Converting Audio Formats

Getting the swing

The command line swings. In this issue of "Command Line" we investigate how you can grab tracks from audio CDs and convert sound files to other formats. No matter whether you choose wav, MP3 or Ogg Vorbis, you can rely on the shell to convert your files. BY HEIKE JURZIK

www.sxc.hu

You do not need a GUI-based tool to convert sound files to different formats – the command line gives you everything you need. To get started, we will be looking at two programs that grab tracks from audio CDs and store them in wav format. Following that, we will be looking at tips and tricks for converting various audio formats and learning how clever use of bash features can speed up the process.

Grabbing with cdda2wav

cdda2wav [1] ("compact disc digital audio to wav") is a program that most distributors include. Debian users can install the tool by running

```
apt-get install cdda2wav
```

Command Line

Although GUIs such as KDE or GNOME are useful for various tasks, if you intend to get the most out of your Linux machine, you will need to revert to the good old command line from time to time. Apart from that, you will probably be confronted with various scenarios where some working knowledge will be extremely useful in finding your way through the command line jungle.

Before you launch *cdda2wav*, check to see if you can access your CD-ROM drive with a non-privileged user account (see Box 1). To grab a track, you need to use the *-t* option and the track number. The program also expects the device name of your CD-ROM drive, which you can supply by adding a *-D* option:

```
$ cdda2wav -D /dev/hdc -t 1
...
recording 276.6666 seconds ster
eo with 16 bits @ 44100.0 Hz ->
'audio'... 0/ 0/ 1        0  11%
```

The output includes information on the titles, the **CDDB** disk ID, and a progress indicator. The wav file is stored as *audio.wav* in the current working directory. Tip: to avoid repeatedly typing the device name, you can temporarily set the *CDDA_DEVICE* environment variable by typing

```
export CDDA_DEVICE=/dev/hdc
```

in the active shell, or you can add the command to your *.bashrc* to make this assignment permanent.

To grab multiple (or all the) tracks from a CD, add *-t* and the numbers of the first and last tracks you want to grab; for example,

```
cdda2wav -t 1+3
```

gives you the first three tracks. To prevent the output from ending up in a single wav file, you might like to add the *-B* option (see Figure 1). This will give you a collection of tracks named *audio_01.wav*, *audio_02.wav* and so on.

cdparanoia

cdparanoia [2] is a useful alternative to *cdda2wav* for grabbing tracks off audio CDs. Again, the tool is included by most distributions. To test the program on the command line, set the *-Q* flag. If you do not see output like the output in Listing 1, or if you see an error message such as:

GLOSSARY

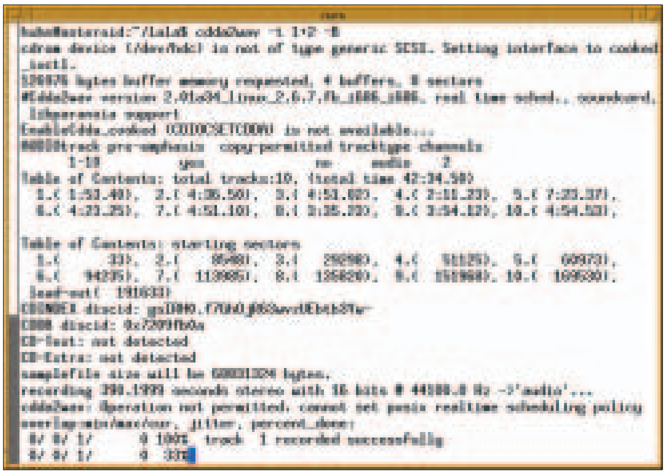
CDDB: *The "**C**ompact **D**isk **D**ata**b**ase" contains information on audio CDs and the titles they contain. Many CD players automatically query a free CCDB server (assuming you have an Internet connection), and add information on the artist, the album name, and the track list to the play list. CDDB is a commercial service, and Linux players more typically access the FreeDB database at www.freedb.org instead.*

```
/dev/cdrom exists ↩
but isn't accessible.
```

check the permissions for the device file. As Box 1 explains, it makes sense to assign the device file to the *cdrom* group, and to add users who need CD access to that group.

