your machine must be able to play the MP3−file at quadruple speed. mpg123 consumes much CPU−time! If you are in doubt, try an empty run with −dummy (keeps the laser switched off). By doing this, you will produce an audio CD which has a 2 seconds of pause between audio tracks.

DAO

If you want to get rid of the pauses between the audio tracks, you have to use disk−at−once (DAO) recording versus the (individual) track−at−once (TAO) recording described above. Support for DAO is currently most advanced in cdrdao. Please see its homepage for details.

Using the parameter read−cd, you are even able to produce 1:1 copies of audio CDs.

3.3 Mixed mode CD−ROMs

There is not much to say about this topic. Just indicate the type of the (subse quent) images with the options −data and −audio. Example:

```
        cdrecord  −v dev=0,6,0  −data  cd_image  −audio  track*.cdr
```

NextPreviousContentsNextPreviousContents

4. Dear Winfried,...

This is the section usually known as "frequently asked questions with answers". If you have a problem with your partner, kids or dog, just send it in, as long as it is related to writing CD−Rs or is otherwise entertaining.

4.1 How sensitive is the burning process?

Test it. Use the option −dummy to do an empty run of cdrecord. Do everything you would do otherwise and watch if the burning process survives.

If you feed cdrecord directly from mkisofs, then disk intensive processes such as updating the *locate* database lower the maximum flow rate and may corrupt the CD. You better check such processes are not started via `cron`, `at` or `anacron` while you burn CD−Rs.

4.2 Has file fragmentation a bad impact on the throughput?

Fragmentation of files is usually so low that its impact isn't noticed. However, you can easily construct pathological cases of fragmentation, which lower the throughput of your hard disks under 100 kbyte/second. So don't do that. :−) Yes, files on a hard disk get fragmented over the years. The faster, the fuller the filesystem is. Always leave 10% or 20% free space, and you should run fine with respect to writing CD−Rs.

If you're uncertain then look at the messages printed while booting. The percentage of fragmentation is reported while checking the filesystems. You can check for this value with the very dangerous command

```
shell> e2fsck −n  /dev/sda5        # '−n' is important!
[stuff deleted −− ignore any errors]
/dev/sda5: 73/12288 files (12.3% non−contiguous)
```

In this example the fragmentation seems to be very high −− but there are only 73 very small files on the filesystem. So the value is *not* alarming.

There is an experimental utility called e2defrag to defragment extended−2 filesystems. The current version does not work reliable enough to use it even for private environments. If you really want to defragment your filesystem, make a backup copy (better: two copies), practise restoring the data, then create a new filesystem (that will destroy the old) and restore the data. This sketch is currently the safest technique.

4.3 Is it possible to store the CD−image on an UMSDOS−filesystem?

Yes. The only filesystem that isn't reliable and fast enough for writing CD−ROMs from is the *network filesystem* (*NFS*). I used UMSDOS myself to share the disk−space between Linux and DOS/Win on a PC (486/66) dedicated for writing CD−ROMs.

4.4 Isn't there some way to get around the ISO−9660 limitations?

Yes. You can put any filesystem you like on the CD. But other operating systems than Linux won't be able to deal with this CD. Here goes the recipe:

- Create an empty file of 650MB size.

```
dd if=/dev/zero of="empty_file" bs=1024k count=650
```

- Create an extended−2 filesystem on this file

```
shell> /sbin/mke2fs  -b 2048  empty_file
empty_file is not a block special device.
Proceed anyway? (y,n) y
```

- Mount this empty file through the loopback devices (you need a reasonable new mount for this; read above).

```
mount −t ext2 −o loop=/dev/loop1 empty_file /mnt
```

- Copy files to /mnt and umount it afterwards.
- Use `cdrecord` on empty_file (which is no longer empty) as if it were an ISO−9660−image.

