

前言

FFMPEG是特别强大的专门用于处理音视频的开源库。你既可以使用它的API对音视频进行处理，也可以使用它提供的工具，如 ffmpeg, ffplay, ffprobe，来编辑你的音视频文件。  
本文将简要介绍一下 FFMPEG 库的基本目录结构及其功能，然后详细介绍一下我们在日常工作中，如何使用 ffmpeg 提供的工具来处理音视频文件。

FFMPEG 目录及作用

- libavcodec：提供了一系列编码器的实现。
- libavformat：实现在流协议，容器格式及其本IO访问。
- libavutil：包括了hash器，解码器和各利工具函数。
- libavfilter：提供了各种音视频过滤器。
- libavdevice：提供了访问捕获设备和回放设备的接口。
- libswresample：实现了混音和重采样。
- libswscale：实现了色彩转换和缩放工能。

FFMPEG基本概念

在讲解 FFMPEG 命令之前，我们先要介绍一些音视频格式的基要概念。

音 / 视频流

在音视频领域，我们把一路音 / 视频称为一路流。如我们小时候经常使用VCD看港片，在里边可以选择粤语或国语声音，其实就是CD视频文件中存放了两路音频流，用户可以选择其中一路进行播放。

容器

我们一般把 MP4、FLV、MOV等文件格式称之为容器。也就是在这些常用格式文件中，可以存放多路音视频文件。以 MP4 为例，就可以存放一路视频流，多路音频流，多路字幕流。

channel

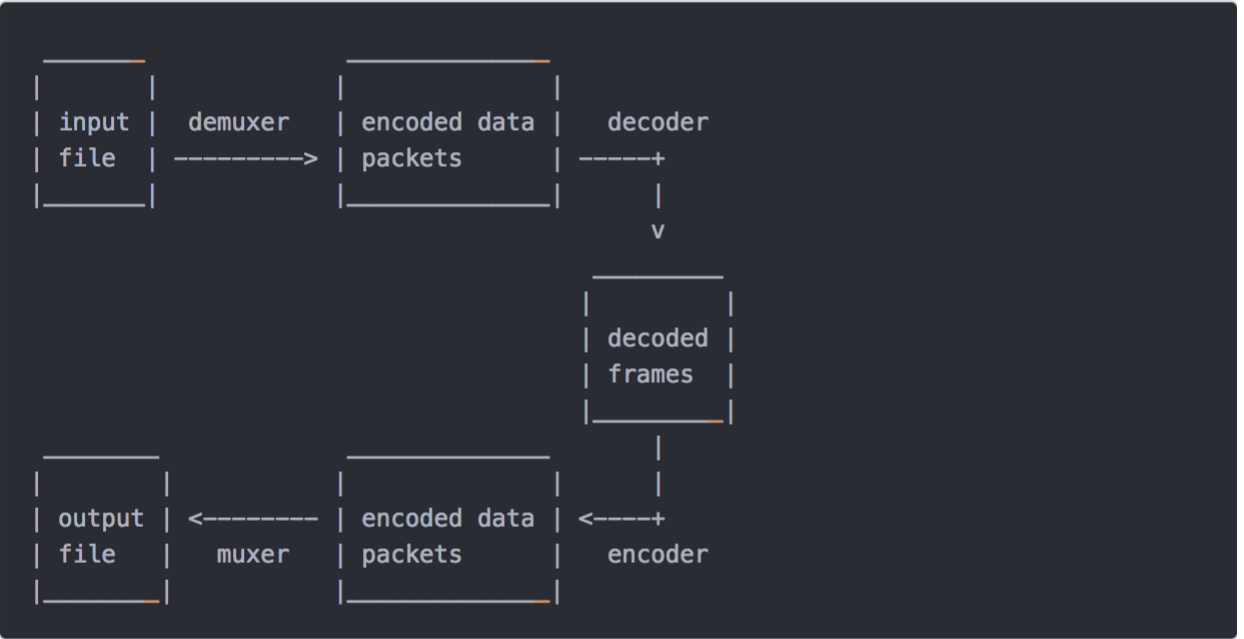
channel是音频中的概念，称之为声道。在一路音频流中，可以有单声道，双声道或立体声。

