Expert Linux

Burning CDs using cdrecord

This month, **David Coulson** looks at CD creation using your Linux system

PCPlus SEE YOUR SUPER DISC

CD burners have become a common piece of computer hardware. Nearly every new workstation will ship with some variety of burner, be it a standard CD-R, a combination of a CD-RW and a DVD drive or, indeed, anything in-between. Since the media for these devices is so inexpensive, they are ideal for archiving relatively small amounts of data or for small backing up purposes.

640MB may sound like a lot, but if you just take a look at your home directory, there will be very few people who have it at less than 500MB – particularly if there is a nice collection of MP3s, source code build directories, and anything else which decides to make a home there. It is plenty for archiving log files or small databases.

However, as always, there is a catch. CD-R drives are not the easiest things in the world to get running properly under Linux. Often, any attempt at doing so without an understanding of what is happening results in a collection of failures which end up in the trash. In true Linux style, there are a dozen different programs which will do the same thing to a CD – and it's extremely difficult to decide which one will work right most of the time. We also have the enjoyable prospect of having to rebuild our kernel. In order to burn data to a CD, we need to fool the software into thinking that we've got a SCSI device, rather than a cheap and cheerful ATAPI drive.

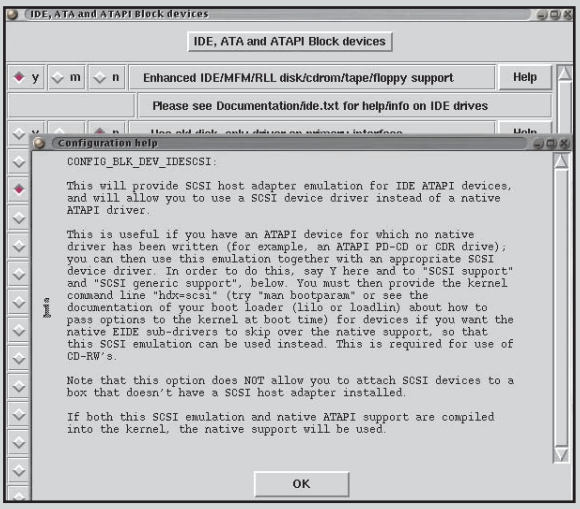
We're going to take a look at cdrecord, which is the most popular CD burning tool for Linux. We will also look at all the other little utilities we need to use to create the finished disc. And, if the command line seems a bit too much just to burn a CD, there are quite a few useful little GUI front-ends for the different utilities which make the whole process easy and painless. Well, almost. PCP

PCPlus SUPER DISC **Related files on SuperDisc 1**
PATH: linux

David Coulson
dcoulson@pcpmag.co.uk

PCPlus NEXT MONTH

We will looking at NIS for authentication, and NFS for home directory sharing



↑CD burning tools assume that you are using a SCSI drive, and since it's quite rare for someone to have a SCSI burner these days, emulation modules are supplied with Linux.

Setting the kernel as a SCSI device

Before burning a CD, we need to set up our kernel to emulate our ATAPI CD drive as a SCSI device...

If you've already got a SCSI CD-R drive, then you will not need to emulate your ATAPI CD drive as a SCSI device. But these are getting quite rare now, so if you're not sure what you've got, it will probably be an ATAPI drive. Within the ATA support section of the kernel configuration, enable the SCSI emulation support. You also need to enable SCSI support within the SCSI section, as well as support for SCSI CD-ROM drives.

Upon building, via the standard make dep && make modules && make modules_install, you'll end up with a ide-scsi.o module, and a few more modules for the SCSI support. This itself doesn't need any options, but we do need to stop the ide-cd module. The ide-cd module handles the ATAPI devices, picking up our drive before the SCSI emulation kicks in. If you have a ide-cd module, you will need to add the following piece of code to /etc/modules.conf:

```
options ide-cd ignore=hda
```

hda is your CD burner. If ide-cd is compiled into the kernel, you need to edit lilo.conf and add the following line:

```
append="hda=ide-scsi"
```

We also need to add a few entries to /etc/modules.conf to ensure that it loads the SCSI modules in the right order when you access /dev/sr0, or /dev/scd0 for 2.2 machines, which is the first SCSI CD-ROM drive:

