Audio unter Linux

sx Audio Debugging für OpenSUSE: https://en.opensuse.org/SDB:Audio_troubleshooting

Das Basis-Framework für Sound ist ALSA mit dem für die Soundkarte passenden Treiber. Moderne Linux Distros transportieren alle Audiodaten über den Pulseaudio Daemon als netzwerk-transparenten Datenstream. Daher kann der Pluseaudio Daemon auch auf entfernte Sound-Hardware im Netz zugreifen. Entsprechend kann man Audiodaten über ALSA oder Pulseaudio abgreifen und weiter verarbeiten. Pulseaudio abstrahiert die Audiodaten vollständig von der darunter liegenden Hardware:

Soundquelle oder -Ziel → PulseAudio → ALSA-Treiber → Soundkarte

Jedes Soundgerät ist entweder Quelle (Source) oder Senke (Sink) für Audiodaten.

Aufnehmen auf der Kommandozeile

Mit arecord/aplay

Auflisten aller Mikrofone

```
arecord -l

**** Liste der Hardware-Geräte (CAPTURE) ****
Karte 1: PCH [HDA Intel PCH], Gerät 0: ALC1150 Analog [ALC1150 Analog]
  Sub-Geräte: 1/1
  Sub-Gerät #0: subdevice #0
Karte 1: PCH [HDA Intel PCH], Gerät 2: ALC1150 Alt Analog [ALC1150 Alt
Analog]
  Sub-Geräte: 1/1
  Sub-Gerät #0: subdevice #0
Karte 3: CinemaTM [Microsoft® LifeCam Cinema(TM)], Gerät 0: USB Audio [USB
Audio]
  Sub-Geräte: 1/1
  Sub-Gerät #0: subdevice #0
Karte 4: MS [Sennheiser USB-ED CC 01 for MS], Gerät 0: USB Audio [USB Audio]
  Sub-Geräte: 1/1
  Sub-Gerät #0: subdevice #0
```

Aufnahme im Sample Format S16_LE (16 bit little endian, 44100, stereo) in zwei Kanälen (-c) mit Rate 48000 Hertz (-r) für 20 Sekunden (-d) von einem bestimmten Mikrofon (–device)

```
arecord -f S16_LE -c 2 -r 48000 -d 20 --device="hw:4,0" tmp/test.wav
```

Mit Pulseaudio

pactl: Control a running PulseAudio sound server

list [short] [TYPE]

```
            Dump all currently loaded modules, available sinks, sources,
streams, etc. TYPE must be one of: modules,
            sinks,  sources,  sink-inputs,  source-outputs,  clients,
samples, cards. If not specified, all info is
            listed. If short is given, output is in a tabular format, for
easy parsing by scripts.
```

Oder über pacmd, Auflisten der Devices

```
thommie@locutus:~> pacmd list-sources | egrep '^\s+name:.*\.monitor'
        name: <alsa_output.usb-Sennheiser_Sennheiser_USB-
ED_CC_01_for_MS_A001010202601978-00.analog-stereo.monitor>
        name: <alsa_output.pci-0000_00_1b.0.analog-stereo.monitor>
        name: <combined.monitor>
        name: <ladspa_output.mbeq_1197.mbeq.monitor>
```

Aufnehmen

```
parecord --channels=1 -d STREAM_NAME filename.wav
```

Aufnehmen aus einer bestimmten Applikation

(ohne Störung durch Notification aus Plasma/KDE usw.)

Auflisten der Quellen

```
pacmd list-sinks | egrep '^\s+name: .*alsa_output'
```

At a high level, here is what you are going to do:

1. Create a “null” sink that you will be recording. Let’s call it `recording`.
2. Create a combined sink that will send its input to both headphones and the `recording` sink. Otherwise, you will be able to record a stream but not hear it yourself. So, let’s call this sink `combined`.
3. Direct the sound from the specific applications you want to record into the combined sink.
4. Record the monitor of the `recording` sink to a file.

Find out the name of your output device by running

```
''pacmd list-sinks | egrep '^\s+name: .*alsa_output'''
```

In my case, it says

```
''name: <alsa_output.pci-0000_00_1f.3.analog-stereo>''
```

To create the two sinks, run these commands (you need to substitute the name of the output device that you learned on the previous step):

```
''pacmd load-module module-null-sink sink_name=recording
sink_properties=device.description=recording
pacmd load-module module-combine-sink sink_name=combined
sink_properties=device.description=combined \
  slaves=recording,alsa_output.pci-0000_00_1f.3.analog-stereo''
```

If you do this often, make it a permanent setup by creating the file `~/.config/pulse/default.pa` with this contents (note that the long `load-module` command should be on a single line, don't wrap it):

```
''.include /etc/pulse/default.pa
load-module module-null-sink sink_name=recording
sink_properties=device.description=recording
load-module module-combine-sink sink_name=combined
sink_properties=device.description=combined
slaves=recording,alsa_output.pci-0000_00_1f.3.analog-stereo''
```

Now, redirect the sound to the `combined` sink:

1. Run the `pavucontrol` command (a graphical window will appear) and go to the "Playback" tab.
2. Start the application you'd like to record.
3. The application should appear in `pavucontrol`. If it doesn't, make sure the application produces some sound. Unfortunately, until the application tries to play something, PulseAudio cannot "see" it.

Quelle: https://ro-che.info/articles/2017-07-21-record-audio-linux

pipewire als pulseaudio Ersatz

https://cubiclenate.com/2021/07/17/pipewire-audio-server-on-opensuse-tumbleweed/

Pakete nachinstallieren

```
sudo zypper in pipewire pipewire-pulseaudio pipewire-alsa''
```

Konflikt

```
Problem: the installed pulseaudio-15.0-6.1.x86_64 conflicts with
'pulseaudio-daemon' provided by the to be installed pipewire-
```

```
pulseaudio-0.3.42-1.1.x86_64
 Solution 1: Following actions will be done:
  deinstallation of pulseaudio-15.0-6.1.x86_64
  deinstallation of pulseaudio-module-bluetooth-15.0-6.1.x86_64
  deinstallation of pulseaudio-module-gsettings-15.0-6.1.x86_64
  deinstallation of pulseaudio-module-x11-15.0-6.1.x86_64
  deinstallation of pulseaudio-module-zeroconf-15.0-6.1.x86_64
  deinstallation of pulseaudio-lang-15.0-6.1.noarch
 Solution 2: do not install pipewire-pulseaudio-0.3.42-1.1.x86_64
```

Diese Pakete werden deinstalliert

```
The following 7 packages are going to be REMOVED:
  alsa-plugins-pulse pulseaudio pulseaudio-lang pulseaudio-module-bluetooth
pulseaudio-module-gsettings
  pulseaudio-module-x11 pulseaudio-module-zeroconf
```

Tuning: https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Migrate-PulseAudio

Crackling/Popping issues on intel-hda soundcard
https://gitlab.freedesktop.org/pipewire/pipewire/-/issues/1569

Neustart Audio-System (z.B: bei Erkennungsproblem Bluetooth Audio):

```
systemctl restart sound.target
```

Audio Cleanup

- Step one: NOISE REDUCTION
- Step two: COMPRESSION
- Step three: LIMITER FILTER
- Step four: EQUALIZER
- Step five: NORMALIZE

From:
https://wiki.netzwissen.de/ - **netzwissen.de Wiki**

Permanent link:
https://wiki.netzwissen.de/doku.php?id=audio

Last update: **17/08/2024 - 07:06**