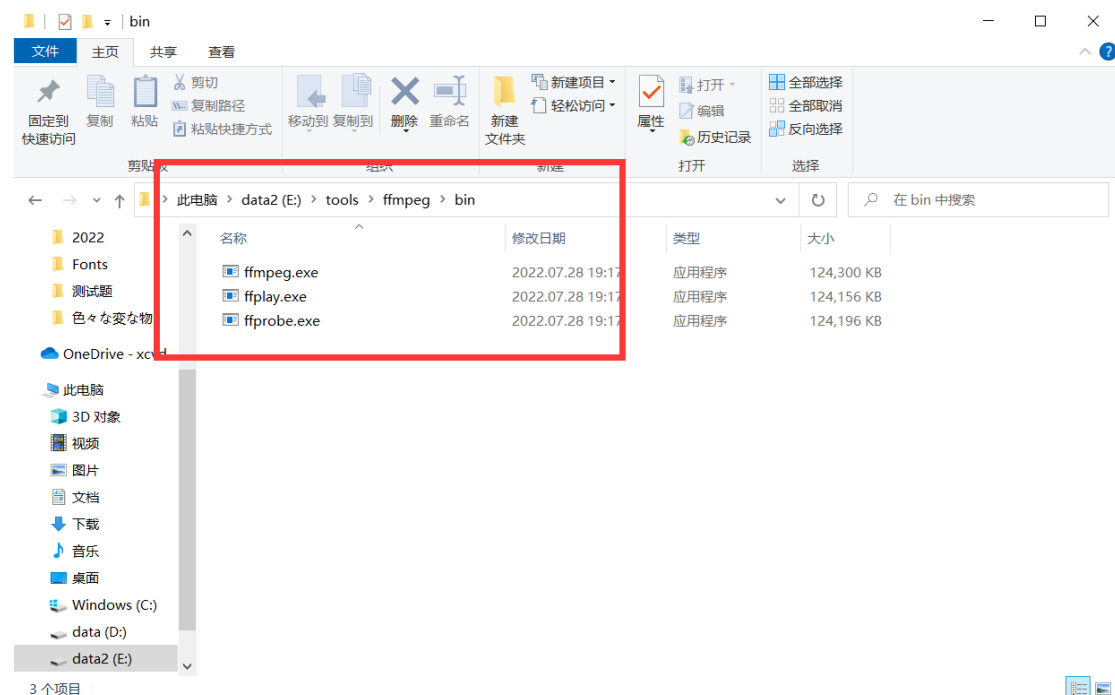


ffmpeg 的安装与使用

在阅读此文之前，请先阅读 VCB-S 的[科普教程](#)

