

Linux Audio

“”

Contents

1	Beeping And Synthesizing	1	12	Adding Sound Effects	6
2	Curiosities	2	13	Using The Sox Tool	6
3	Sound Settings	2	14	Fidelity Or Sound Quality	7
4	Jukeboxes	2	15	Sound Volume	7
5	Internet Radio	2	16	Extracting Audio From Video	8
6	Playing Sound Files	2	17	Splitting Sound Files	9
7	Editing Sound Files	3	18	Play Lists	9
8	Finding Sound Files	3	19	Synth Music	9
9	Recording Sound	3	20	Metadata	9
10	Audio Cds	4	21	Speech Synthesis	9
10.1	Playing Audio Compact Disks	4	22	Internet Phone Calls	9
10.2	Copying Cd Audio	4	23	Curiosities	9
10.3	Burning Compact Discs	5	24	Notes	9
11	Converting Sound Formats	5			

This book tries to contain succinct recipes for using audio files with the linux operating system, with a special emphasis on using the command line (bash shell).

waon:
converts waves to notes

timidity:
converts notes to waves

To select a device for recording, use set followed by the name of the

sound jargon

gain	Like the volume
channels	Whether sound is mono, stereo or surround sound
herz	The number of samples per second
samples	
midi	Synthesized music or sound

Unencrypted voicechat

```
[0n PC1] nc -l -p 6666 > /dev/dsp
[0n PC2] cat /dev/dsp | nc <PC1's IP> 6666
```

Keep from having to adjust your volume constantly

```
find . -iname \*.mp3 -print0 | xargs -0 mp3gain -krd 6 && vorbisgain -
=> rfs .
```

Tune your guitar from the command line.

```
for n in E2 A2 D3 G3 B3 E4;do play -n synth 4 pluck $n repeat 2;done
```

Beeping And Synthesizing*Ring the bell*

```
alias beep='echo -en "\007"'
```

MP3 SOUND FILES*Reduce mp3 bit-rate (and size, of course)*

```
lame --mp3input -m m --resample 24 input.mp3
```

Play the mp3 sound file 'test.mp3'

```
mpg123 test.mp3
```

```
mpg321 test.mp3 ~ (more efficient where no floating-point unit)
```

Play the MP3 stream at the url

```
mpg321 http://example.net/broadcast/live.mp3
```

When not recording sound, keep the inputs muted

Change the sampling rate of 'old.wav' to 7,000 Hz, and write to 'new.wav'

```
sox old.wav -r 7000 new.wav
```

This command writes a new file, 'new.cdr', in the audio CD format; the original file, 'new.wav', is not altered.

Convert the file 'new.wav' to the audio CD format, save in 'cd-single'

```
sox new.wav -t cdr cd-single
```

Encode an MP3 file from a WAV file called 'september-wind.wav'

```
lame september-wind.wav september-wind.mp3
```

It usually takes some time to encode an MP3 file.

Convert the MP3 file 'remix.mp3' to a WAV file 'remix.wav'

```
mpg321 -b 10000 -s remix.mp3 | sox -t raw -r 44100 -s -w -c 2 - remix.
⇒ wav
```

Output your microphone to a remote computer's speaker

```
dd if=/dev/dsp | ssh -c arcfour -C username@host dd of=/dev/dsp
```

Use curl to save an MP3 stream

```
curl -sS -o $outfile -m $showlengthinseconds $streamurl
```

Generate White Noise

```
cat /dev/urandom > /dev/dsp
```

Curiosities*Beep siren*

```
tempo=33; slope=10; maxfreq=888; function sinus { echo "s($1/$slope)*
⇒ $maxfreq"|bc -l|tr -d '-'; }; for((i=1;;i++)); do beep -l$tempo -f'
⇒ sinus $i'; done
```