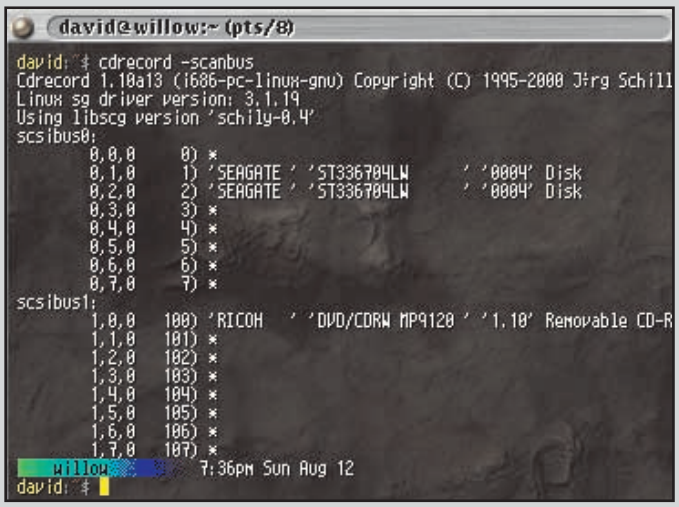
```
pre-install sg      modprobe ide-scsi
pre-install sr_mod  modprobe ide-scsi
pre-install ide-scsi     modprobe ide-cd
```

Once that's all up and running, you should be able to mount a CD using the following piece of code:

```
mount -t iso9660 /dev/sr0 /mnt/cdrom
```

You may also want to create the appropriate symlinks in /dev, or change existing ones to use sr0 rather than the hda device. Usually, /dev/cdrom is all that is needed.

We can check that our CD drive is running under SCSI emulation by checking /proc/scsi/scsi using cat:



↑Even if you know your drive is /dev/sr0, cdrecord needs a bit more information, and cdrecord -scanbus is the way to get it.

↑Yes, mkisofs really does have that many options, but we only need a few of them.

```
Host: scsi1 Channel: 00 Id: 00 Lun: 00
  Vendor: RICOH Model: DVD/CDRW MP9120 Rev: 1.10
  Type: CD-ROM ANSI SCSI revision: 02
```

The host will be scsi0 if you have no real SCSI hardware, and the Id will increase if you have more than one CD drive running under the emulation. These numbers are important, as they are a unique indentifier for the drive within the SCSI module, so it's worth figuring out what is what before jumping in.

An alternative method is to use cdrecord (**freshmeat.net/projects/cdrecord**) to scan the SCSI buses and locate any devices which are attached:

```
$ cdrecord -scanbus
Cdrecord 1.10a13 (i686-pc-linux-gnu) Copyright
(C) 1995-2000 J^rg Schilling
Linux sg driver version: 3.1.19
Using libscg version 'schily-0.4'
scsibus1:
  1,0,0 100) 'RICOH ' 'DVD/CDRW MP9120 ' '1.10'
Removable CD-ROM
```

1,0,0 can be interpreted as device 0, on channel 0, on the second SCSI bus, bearing in mind that everything starts from zero. However, 1,0,0 is all that we need to tell cdrecord to use that drive. cdrecord can either take a dev=1,0,0 option, but as the drive will always live in exactly the same place, it's best to set it within the cdrecord configuration file, /etc/default/cdrecord:

```
CDR_DEVICE=1,0,0
CDR_SPEED=4
```

The speed option can be set at anything that your drive is capable of, but it's best to start quite low until you're sure that your drive is working as it should.

There are two common methods for writing data to a CD – Disc-at-Once (DAO) and Track-at-Once (TAO). TAO is certainly the most widely used method for both audio and data, and that's all we will be looking at here. If you have already got yourself an ISO to burn, either downloaded off the internet or copied from another CD, writing it to the CD-R device can be done with just the cdrecord tool:

```
cdrecord -v -data /path/to/something.iso
```

FORUM www.futureforums.co.uk/pcplus

Audio CD creation

You could use gtoaster to burn an audio CD, but a five-line bash script will do the job...

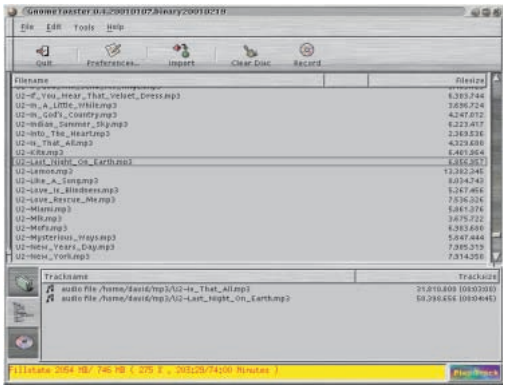
MP3 audio files are great – relatively small in size with little noticeable loss of quality and, of course, they are nice and portable. However, when you're not stuck in front of the computer and you want to listen to music, you either have to buy an MP3 player, which can probably hold 32MB – 64MB of MP3s, and if you like 320kbit MP3s it isn't that much, or you have to stick to your CDs and your portable CD player.

