

## FFmpeg: 视频转码、剪切、合并、播放速调整

2016-01-08

前阵子帮导师处理项目 ppt, 因为插入视频的格式问题被搞得焦头烂额。Windows 内置解码器不够全, PowerPoint 原生支持的内嵌视频格式有限。更麻烦的是新版本的 PowerPoint 支持的格式 (比如 MPEG4 编码的 .mp4) 旧版本还不支持, 而你对手头的 ppt 即将播放的场所用什么系统什么版本的软件又无能为力, 所以最稳妥的方式是, 把所有视频转为基本所有版本的 Windows 和 Office 都支持的格式——wmv。

我用过各种 online 的网页版转码工具, 需要上传下载太麻烦, 还有文件大小限制。Windows 下面的 Movie Maker 免费, 勉强可用, 还是不顺手, 而且除了 4:3 和 16:9, 没法直接输出和原视频相同的画面比例。

之后在 Linux 下用 RecordMyDesktop 录屏, 也面临把 .ogv 格式的视频转成其他格式, 以及剪切视频片段的问题。在 Ubuntu Software Center 下载推荐度比较高的两个软件, 还是不顺手, 最终发现了终极解决方案: ffmpeg。

### 安装

去[官网](#)按提示安装即可, 支持三大操作系统。唯一要吐槽的是, Ubuntu 14.04 等较新的版本, 从默认软件列表里移除了 ffmpeg, 转而支持 libav, 可是 libav 又没弄好, 很难用——开源社区瞎折腾起来真是无力吐槽。

2017 更新: Ubuntu 16.04, FFmpeg 又回到了官方仓库:

```
sudo apt install ffmpeg
```

## 转码

最简单命令如下:

```
ffmpeg -i out.ogv -vcodec h264 out.mp4
ffmpeg -i out.ogv -vcodec mpeg4 out.mp4
ffmpeg -i out.ogv -vcodec libxvid out.mp4
ffmpeg -i out.mp4 -vcodec wmv1 out.wmv
ffmpeg -i out.mp4 -vcodec wmv2 out.wmv
```

-i 后面是输入文件名。-vcodec 后面是编码格式, h264 最佳, 但 Windows 系统默认不安装。如果是要插入 ppt 的视频, 选择 wmv1 或 wmv2 基本上万无一失。

附加选项: -r 指定帧率, -s 指定分辨率, -b 指定比特率; 于此同时可以对声道进行转码, -acodec 指定音频编码, -ab 指定音频比特率, -ac 指定声道数, 例如

```
ffmpeg -i out.ogv -s 640x480 -b 500k -vcodec h264 -r 29.97 -acodec libf
```

## 剪切

用 -ss 和 -t 选项, 从第 30 秒开始, 向后截取 10 秒的视频, 并保存:

```
ffmpeg -i input.wmv -ss 00:00:30.0 -c copy -t 00:00:10.0 output.wmv  
ffmpeg -i input.wmv -ss 30 -c copy -t 10 output.wmv
```

达成相同效果，也可以用 `-ss` 和 `-to` 选项，从第 30 秒截取到第 40 秒：

```
ffmpeg -i input.wmv -ss 30 -c copy -to 40 output.wmv
```

值得注意的是，ffmpeg 为了加速，会使用关键帧技术，所以有时剪切出来的结果在起止时间上未必准确。通常来说，把 `-ss` 选项放在 `-i` 之前，会使用关键帧技术；把 `-ss` 选项放在 `-i` 之后，则不使用关键帧技术。如果要使用关键帧技术又要保留时间戳，可以加上 `-copyts` 选项：

```
ffmpeg -ss 00:01:00 -i video.mp4 -to 00:02:00 -c copy -copyts cut.mp4
```

参考：[ffmpeg.org](https://ffmpeg.org)

## 合并

把两个视频文件合并成一个。

简单地使用 `concat` demuxer，示例：

```
$ cat mylist.txt  
file '/path/to/file1'  
file '/path/to/file2'  
file '/path/to/file3'
```

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

更多时候, 由于输入文件的多样性, 需要转成中间格式再合成:

```
ffmpeg -i input1.avi -qscale:v 1 intermediate1.mpg  
ffmpeg -i input2.avi -qscale:v 1 intermediate2.mpg  
cat intermediate1.mpg intermediate2.mpg > intermediate_all.mpg  
ffmpeg -i intermediate_all.mpg -qscale:v 2 output.avi
```

参考: [stackoverflow](#); [ffmpeg.org](#)。

## 调整播放速度

加速四倍:

```
ffmpeg -i TheOrigin.mp4 -vf "setpts=0.25*PTS" UpTheOrigin.mp4
```

四倍慢速:

```
ffmpeg -i TheOrigin.mp4 -vf "setpts=4*PTS" DownTheOrigin.mp4
```

参考: [bogotobogo](#)。

使用愉快!

标签: [推荐](#) [FFmpeg](#)

留言请用 [Github Issues](#)

聊天请在 [Gitter/fan-farm](#)

授权协议 (CC) BY-NC-SA | 订阅 [RSS](#) | 邮箱 [hi@fzheng.me](mailto:hi@fzheng.me)