FFmpeg - The Media Swiss Army Knife

Jess Portnoy jess.portnoy@kaltura.com, Kaltura, Inc

Abstract

FFmpeg is a FOSS, cross-platform, solution to record, convert and stream audio and video. This session will focus on using the CLI tools included in this project [ffmpeg and ffmprobe] to accomplish everyday video manipulation and streaming tasks.

Session Overview

The session will include a live demo showing how to achieve the following goals:

- Extract metadata from video and audio files
- Convert between different media types
- Live Stream over RTMP
- Record screen display
- Basic editing operations [concatenation of several video files into one, extracting the audio track from a video, etc]

Installation

FFmpeg is written in C.
Being FOSS [licensed under LGPLv2.1], it can be compiled from source for most modern OSes.
The FFmpeg site also includes download links for binaries built for Linux, Mac and Windows.

Many Linux distros include ffmpeg packages in their official repos.

Important note about compilation and distribution:

FFmpeg incorporates several optional parts and optimizations that are covered by the GNU General Public License (GPL) version 2 or later. If those parts get used the GPL applies to all of FFmpeg.

Installation - cont'd

Your choice of FFmpeg distribution will vary depending on your needs.
If your OS [or your specific distribution in the case of Linux], provides official packages that your project can work against, then it's probably the best way to go.

However, there are situations in which you will require a specific FFMpeg version or options that will warrant a custom build.

ffprobe

ffprobe gathers information from multimedia streams and prints it in human- and machine-readable fashion.
This is a very handy tool if your application handles different media files and needs to analyse their contents.

For example it can be used to check the format of the container used by a multimedia stream and the format and type of each media stream contained in it.

ffprobe CLI example

```
$ ffprobe -v error -show_format -show_streams \
/path/to/media/file
```

Truncated output:

```
[STREAM]
codec_name=h264
codec_long_name=H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10
codec_type=video
codec_tag_string=avc1
width=1280
height=720
display_aspect_ratio=16:9
pix_fmt=yuv420p
```

ffmpeg CLI - extract audio track

```
$ ffmpeg -i /path/to/vid/file -ss 00:03:05 -t 00:00:45.0
-q:a 0 -map a /tmp/out_audio.mp3
```

Flag explanation:

> -ss: the starting timestamp
>
> -t: duration
>
> -q🅰 alias for -qscale:a
>
> -map a: designate the audio input streams as a source for the output file

Quality values are encoder specific, so for libmp3lame the range is 0-9 where a lower value is a higher quality. `-q:a 0` means the highest quality - VBR from 22 to 26 KB/s.

ffmpeg CLI - captions

A simple example:

```
$ ffmpeg -i /path/to/mp4 \
 -i /path/to/srt -c copy -c:s mov_text \
 /tmp/outfile.mp4
```

This will copy /path/to/mp4 onto /tmp/outfile.mp4 while embedding /path/to/srt into it.

ffmpeg CLI - captions cont'd

Convert SRT to ASS:

```
$ ffmpeg -i /path/to/srt /tmp/horse.ass
```

Create an MKV from the original MP4 file and embed ASS caption file into the video:

```
$ ffmpeg -i /path/to/source/mp4 -i \
/tmp/horse.ass -c:v copy -c:a copy -c:s copy \
-y /tmp/out.mkv
```

ffmpeg CLI - live streaming your screen display

Stream screen display over RTMP:

```
ffmpeg -f x11grab -s 1920x1080 -framerate 15 -i \
 :0.0 -c:v libx264 -preset fast -pix_fmt yuv420p \
 -s 1280x800 -threads 0 -f flv $STREAM_URL
```

> -i :0.0: 0.0 is display.screen number of the X11 server [same as $DISPLAY ENV var]
>
> -f x11grab: grab the X11 display with ffmpeg
>
> -s 1920x1080: set frame size to 1920x1080
>
> -c:v libx264: transcode using libx264
>
> -pix_fmt yuv420p: set pixel format to yuv420p

Thank you && Questions

Appendix - Useful Resources

- FFmpeg's index page
- FFmpeg License and Legal Considerations
- FFmpeg's documentation
- ffprobe
- FFmpeg MP3 Encoding Guide
- Comparison of video container formats
- Streaming Guide
- Statically linked FFmpeg build