How to trim a video using FFmpeg

A common feature of video editing applications is the ability to cut/trim videos. When editing a video, you might want to cut out some parts or you might want to stitch together different videos by cutting sections from different sources and concatenating them into a single video.

We'll see how to do exactly that using FFmpeg — a command-line utility that can be used to create, edit and process different types of media. We'll look at some FFmpeg commands that you can use to trim a video into different parts and then see how you can take different videos and concatenate them into a single video. We'll finish off by looking at how you can achieve the same results using Shotstack — a cloud-based video editing API.

BY JOYCE ECHESSA

27th June, 2021

Cut/trim a video with FFmpeg

FFmpeg offers different commands that you can use to split up a video. We'll take a look at how to use the seeking parameter `-ss`, but you can also use other commands such as the trim filter.

To cut a specific part of a video, you use the seeking option `-ss` to get to a specific part that you want to cut. `-ss` can be used in different ways, depending on how you want to cut the video. Let's take a look at some examples.

Cut using a duration

```
$ ffmpeg -i input.mp4 -ss 00:05:20 -t 00:10:00 -c:v copy -c:a copy output1.mp4
```

The above command will take the input video `input.mp4`, and cut out 10 minutes from it starting from `00:05:20` (5 minutes and 20 second mark), i.e. the output video will be from `00:05:20` to `00:15:20`.

`-ss` specifies the starting position and `-t` specifies the duration from the start position. In the above command, we cut 10 minutes from the `00:05:20` mark.

The `-c:v copy -c:a copy` commands copy the original audio and video without re-encoding.

To specify time, you can use two different time unit formats: sexagesimal (`HOURS:MM:SS.MILLISECONDS`, e.g. `01:23:45.678`), or in seconds. If using the former, you can leave out the milliseconds `HOURS:MM:SS` as we did in our example.

If you specify a duration that will result in a stop time that is beyond the length of the input video, the output video will end where the input video ends.

Cut using a specific time

```
$ ffmpeg -i input.mp4 -ss 00:05:10 -to 00:15:30 -c:v copy -c:a copy output2.mp4
```

The above command uses `-to` to specify an exact time to cut to from the starting position.

The cut video will be from `00:05:10` to `00:15:30`, resulting in a 10 minutes and 20 seconds video.

If you specify a time `-to` that is longer than the input video, e.g. `-to 00:35:00` when the input video is 20 minutes long, the cut video will end where the input video ends. If you specify a `-to` that is smaller than `-ss`, then the command won't run. You'll get the following error: `Error: -to value smaller than -ss; aborting.`

Note that if you specify `-ss` before `-i`, `-to` will have the same effect as `-t`, i.e. it will act as a duration.

```
$ ffmpeg -ss 00:05:20 -i input.mp4 -t 00:10:00 -c:v copy -c:a copy output3.mp4

$ ffmpeg -ss 00:05:20 -i input.mp4 -to 00:10:00 -c:v copy -c:a copy output4.mp4
```

The above commands both result in videos that are 10 minutes long that are from `00:05:20` to `00:15:20` of the input video.

When using seek, you might have noticed that sometimes the output files might not be the exact length you were expecting, they might be off by a few seconds. For most video formats, it's not possible to seek exactly. FFmpeg seeks to the closest seek point before the position that you specified. Accuracy can be improved by transcoding the video with `-accurate_seek` enabled. With this, the extra segment between the seek point and the position specified will be decoded and discarded. When `-noaccurate_seek` is used, it will be preserved.

Cut the end of a video

The seeking command has another variant `-sseof` that you can use to cut the last `N` seconds from a video. It uses negative values to indicate positions relative to the EOF (end of file). Position `0` is at EOF.

```
$ ffmpeg -sseof -600 -i input.mp4 -c copy output5.mp4

$ ffmpeg -sseof -00:10:00 -i input.mp4 -c copy output6.mp4
```

Both of the above two commands will make a cut of the last 10 minutes of the input video.

If you use a time that is longer than the input video, e.g. `-01:10:00` for a 20 minute video, the command will still run. The output video will be the same length as the input.

Cut with re-encoding

When you leave out the `-c copy` option when trimming a video, FFmpeg will automatically re-encode the output video and audio according to the format you chose. The operation will take longer to complete compared to the previous commands that we've looked at but will give a more frame-accurate result.

```
$ ffmpeg -ss 00:05:20 -accurate_seek -i input.mp4 -t 00:10:00 -c:v libx264 -c:a aac out
```

In the above command, we re-encode the audio and video when cutting the video and we also use the `-accurate_seek` flag which will result in a more accurate length for the output video.