Windows 下的安装

下载 [ffmpeg](#)，将其解压到一个不会去移动/删除的文件夹，并记住其子文件夹「bin」的路径



首先，右键「此电脑」，点击「属性」，找到「高级系统设置」，打开「环境变量」



Linux 下的安装

使用 apt / yum 即可

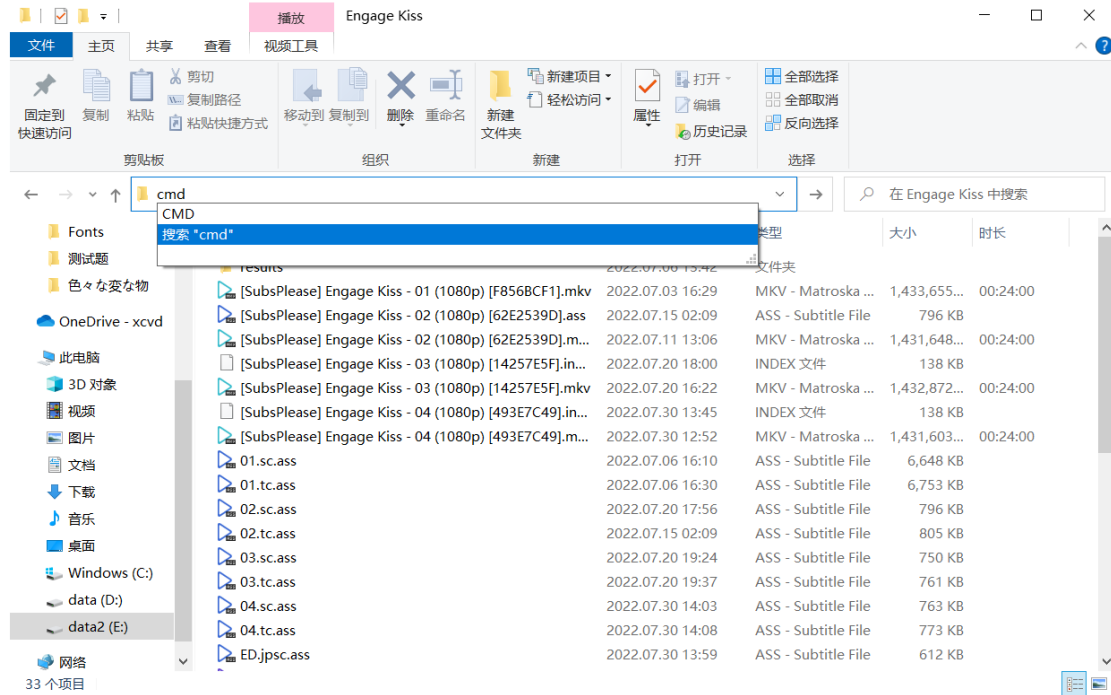
```
sudo apt-get install ffmpeg
```

```
sudo yum -y install ffmpeg
```

使用

首先我们需要理解「命令行工具」，就是只能用命令行调用的工具；在 Windows 下通过 cmd 或者 PowerShell 调用，在 Linux / macOS 下通过终端调用

附：Windows 下快速在一个文件夹打开 cmd 窗口的方法



ffmpeg 就是一个命令行工具，其基本语法为：

```
ffmpeg [options] [[infile options] -i infile]... {[outfile options] outfile}...
```

常用参数有：

- c[:a|v|s] 指定编/解码器
- b[:a|v] 制定音频/视频码率
- vf 指定滤镜
- an -vn 使输出文件不带音轨/视频轨
- ar 指定音频采样率
- map 制定输入文件中的流
- etc. （详见帮助文档或者使用必应/谷歌等搜索引擎搜索）

例如，想要将一个包含 flac 音轨的 114514.mkv 视频文件的音轨压缩至 aac 256Kbps，视频轨保持不变，输出至 1919810.mp4，可以使用指令：

```
ffmpeg -i 114514.mkv -c:a aac -b:a 256k -c:v copy 1919810.mp4
```

（需要注意的是，mp4 不能封装 flac 音轨；虽然 ffmpeg 的实验性功能提供将其封装进去的选项，但是很有可能出现兼容性问题；如需封装无损音轨，请使用 alac）

如果想要将一个包含两个音频轨的 114514.mkv 的第二条音频轨拆出来，应先使用

```
ffprobe 114514.mkv
```

查询该音频轨的编码和流序号；假设查询得知此音频为 aac 编码，编号为#0:2，那么可以使用指令：

```
ffmpeg -i 114514.mkv -map 0:2 -c copy 114514.aac
```

```
ffmpeg -i 114514.mkv -map 0:2 -c:a copy -vn 114514.m4a
```

```
ffmpeg -i 114514.mkv -map 0:2 -c:a copy -vn 114514.mka
```

既可以将其封装进 aac 文件，也可以封装进 m4a 和 mka

如果需要拆的流为默认轨道，或者文件中只有一个视频轨/音频轨，那么可以将-map 去掉；抽取字幕流/视频流同理

如果需要将音频 1.flac 从 0:1:1.400 到 0:4:05.14 截取出来，并保存为 2.flac，可以使用指令：

```
ffmpeg -i 1.flac -ss 0:1:1.4 -to 0:4:05.14 2.flac
```

```
ffmpeg -ss 0:1:1.4 -to 0:4:05.14 -i 1.flac 2.flac
```

也可以使用-t 取代-to，其参数变为持续时间

将-ss，-to，-t 放在-i 前后的区别是，如果放在前面，ffmpeg 将直接从这个时间点开始解码编码，直至结束的时间点；如果放在后面，ffmpeg 将从头开始解码，从指定的开始时间点开始编码，直至结束的时间点

