<https://mp.weixin.qq.com/s/FNddgYrOXs_-Sz8hp83w4A>

一、前言：

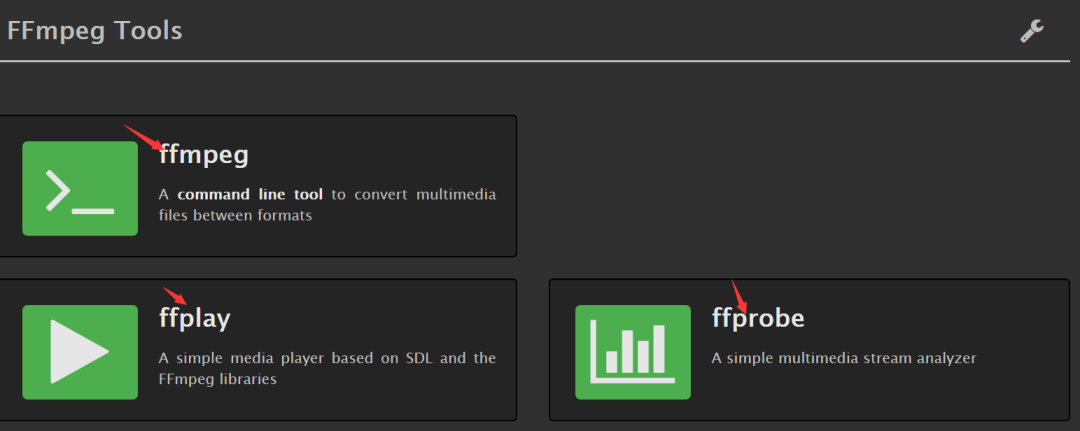
今天主要给大家分享ffmpeg和ffplay相关命令的使用！

1. ffmpeg命令汇总

1.ffmpeg工具：

ffmpeg工具主要分三个，从官网我们可以看到：

<https://www.ffmpeg.org/about.html>



* ffmpeg:(Hyper fast Autio and Video encoder)超快音视频编码器（类似于爱剪辑）。
* ffplay:(Simple media player)简单媒体播放器
* ffprobe:(Simple multimedia streams analyzer)简单多媒体流分析器

在上面官网图片里面中，如果大家要详细了解ffmpeg、ffplay、ffprobe三个工具的使用，可以点击进去，里面有非常详细的介绍！这里接下来我给大家总结一下这段时间我学习里面的一些常用命令，再加上一些视频播放演示,这样学习起来就不会太枯燥了.

### **2、ffmpeg命令使用总结**

(1)ffmpeg命令查看帮助文档

1.基本信息:ffmpeg -h

Print help / information / capabilities:  
-L              show license  
-h topic           show help  
-? topic           show help  
-help topic         show help  
--help topic         show help  
-version           show version  
-buildconf          show build configuration  
-formats           show available formats  
-muxers            show available muxers  
-demuxers           show available demuxers  
-devices           show available devices  
-codecs            show available codecs  
-decoders           show available decoders  
-encoders           show available encoders  
-bsfs             show available bit stream filters  
-protocols          show available protocols  
-filters           show available filters  
-pix\_fmts           show available pixel formats  
-layouts           show standard channel layouts

2.高级信息：ffmpeg -h long

Global options (affect whole program instead of just one file:  
-loglevel loglevel   set logging level  
-v loglevel       set logging level  
-report          generate a report  
-max\_alloc bytes     set maximum size of a single allocated block  
-y            overwrite output files  
-n            never overwrite output files  
-ignore\_unknown     Ignore unknown stream types  
-filter\_threads     number of non-complex filter threads  
-filter\_complex\_threads number of threads **for** -filter\_complex  
-stats           print progress report during encoding

-max\_error\_rate maximum error rate  ratio of errors (0.0: no errors, 1.0: 100% errors) above which ffmpeg returns an error instead of success.