Notwithstanding the legal implications, we can create ourselves a custom CD, containing all our favourite tracks, rather than having to drag around a whole collection of albums. gtoaster will drag MP3s into its track window, but if you know exactly what you want on the CD, it's a little cumbersome. Instead, we can use mpg123 to create the type of audio a CD player likes, then burn it using cdrecord, which is what gtoaster uses anyway.

If we just create a directory and symlink the MP3s we want into it, we can use a little script to look through the directory and burn the contents to a CD:

```
#!/bin/sh
for f in ~/to-burn/*.mp3; do
  mpg123 -cdr - "$f" | cdrecord -
audio -pad -nofix -
done
cdrecord -fix
```

The tracks will be burnt in the order they appear when you do ls *.mp3 within the directory. So, if you name them 01 through to 15, they will be burnt in numerical order. If you prefer the OGG format to MP3, you can use ogg123 to produce the CD audio format, then proceed to burn that to CD.



↑Burning audio CDs is quick and easy using a simple script to search the directory and write the contents, instead of using gtoaster.

It's rather useful just to do a test burn, but fortunately there is a neat way to do that without wrecking a CD. cdrecord has a -dummy option which will perform the burning process without the CD drive turning its laser on.

Creating your own ISO is another matter entirely. We need to use the mkisofs (**freshmeat.net/projects/mkisofs**), which is reasonably basic to use, to create our ISO9660 filesystem on our hard disk, from an existing directory structure.

```
mkisofs -r -o image.iso directory/
```

This will produce an ISO9660 filesystem in the image.iso file, building the contents from the structure of directory/. We need to use the -r option to ensure that the data in the filesystem has public readable permissions, rather than that of the directory/ directory. This also enables what is known as the RockRidge extensions, which enables the CD to store more file information than the standard DOS 8.3 format. If you want to use the CD under Windows, you can use the -J option instead, which creates a CD with Joliet extensions that is readable under Linux, of course, or you can use mkhybrid. Before we burn the image to CD, it's best to check it actually looks how it should, and we can mount the image, just as if it were a regular CD. This does, however, need loop device support, which appears within the Block devices section of the kernel configuration. This should not be confused with the loopback network interface, which is something entirely different. You also may need to run /dev/MAKEDEV loop to create the /dev/loop? devices. The loop.o module needs to mount the filesystem:

```
mount -t iso9660 -o ro,loop image.iso ~/mnt/cdrom
```

From here, we can access ~/mnt/cdrom and experience what it will look like once the ISO9660 image is burnt to our CD-R. Once we're happy, we can burn it with cdrecord -v -data image.iso and wait a while. Once it's done, we can mount /dev/cdrom, and our image.iso is burned to the CD-R.

Creating CDs is a two-stage process, but we can combine these into a single command by piping the output of mkisofs into cdrecord. This requires a little messing around, as cdrecord can't figure out how big the image is, so wrapping it all up into a little script can make it much less painful:

```
#!/bin/sh
DIR=$1
IMG_SIZE=`mkisofs -R -q -print-size $DIR 2>&1 |
sed -e "s/.* = //"`
[ "0$IMG_SIZE" -ne 0 ] && mkisofs -r $DIR |
cdrecord tsize=${IMG_SIZE}s -data -
```

And we just pass the directory that we want burning as an argument to the script, ./burn.sh directory/. This neglects the whole testing stage, but once you're familiar with what your drive is capable of, it's not such a problem. Burning directly from a piped mkisofs process can take a fair bit of CPU and I/O time, so unless you've got a fairly powerful box along the lines of a of a PII 400MHz, you'll find you encounter more problems than it's worth trying to make it do everything at once. But, if you're short on disk space and think your machine can handle it, it's worth trying out.

CD-RW devices are slightly different, since you need to blank the current contents before writing new data. Much like any other type formatting, there is a quick way which just deletes the file references from the superblock. In the case of a CD, it is the Table of Contents (TOC) and the slow method which blanks the entire media. cdrecord has a blank= option, which can be used when burning new data to remove any existing data on the CD. blank=all will wipe the entire CD, and bank=fast will just delete the TOC. Other than this, writing to the CD is exactly the same as with any other CD drive. If you intend to burn audio information to a CD, it's worth remembering that many types of CD player will refuse to play CD-RW media.