FFMPEG 命令

我们按使用目的可以将 FFMPEG 命令分成以下几类：

- 基本信息查询命令
- 录制
- 分解/复用
- 处理原始数据
- 滤镜
- 切割与合并
- 图 / 视互转
- 直播相关

除了 FFMPEG 的基本信息查询命令外，其它命令都按下图所示的流程处理音视频。



然后将编码的数据包传送给解码器（除非为数据流选择了流拷贝，请参阅进一步描述）。解码器产生未压缩的帧（原始视频/ PCM音频/ ... ），可以通过滤波进一步处理（见下一节）。在过滤之后，帧被传递到编码器，编码器并输出编码的数据包。最后，这些传递给复用器，将编码的数据包写入输出文件。  
默认情况下，ffmpeg只包含输入文件中每种类型（视频，音频，字幕）的一个流，并将其添加到每个输出文件中。它根据以下标准挑选每一个的“最佳”：对于视频，它是具有最高分辨率的流，对于音频，它是具有最多channel的流，对于字幕，是第一个字幕流。在相同类型的几个流相等的情况下，选择具有最低索引的流。  
您可以通过使用 -vn / -an / -sn / -dn选项来禁用某些默认设置。要进行全面的手动控制，请使用 -map选项，该选项禁用刚描述的默认设置。  
下面我们就来详细介绍一下这些命令。

基本信息查询命令

FFMPEG 可以使用下面的参数进行基本信息查询。例如，想查询一下现在使用的 FFMPEG 都支持哪些 filter，就可以用 ffmpeg -filters 来查询。详细参数说明如下：  
-version 显示版本。

-formats 显示可用的格式（包括设备）。  
-demuxers 显示可用的demuxers。  
-muxers 显示可用的muxers。  
-devices 显示可用的设备。  
-codecs 显示libavcodec已知的所有编解码器。  
-decoders 显示可用的解码器。  
-encoders 显示所有可用的编码器。  
-bsfs 显示可用的比特流filter。  
-protocols 显示可用的协议。  
-filters 显示可用的libavfilter过滤器。  
-pix\_fmts 显示可用的像素格式。  
-sample\_fmts 显示可用的采样格式。  
-layouts 显示channel名称和标准channel布局。  
-colors 显示识别的颜色名称。

接下来介绍的是 FFMPEG 处理音视频时使用的命令格式与参数。

## 命令基本格式及参数

下面是 FFMPEG 的基本命令格式：  
ffmpeg [global\_options] {[input\_file\_options] -i input\_url} ...  
[output\_file\_options] output\_url] ...

ffmpeg 通过 -i 选项读取任意数量的输入“文件”（可以是常规文件，管道，网络流，抓取设备等，并写入任意数量的输出“文件”。原则上，每个输入/输出“文件”都可以包含任意数量的不同类型的视频流（视频/音频/字幕/附件/数据）。流的数量和/或类型是由容器格式来限制。选择从哪个输入进入到哪个输出将自动完成或使用 -map 选项。  
要引用选项中的输入文件，您必须使用它们的索引（从0开始）。例如。第一个输入文件是0，第二个输入文件是1，等等。类似地，文件内的流被它们的索引引用。例如。2：3是指第三个输入文件中的第四个流。  
上面就是 FFMPEG 处理音视频的常用命令，下面是一些常用参数：

## 主要参数

-f fmt（输入/输出）强制输入或输出文件格式。格式通常是自动检测输入文件，并从输出文件的文件扩展名中猜测出来，所以在大多数情况下这个选项是不需要的。  
-i url（输入）输入文件的网址  
-y（全局参数）覆盖输出文件而不问。  
-n（全局参数）不要覆盖输出文件，如果指定的输出文件已经存在，请立即退出。  
-c [: stream\_specifier] codec（输入/输出，每个流）选择一个编码器（当在输出文件之前使用）或解码器（当在输入文件之前使用时）用于一个或多个流。codec 是解码器/编码器的名称或 copy（仅输出）以指示该流不被重新编码。如：ffmpeg -i INPUT -map 0 -c:v libx264 -c:a copy OUTPUT  
-codec [: stream\_specifier] 解码器（输入/输出，每个流）同 -c  
-t duration（输入/输出）当用作输入选项（在 -i 之前）时，限制从输入文件读取的数据的持续时间。当用作输出选项时（在输出url之前），在持续时间到达持续时间之后停止输出。  
-ss位置（输入/输出）当用作输入选项时（在 -i 之前），在这个输入文件中寻找位置。请注意，在大多数格式中，不可能精确搜索，因此ffmpeg将在位置之前寻找最近的搜索点。当转码和-accurate\_seek被启用时（默认），搜索点和位置之间的这个额外的分段将被解码和丢弃。当进行流式复制或使用-noaccurate\_seek时，它将被保留。当用作输出选项（在输出url之前）时，解码但丢弃输入，直到时间戳到达位置。  
-frames [: stream\_specifier] framecount（output，per-stream）停止在帧计数帧之后写入流。  
-filter [: stream\_specifier] filtergraph（output，per-stream）创建由filtergraph指定的过滤图，并使用它来过滤流。filtergraph是应用于流的filtergraph的描述，并且必须具有相同类型的流的单个输入和单个输出。在过滤图图形中，输入与标签中的标签相关联，标签中的输出与标签相关联。有关filtergraph语法的更多信息，请参阅ffmpeg-filters手册。