-bits\_per\_raw\_sample number  set the number of bits per raw sample  
-vol volume          change audio volume (256=normal)

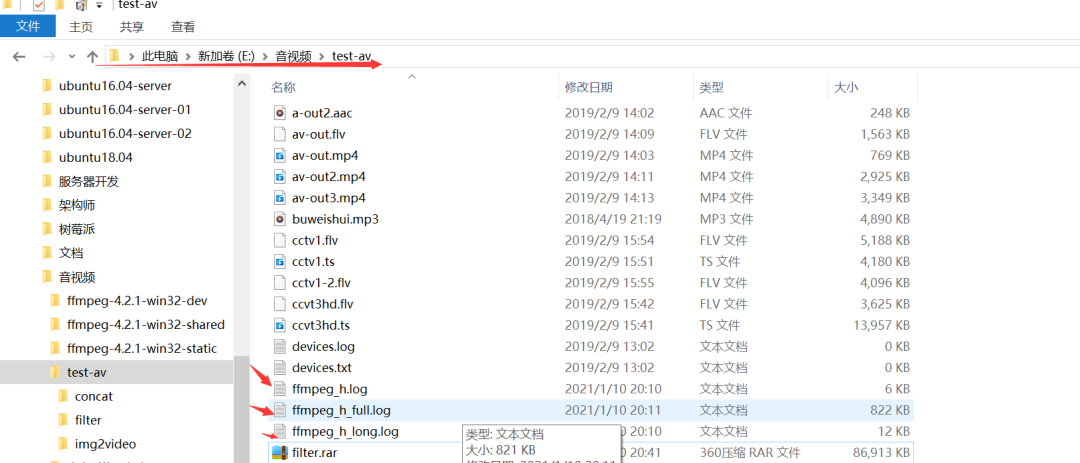
3.所有信息命令：ffmpeg -h full

magicyuv AVOptions:  
  -pred              <int>        E..V..... Prediction method (from 1 to 3) (default left)  
     left                         E..V.....  
     gradient                     E..V.....  
     median                       E..V.....  
还有很多信息没有截取出来，这里只是一部分信息

注意：我们可以看到上面的命令查询帮助信息打印出来太多了，不方便阅读，所以我们把它们都放在单独的一个日志文件里面，这样就方便查看阅读：

E:\音视频\test-av>ffmpeg -h > ffmpeg\_h.log  
  
E:\音视频\test-av>ffmpeg -h long  > ffmpeg\_h\_long.log  
  
E:\音视频\test-av>ffmpeg -h full  > ffmpeg\_h\_full.log

最终就会在当前目录下生成三个日志文件：



同时我们要注意，在ffplay和ffprobe里面，下面这两条命令就表示查看所有信息，这和ffmpeg有点差异：

ffplay -h  
  
ffprobe -h

(2)ffmpeg命令分类查询：

1.version:显示ffmpeg版本型号：

E:\音视频\test-av>ffmpeg -version  
ffmpeg version 4.2.1 Copyright (c) 2000-2019 the FFmpeg developers  
built with gcc 9.1.1 (GCC) 20190807

configuration: --disable-static --enable-shared --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libdav1d --enable-libbluray --enable-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxml2 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --enable-libvo-amrwbenc --enable-libmysofa --enable-libspeex --enable-libxvid --enable-libaom --enable-libmfx --enable-amf --enable-ffnvcodec --enable-cuvid --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-libopenmpt

libavutil       56. 31.100 / 56. 31.100  
libavcodec      58. 54.100 / 58. 54.100  
libavformat     58. 29.100 / 58. 29.100  
libavdevice     58.  8.100 / 58.  8.100  
libavfilter      7. 57.100 /  7. 57.100  
libswscale       5.  5.100 /  5.  5.100  
libswresample    3.  5.100 /  3.  5.100  
libpostproc     55.  5.100 / 55.  5.100