More information about creating CDs, including troubleshooting tips and plenty of little short-cuts to help you along the way are in the CD-Writing HOW-TO over at **http://linuxdoc.org/HOWTO/CD-Writing-HOWTO.html**

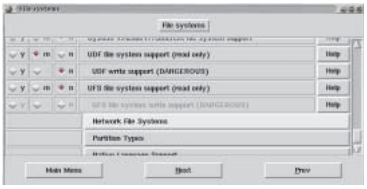
Packet writing

As well as burning a whole ISO, we can read and write individual files...

Usually, when creating a CD we'll burn a disc, such as an ISO or, in the case of an audio CD, we'll burn a set of tracks. However, burning an entire filesystem isn't always ideal for back ups or for log file archiving.

Using the UDF (Universal Disk Format) filesystem, which is used for file storage on DVDs, we can use packet writing. This enables us to write to a disk just as with any other filesystem. Just mount it and then copy whatever you want onto it without messing around with mkisofs or cdrecord. It also makes it considerably easier to store log files on it, as we could then configure syslog or Apache to log straight to the UDF filesystem.

Packet writing support does require a patch from **http://packet-cd.sourceforge.net/**. This is experimental

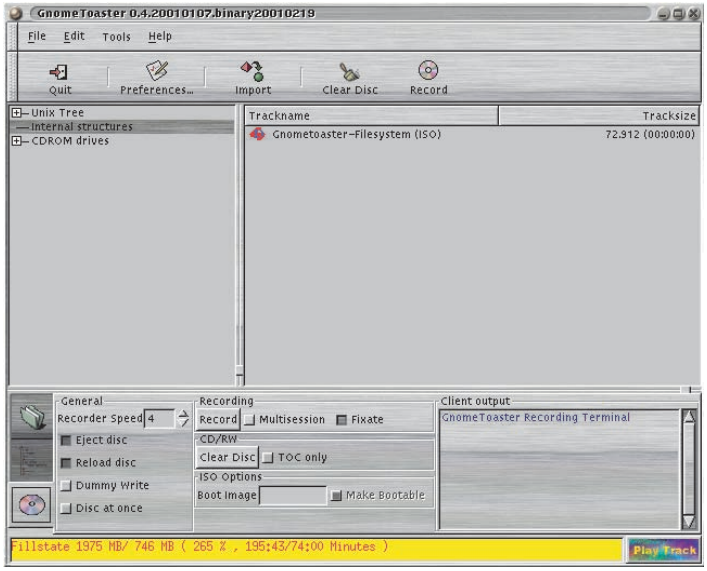


As well as Track-at-Once and Disc-at-Once, we can burn data in packets using the UDF filesystem and packet writing kernel patches.

at best, so it is advisable that you probably shouldn't use it if the data is critical. UDF support is included within the 2.4 kernel tree, but for 2.2 users, patches and binary modules can be downloaded from **www.trylinux.com/projects/udf/index.htmlloop.png**. When we want to mount an ISO image, we need to use the loop device, as it is not a block device.

Graphical CD burning

Tired of having to use command line tools? gtoaster is an excellent front-end to cdrecord and mkisofs...



gtoaster is an ideal graphical CD burning application for those people who are a little afraid of command line applications.

cdrecord and mkisofs are extremely good at what they do, purely because so many people have used them over the years. However, they're not particularly straightforward to use, not to mention that you have to remember all of the different switches, flags and options to make them do what you want. Some clever people noticed this and, rather than rewriting cdrecord and mkisofs, they decided to make a simple GTK+ based front-end to those tools, which makes CD burning a painless task.

As it simply launches cdrecord when it wants to burn a CD, we don't need to configure gtoaster with any special options. cdrecord will use the existing configuration file in /etc/default unless, of course, gtoaster passes it device information when it is run.

The entire gtoaster interface is drag-and-drop, so you just drag a file or directory into the Filesystem section, and it adds it to the contents of the ISO9660 image. If you want to burn an ISO image downloaded from the net, drag it into the tracks section and it will burn it as a track. And, for those of us who want to burn audio CDs, it is intelligent enough to know that it should convert MP3s, wavs and any other audio format you configure it to understand, into CD audio before burning it.