

# FFMPEG

## 音视频前置知识

很完整的学习笔记：<https://blog.csdn.net/leixiaohua1020/article/details/18893769>

- 封装格式：不同的音视频文件的后缀（avi、rmvb、mp4、mkv等）代表不同的封装格式，就是把视频数据和音频数据打包成一个文件的规范。
- 视频播放器播放一个互联网上的视频文件，需要经过以下几个步骤：解协议，解封装，解码视音频，视音频同步。如果播放本地文件则不需要解协议，为以下几个步骤：解封装，解码视音频，视音频同步。他们的过程如图所示。
  - 解协议：**将流媒体协议的数据，解析为标准的相应的封装格式数据。**视音频在网络上传播的时候，常常采用各种流媒体协议，例如HTTP，RTMP，或是MMS等等。这些协议在传输视音频数据的同时，也会传输一些信令数据。这些信令数据包括对播放的控制（播放，暂停，停止），或者对网络状态的描述等。解协议的过程中会去除掉信令数据而只保留视音频数据。
  - 解封装：**将输入的封装格式的数据，分离成为音频流压缩编码数据和视频流压缩编码数据。**封装格式种类很多，例如MP4，MKV，RMVB，TS，FLV，AVI等等，它的作用就是将已经压缩编码的视频数据和音频数据按照一定的格式放到一起。例如，FLV格式的数据，经过解封装操作后，输出H.264编码的视频码流和AAC编码的音频码流。
  - 解码音视频：**将视频/音频压缩编码数据，解码成为非压缩的视频/音频原始数据。**音频的压缩编码标准包含AAC，MP3，AC-3等等，视频的压缩编码标准则包含H.264，MPEG2，VC-1等等。解码是整个系统中最重要也是最复杂的一个环节。通过解码，压缩编码的视频数据输出成为非压缩的颜色数据，例如YUV420P，RGB等等；压缩编码的音频数据输出成为非压缩的音频抽样数据，例如PCM数据。
  - 音视频同步：**根据解封装模块处理过程中获取到的参数信息，同步解码出来的视频和音频数据，并将视频音频数据送至系统的显卡和声卡播放出来。**



## 介绍

**FFmpeg** 是一款开源软件，用于生成处理多媒体数据的各类库和程序。FFmpeg可以转码、处理视频和图片（调整视频、图片大小，去噪等）、打包、传输及播放视频。作为最受欢迎的视频和图像处理软件，它被来自各行各业的不同公司所广泛使用。

主要用途：

- 视频处理：FFmpeg可以用于各种视频处理任务，比如去噪、模糊视频、色彩转换，视频旋转、提取帧、缩放视频尺寸等。
- 视频压缩：FFmpeg 包含为各种编解码器[如 JPEG、MPEG-1/2/4、H263+ AAC (MPEG)、Theora (Ogg Vorbis)、AVS+、VP8 (WebM)、H.264/AVC、HEVC、AV1 等]提供接口的库
- 视频打包：FFmpeg还完全支持使用HLS和MPEG-DASH打包视频，它还可以配置为使用RTMP或其他协议来传输视频。
- 支持视频、音频容器格式：可以用来读、写及转换各种容器格式，比如avi、mp4、mp3、wma、wav、ts、flv、mkv以及许多其他未知格式。
- ffprobe：构建了一个名为 ffprobe 的命令行，可用于查看和分析视频文件。你可以使用 ffprobe 从视频中提取各种元数据、计算帧数、查找 IDR 帧的位置等等。
- ffmpeg：命令行工具，用于播放视频

许多文档和使用指南：[https://github.com/0voice/ffmpeg\\_develop\\_doc/tree/main](https://github.com/0voice/ffmpeg_develop_doc/tree/main)

博客：<https://blog.csdn.net/leixiaohua1020>

官方文档：<https://ffmpeg.org/documentation.html>

### 三、FFmpeg的主要工作流程

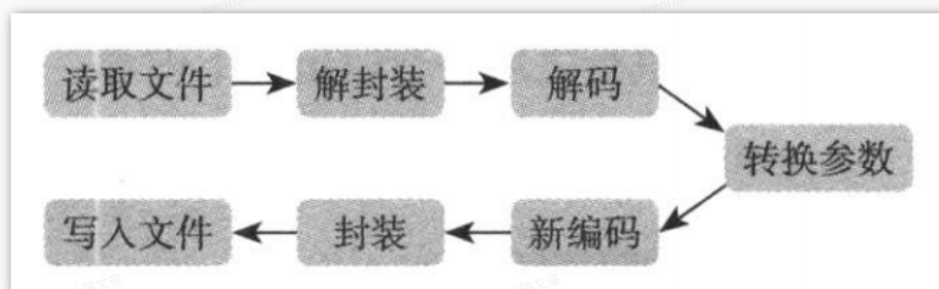
FFmpeg 的主要工作流程相对比较简单，具体如下：

- 1) 解封装 (Demuxing)
- 2) 解码 (Decoding)
- 3) 编码 (Encoding)
- 4) 封装 (Muxing)

其中需要经过 6 个步骤，具体如下：

- 1) 读取输入源
- 2) 进行音视频的解封装
- 3) 解码每一帧音视频数据
- 4) 编码每一帧音视频数据
- 5) 进行音视频的重新封装
- 6) 输出到目标

FFmpeg 整体处理的工作流程与步骤如下图所示：



## FFMPEG使用

命令参数：

- ffmpeg: <https://blog.csdn.net/leixiaohua1020/article/details/12751349>
- ffplay: <https://blog.csdn.net/leixiaohua1020/article/details/15186441>
- ffprobe: <http://ffmpeg.org/ffprobe.html>

