

ffmpeg

ffmpeg is a integrated tool for video encoding/decoding processing as open source project - <http://ffmpeg.org>

Typical syntax of ffmpeg command is like below:

```
ffmpeg [global_options] {[input_file_options] -i input_url} ... {[output_file_options] output_url} ...
```

1. Get Video Information

```
$ ffmpeg -i video.mp4
```

2. Get Video Information without banner

```
$ ffmpeg -i video.mp4 -hide_banner
```

3. Converting video to mp4

```
$ ffmpeg -i video.avi video.mp4
```

4. Converting video to mkv

```
$ ffmpeg -i video.avi video.mkv
```

5. Converting video to mp4 without any quality loss

```
$ ffmpeg -i input.webm -qscale 0 output.mp4
```

6. Displaying the supported formats by ffmpeg

```
$ ffmpeg -formats
```

7. Converting video files to audio files

```
$ ffmpeg -i input.mp4 -vn -ab 320 output.mp3
```

8. Change resolution of video files

```
$ ffmpeg -i input.mp4 -filter:v scale=1280:720 -c:a copy output.mp4
```

or

```
$ ffmpeg -i input.mp4 -s 1280x720 -c:a copy output.mp4
```

9. Compressing video files

```
$ ffmpeg -i input.mp4 -vf scale=1280:-1 -c:v libx264 -preset veryslow -crf 24 output.mp4
```

Note crf value 24 is too aggressive, you'd better lower its value,

10. Compressing audio files

```
$ ffmpeg -i input.mp3 -ab 128 output.mp3
```

11. Extracting images from video

```
$ ffmpeg -i input.mp4 -r 1 -f image2 image-%2d.png
```

- -r – Set the frame rate. I.e the number of frames to be extracted into images per second. The default value is 25.
- -f – Indicates the output format i.e image format in our case.
- image-%2d.png – Indicates how we want to name the extracted images. In this case, the names should start like image-01.png, image-02.png, image-03.png and so on. If you use %3d, then the name of images will start like image-001.png, image-002.png and so on.

12. Split video files into multiple parts

```
$ ffmpeg -i input.mp4 -t 00:00:30 -c copy part1.mp4 -ss 00:00:30 -codec copy part2.mp4
```

Below is an advanced version of splitting video files showing the way to implement one command instead of multiple command even though that does not reduce efforts to type the necessary options.

```
echo "Two commands"
time ffmpeg -v quiet -y -i input.ts -vcodec copy -acodec copy -ss 00:00:00 -t 00:30:00 -sn test1.mkv
time ffmpeg -v quiet -y -i input.ts -vcodec copy -acodec copy -ss 00:30:00 -t 01:00:00 -sn test2.mkv
echo "One command"
time ffmpeg -v quiet -y -i input.ts -vcodec copy -acodec copy -ss 00:00:00 -t 00:30:00 \
-sn test3.mkv -vcodec copy -acodec copy -ss 00:30:00 -t 01:00:00 -sn test4.mkv
```

13. Joining multiple video parts into one

```
$ ffmpeg -f concat -i join.txt -c copy output.mp4
```

For your information, join.txt contains following video files

```
C:\Users\LifePlanet\Downloads\part1.mp4
C:\Users\LifePlanet\Downloads\part2.mp4
C:\Users\LifePlanet\Downloads\part3.mp4
C:\Users\LifePlanet\Downloads\part4.mp4
```

14. Adding subtitles to a video file

```
$ ffmpeg -i input.mp4 -i subtitle.srt -map 0 -map 1 -c copy -c:v libx264 -crf 23 -preset veryfast output.mp4
```

15. Adding multi subtitles to a video file

```
$ ffmpeg -i $movie.mov -i $sub_en.srt -i $sub_de.srt -map 0:v -map 0:a -map 1 -map 2 \  
-c:v copy -c:a copy -c:s srt \  
-metadata:s:s:0 language=eng -metadata:s:s:1 language=ger \  
$output.mkv
```

- [Add an image overlay in front of video using ffmpeg](#) – ffmpeg enables you to overlay image on top of the video.
- [Burnt-in SRT subtitle based on ffmpeg](#) – ffmpeg provides the straight and easy way to burn-in SRT subtitle. By force_style, you can customize your subtitle format like font, fontsize, outline, outlinecolor, borderstyle and so on. The disadvantage of burnt-in caption is that can make bad TV experience for those who does not want to see the subtitle on top of the screen. On the contrary, the benefit of the burnt-in subtitle is that doesn't require any additional technical tools/modules on the client.
- [CEA-608/708 Closed Caption](#) – CEA-608/708 captions are the standard for Digital Video Broadcasting supporting closed caption as a part of video stream. CEA-608 captions are for NTSC videos only and its captions are encoded in the signal's VBI for NTSC. CEA-708 captions are considered for digital television only and are encoded directly into the MPEG data packages. The captions follow the CEA-708-D FCC recommendations.
- [Creating a master HLS playlist.m3u8 in FFmpeg](#) – FFmpeg kindly provides a straightforward way can create master HLS playlist. You can simply do it by adding master_pl_name as a part of ffmpeg encoding options. Make sure to require a closed GOP when encoding and to set the GOP size to fit your segment time constraint. For more information, please refer at <https://ffmpeg.org/ffmpeg-formats.html#hls> <https://ffmpeg.org/ffmpeg-formats.html#hls>
- [Encode mp4 with multi subtitles in ffmpeg](#) – ffmpeg can embed subtitles as a part of tracks in a single file.
- [Encoding videos in .ts by FFmpeg](#) – Below example shows how to encode video in .ts by FFmpeg
- [ffmpeg overlay effect with fade in/out and alphablending](#) – ffmpeg has a feature to overlay image on top of the video with fade-in and fade-out. It also able to combine with alpha value, so its interaction could be more smoother.
- [ffmpeg with NVIDIA CUDA GPU](#)
- [Installing FFmpeg on CentOS 7](#) – sdsds
- [Splitting video file by size in bash](#)