If you want to make an entry in `/etc/fstab` for such a CD, then disable the checking of the device file on system startup. For example:

```
/dev/cdrom  /cdrom  ext2  defaults,ro  0 0
```

The first 0 means "don't include in dumps", the second (=important) one means "don't check for errors on startup" (fsck will fail to check the CD for errors).

4.5 How to read the tracks from audio CDs?

There are several software packages available. The newest one is "cdpranoia" and can be downloaded from

http://www.xiph.org/paranoia/

Or you want to try the combination of "cdda2wav" and "sox", available from sunsite and its mirrors:

ftp://sunsite.unc.edu/pub/Linux/apps/sound/cdrom/cdda2wav0.71.src.tar.gz

ftp://sunsite.unc.edu/pub/Linux/apps/sound/convert/sox−11gamma−cb3.tar.gz

`cdda2wav` enables you to get a specific interval (or a whole track) from your audio CD and converts it into a .wav−file. `sox` converts the WAV files back into the (audio CD) cdda−format so it can be written to the CD−R using `cdrecord`. You don't necessarily need `sox` if you use a recent version of `cdrecord`, because it has built−in support for .au and .wav files.

4.6 How to probe for SCSI devices after boot?

The file `drivers/scsi/scsi.c` contains the information

```
/*
 * Usage: echo "scsi add−single−device 0 1 2 3" >/proc/scsi/scsi
 * with  "0 1 2 3" replaced by your "Host Channel Id Lun".
 * Consider this feature BETA.
 *     CAUTION: This is not for hotplugging your peripherals. As
 *     SCSI was not designed for this you could damage your
 *     hardware !
 * However perhaps it is legal to switch on an
 * already connected device. It is perhaps not
 * guaranteed this device doesn't corrupt an ongoing data transfer.
 */
```

Please note that this should only be used if your add SCSI devices to the end of the chain. Inserting new SCSI devices into an existing chain disturbs the naming of devices (directory /dev) and may destroy the complete content of your hard disk.

4.7 Is it possible to make a 1:1 copy of a data CD?

Yes. But you should be aware of the fact that any errors while reading the original (due to dust or scratches) will result in a defective copy. Please note that both methods will fail on audio CDs! You have to use cdrdao or cdda2wav on audio CDs.

First case: you have a CD−writer and a separate CD−ROM drive. By issuing the command

```
cdrecord −v dev=0,6,0 speed=2 −isosize /dev/scd0
```

you read the data stream from the CD−ROM drive attached as `/dev/scd0` and write it directly to the CD−writer.

Second case: you don't have a separate CD−ROM drive. In this case you have to use the CD−writer to read out the CD−ROM first:

```
dd if=/dev/scd0 of=cdimage
```

This command reads the content of the CD−ROM from the device /dev/scd0 and writes it into the file "cdimage". The contents of this file are equivalent to what `mkisofs` produces, so you can proceede as described earlier in this document (which is to take the file cdimage as input for cdrecord). If you want to see a progress−meter and other fancy stuff, then you can also use Jörg Schillings sdd.

In case you run into errors, then install a recent version of cdrecord, which ships a tool called "readcd" (found under misc/). It gives you the same result as dd, but reads sectors on the CD−ROM several times in case of errors.

4.8 Can Linux read Joliet CD−ROMs?

Yes. Newer Kernels (2.0.36 and the upcoming 2.2) have built−in support for the joliet format. Remember you have to use both options in your /etc/fstab: the keywords iso9660 and joliet (later is really an extension). For more details, see http://www−plateau.cs.berkeley.edu/people/chaffee/joliet.html.