To grab a single track with *cdparanoia*, simply stipulate the track number, for example

```
cdparanoia 1
```



```
huhn@asteroid:~/LaLaB cdda2wav -t 1+2 -B
cdrom device (/dev/hdc) is not of type generic SCSI. Setting interface to cooked
_ioctl.
520976 bytes buffer memory requested, 4 buffers, 8 sectors
#Cdda2wav version 2.01a34_linux_2.6.7_fb_i686_i686, real time sched., soundcard,
 libparanoia support
Enable(Cdda_cooked (CDIOCSETCDDA)) is not available...
AUDIOtrack pre-emphasis  copy-permitted tracktype channels
      1-10         yes                no      audio    2
Table of Contents: total tracks:10, (total time 42:34.58)
  1.( 1:53.40),  2.( 4:36.50),  3.( 4:53.02),  4.( 2:11.23),  5.( 7:23.37),
  6.( 4:23.25),  7.( 4:51.10),  8.( 3:35.23),  9.( 3:54.12), 10.( 4:54.53),

Table of Contents: starting sectors
  1.(      33),  2.(   8540),  3.(   29290),  4.(   51125),  5.(   60973),
  6.(   94235),  7.(  113985),  8.(  135820),  9.(  151968), 10.(  169530),
 lead-out(  191633)
CDINDEX discid: [illegible]
CDDB discid: 0x7209fb0a
CD-Text: not detected
CD-Extra: not detected
samplefile size will be 60031324 bytes.
recording 340.1999 seconds stereo with 16 bits @ 44100.0 Hz ->'audio'...
cdda2wav: Operation not permitted. cannot set posix realtime scheduling policy
overlap:min/max/cur, jitter, percent_done:
 0/ 0/ 1/     0 100%  track  1 recorded successfully
 0/ 0/ 1/     0  33%
```

Figure 1: Grabbing audio tracks in next to no time with *cdda2wav*.

It is just as easy to define a list of tracks. If you want to grab the first five tracks, for example, you can do so by specifying *1-5*. If you are grabbing multiple tracks, again you need to set the *-B* flag to avoid storing everything in a single wav file, as in

```
$ cdparanoia -B 1-2
...
outputting to track01.cdda.wav
```

cdparanoia names the wav files *track01.cdda.wav*, *track02.cdda.wav*, etc.

Converting wav to MP3

There are quite a few command line tools for converting wav files to MP3, and *lame* [3] is one of the more popular conversion tools. The project homepage offers a source code download, but search machines such as *rpmseek.com* [4] will point you to pre-compiled packages for Debian and RPM-based distributions. Unfortunately, the link to the Debian package is broken, but you can check out [5] for packages for all three Debian branches (*stable*, *testing*, and *unstable*).

If you run *lame* against a wav file without specifying any options, the tool produces a MP3 file with a **bit rate** of 128Kbps. And if you fail to supply a name for the MP3 file, *lame* will just pick a name for you:

```
$ lame audio_01.wav
...
Encoding audio_01.wav t↩
o audio_01.wav.mp3
...
```

Of course, you can specify a name for the output file:

```
lame audio_01.wav 01.mp3
```

If you need better sound quality, you can change the bit rate by setting the *-b* flag. For example, *lame* uses a bit rate of 192Kbps when you type the following

```
lame -b 192 1.wav 1.mp3
```

As *lame* does not support wildcards by default (that is, a command such as *lame *.wav* will display an error message and quit), you will need a *for* loop to harness the full power of the command line and convert all the wav files in a directory, using the *basename* command to remove the *.wav* extension, and assign the correct extension to your MP3 files.

```
for i in *.wav; do lame $i ↩
`basename $i .wav`.mp3; done
```