Sound Settings**Jukeboxes**

```
gmusicbrowser
@@
```

Internet Radio**internet radio tools**

get-iplayer	Bbc iplayer radio stations
mimms	Down load mms:// streams

Listen to BBC Radio from the command line.

```
bbcradio() { local s;echo "Select a station:";select s in 1 1x 2 3 4 5
⇒ 6 7 "Asian Network an" "Nations & Local lcl";do break;done;s=($s);
⇒ mplayer -playlist "http://www.bbc.co.uk/radio/listen/live/r"${s[@]:
⇒ -1} ".asx";}
```

Listen to an internet radio station with 'mplayer'

```
mplayer mms://server:port/path
```

Playing Sound Files

Before you begin playing sound, make sure you've set the master and PCM volume levels with the mixer

Play the file 'pentastar.aiff'

```
play pentastar.aiff
```

Play the MP3 file 'september-wind.mp3'

```
mpg321 september-wind.mp3
```

Select a 'wav' sound file in the current folder to play

```
PS3="Enter a number: "; select f in *.wav;do play $f; break; done
```

Buffering audio may be useful for when the system is running many processes or otherwise has a lot of activity,

Randomly play a mp3 file

```
mpg123 "'locate -r '\.mp3$' | awk '{a[NR]=$0}END{print a["$RANDOM" %
⇒ NR]}'"
```

Play the MP3 stream at the url

```
mpg321 http://example.net/broadcast/live.mp3
```

Play the MP3 stream from a url with a 2MB audio buffer

```
mpg321 -b 2048 http://example.net/broadcast/live.mp3
```

Play the MIDI file 'copa-cabana.mid'

```
playmidi copa-cabana.mid
```

Play the MIDI file 'copa-cabana.mid' on a non-MIDI sound card

```
playmidi -f copa-cabana.mid
```

Editing Sound Files

Open the sound file 'mixdown.wav' in the program Snd

```
snd mixdown.wav
```

sound editing tools

audacity	Sound editor
jokosher	A usable sound editor

Finding Sound Files

Find 'mp3' and 'wav' files in the users folder tree

```
find ~ -iname '*.mp3' -o -iname '*.wav' | less
```

Find all audio files in the users home folder tree

```
find ~ | file -f - | grep audio | less
```

Recording Sound

Select the *LINE IN* jack as the recording source

```
amixer set Line capture
```

Select the microphone jack as the recording source

```
amixer set MIC capture
```

Record the input of your sound card into ogg file

```
rec -c 2 -r 44100 -s -t wav - | oggenc -q 5 --raw --raw-chan=2 --raw-
  => rate=44100 --raw-bits=16 - > MyLiveRecording.ogg
```

Record a WAV sample from the microphone and save it to a file 'hello.wav'

```
rec hello.wav ~ (begins an 8,000 Hz, monaural 8-bit WAV recording)
```

Make a high-fidelity recording from the microphone and save it to 'goodbye.wav'

```
rec -s w -c 2 -r 44100 goodbye.wav
```

Make a recording in CD audio format, and save the output to 'goodbye.cdr'

```
rec goodbye.cdr
```

Audio Cds

```
debian
  package: cdtool
```

Show a list of tracks on a compact disc

```
cdirc ~ (might print 'unknown cd - 43:14 in 8 tracks...')
```

Eject a CD

```
cdeject ~ (This command will also eject an unmounted CD-ROM (data CD))
```

Convert the file 'new.wav' to an audio CD format file

```
sox new.wav new.cdr
```

Use *cdinfo* to display information about an audio CD

10.1 Playing Audio Compact Disks

Play an audio CD

```
cdplay
```

Play an audio CD, beginning with the third track

```
cdplay 3
```

Play an audio CD, from the 1st track to the 4th track

```
cdplay 1 4
```

Play only the third track of an audio CD

```
cdplay 3 3
```

Pause the current CD playback

```
cdpause
```

Restart a paused CD

```
cdplay
```

Restart a paused CD from the beginning

```
cdplay x
```

Stop the current CD playback

```
cdstop
```

Use *cdplay* with the 'shuffle' argument to play the CD tracks in random

```
cdplay shuffle
```

10.2 Copying Cd Audio

Copy track 7 of audio CD to a CD-quality WAV file in current directory

```
█ cdda2wav -t7 -d0 -x -D /dev/cdrom
```