## 视频参数

-vframes num（输出）设置要输出的视频帧的数量。对于 -frames : v，这是一个过时的别名，您应该使用它。  
-r [: stream\_specifier] fps（输入/输出，每个流）设置帧率（Hz值，分数或缩写）。作为输入选项，忽略存储在文件中的任何时间戳，根据速率生成新的时间戳。这与用于-framerate选项不同（它在Ffmpeg的旧版本中使用的是相同的）。如果有疑问，请使用-framerate而不是输入选项-r。作为输出选项，复制或丢弃输入帧以实现恒定输出帧频fps。  
-s [: stream\_specifier] 大小（输入/输出，每个流）设置窗口大小。作为输入选项，这是video\_size专用选项的快捷方式，由某些分帧器识别，其帧尺寸未被存储在文件中。作为输出选项，这会将缩放视频过滤器插入到相应过滤器图形的末尾。请直接使用比例过滤器将其插入到开头或其他地方。格式是' wxh'（默认 - 与源相同）。  
-aspect [: stream\_specifier] 宽高比（输出，每个流）设置方面指定的视频显示宽高比。aspect可以是浮点数字串，也可以是num : den形式的字符串，其中num和den是宽高比的分子和分母。例如“4 : 3”，“16 : 9”，“1.3333”和“1.7777”是有效的参数值。如果与-vcodect副本一起使用，则会影响存储在容器级别的宽高比，但不会影响存储在编码帧中的宽高比（如果存在）。  
-vn（输出）禁用视频录制。  
-vcodec编解码器（输出）设置视频编解码器。这是-codec : v的别名。  
-vf filtergraph（输出）创建由filtergraph指定的过滤图，并使用它来过滤波。

## 音频参数

-aframes（输出）设置要输出的音频帧的数量。这是-frames : a的一个过时的别名。  
-ar [: stream\_specifier] freq（输入/输出，每个流）设置音频采样频率。对于输出流，它默认设置为相应输入流的频率。对于输入流，此选项仅适用于音频捕获设备和原始分路器，并映射到相应的分路器选项。  
-ac [: stream\_specifier] 通道（输入/输出，每个流）设置音频通道的数量。对于输出流，它默认设置为输入音频通道的数量。对于输入流，此选项仅适用于音频捕获设备和原始分路器，并映射到相应的分路器选项。  
-an（输出）禁用录音。  
-acodec编解码器（输入/输出）设置音频编解码器。这是-codec的别名：a。  
-sample\_fmt [: stream\_specifier] sample\_fmt（输出，每个流）设置音频采样格式。使用-sample\_fmts获取支持的样本格式列表。  
-af filtergraph（输出）创建由filtergraph指定的过滤图，并使用它来过滤波。  
了解了这些基本信息后，接下来我们看看 FFMPEG 具体都能干什么吧。

## 录制

首先通过下面的命令查看一下 mac 上都有哪些设备。  
ffmpeg -f avfoundation -list\_devices true -i ""

## 录屏

ffmpeg -f avfoundation -i 1 -r 30 out.yuv  
-f 指定使用 avfoundation 采集数据。  
-i 指定从哪儿采集数据，它是一个文件索引号。在我的MAC上，1代表桌面（可以通过上面的命令查询设备索引号）。  
-r 指定帧率。按ffmpeg官方文档说-r与-framerate作用相同，但实际测试时发现不同。-framerate 用于限制输入，而-r用于限制输出。  
注意，桌面的输入对帧率没有要求，所以不用限制桌面的帧率。其实限制了也没用。