4.9 How do I read/mount CD−ROMs with the CD−writer?

Just as you do with regular CD−ROM drives. No tricks at all. Note that you have to use the scd devices (SCSI CD−ROM) to mount CD−ROMs for reading, even if you have an ATAPI CD−ROM (remember you configured your ATAPI devices to act like SCSI). Example entry for /etc/fstab:

```
/dev/scd0  /cdrom  iso9660  ro,user,noauto  0  0
```

4.10 How to put even more data on the CD−R?

Use bzip2 instead of any other compressor like gzip or pkzip. It will save you up to 30% of disk−space for larger (>100kb) files. You can download it from

http://www.muraroa.demon.co.uk/

Instead of writing a true audio CD, you can optionally convert your WAV audio files into MP3 audio files and store them on a ISO−9660 filesystem as regular files. Usually MPEG III give you a compression of 1:10. Of course, most CD−players are not able to read files... this is the drawback. On the other hand, why not running the music for your next party from hard disk? 18 Gbyte are enough for 3000−4000 titles. :−)

A software MPEG III−encoder is available from

http://www.stud.ifi.uio.no/~larsi/other/8hz−mp3−cheng.tar.gz

A MPEG III−player is available from

> http://homepages.uni−tuebingen.de/student/michael.hipp/mpg123/

For recorded speech, you may want to try to reduce its size using `shorten` or "GSM lossy speech compression":

> ftp://svr−ftp.eng.cam.ac.uk/pub/comp.speech/
>
> http://kbs.cs.tu−berlin.de/~jutta/toast.html

4.11 How to make bootable CD−ROMs?

You must have an 1.44 MB bootable floppy−disk. Create an exact image of this floppy−disk by issuing the command

```
dd if=/dev/fd0 of=boot.img bs=18k
```

Place this floppy image into the directory holding the collection of your files (or into a subdirectory of it, just as you like). Tell mkisofs about this file with the option '−b' and also use '−c'. For details read the file README.eltorito in the mkisofs−distribution.

An interesting application for a custom bootable CD is as a virus safe DOS− or Windows−system. It saves you the money for the hard disks (if you have a network and use samba to put the user−data on a fileserver). However, this is purely theoretical as nobody reported an actual recipe to me.

Some details about the bootable RedHat CD−ROM is available from http://members.bellatlantic.net/~smithrod/rhjol−technical.html.

4.12 How to make CD−ROMs writable somehow?

There is an *overlay filesystem* available for Linux, which is mounted over the CD−ROM and intercepts all writing operations. New and modified files are stored elsewhere, but for the user it looks like the CD−ROM is modified. For more information, see http://home.att.net/~artnaseef/ovlfs/ovlfs.html.

If that is not enough for your needs: wait for the UDF−filesystem to be supported by Linux or help developing it (see http://trylinux.com/projects/udf/.

4.13 Is it possible to use several writers at once?

Yes. However, it has been only tested with two writers yet. You need either a recent version of the Linux kernel (2.2.10 as of writing) or a a kernel patch for more buffers in the SCSI generic driver (ftp://ftp.fokus.gmd.de/pub/unix/cdrecord/alpha; works up to 2.2.5).

4.14 Which media is the best?

The german computer magazine "c't" had a list of tips regarding the blank CD−Rs in their november 1996 issue:

- "No name" discs are generally not of highest quality and should better not be used.
- If a recordable CD is defective, this is likely to apply to the whole batch (if you bought more than one at a time); maybe you are lucky and can at least use the first 500MB of such CD−Rs ...
- Don't touch the CD−Rs at their shimmering side before writing.

4.15 What about Solaris, *BSD, AIX, HP−UX, etc.?

Only chapter 2 is Linux−specific. You can apply chapter 3 and 4 even if you run another family of operating systems than Linux. Please see the files README.NetBSD, README.aix, README.hpux, README.next, README.solaris, README.sunos, README.vms or README.xxxBSD from the cdrecord−distribution.