需要注意的是，当使用-c:v copy 剪切视频时，由于 H264 (AVC) 与 H265 (HEVC)存在「关键帧」的概念，因此必须精确地从关键帧开始切割，否则：

如果将-ss 参数放在-i 后面，从视频开始直至指定的时间点之后的第一个关键帧，画面都会是静止；

如果将-ss 参数放在-i 前面，剪切出来的视频将会提前一段时间开始，并且如果指定了音频重编码，直至指定的时间点为止都是静音

如果剪切视频时使用了-c:a copy，由于 AAC 等格式也存在「关键帧」的概念，ffmpeg 会在封装时自动加上一个延时的参数，使音视频同步；但是部分播放器或者解码器并不支持这一参数，因此会造成音视频不同步（典型的例子是将这样剪切出来的视频直接往 B 站投稿，可能会导致音视频不同步）

然而在实操过程中，即使使用 Aegisub 等工具找准了时间点，往往也会发生上述现象。因此，我个人认为比较合适的剪切视频比较好的方法是：使用 ffmpeg 的 hls 生成 m3u8 切片，将-ss 设置在开始前一段时间，并且设置好合适的-hls_time

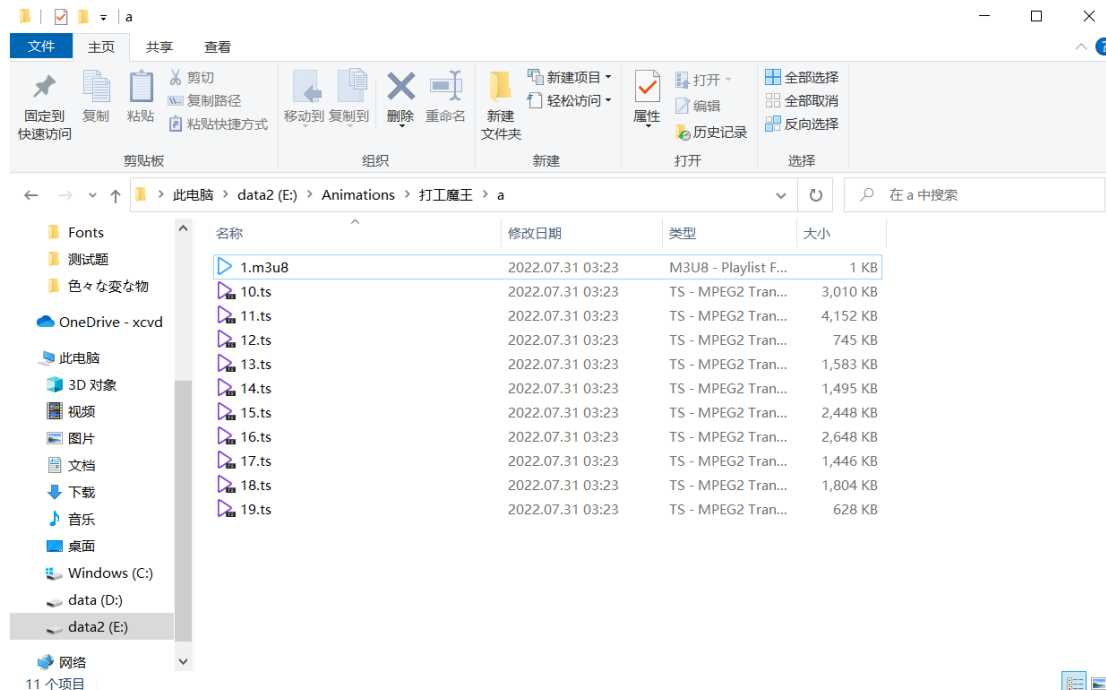
如果这么操作，那么需要首先找到需要开始切割的关键帧的前一个关键帧，可以使用 Aegisub 得知（如图，默认设置的情况下，进度条会显示关键帧位置，播放到那一帧时表示时间的方框会变色）



假如我们要从这一帧开始切割，查询得知上一个关键帧的时间点为 0:14:07.888，假设我们要切割至 0:15:30.0，那么我们首先插件一个叫「a」的文件夹，然后使用指令：

