

# FFmpeg Tips and Tricks

Adrian Kosmaczewski

2021-10-15

Here's a small selection of cool things you can do on Linux with FFmpeg.

- Convert Between Video Formats
- Create Animated GIFs
- Trim Video
- Make Video Smaller
- Make a Video compatible with QuickTime
- Make Videos Compatible with Windows Media
- Split Video
- Concatenate Video Files
- Better Concatenation
- Record your Webcam

## Convert Between Video Formats

The most basic, simple, and useful thing to do with FFmpeg:

```
$ ffmpeg -i input.m4v output.mp4
```

For example, to convert MOV to MP4

```
$ ffmpeg -i movie.mov -vcodec copy -acodec copy out.mp4
```

You can also use Handbrake to drive these conversions through a GUI.

## Create Animated GIFs

There are two ways to do this: first by generating the frames in a subfolder, and then concatenating everything with `convert`:

```
$ ffmpeg -i video.mp4 -r 5 'frames/frame-%03d.jpg'  
$ convert -delay 20 -loop 0 frames/*.jpg myimage.gif
```

Where `-r` in the first command stands for the frames per second caught in the GIF.

But can do this without `convert` in one shot:

```
$ ffmpeg -i portal.mp4 -r 5 portal.gif
```

## Trim Video

Generate a subclip of a video, removing frames before and after some reference points:

```
$ ffmpeg -i input.mp4 -ss 00:01:15 -t 00:00:30 -async 1 output.mp4
```

- The `-ss` parameter is the starting point.
- The `-t` provides the length of the clip.

## Make Video Smaller

Smaller in bytes:

```
$ ffmpeg -i input.mp4 -vcodec libx265 -crf 20 output.mp4
```

Smaller in width and height:

```
$ ffmpeg -i input.avi -s 720x480 -c:a copy output.mkv
```

## Make a Video compatible with QuickTime

This should make a video that is Mac-friendly:

```
$ ffmpeg -i input.webm -f mp4 -vcodec libx264 \
    -pix_fmt yuv420p output.mp4
```

## Make Videos Compatible with Windows Media

This creates an AVI file that is compatible... with Windows 95!

```
$ ffmpeg -i input.mp4 -vcodec msvideo1 -acodec adpcm_ms \
    -vf scale=320:240 -f avi output.avi
```

- The `scale` information is required, and width and height must be multiples of 4 for the `msvideo1` encoder.
- The `avi` parameter selects the container format.

## Split Video

As the title implies:

```
ffmpeg -i largefile.mp4 -t 00:50:00 -c copy smallfile1.mp4 \
    -ss 00:50:00 -c copy smallfile2.mp4
```

## Concatenate Video Files

The inverse operation as above; create a `playlist.txt` file with this structure:

```
# Playlist
file 'part-1.mkv'
```

```
file 'part-2.mkv'
file 'part-3.mkv'
```

And then concatenate all movies together:

```
$ ffmpeg -f concat -safe 0 -i playlist.txt -c copy output.mkv
```

To make sure the audio works in QuickTime / iOS when using a playlist:

```
ffmpeg -f concat -safe 0 -i playlist.txt -c:v copy -c:a aac output.mp4
```

## Better Concatenation

The script below performs a better concatenation using intermediate files.

```
#!/usr/bin/env bash
```

```
# This script merges all video files into one.
```

```
# It requires ffmpeg:
```

```
# Linux: sudo apt install ffmpeg
```

```
# Mac: brew install ffmpeg
```

```
ffmpeg -i part0.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts int0.ts
ffmpeg -i part1.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts int1.ts
ffmpeg -i part2.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts int2.ts
ffmpeg -i part3.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts int3.ts
ffmpeg -i part5.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts int5.ts
ffmpeg -i part6.mp4 -c copy -bsf:v h264_mp4toannexb -f mpegts int6.ts
ffmpeg -i "concat:int0.ts|int1.ts|int2.ts|int3.ts|int5.ts|int6.ts" \
    -c copy -bsf:a aac_adtstoasc output.mp4
rm *.ts
```

Concatenating video files requires the audio to be in the same format, for example in AAC:

```
$ ffmpeg -i input.mp4 -c:v copy -c:a aac output.mp4
```

## Record your Webcam

This command records your webcam and generates a video file:

```
$ ffmpeg -f oss -i /dev/dsp -f video4linux2 -s 320x240 \
    -i /dev/video0 out.mpg
```

The `/dev/video0` part should match your own webcam; to find it, for example if your laptop has an integrated webcam but you also have a USB Logitech camera connected:

```
$ sudo apt install v4l-utils
$ v4l2-ctl --list-devices
```

Logitech Webcam C930e (usb-0000:00:14.0-3.1):  
/dev/video2  
/dev/video3

Integrated Camera: Integrated C (usb-0000:00:14.0-8):  
/dev/video0  
/dev/video1