Copy all tracks on an audio CD to CD-quality CD audio-format files

```
█ cdda2wav -D /dev/cdrom -x -O cdr -d0 -B
```

Sample the third track from a scratched audio CD in the default CD-ROM drive using “paranoid” data verification, and write the output to a WAV format file in the current directory

```
█ cdparanoia -w 3-3
```

Sample the entire audio CD using “paranoid” data verification

```
█ cdparanoia -w -B
```

10.3 Burning Compact Discs

Burn 'cello.cdr' to the disc in the CD-R drive whose target ID is 2 on the primary SCSI bus

```
█ cdrecord dev=0,2,0 -audio cello.cdr
```

Burn all '.cdr' files in this folder at double speed to the CD-R drive whose target ID is 2 on the primary SCSI bus, and give verbose output

```
█ cdrecord dev=0,2,0 speed=2 -v -audio *.cdr
```

Run a test burn of 'symphony.cdr' to the disc in the CD-R drive, target ID is 6 (LUN 1) on the primary SCSI bus

```
█ cdrecord dev=0,6,1 -dummy -audio symphony.cdr
```

Burn the data track 'band-info' and all the audio tracks in the current directory with a '.cdda' extension to the CD-R drive whose target ID is 2 on the primary SCSI bus

```
█ cdrecord dev=0,2,0 band-info -audio *.cdda
```

Section 11

Converting Sound Formats

```
glame,  
mame, lame, sox
```

An important distinction in sound file formats is between the file format (wav, etc) and the encoding format (mp3, pcm, etc) The encoding format relates to the compression of each individual sample in the sound file. mpg321 and mpg123 are very similar, except that mpg321 is optimized for computers which don't have a floating point processor. Also mpg321 does not have all the functionality of mpg123.

Convert a sound file from 'wav' format to 'mp3'

```
█ lame test.wav test.mp3 ~(this is slow)
```

Convert 'sound.mp3' into a wav file 'new.wav' (a new file is created)

```
█ mpg321 -w new.wav old.mp3 ~(the file 'old.mp3' is unchanged)
```

```
█ mpg123 -w new.wav old.mp3 ~(the same)
```

Convert mp3 into mb4 (audiobook format)

```
█ mpg123 -s input.mp3 | faac -b 80 -P -X -w -o output.m4b -
```

Mp3 streaming

```
█ nc -l -p 2000 < song.mp3
```

Play random music from blip.fm

```
█ mpg123 'curl -s http://blip.fm/all | sed -e 's#"#\n#g' | grep mp3$ |  
⇒ xargs '
```

Convert wav files to flac

```
flac --best *.wav
```

Converts a single FLAC file with associated cue file into multiple

```
cuebreakpoints "$2" | shnsplit -o flac "$1"
```

An old method of converting to 'wav' format, but it doesnt work!

```
mpg321 -b 10000 -s remix.mp3 | sox -t raw -r 44100 -s -w -c 2 - remix.  
⇒ wav
```

If sox was compiled with mp3 support, you can convert with

```
sox test.mp3 test.wav ~(see above for how to compile sox)
```

```
sox -g | grep AUDIO ~(to see if sox has mp3 support)
```

Convert mp3 to ogg

```
mp32ogg file.mp3
```

Section 12

Adding Sound Effects

Listen to 'test.wav' with reverb added

```
sox test.wav -d .5 1000 100
```

Add basic reverb to file 'old.wav' and write the output to file 'new.wav'

```
sox old.wav new.wav .5 1000 100
```

Add a spacey, echoing reverb to file 'test.wav' and save to 'new.wav'

```
sox test.wav new.wav reverb 1 1000 333 333 333 333 (no ???)
```

Add a 100 millisecond echo to the 'old.wav' and write output to 'new.wav'

```
sox old.wav new.wav echo .5 .5 100 .5
```

Add a one-second echo to 'old.wav' and save changes in 'new.wav'

```
sox old.wav new.wav echo .5 .5 1000 .5
```