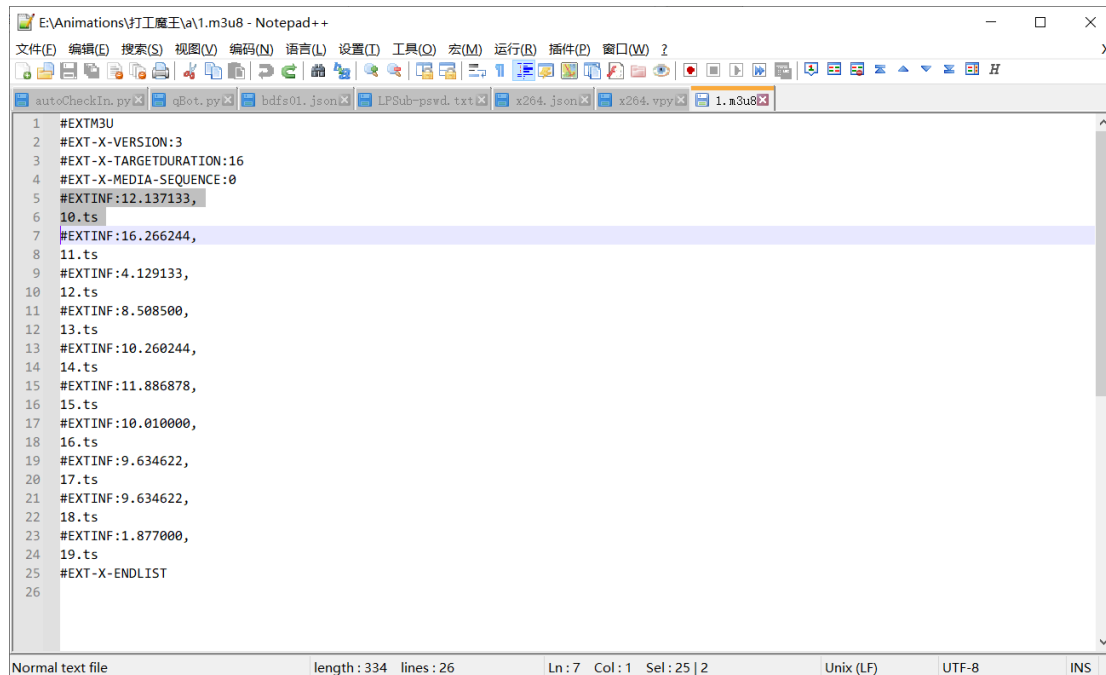
```
ffmpeg -ss 0:14:07.8 -to 0:15:30 -i 1.mkv -hls_time 10 -hls_list_size 0 -vbsf h264_mp4toannexb -c copy a/1.m3u8
```

这时我们打开文件夹「a」，可以看到这些文件



打开视频查看可知，10.ts 为指定的开始时间至我们所需切割的开始时间的文件，因此我们不需要这个文件；而 11.ts 直至 19.ts 都是我们需要的文件

因此这时我们需要用文本编辑器打开那个 m3u8 文件，将 10.ts 的信息删掉，保存



然后使用指令

```
ffmpeg -i a/1.m3u8 -c copy 1_clip.mkv
```

就得到了切割完成的文件（此方法有小几率失败，可能是由于时间码有关的玄学问题）

如果需要指定解码器/编码器，也可以使用-c

例如，假如要用 NVDEC 硬解视频流为 HEVC 的 114514.mkv，并使用 x264 的 medium 预设以 crf 18 重编码成 1919810.mkv，同时复制音频，那么可以使用：

```
ffmpeg -c:v hevc_cuvid -i 114514.mkv -preset medium -crf 18 -c:a copy 1919810.mkv
```

假如要编码成 10bit，那么可以使用：

```
ffmpeg -c:v hevc_cuvid -i 114514.mkv -preset medium -crf 18 -c:a copy -pix_fmt yuv420p10 1919810.mkv
```

假如同时要设置线程数为 12，那么就变为：

```
ffmpeg -c:v hevc_cuvid -i 114514.mkv -preset medium -crf 18 -c:a copy -pix_fmt yuv420p10 -threads 12 1919810.mkv
```