2.buildconf:显示编译配置：

configuration:  
    --disable-static  
    --enable-shared  
    --enable-gpl  
    --enable-version3  
    --enable-sdl2  
    --enable-fontconfig  
    --enable-gnutls  
    --enable-iconv  
    --enable-libass  
    --enable-libdav1d  
    --enable-libbluray  
    --enable-libfreetype  
    --enable-libmp3lame  
    --enable-libopencore-amrnb  
    --enable-libopencore-amrwb  
    --enable-libopenjpeg  
    --enable-libopus  
    --enable-libshine  
    --enable-libsnappy  
    --enable-libsoxr  
    --enable-libtheora  
    --enable-libtwolame  
    --enable-libvpx  
    --enable-libwavpack  
    --enable-libwebp  
    --enable-libx264  
    --enable-libx265  
    --enable-libxml2  
    --enable-libzimg  
    --enable-lzma  
    --enable-zlib  
    --enable-gmp  
    --enable-libvidstab  
    --enable-libvorbis  
    --enable-libvo-amrwbenc  
    --enable-libmysofa  
    --enable-libspeex  
    --enable-libxvid  
    --enable-libaom  
    --enable-libmfx  
    --enable-amf  
    --enable-ffnvcodec  
    --enable-cuvid  
    --enable-d3d11va  
    --enable-nvenc  
    --enable-nvdec  
    --enable-dxva2  
    --enable-avisynth  
    --enable-libopenmpt

3.formats:显示可用格式(muxers+demuxers)，muxers表示显示可用复用器；demuxers表示显示可用解复用器:

File formats:  
 D. = Demuxing supported  
 .E = Muxing supported

当然我们可以专门查找flv格式的文件格式：

E:\音视频\test-av>ffmpeg -muxers | findstr flv  
  E flv             FLV (Flash Video)

E:\音视频\test-av>ffmpeg -demuxers | findstr flv  
 D  flv             FLV (Flash Video)  
 D  live\_flv        live RTMP FLV (Flash Video)

4.codecs:显示可用编解码器(-decoders表示显示可用解码器；-encoders表示显示可用编码器)，下面是打印查找的一部分信息:

D.VI.S 012v                 Uncompressed 4:2:2 10-bit  
  D.V.L. 4xm                  4X Movie  
  D.VI.S 8bps                 QuickTime 8BPS video  
  .EVIL. a64\_multi            Multicolor charset **for** Commodore 64 (encoders: a64multi )  
  .EVIL. a64\_multi5           Multicolor charset **for** Commodore 64, extended with 5th color (colram) (encoders: a64multi5 )  
 D.V..S aasc                 Autodesk RLE  
 D.V.L. agm                  Amuse Graphics Movie  
 D.VIL. aic                  Apple Intermediate Codec  
 DEVI.S alias\_pix            Alias/Wavefront PIX image  
 DEVIL. amv                  AMV Video  
 D.V.L. anm                  Deluxe Paint Animation  
 D.V.L. ansi                 ASCII/ANSI art

5.bsfs:显示可用比特流filter,打印信息比较多，截取了一部分，后期的命令打印都只截取一部分：

Bitstream filters:  
aac\_adtstoasc  
av1\_frame\_split  
av1\_metadata  
chomp  
dump\_extra  
dca\_core  
eac3\_core  
extract\_extradata  
filter\_units  
h264\_metadata  
h264\_mp4toannexb  
h264\_redundant\_pps  
hapqa\_extract  
hevc\_metadata  
hevc\_mp4toannexb  
imxdump  
mjpeg2jpeg  
mjpegadump  
mp3decomp  
mpeg2\_metadata  
mpeg4\_unpack\_bframes  
mov2textsub  
noise  
null  
prores\_metadata  
remove\_extra  
text2movsub  
trace\_headers  
truehd\_core  
vp9\_metadata  
vp9\_raw\_reorder  
vp9\_superframe  
vp9\_superframe\_split

6.protocols:显示可用的协议：

Supported file protocols:  
Input:  
  async  
  bluray  
  cache  
  concat  
  crypto  
  data  
  ffrtmpcrypt  
  ffrtmphttp  
  file  
  ftp  
  gopher  
  hls  
  http  
  httpproxy  
  https  
  mmsh  
  mmst  
  pipe  
  rtmp  
  rtmpe  
  rtmps  
  rtmpt  
  rtmpte  
  rtmpts  
  rtp  
  srtp  
  subfile  
  tcp  
  tls  
  udp  
  udplite  
Output:  
  crypto  
  ffrtmpcrypt  
  ffrtmphttp  
  file  
  ftp  
  gopher  
  http  
  httpproxy  
  https  
  icecast  
  md5  
  pipe  
  prompeg  
  rtmp  
  rtmpe  
  rtmps  
  rtmpt  
  rtmpte  
  rtmpts  
  rtp  
  srtp  
  tee  
  tcp  
  tls  
  udp  
  udplite

