

ffmpeg and audio manipulation

A .wav file (**Waveform Audio File Format** (also called WAVE)) is an uncompressed audio file format. In order to deal with the long recordings of underwater activity we need to cut the audio files into smaller bits.

1- Processing audio with ffmpeg

ffmpeg is a command-line tool that converts audio or video formats. It can also capture and encode in real-time from various hardware and software sources such as a TV capture card.

ffmpeg can be installed with:

```
sudo apt install ffmpeg
```

and check the installation with:

```
ffmpeg -version
```

The ffmpeg commands can be used as bash or called in Python via Subprocess.

Below is the code for cutting a .wav file into enumerated 1min pieces:

```
ffmpeg -i somefile.wav -f segment -segment_time 60 -c copy out%03d.wav
```

The segment_time specifies the time in seconds (here 60sec for 1min. pieces). For a file which is 5min30 seconds long, there will be a last .wav file of 00:30sec duration.

ffmpeg can also be used to record audio. For this, the devices that can record or output audio should be listed:

on Windows the argument "dshow" is used to list devices:

```
ffmpeg -list_devices true -f dshow -i dummy
```

The retrieved device name can be used in the following command to record the audio as a .wav file:

```
ffmpeg -f dshow -i audio="Device name" path-to-file\file-name.wav
```

On Linux machines (which the actual topic for our project) the audio input devices can be listed with (option L is to list (it is not 1 🤔)):

```
arecord -l
```

Then the retrieved device name (let's say it is Subdevice #0) can be used in the following command:

```
ffmpeg -f alsa -i hw:0 alsaout.wav
```

The output file path can be replaced.

Creating recordings of 10 second segments:

```
ffmpeg -f alsa -i hw:0 -segment_time 10 -f segment %03d.wav
```