

Multimedia documentation

Documents how to do some multimedia tasks

Contents

1	Convert wma to ogg	1
2	Convert ogg to mp3	2
3	Ripping a DVD	2
4	Ripping an audio CD	2
5	Merging two avi files	2
6	Merging two mkv files	3
7	Playing with subtitles	3
8	Sound settings in mplayer	3
9	Extract the audio content of a video file	3
10	Extract images from a video file	4
11	Renaming an audio file based on ID3 tags	4
11.1	References	4

1 Convert wma to ogg

You need to dump the wma file into an uncompressed format (**wav**):

```
$ mplayer -ao pcm:file=/tmp/out.wav $wma_file
```

Then encode it into ogg:

```
$ oggenc -o ${wma_file%.wma}.ogg /tmp/out.wav
```

2 Convert ogg to mp3

Same as previous section for the dumping part, but use:

```
$ lame /tmp/out.wav ${ogg_file%.ogg}.mp3
```

for the encoding part.

3 Ripping a DVD

Download Handbrake¹ (command line tool should be enough), and dvdbackup²

Rip the dvd into a temporary directory (here the current directory):

```
dvdbackup -i /dev/dvd -o . -M
```

Then list the titles:

```
handbrake -i ~/tmp/where_was_ripped -t 0
```

Do the rip:

```
handbrake -i ~/tmp/where_was_ripped -t 1 -o $outfile.mkv \
--size 700 --two-pass --audio 1
```

4 Ripping an audio CD

There is a script called abcde³ that will do this very well. Download and install it and insert the CD you want to rip. The configuration is made using a configuration file: `$HOME/.abcde.conf`. Mine just contains:

```
OUTPUTTYPE="flac"
```

Since I wanted to use FLAC⁴ instead of the default, Ogg Vorbis⁵. The manpage documents all the configuration file options.

Once you are done configuring abcde, just cd to the directory where you want to rip and run:

```
$ abcde
```

The script will automatically download the disc's metadata (artist, album and song names, genre...), ask you to confirm them and then rip and encode the music.

5 Merging two avi files

To merge two avi file *that have the same format* (because e.g they are splitted into two CDs):

```
mencoder -oac copy -ovc copy file1.avi file2.avi -o output.avi
```

If mencoder tells you the two files do not have the same format, then the simplest method I have seen is to use avimerge, from transcode⁶:

¹<http://handbrake.fr/>

²<http://dvdbackup.sourceforge.net/usage.html>

³<http://code.google.com/p/abcde/>

⁴<http://flac.sourceforge.net/>

⁵<http://www.vorbis.com/>

⁶<http://www.transcoding.org/transcode>

```
avimerge -o output.avi -i file1.avi file2.avi
```

If you see messages like:

```
No audiodata left for track 0->0 (6078000.00=6078000.00)
```

And you see that audio is out of sync, you could use the `-c` flag. See the manpage for more information.

6 Merging two mkv files

To merge two mkv files, you should use `mkvmerge` from MKVToolnix⁷:

```
$ mkvmerge --append-mode track -o full.mkv file1.mkv '+file2.mkv'
```

7 Playing with subtitles

If you want to edit the subtitles or re-time them, the one utility I found useful is `Subtitles`⁸, and its associated command line utility `subs`. As an example, to shift the times in a `.srt` file 60 seconds later, one can just do:

```
$ subs -b 60 file.srt
```

8 Sound settings in mplayer

Use 0 and 9 to respectively increase or decrease the volume while `mplayer` is playing. You can use `-softvol-max` to adjust the maximum volume. The argument that is taken is the maximum amplification level (in percent). For example:

```
$ mplayer -softvol-max 2000 somefile.avi
```

will allow an amplification of 20 times, at the cost of some lost quality and granularity.

9 Extract the audio content of a video file

To get only the audio track of a video file, use `mplayer`'s `-dumpaudio`:

```
$ mplayer -dumpaudio $videofile -dumpfile $audiofile
```

Since this is not always very stable, here is an alternative method using `ffmpeg`⁹:

```
$ ffmpeg -i $videofile $audiofile
```

Where `$audiofile` should have the extension you want, e.g `mp3`.

⁷<http://www.bunkus.org/videotools/mkvtoolnix/>

⁸<http://karasik.eu.org/software/>

⁹<http://ffmpeg.org>

10 Extract images from a video file

To extract images from a video file, use mplayer's `-vo` option:

```
$ mplayer -ss $interval -vo $format -frames $nframes -nosound $file
```

Where:

- `$interval` is the number of seconds to wait between two frames. Can be a decimal number, like 0.01.
- `$format` is the picture format to use. Can be for example `jpeg`, `png`, `gif`...
- `$nframes` is the number of frames to take
- `$file` is the path to your file.

Example:

```
$mplayer -ss 0.01 -vo jpeg -frames 5 -nosound mvi_1234.avi
```

will dump the first five frames of the movie `mvi_1234.avi` encoding them in the JPEG¹⁰ format, waiting 0.01 seconds between each frame.

11 Renaming an audio file based on ID3 tags

You can use the `id3`¹¹ tool to edit / read and rename file according to their ID3 tags. One command I often use to rename all files to a `$track-$title.mp3` format is:

```
$ id3 -f '%#n-%t.mp3' *.mp3
```

11.1 References

- The extracting image from a movie¹² post on mplayer's mailing list.

¹⁰<http://en.wikipedia.org/wiki/JPEG>

¹¹<http://home.wanadoo.nl/squell/id3.html>

¹²<http://lists.mplayerhq.hu/pipermail/mplayer-users/2004-August/047452.html>