Add a deep, "alien-sounding" chorus to 'old.wav' and save to 'new.wav'

```
sox old.wav new.wav chorus 1 .5 100 1 5 9 -t
```

Add a subtle Vibro-Champ effect to the file 'old.wav' and write the output to 'new.wav'

```
sox old.wav new.wav vibro 1
```

Add an effect of a "maxed-out Vibro-Champ" to the file 'old.wav'

```
sox old.wav new.wav vibro 30 1
```

Add an "underwater" flange to the file 'old.wav' and save to 'new.wav'

```
sox old.wav new.wav flanger .5 .5 4 .5 1 -t
```

Add a phased "breathing" effect to 'old.wav' and save to 'new.wav'

```
sox old.wav new.wav phaser .5 .5 .5 .9 .5 -t
```

Using a decay greater than .5 may result in feedback

Add a 100 millisecond chorus to 'old.wav' and save to 'new.wav'

```
sox old.wav new.wav chorus 1 .5 100 1 1 1 -t
```

Add a "tin-can" echo effect to 'old.wav' and save in 'new.wav'

```
sox old.wav new.wav echo 1 .5 5 .5 ~(see also 'echos')
```

Add "wah-wah"-like flange to 'old.wav' and write the output to 'new.wav'

```
sox old.wav new.wav flanger .5 .5 .5 1 2 -t
```

Add a heavy phase to the file 'old.wav' and write the output to 'new.wav'

```
sox old.wav new.wav phaser 1 .5 4 .5 1 -s
```

Using The Sox Tool

Compile and install the latest version of the 'sox' sound tool

```
remove any current sox version
download sox from http://sox.sourceforge.net
unpack it with:
  tar zxvf sox-14.2.0.tar.gz
install the libmed, libmad? and lame libraries for mp3
do
  ./configure; make; make install    ##(as root)
```

See if sox has support for mp3 files

```
sox -h | grep AUDIO
```

look through the list of file types for 'mp3'

If sox has no mp3 support, just convert to 'wav' format with

```
mpg321 -w new.wav sound.mp3
```

Find the duration of the sound file 's.wav'

```
soxi s.wav
```

```
soxi s.wav | grep -i duration    ~(the same)
```

```
soxi s.wav | grep -i duration | sed 's/^[^:]*:;//;s/=.*//'    ~(just hh:mm:
=> ss)
```

Play the sound file 's.wav'

```
play s.wav
```

Make the volume level in sound files more or less the same

```
normalize
```

Reverse the sound in the file 'old.wav' and write the output to 'new.wav'

```
sox old.wav new.wav reverse
```

Fidelity Or Sound Quality

The quality or 'fidelity' of a sound file is affected by the number of samples per second, the number of channels (mono, stereo etc) the encoding (mp3, pcm, etc) and the bit size of each sample. Generally telephone sound data is of a low quality and cds high

Change the sampling rate of 'old.wav' to 7,000 Hz, and write to 'new.wav'

```
sox old.wav -r 7000 new.wav
```

Sound Volume

deb:

normalize-audio a package for 'normalising' (evening out) the sound volume in an audio file or files

In the world of sound, volume is often referred to a 'gain' and is stored as 'metadata' within the sound file, in other words, the gain or volume has nothing to do with the amplitude of the sound wave.

```
aumix, amixer, alsamixer, mplayer
```

Peruse the current mixer settings

```
amixer | less
```

Output the microphone settings

```
amixer get MIC
```

Output the second PCM settings

```
amixer get PCM,1
```

To change a mixer setting, give the amixer 'set' command as an option

Set the master volume to 75 percent

```
amixer set Master 75%
```

Set the PCM volume to 30

```
amixer set PCM 30
```

The special 'mute' and 'unmute' arguments are used for muting the

Unmute the microphone and turn it on for recording

```
amixer set MIC unmute capture
```

Mute the microphone

```
amixer set MIC mute
```

Unmute the master volume and set it to 80 percent volume

```
amixer set Master 80% unmute
```

Increase mplayer maximum volume

```
mplayer dvd:// -softvol -softvol-max 500
```

