# 描述

FFmpeg是一个非常快的视频和音频转换器，可以从现场抓取音频/视频源。它可以转任意比特率和大小的视频。

ffmpeg读取任意数量的输入文件（可以是普通文件，管道，网络流，抓取设备等等），由-i选项指定，并写入任意数量的输出文件，由一个简单的输出文件名指定。在命令行上的任何不能识别的操作将会被认为是一个输出的文件的文件名。

原则上，每个输入或输出文件都可以包含任意数量的不同类型的流（视频/音频/字符/附件/数据）。允许的流类型的数量对于格式化容器可能是有限制的。选择从哪个流输入到输出是自动完成的，或者可以使用-map操作选项（见流选择章节）。

指定输入文件时，使用它们以0开始索引。例如：第一个输入文件是0，第二个是1，等等。同样，一个文件中的流也是通过索引使用。例如：2:3是指在第三个输入文件的底4个流。也可以看流章节。

作为一个一般规则，操作选择被应用到下一个指定的文件。因此排序是重要的，你可以在命令行出现多次相同的选项。每一个事件应用到下一个输入或输出文件。这个规则的例外是全局选项，应该首先指定。

不要混合输入和输出文件，首先要指定所有的输入文件，然后所有的输出文件。也不要混合输入不同文件的操作选项，所有选项只适用于下一个输入或输出文件，并在文件之间重置。

* 设置输出文件的视频比特率为64kbit/s：

ffmpeg -i input.avi -b:v 64k -bufsize 64k output.avi

* 强制输出文件的帧率为24帧

ffmpeg -i input.avi -r 24 output.avi

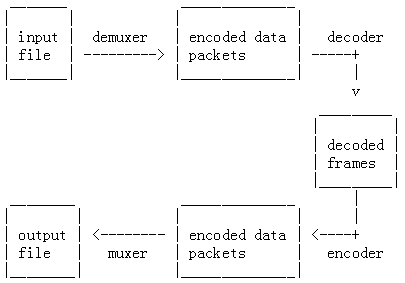
* 强制输入文件的帧率(原格式需要有效)1FPS和输出文件的帧率为24FPS

ffmpeg -r 1 -i input.m2v -r 24 output.avi

原始输入文件可能需要格式选项。

# 详细说明

ffmpeg对每个文件的转码过程可描述为下图：



ffmpeg库调用libavformat库(含分配器)读取文件，并得到包含编码的数据包。当输入多个文件的时候，ffmpeg在活动的流上通过时间戳保持同步。

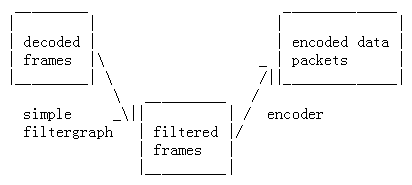
编码的数据包同过解码器(除非选择的流是拷贝选项)。解码器产生未压缩的帧（视频/音频/..）并可以通过过滤进一步处理。过滤后，帧被传递到编码器，编码器进行编码，并输出编码的数据包。最后通过复用器将编码包输出到文件。

## 过滤

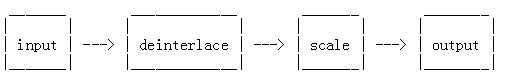
在编码之前，ffmpeg可以使用libavfilter库对原始音频和视频帧进行过滤。

### 简单过滤

简单的过滤是那些有一个输入和一个输出，并且输入输出类型相同。在图表中，可以简单的表示为在解码和编码间插入一个额外的步骤：



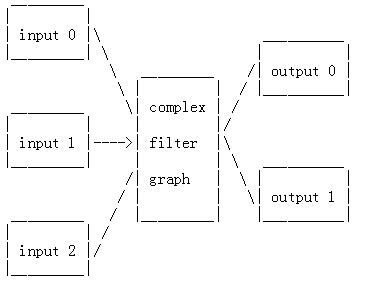
简单过滤对每个流进行 -filter选项配置（-vf是视频，-af是音频），视频的一个简单过滤图如下图：



注意，一些过滤器只改变帧属性不改变帧内容。例如：上面的例子fps过滤器只改变帧的数量变化，但不改变帧的内容。另一个例子，setpts filter，仅设置时间戳，通过的帧不变。

### 复杂过滤器视图

复杂的过滤是那些不能被简单的描述为一个线性的处理链应用到一个流上。例如有一个以上的输入或输出或输入输出的流类型不同。可以利用下图来表示：



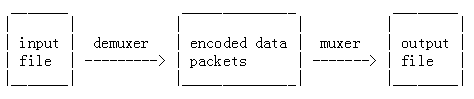
复杂过滤使用 -filter\_complex操作选项配置。这个操作选项是全局的，因为一个复杂的过滤，不能简简单单的与一个流或文件关联。

-lavfi 操作等价于 -filter\_complex

关于一个复杂过滤的简单例子，覆盖过滤，其中两个视频输入一个视频输出，将一个视频叠加到另一个视频上。其音频对应的是滤波器。

## 流拷贝

流拷贝是通过向编解码器提供一种复制参数的一种模式。他让ffmpeg解码和编码步骤省略为指定的流，所以它是唯一的解复用和多路复用。它用于修改容器格式或修改容器级元数据。流程可以简化为如下图表：



由于没有解码或编码，速度非常快，没有质量损失。然而因为许多因素它可能不工作。运用过滤器显然也不可能，因为过滤器工作在未压缩的数据上。

# 流的原则

默认情况下，在输入文件中ffmpeg只包含（视频/音频/字幕）这些类型中一个流，并将他们添加到输出文件。根据以下标准选择视频流的最佳分辨率，音频的最佳频率，对于字幕来说是第一个字幕流。同一类型，比特率相同，选择最低索引的流。

通过 -vn/-an/-sn/-dn选项可以禁用一些默认选项。对于完整控制使用 -map选项，它将禁用刚刚描述的默认值。

# 选项

所有的数值选项，如果没有指定，否则接收一个字符串代表一个数作为输入，并且可能跟一个单位的前缀，例如："K", "M",或"G"。如果 'i'被解释成一个前缀，完整的前缀被解释成一个二进制的倍数基于1024。

# 流说明符

一下选项适用于每一个流，比如比特率，编解码器。流说明符用于精确的指定流给以选项。一个流说明字符串通常附加选项名称，并以冒号分隔。例如 -codec:a:1 ac3 包含了a:1流说明符，它将匹配第二个音频流。因此他会选择第二个音频流的AC3解码器。一个流说明符可以匹配一系列流，所以这个选项适合所有。例如：-b:a 128k匹配所有的音频流。

一个空的流说明符匹配所有的流。例如： -codec copy or -codec : copy 将复制所有数据。

流说明符可能的形式：

*stream\_index*

匹配该索引对应的流