Converting MP3 back to wav

lame also has an option for converting MP3 files back to wav – of course, lossy MP3 encoding means that the sound quality of the wav file will not compare well to the original audio CD. The *--decode* option allows you to turn the tables, and convert MP3 back to wav:

```
$ lame --decode 01.mp3
input:  01.mp3  (44.1 kHz, 2 ch↩
annels, MPEG-1 Layer III)
output: 01.mp3.wav  (16 bit, Mi↩
crosoft WAVE)
skipping initial 1105 samples ↩
(encoder+decoder delay)
Frame#   807/4348  128 kbps  L R
```

Box 1: Access Privileges for the CD-ROM Device

The various distributions have different ways of defining which users should have access to the CD-ROM drive. *ls -l* outputs the permissions for the device:

```
$ ls -l /dev/cdrom
lrwxrwxrwx  1 root  root     3
Jun  7 00:41 /dev/cdrom -> hdc
```

As is to be expected, */dev/cdrom* is a symbolic link that points to the device file: *hdc* in our case. You need another *ls* command to display the permissions here:

```
$ls -l /dev/hdc
brw-rw----  1 root  disk 22, 0
Mar 14  2002 /dev/hdc
```

Debian Specifics

While most distributions simply change the owner of the drive when a user logs onto a system, Debian has a different approach. The drive belongs to the *disk* group by default. As it is not advisable to add users to the *disk* group for security reasons (since it would give them write access to hard disk partitions), the administrator, *root*, needs to change the group assignment for *cdrom*:

```
chown root.cdrom /dev/hdc
```

After doing so, and still working as the admin user, edit */etc/group*, adding the user account that needs access to the drive to the *cdrom* group. You can add multiple accounts by separating them with a comma (avoiding spaces), for example:

```
cdrom:x:24:huhn,petronella
```

To apply these changes, the user needs to log off and back on to the system. You can run the *groups* command to check if everything has worked out; the command displays a user's group memberships on the command line:

```
$groups
audio cdrom video
```

GLOSSARY

Bit rate: *When compressing sound files, the bit rate (which is typically given in Kbits per second) defines the sound quality of the MP3 file. The higher the bit rate, the better the MP3 file will sound. On the downside, the file size increases to match. Under normal circumstances, bit rates between 128 and 192Kbps make sense.*

In this scenario *lame* again chooses the filenames, unless you explicitly specify them. And again, a *for* loop can help you convert multiple files with one fell swoop.

The *mpg123* [6] command line player is another useful alternative. The player has a *-w* option for converting MP3s to wav files:

```
$ mpg123 -w file.wav 01.mp3
...
Playing MPEG stream from 01.mp3
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
[1:53] Decoding of 01.mp3 finished.
```

Note the importance of the order of the input and output files. You need to specify the output file first, using the *-w* option, and then the name of the MP3 file you want to convert.

From wav to Ogg

Ogg Vorbis is a free alternative to MP3 – free in the sense that the codecs are patent-free and released under a free license. Additionally, Ogg Vorbis supports superior sound quality. If you install the *vorbis-tools* from your distribution media, you should have the *ogg123* command line player, along with an encoder and decoder. *oggenc* is the tool you need to convert wav files to Ogg Vorbis format:

```
$ oggenc 01.wav
Opening with wav module: WAV file reader
Encoding "01.wav" to
         "01.ogg"
at quality 3.00
        [ 17.8%] [ 1m24s remaining]
```

Again the tool automatically chooses filenames, but in contrast to the programs we have looked at so far, *oggenc* supports wildcards and can convert all the wav files in a directory with a single command (*oggenc *.wav*) (Figure 2).

