Cd-Writing under Linux

Alon Altman

8th July 2002

What's new?

Old Lecture:

- Installation
- Iso9660 (data CD) generation
- Advanced Topics
  - Multisession
  - CD-RW Blanking
  - Bootable CDs
  - Mp3 to Audio CD
- Today - All the above and...
  - BIN/CUE images (raw CDs)
  - CD-to-CD copying
  - VCD burning

Concepts

- **CD Image** - A file which includes the entire contents of a CD-ROM or a CD-ROM track ready for writing.
- CD-ROM Filesystems:
  - **iso9660** - "Volume and file structure of CD-ROM for information interchange".
  - **Joliet** - Micro$oft extension of iso9660 to support Unicode and longer file names.
  - **UDF** (Universal Disk Format) - New standard, used mosly in DVDs, supporting packet read/write on CD-RW. Under Linux, UDF is still experimental, and supports only loopback (not CD).

What do you need?

- A *CD-Writer* (SCSI or IDE), that is supported in Linux.
- A blank CD-R or CD-RW *media.*
- *Software:* `mkisofs` and `cdrecord` (and maybe also `cdrdao)`
- *Data* you want to write as files on your HD, and/or audio tracks as wav files (data can actually be piped over the network, but it is strongly advised to keep the data on the local HD).
- About 700Mb of *free HD space* for CD-Image (not really required, but preferable).

Documentaion

- **CD-Writing-HOWTO** - excellent document all about writing CDs in Linux.
- **man pages** of `mkisofs`, `cdrecord`, `cdrdao`, and `bchunk`.
- **/usr/share/doc/cdrecord*** - `cdrecord` documentation, also available on the Web.
- CDRecord Homepage - `http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html`

The CD Writing Process

- Prepare your system for writing CDs (done only once)
- Prepare source directory/wav files
- Make the ISO image.
- Write the ISO image to a CD.

Preparation

- Install your CD Writer Drive physically. **Note:** Make sure your CD Writer drive is *not* on the same IDE cable as your HD with the CD-images, otherwise collisions will occur and create buffer underruns.
- Install kernel modules `ide-scsi` (for IDE only), `sg` and `sr_mod`. You can either re-compile the kernel, or insmod the modules.
- **For IDE only** - edit `/etc/lilo.conf` and add in your configuration a line `append=''hdd=ide-scsi''`, while hdd is the name of your CD-Writer.
- Reboot your computer. Run as root: `cdrecord --scanbus` and make sure you see your cd-writer listed.

Preparation of Source Directory

In order to write CDs you need the source data be put in one directory (actually, the newest version of `mkisofs` knows to handle multiple source dirs). If your files are scattered all over the hard disk, you can eiter copy the files to one source directory, or use symbolic links, and create an image directory that includes only symbolic links to the source files.

For writing CD-Audio tracks, you need to prepare WAV files that will be converted to CD-Audio.

Running mkisofs

mkisofs is the program that converts files on your filesystem to a CD image. The syntax is:

```
mkisofs [options] -o filename pathspec
```

The -o option specifies the name output *image file* to be written.

The path specification is the name of the source directory.

To use multiple source directories scattered arbitrarily on the filesystem, you may use the -graft-points option. man mkisofs for details.

mkisofs options

`mkisofs` has *lots* of options, such as:

`-v` Enable verbose mode - always use this!

`-r` Enable Rock Ridge (UNIX permissions) - for backups.

`-J` Enable Joliet (Windows-Readable LFNs)

`-jcharset` Specify charset for file names in Windows. Use charset `iso8859-8` for standard Hebrew support.

`-f` Follow symbolic links - Important if you've symlinked your source directory.

`-l` Allow long file names (31 characters)

`-D` Do not relocate deep directories (non-standard).

`-L` Allow filenames to begin with a period (non-standard)

Refer to `man mkisofs` for complete list of options.

Testing the CD-Image

Linux lets you `mount` a CD-Image as if it is a real CD. This is by using the special `loop` device in the kernel. This topic was more throughly discussed in the Rescue disk lecture.