7.filters:显示可用的过滤器：

E:\音视频\test-av>ffmpeg -filters  
Filters:  
  T.. = Timeline support  
  .S. = Slice threading  
  ..C = Command support  
  A = Audio input/output  
  V = Video input/output  
  N = Dynamic number and/or type of input/output  
  | = Source or sink filter

8.pix\_fmts:显示可用的像素格式

E:\音视频\test-av>ffmpeg -pix\_fmts  
Pixel formats:  
I.... = Supported Input  format **for** conversion  
.O... = Supported Output format **for** conversion  
..H.. = Hardware accelerated format  
...P. = Paletted format  
....B = Bitstream format  
FLAGS NAME            NB\_COMPONENTS BITS\_PER\_PIXEL  
-----

9.layouts：显示标准声道名称

E:\音视频\test-av>ffmpeg -layouts  
Individual channels:  
NAME           DESCRIPTION  
FL             front left  
FR             front right  
FC             front center  
LFE            low frequency  
BL             back left  
BR             back right  
FLC            front left-of-center  
FRC            front right-of-center  
BC             back center  
SL             side left  
SR             side right  
TC             top center  
TFL            top front left  
TFC            top front center  
TFR            top front right  
TBL            top back left  
TBC            top back center  
TBR            top back right  
DL             downmix left  
DR             downmix right  
WL             wide left  
WR             wide right  
SDL            surround direct left  
SDR            surround direct right  
LFE2           low frequency 2  
  
Standard channel layouts:  
NAME           DECOMPOSITION  
mono           FC  
stereo         FL+FR  
2.1            FL+FR+LFE  
3.0            FL+FR+FC  
3.0(back)      FL+FR+BC  
4.0            FL+FR+FC+BC  
quad           FL+FR+BL+BR  
quad(side)     FL+FR+SL+SR

10.sample\_fmts:显示可用的音频采样格式：

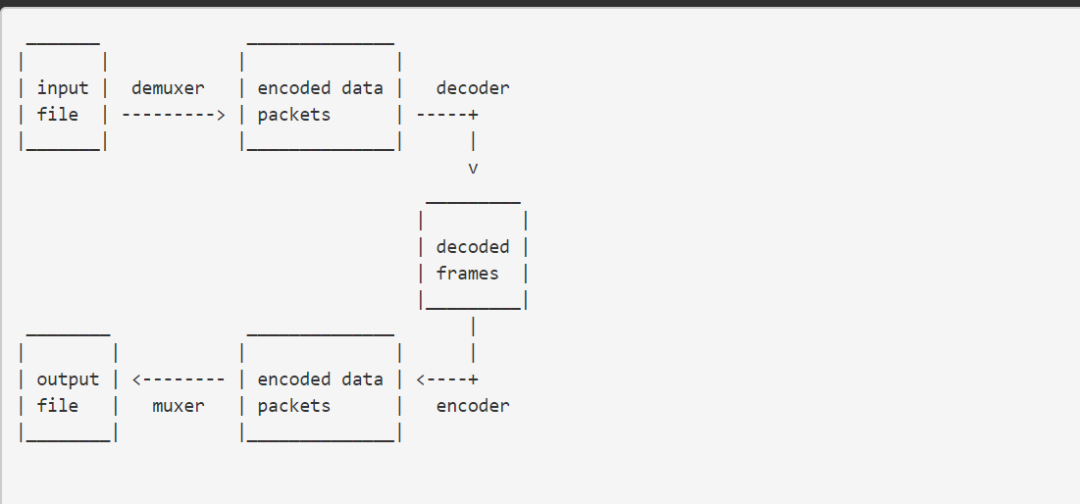
E:\音视频\test-av>ffmpeg -sample\_fmts  
name   depth  
u8        8  
s16      16  
s32      32  
flt      32  
dbl      64  
u8p       8  
s16p     16  
s32p     32  
fltp     32  
dblp     64  
s64      64  
s64p     64