To choose a different name for the output file, you need to set the *-o* flag:

```
oggenc 01.wav -o bla.ogg
```

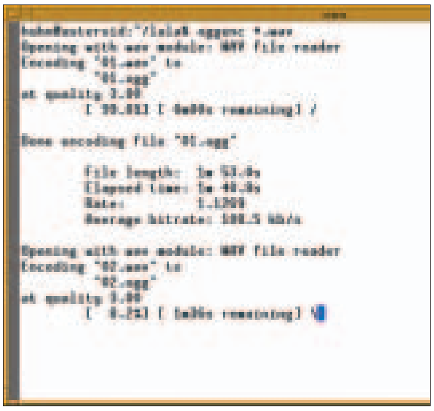
One of *oggenc*'s most practical features is its ability to read information from command line parameters for the artist, (*-a*), album name (*-l*), title (*-t*), genre (*-G*), and date (*-d*) while converting a file:

```
oggenc 01.wav -a "The Alan Parsons Project" -t "Sirius" -l "Eye in the Sky" -G "Pop" -d "1981" -o blubb.ogg
```

This allows current players, such as XMMS, to display this information when playing the tracks later by reading the ID3 tags.

Listing 1: *cdparanoia -Q* Output

```
$ cdparanoia -Q
cdparanoia III release 9.8 (March 23, 2001)
(C) 2001 Monty <monty@xiph.org> and Xiphophorus
Report bugs to paranoia@xiph.org
http://www.xiph.org/paranoia/
Table of contents (audio tracks only):
track        length               begin        copy pre ch
===========================================================
  1.     8515 [01:53.40]        33 [00:00.33]    no   no  2
  2.    20750 [04:36.50]      8548 [01:53.73]    no   no  2
  3.    21827 [04:51.02]     29298 [06:30.48]    no   no  2
  4.     9848 [02:11.23]     51125 [11:21.50]    no   no  2
  5.    33262 [07:23.37]     60973 [13:32.73]    no   no  2
  6.    19750 [04:23.25]     94235 [20:56.35]    no   no  2
  7.    21835 [04:51.10]    113985 [25:19.60]    no   no  2
  8.    16148 [03:35.23]    135820 [30:10.70]    no   no  2
  9.    17562 [03:54.12]    151968 [33:46.18]    no   no  2
 10.    22103 [04:54.53]    169530 [37:40.30]    no   no  2
TOTAL  191600 [42:34.50]    (audio only)
```



```
[illegible]:~/[illegible]$ oggenc *.wav
Opening with wav module: WAV file reader
Encoding "01.wav" to
         "01.ogg"
at quality 3.00
        [ 99.8%] [ 0m00s remaining] /

Done encoding file "01.ogg"

        File length:  1m 53.0s
        Elapsed time: 1m 40.0s
        Rate:         1.1300
        Average bitrate: 100.5 kb/s

Opening with wav module: WAV file reader
Encoding "02.wav" to
         "02.ogg"
at quality 3.00
        [  0.2%] [ 1m00s remaining]
```

Figure 2: *oggenc* supports wildcards.

From Ogg to wav

Converting Ogg Vorbis to wav is just as simple with *oggdec*. Again you need the *-o* option if you want to specify a filename. And again, the program supports wildcards. The following command

```
oggdec *.ogg
```

converts all the Ogg Vorbis files in a directory into wav files.

Conclusion

Converting audio files on the command line is easy: if the tool you are using does not support wildcards, bash gives you a workaround in the form of a simple *for* loop. If all of this sounds like too much effort, or too much typing, check out the *abcde* [7] script, which uses the commands we have looked at in this issue, adding comments or ID3 tags, and even querying the CDDB database to retrieve names and track titles. ■

INFO

[1] *cdda2wav* homepage: *http://www.cdda2wav.de/*

[2] *cdparanoia* website: *http://www.xiph.org/paranoia/*

[3] *lame* project: *http://lame.sourceforge.net/*

[4] Search machine for RPMs and Debian packages: *http://rpmseek.com/*

[5] *lame* for Debian Linux: *ftp://ftp.nerim.net/debian-marillat/*

[6] Command line MP3 player *mpg123*: *http://www.mpg123.de/*

[7] Heike Jurzik: "Music was my first love: abcde – A Better CD Encoder", Linux Magazine #30 / May 2003, p82.