Automatically adjust the volume for a set of mp3 files in a folder

```
mp3gain -a -k *mp3
```

Automatically adjust the volume for a single mp3 sound file

```
mp3gain -r -k *mp3
```

Undo the changes made by mp3gain on the sound file 's.mp3'

```
mp3gain -u s.mp3
```

Increase the default volume (89db) by 2 decibels of the sound file 'quiet.mp3'

```
mp3gain -r -d 2.0 quiet.mp3
```

Increase the volume by a gain of 3 the sound file 'quiet.mp3'

```
mp3gain -g 3 quiet.mp3
```

Adjust the volume in sound files

```
normalize-audio
```

Amixer : raise volume and unmute if necessary

```
amixer -c 0 set Master 1+ unmute
```

Raise the volume of file 'old.wav' twofold and write the output to 'new.wav'

```
sox -v3 old.wav new.wav
```

Lower volume of file 'old.wav' by half and write the output to 'new.wav'

```
sox -v.5 old.wav new.wav
```

Raise the volume of 'quiet.cdr' as high as possible without distortion

```
sox quiet.cdr loud.cdr stat -v ~ (might print '3.125')
```


Extracting Audio From Video

Extract an audio track from a multilingual video file, for a

```
mencoder -aid 2 -oac copy file.avi -o english.mp3
```

Extract audio stream from an AVI file using mencoder

```
mencoder "${file}" -of rawaudio -oac mp3lame -ovc copy -o audio/"${file}
⇒ /%avi/mp3}"
```

Extract audio track from a video file using mencoder

```
mencoder -of rawaudio -ovc copy -oac mp3lame -o output.mp3 input.avi
```

Extract audio from Mythtv recording to Rockbox iPod using ffmpeg

```
ffmpeg -ss 0:58:15 -i DavidLettermanBlackCrowes.mpg -acodec copy
⇒ DavidLettermanBlackCrowes.ac3
```

Dump an audio stream from flv (using mplayer)

```
mplayer -dumpaudio -dumpfile test.mp3 test.flv
```

Splitting Sound Files

Copy the first 40 seconds of 'old.wav' in the sound file 'new.wav'

```
sox old.wav new.wav trim 0 40
```

Split lossless audio (ape, flac, wav, wv) by cue file

```
cuebreakpoints <cue file> | shnsplit -o <lossless audio type> <audio
⇒ file>
```

Play Lists

Add 10 random unrated songs to xmms2 playlist

```
xmms2 mlib search NOT +rating | grep -r '^ [0-9]' | sed -r 's/^( [0-9]+)
⇒ .*/\1/' | sort -R | head | xargs -L 1 xmms2 addid
```

Generate a playlist of all the files in the directory, newer first

```
find . -type f -print0 | xargs -r0 stat -c %Y\ %n | sort -rn | gawk '{
⇒ sub(/.\//, "", $2); print $2}' > /tmp/playlist.m3u
```

Synth Music

Play musical notes from octave of middle C

```
man beep | sed -e '1,/Note/d; /BUGS/, $d' | awk '{print $2}' | xargs -IX
```

Metadata

Sort your music

```
for file in *.mp3; do mkdir -p "$(mp3info -p "%a/%l" "$file")" && ln -s
⇒ "$file" "$(mp3info -p "%a/%l/%t.mp3" "$file")";done
```

Speech Synthesis

Stdin speaker via espeak

```
awk '{print}' | espeak -v pt -stdin
```

Espeak, festival Mp3gain

Internet Phone Calls

XMPP/Jingle and SIP are voice over ip protocols

One can use a 'sip' protocol client such as

```
█ ekiga, twinkle, wengophone
```

Or a proprietary application such as skype

Curiosities

What is the sound of your memory ?

```
█ sudo cat /dev/mem > /dev/dsp
```

Notes

This toggles mute on the Master channel of an alsa soundcard

```
█ amixer sset Master toggle
```

Normalize volume in your mp3 library

```
█ find . -type d -exec sh -c "normalize-audio -b \"{}\"/*mp3" \;
```