假如要使用 x265 的 slow 预设并且以 crf 19 编码成 10bit，可以使用：

```
ffmpeg -c:v hevc_cuvid -i 114514.mkv -preset slow -crf 19 -c:v libx265 -c:a copy -pix_fmt yuv420p10 1919810.mkv
```

假如要使用 NVENC 硬编 Ma10p(非常不推荐使用硬编)，并且指定固定码率 20Mbps，那么可以用：

```
ffmpeg -c:v hevc_cuvid -i 114514.mkv -preset slow -crf 19 -c:v hevc_cuvid -b:v 20M -c:a copy -pix_fmt yuv420p10 1919810.mkv
```

在使用硬编/硬解之前，请先确认自己的显卡是否支持这个格式！

同时请注意，默认的 YUV 采样模式和位深为输入视频的参数；例如输入视频为 Hi10p 或者 Ma10p，那么输出默认也是 10bit；如果需要输出 8bit，需要手动指定-pix_fmt yuv420p（我想用 ffmpeg 也不至于去编码 yuv444 吧）；这点在硬编/解的兼容性上面会体现出来，因为较老的显卡不支持 Hi10p，更老的显卡不支持 Ma10p；而对于 YUV444 来说，各家显卡的兼容性就更差了。

如果需要将 1920*1080 的 Hi10p 视频用 lanczos 缩小到 720p，并且使用 x264 8bit crf 18，预设 veryslow，那么可以使用：

```
ffmpeg -i 114514.mp4 -c:a copy -crf 18 -preset veryslow -s 1280x720 -sws_flags lanczos -pix_fmt yuv420p 1919810.mp4
```

也可以使用：

```
ffmpeg -i 114514.mp4 -c:a copy -crf 18 -preset veryslow -vf scale=1280:-1 -sws_flags lanczos 1919810.mp4
```

-vf 还有别的各种用处，例如加字幕、旋转视频和倒放音频等（不如试试把寒蝉第二季的 OP TV ver 倒放听一听）

例如，如果要将生肉 1.mkv 以 x264 8bit crf 20 preset slow 来压制，同时内嵌 1.ass，同时将音频复制，那么可以使用：

```
ffmpeg -i 1.mkv -c:a copy -crf 20 -preset slow -vf subtitles=1.ass 1919810.mp4
```

假如文件名带有空格（如：1 1.mkv 1 1.ass），那么修改成：

```
ffmpeg -i "1 1.mkv" -c:a copy -crf 20 -preset slow -vf "subtitles='1 1.ass'" "1919810.mp4"
```

需要特别注意的是，ffmpeg 自带的 libass 对华康的字形无法正确选择（可能是因为版本原因）。例如：指定字体为 DFPSNGyoSho-W5，实际压制时会选择成 DFSNGyoSho-W5 这一点在基于 ffmpeg 的 GUI 压制工具 ShanaEncoder 中得到修复，原因未知。

另外，由于 ffmpeg 的字幕滤镜为 libass，因此不支持 VsFilterMod 的标签；如需使用

vsfm，请使用 AviSynth 或 VapourSynth。

ffmpeg 也可以处理其他格式。例如，如果需要将 1.jpg 转换成 1.png，可以使用指令：

```
ffmpeg -i 1.jpg 1.png
```

有人会问，png，bmp 和 jpg 的区别在哪里呢？这三个格式都支持无损，区别在于，png 是 RGB 格式，jpg 是 YUV 格式；而 bmp 则是未压缩的格式。

ffmpeg 作为一个万能的格式转换软件，并不是专业的视频/音频处理工具，其封装或处理可能存在不合标准之处，对输入文件也有较高的标准要求；有的时候 ffmpeg 并不能正确处理视频文件，可能会有各种各样的 bug；这时请使用别的工具替代，如 MeGUI 和 MKVToolNix 等

以上所述的功能仅为方便日常调用而介绍；如需进行专业的处理，请务必使用对应的专业工具，如 AVS/VS，QAAC，MKVToolNix 等

更多的使用方法请善用搜索引擎

——Lambholl@LPSub