4.16 Where to store the local configuration permanently?

You have two options. Either you use the built−in configuration file for cdrecord, or you use a shell wrapper like the one shown below. This shell script reads a configuration file, which lists the options and parameters for cdrecord line by line. The names are exactly the same as on the commandline, but without the leading dash. Comments are allowed. Example:

```
# be verbose
v
# set the speed of the writer
speed=2
# the device−coordinates in the form BUS,ID,LUN
dev=0,6,0
```

The configuration files for the wrapper belong into /etc/cdrecord and must be referenced on the commandline. Example: if you want to refer to the configuration /etc/cdrecord/mywriter.cfg, then you can issue the command "cdrecord.sh mywriter.cfg −audio track1...". Everthing after mywrite.cfg is passed to cdrecord.

```
#! /bin/bash

CFGDIR="/etc/cdrecord"

CFG="$1"
shift
ARGS_LEFT="$@"

if [ ! −f "$CFGDIR/$CFG" ]
then
    echo "Configuration file $CFGDIR/$CFG not found. Exiting."
    exit 1
fi

while read LINE
do
    case $LINE in
        \#*|"") continue;;
    esac
    old_IFS="$IFS"
    IFS="$IFS="
    set −− $LINE
    IFS="$old_IFS"
    O_NAME="$1"
    O_VALUE=""
    while shift
    do
        case $1 in
            "") continue;;
        esac
        O_VALUE="$1"
    done

    if [ −z "$O_VALUE" ]
    then
        O_CDRECORD="$O_CDRECORD −$O_NAME "
        continue
    fi
    O_CDRECORD="$O_CDRECORD $O_NAME=$O_VALUE "

done < "$CFGDIR/$CFG"

set −x    #DEBUG
exec cdrecord $O_CDRECORD $ARGS_LEFT
echo "Execution of cdrecord failed."
```

4.17 How can the CD−info be retrieved?

Somewhere behind the first 32 k on the CD, a block with information about the CD is located. You can extract the information with the following shell script:

```
#! /bin/bash

RD=/dev/cdrom
for i in 32768,7   32776,32 32808,32 32958,128 33086,128 33214,128 \
         33342,128 33470,32 33581,16 33598,16  33615,16  33632,16
do
    old_IFS="$IFS"
    IFS=","
    set −− $i
    IFS="$old_IFS"
    OFFSET=$1
    LENGTH=$2
    echo "*`dd if=$RD bs=1 skip=$OFFSET count=$LENGTH   2> /dev/null`#"
done
```

4.18 What about re−writing

When overwriting CD−RW media, specify the parameter blank=fast to cdrecord. That's all. See the man page of cdrecord for details about this parameter.

4.19 How to create a multi−session CD?

First of all, the image for a multi−session CD must be formatted using the ISO−9660 filesystem using RockRidge−extensions. And you must use the option −multi for cdrecord as long as you want to add further sessions. So at least for the first session, you must use −multi.

The images for the second and subsequent sessions are a little bit more complicated to generate. Mkisofs must know where the free space on the CD−R begins. That information can be gathered by using the option −msinfo on cdrecord (see example below).

```
shell> NEXT_TRACK=`cdrecord −msinfo dev=0,6,0`
shell> echo $NEXT_TRACK
shell> mkisofs −R −o cd_image2 −C $NEXT_TRACK −M /dev/scd5
         private_collection/ \
```

For more information, please read the file README.multi, which is distributed with cdrecord.

NextPreviousContentsNextPreviousContents

5. Troubleshooting

Always remember, that you can still use corrupt CD−ROMs as coasters. :−)

5.1 It doesn't work: under Linux

Please check first, that the writer works under the software it is shipped with (=under another operating system). Concretely:

- Does the controller recognize the writer as a SCSI device?
- Does the driver software recognize the writer?
- Is it possible to make a CD using the accompanied software?

If "it doesn't even work" with the accompanied software you have a hardware conflict or defective hardware. If it works and you use loadlin to boot Linux, then that is your problem. Loadlin makes a warm−boot with most of the hardware already initialized and that can confuse the Linux kernel.