To test a cdrom image issue the command:

```
# mount -o loop,ro -t iso9960 image.iso /mnt/cdimage
```

Don't forget to unmount the image after testing:

```
# umount /mnt/cdimage
```

Running cdrecord

`cdrecord` is program that actually writes the disk. The syntax is:

```
cdrecord  [general options] dev=device [track options]
          track1 ... trackn
```

'device' is the SCSI id of the device, in the format scsibus,target,lun. This data can be found by running `cdrecord --scanbus`. For IDE drives, this usually is 0,0,0.

General options:

The most important options include:

`-v`	Enable verbose mode - always use this!
`-dummy`	Simulation mode. Try this before real writing.
`-eject`	Eject after write.
`speed=#`	Set writing speed.

General options (cont.)

`fs=#` Set FIFO size (buffer). Default is 4MB. Mostly used when piping data directly from `mkisofs`.

`-dao` Disk-At-Once mode, eliminates silence between audio tracks.

Track options

`-audio` All subsequent tracks are audio tracks (au or wav files).

`-data` All subsequent tracks are data tracks (ISO images).

`-pad` Pad data/audio tracks to frame. Needed when writing audio.

Refer to `man cdrecord` for detailed explanation of all options.

GUIs for CD-Writing (or: How to do it easily?)

- xcdroast - `http://www.xcdroast.org`
- gtoaster - `http://gnometoaster.rulez.org/`
- kreatecd - `ftp://ftp.kde.org/pub/kde/unstable/apps/multimedia/cdrom`
- and much much more..

Personally, I use xcdroast and it works very well. All these frontends make it much easier to write your CDs under Linux, at the cost of loss of some ultra-advanced functionallty.

Writing BIN/CUE images

Sometimes, CD images are supplied in a BIN/CUE format and not ISO. This format is used to specify the contents of an entire disc as opposed to just one track. Therefore, the BIN/CUE format is much more flexible and allows for creation of *any* type of CD.

To write these raw images, we use a different CD-Writing tool called `cdrdao`:

```
cdrdao   write --device 0,0,0 [--simulate] -v 2
         --buffers 64 --speed speed image.cue
```

Note: The BIN file must be in the same directory as the CUE file for this to work.

Converting BIN/CUE to ISO+WAV

There are things that cannot be done with a BIN/CUE image. For example, loopback mounting of the image file and burning the image alongside other audio (or data) tracks.

In these special cases, you can use `bchunk` to extract the ISO (=data) and WAV (=audio) tracks from the BIN and CUE combination.

Usage: `bchunk -w` *`image.bin image.cue basename`*

This generates files with the given basename with the data and audio tracks of the CD

Copying a CD

To copy a complete CD 1-to-1, we also use `cdrdao` with the `copy` command. Use the `--source-device` and `--on-the-fly` to copy CD to CD without creating an image file.

On older kernels, you might need to install a patch to enable CD to CD copying due to SCSI implementation details.

CD-RW blanking

In order to blank a CD-RW (ReWriteable) disc, you should use the `blank` option of `cdrecord`.

The syntax is: `cdrecord dev=0,0 blank=`*`type`*

Blank types are:

all	Blank entire disk
fast	Minimally blank entire disk (best choice)
track	Blank a track
unclose	Un-close last session
session	Blank last session

Writing VCDs

VCD is a disc format used for distribution of MPEG-1 encoded video. A VCD disc includes a rigidly defined iso9660 filesystem which defines interaction and control of the VCD, and tracks of MPEG-1 encoded data. The main advantage of VCDs is that they are playable by most consumer-level DVDs.

AFAIK, The easiest way to author a VCD is to use `VCDImage` to convert MPEG-1 encoded video in the proper format to a BIN/CUE image for the VCD.

With the BIN/CUE image created, you can use `cdrdao` to write the image to a CD.

The tools for generation of MPEG-1 VCD streams are outside the scope of this lecutre.