## 录屏+声音

ffmpeg -f avfoundation -i 1:0 -r 29.97 -c:v libx264 -crf 0 -c:a libfdk\_aac -profile:a aac\_he\_v2 -b:a 32k out.flv  
-i 1:0 冒号前面的“1”代表的屏幕索引号。冒号后面的“0”代表的声音索引号。

-c:v 与参数 -vcodec 一样，表示视频编码器。c 是 codec 的缩写，v 是video的缩写。  
-crf 是 x264 的参数。0 表式无损压缩。  
-c:a 与参数 -acodec 一样，表示音频编码器。  
-profile 是 fdk\_aac 的参数。aac\_he\_v2 表式使用 AAC\_HE v2 压缩数据。  
-b:a 指定音频码率。b 是 bitrate的缩写，a是 audio的缩与。

## 录视频

```
ffmpeg -framerate 30 -f avfoundation -i 0 out.mp4
```

-framerate 限制视频的采集帧率。这个必须要根据提示要求进行设置，如果不设置就会报错。  
-f 指定使用 avfoundation 采集数据。  
-i 指定视频设备的索引号。

## 视频+音频

```
ffmpeg -framerate 30 -f avfoundation -i 0:0 out.mp4
```

## 录音

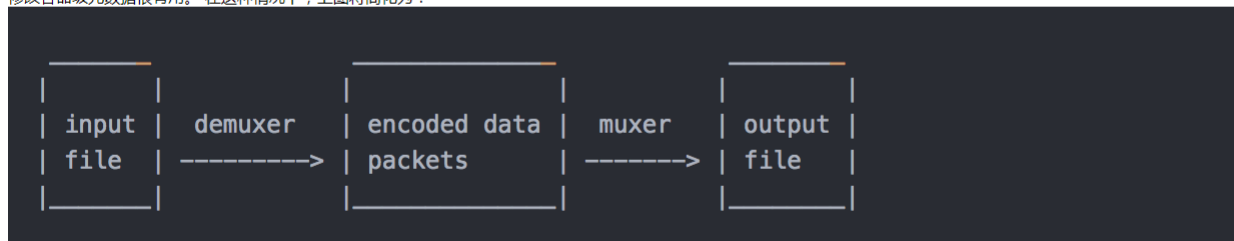
```
ffmpeg -f avfoundation -i :0 out.wav
```

## 录制音频裸数据

```
ffmpeg -f avfoundation -i :0 -ar 44100 -f s16le out.pcm
```

## 分解与复用

流拷贝是通过将 copy 参数提供给 -codec选项来选择流的模式。它使得ffmpeg省略了指定流的解码和编码步骤，所以它只能进行多路分解和多路复用。这对于更改容器格式或修改容器级元数据很有用。在这种情况下，上图将简化为：



由于没有解码或编码，速度非常快，没有质量损失。但是，由于许多因素，在某些情况下可能无法正常工作。应用过滤器显然也是不可能的，因为过滤器处理未压缩的数据。

## 抽取音频流

```
ffmpeg -i input.mp4 -acodec copy -vn out.aac
```

acodec: 指定音频编码器，copy 指明只拷贝，不做编解码。  
vn: v 代表视频，n 代表 no 也就是无视频的意思。

## 抽取视频流

```
ffmpeg -i input.mp4 -vcodec copy -an out.h264
```

vcodec: 指定视频编码器，copy 指明只拷贝，不做编解码。  
an: a 代表视频，n 代表 no 也就是无音频的意思。

## 转格式

```
ffmpeg -i out.mp4 -vcodec copy -acodec copy out.flv
```

上面的命令表式的是音频、视频都直接 copy，只是将 mp4 的封装格式转成了 flv。

## 音视频合并

```
ffmpeg -i out.h264 -i out.aac -vcodec copy -acodec copy out.mp4
```

处理原始数据

## 提取YUV数据

```
ffmpeg -i input.mp4 -an -c:v rawvideo -pixel_format yuv420p out.yuv
```

ffplay -s wxh out.yuv  
-c:v rawvideo 指定将视频转成原始数据  
-pixel\_format yuv420p 指定转换格式为yuv420p