5.2 Error−message: No read access for 'dev=0,6,0'.

Under Linux, some versions of the C−library are incompatible (buggy), so that an application linked against one version will not work with another. An example for an error triggered by pre−compiled binaries is the following:

```
[root@Blue /dev]# cdrecord −eject dev=0,6,0
cdrecord: No such file or directory. No read access for 'dev=0,6,0'.
```

5.3 It doesn't work: under DOS and friends

Try to use Linux. Installation and configuration of SCSI drivers for DOS is the hell. Linux is too complicated? Ha!

5.4 SCSI errors during the burning phase

Most likely those errors are caused by

- missing dis−/reconnect feature on the SCSI bus
- insufficiently cooled hardware
- defective hardware (should be detected by 5.1.)

Under various circumstances SCSI devices dis− and reconnect themselves (electronically) from the SCSI bus. If this feature is not available (check controller and kernel parameters) some writers run into trouble during burning or fixating the CD−R.

Especially the NCR 53c7,8xx SCSI driver has the feature disabled by default, so you might want to check it first:

```
NCR53c7,8xx SCSI support                   [N/y/m/?] y
   always negotiate synchronous transfers [N/y/?] (NEW) n
   allow FAST−SCSI [10MHz]                [N/y/?] (NEW) y
   allow DISCONNECT                       [N/y/?] (NEW) y
```

NextPreviousContents Next PreviousContents

6. Credits

Many thanks go to the readers of this HOWTO, who contributed actively to its contents. As I don't had access to a CD−writer myself for several years, reports about real live setups and experience have always been of great value to me.

Doug Alcorn <doug@lathi.net>

helped to improve the handling of newer kernels

Kalle Andersson <kalle@sslug.dk>

How to write audio CDs directly from MP3.

Alan Brown <alan@manawatu.net.nz>

Rick Cochran <rick@msc.cornell.edu>

hint about dis−/reconnect disabled by default in the ncr driver

Robert Doolittle <bob.doolittle@sun.com>

good arguments for dropping cdwrite from the HOWTO

Markus Dickebohm <m.dickebohm@uni−koeln.de>

Thomas Duffy <tduffy@sgi.com>

major cleanup of syntax and spelling

Jos van Geffen <jos@tnj.phys.tue.nl>

noted the problem in 4.9.

Bernhard Gubanka <beg@ipp−garching.mpg.de>

noticed the need of a recent version of mount to utilize the loopback device

Stephen Harris <sweh@mpn.com>

contributed hint about writing audio CDs

Janne Himanka <shem@oyt.oulu.fi>

pointer to kernel patch to read Joliet CD−ROMs

Stephan Noy <stnoy@mi.uni−koeln.de>

information and experience about writing audio CDs

Don H. Olive <don@andromeda.campbellsvil.edu>

URL of the mkhybrid tool

Jesper Pedersen <jews@imada.ou.dk>

Pierre Pfister <pp@uplift.fr>

helped to develop the recipe on 1:1 copies.

Daniel A. Quist <dquist@cs.nmt.edu>

information about IDE CD−R and newer kernel versions

Martti.Rahkila@hut.fi

Reported problem with pre−initialized writers when booting via loadlin.

Dale Scheetz <dwarf@polaris.net>

Joerg Schilling <schilling@fokus.gmd.de>

information about cdrecord

Martin Schulze <joey@Infodrom.North.DE>

gave information about the cdwrite−mailinglist

Gerald C Snyder <gcsnyd@loop.com>

tested writing of an ext2 CD−ROM (see 4.4)

Art Stone <stone@math.ubc.ca>

had the idea to put non−ISO−9660 filesystems on a CD

The Sheepy One <kero@escape.com>

suggested using defective CD−ROMs as coasters for drinks

Erwin Zoer <ezoer@wxs.nl>

Futhermore, I would like to thank the following people for reporting spelling mistakes: Bartosz Maruszewski <B.Maruszewski@zsmeie.torun.pl>, Ian Stirling <ian@opus131.com>, Brian H. Toby.

End of the Linux CD−Writing HOWTO. (You can stop reading here.)

Next PreviousContents