11.colors：显示可用的颜色名称：

E:\音视频\test-av>ffmpeg -colors  
name                             #RRGGBB  
AliceBlue                        #f0f8ff  
AntiqueWhite                     #faebd7  
Aqua                             #00ffff  
Aquamarine                       #7fffd4  
Azure                            #f0ffff  
Beige                            #f5f5dc  
Bisque                           #ffe4c4  
Black                            #000000  
BlanchedAlmond                   #ffebcd  
Blue                             #0000ff  
BlueViolet                       #8a2be2  
Brown                            #a52a2a  
BurlyWood                        #deb887

三、ffmpeg音视频处理流程：

我们可以去ffmpeg官网介绍里面看到它的处理流程图：



为了说明这个处理过程，我们以下面一条命令来举例说明：

ffmpeg -i test\_1920x1080.mp4 -acodec copy -vcodec libx264 -s 1280x720 test\_1280x720.flv

说明：

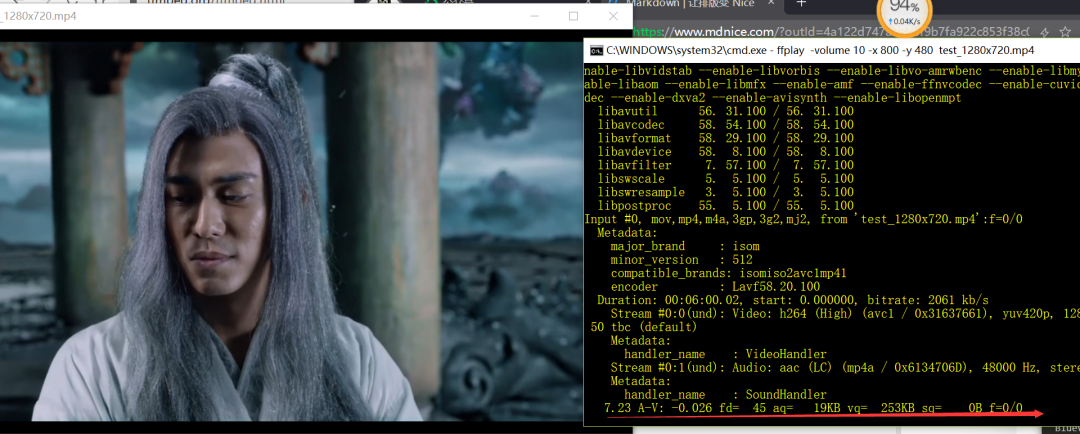
* (input file)这里输入文件为test\_1920x1080.mp4格式的视频文件格式
* demuxer：通过可用解复用器把输入文件，也就是音视频文件，分离出音频包和视频包
* 编码数据包；通过上面的demuxr，就得到了音频包和视频包
* decoder:然后通过解码器解码后，就生成了解码数据帧,这里要通过过滤处理，怎样过滤处理具体可以看官方的说明，这里暂时不说明，如果有时间，研究完官方说明，再来讲
* encoder:过滤处理完后的解码数据帧通过编码器处理后，会生成编码数据包
* muxer:编码数据包再次通过可用复用器处理，就生成了一个新格式的音视频文件，比如这里是test\_1280x720.flv

四、总结：

更多命令学习可以参考官网手册来学习：

<https://www.ffmpeg.org/ffmpeg.html>

本次就暂时不分享关于ffplay来演示视频播放了，演示效果如下：



文章相关参考：

<https://www.ffmpeg.org/ffmpeg.html>

<https://www.ffmpeg.org/about.html>

<https://ke.qq.com/webcourse/index.html#cid=468797&term_id=100561187&taid=4217133899130685&vid=5285890796072844516>