Writing with no image (on-the-fly)

In order to write a CD without creating an image on the HD, all you need to do is pipeline to output of `mkisofs` to `cdrecord`.

The syntax is: `mkisofs [options] | cdrecord [options] -`

- This will create the image and write the CD on-the-fly.
- It is strongly advised to do a dry-run with `-dummy` before doing the real write.
- Make sure that the CD-Writer is NOT on the same IDE as the HD with image data, otherwise this will not work.
- If the machine is tight on CPU, you might want to `nice` the mkisofs to priority -18 by typing: `nice --18 mkisofs` instead of `mkisofs`.
- If you get buffer underruns, use `fs=`*`size`* in cdrecord to increase buffer size above 4 megs.

Creating bootable CDs

`cdrecord` supports several types of bootable CDs, however the most commonly used is "El Torito" bootable CDs.

The "El Torito" standard specifies that the CD should include an image of a floppy or hard drive, that will replace the real floppy/hard drive when booting. The floppy drive option is most commonly used, to boot the system from CD, still keeping HDD support.

In order to create a bootable CD, you first need a floppy image. More information about floppy images may be found in the "Rescue Disk" lecture, previously given in this forum.

If you already have a bootable floppy, you can create a floppy image using: `cat /dev/fd0 > image`

In order to make a CD bootable, you need to put the floppy image somewhere in the CD image directory, and tell `mkisofs` about it using the `-b` flag. In addition you need to specify a name for the boot catalog file with the `-c` option. This usually is "`boot.catalog`".

For example, if your floppy image is on `/bootimage` relative to the CD image, then the command should look like:

```
mkisofs [options] -b /bootimage -c boot.catalog path
```

This will create the CD image, bootable by the given boot image.

Multisession CDs

Multisession CDs allow you to add data to already-burned CDs in an additional 'session'

To write multi-session CDs, you need to mark each session (possibly except the last) as ready for multi-session using the `-multi` flag on `cdrecord`.

To add a session to a written disk, you have to first get the parameters of the first session using:

```
cdrecord dev=x,y -msinfo.
```

This will return two numbers seperated by a comma "`xxx,yyy`".

To make the image for the second session, you will need to add the following flags to `mkisofs` (Persuming your CD-Writer drive is /dev/scd0):

```
mkisofs -M /dev/scd0 -C xxx,yyy [more options]
```

This will create the image. This image can be written as usual. If you wish to be able to add additional sessions, you must use the `-multi` option in `cdrecord`.

For more detailed informaion, see `README.multi` in your cdrecord documentation.

Audio CD from mp3 on the fly

Suppose you have some (legal) mp3 files on your hard disk, and you would like to make an audio CD out of them, *without* needing to save the temporary wav files on your HD. How can this be done?

To convert and mp3 file to a wav file, you would usually use this command:

```
mpg123 -s song.mp3 |
sox -s -t raw -w -r 44100 -c 2 - song.wav
```

To write the mp3 files direcly, all you have to do is create *named pipes* on the filesystem (calling them `song1`.raw, `song2`.raw, etc.) and call the decoding command in the background into those pipes, then write from the pipes.

To create the pipes, use (for each song):

```
mknod songn.raw p
```

Then, do the decoding for each file in the background (There is no need to convert to wav format):

```
mpg123 -s songn.mp3 > songn.raw &
```

Then, run `cdrecord` as usual, with the `-swab` option added to indicate you are using swapped byte order (little-endian):

```
cdrecord -v dev=xx,yy speed=2 -swab -audio song*.raw
```

Summary

- Creating CDs under Linux is quite an easy process.
- The easiest way to create CDs is with a GUI.
- BIN/CUE images should be written using `cdrdao`, and can be converted to ISO using `bchunk`.
- Linux allows you to do advanced tricks (such as mp3 on the fly) using the OS's pipe feature in conjunction with `cdrecord`.
- Have fun, and don't forget - use `-dummy` first!