

# ffmpeg 常用指令

---

## 去除logo

```
ffplay -i input.mp4 -vf delogo=x=72:y=32:w=168:h=86:show=1
```

```
ffmpeg -i output.mp4 -vf delogo=x=72:y=32:w=168:h=86 output.mp4
```

## 基础指令

### 拷贝视频轨道

```
-vcodec copy
```

### 拷贝音轨

```
-acodec copy
```

### 指定码率

```
-b:v 3000k
```

## 帧率 fps

- frame per second
- 每一秒钟的影片含有多少张静态图片

## 分辨率

显示分辨率（屏幕分辨率）是屏幕图像的精密度，是指显示器所能显示的像素有多少。

图像分辨率则是单位英寸中所包含的像素点数

## 比特率

比特率经常在电信领域用作连接速度、传输速度、信息传输速率和数字带宽容量的同义词

比特率越高，传送的数据越大，还原后的音质、画质就越好，在视频领域,比特率常翻译为码率

## 编码格式

H.264的特点就是相较于其他编码格式来说，它具有低码率、容错率高、网络适应性强以及在同等条件下画面的质量比其他编码高。并且在同样的画面质量、码率的情况下，使用它编码的视频占用的内存小。

## 加解密

- 加密

```
ffmpeg -y -i 9af1ce88cb3afae94600496d56b0898b.mp4 -vcodec copy -acodec copy -encryption_key 38453532423944464234453338424435 -encryption_kid 34373237656539333935663332346364 -encryption_scheme cenc-aes-ctr -f mp4 -movflags faststart enc_9af1ce88cb3afae94600496d56b0898b.mp4
```

## 解密

```
ffmpeg -y -decryption_key 38453532423944464234453338424435 -i enc_9af1ce88cb3afae94600496d56b0898b.mp4 -vcodec copy -acodec copy -f mp4 -movflags faststart new_9af1ce88cb3afae94600496d56b0898b.mp4
```

## 原始GPU指令（原码率）

```
/root/bin/ffmpeg -hwaccel cuvid -hwaccel_output_format cuda -c:v h264_cuvid -noautorotate -threads 8 -i /data/zt_transcode/download/test2.mp4 -c:v h264_nvenc -r 24 -b:v 86288 -ar 32000 -max_muxing_queue_size 99999 -f mp4 -movflags faststart -y ori_trans_7a4676ee611d7ae8283fe5b27fabdb44.mp4
```

保持原视频时间不变

```
/root/bin/ffmpeg -hwaccel cuvid -hwaccel_output_format cuda -noautorotate -threads 8 -i /data/zt_transcode/download/test2.mp4 -r 24 -copyts -c:v copy -c:a copy output-copy.mp4
```

## 720格式转码

```
ffmpeg -threads 8 -i d251cb626bb3a487960fb66db22046d5.mp4 -r 24 -b:v 740k -b:a 60k -maxrate 1200k -bufsize 1200k -bf 2 -g 24 -c:v libx264 -vsync cfr -strict -2 -vf scale=-2:720 -max_muxing_queue_size 99999 -ar 32000 -ac 2 -copytb 1 -f mp4 -movflags faststart -preset veryfast 720_d251cb626bb3a487960fb66db22046d5.mp4
```

```
ffmpeg -threads 8 -i 7a4676ee611d7ae8283fe5b27fabdb44.mp4 -r 24 -af asetpts=PTS -b:v 740k -b:a 60k -maxrate 1200k -bufsize 1200k -bf 2 -g 24 -c:v libx264 -vsync cfr -strict -2 -vf scale=-2:720 -max_muxing_queue_size 99999 -ar 32000 -ac 2 -copytb 1 -f mp4 -movflags faststart -preset veryfast 720_7a4676ee611d7ae8283fe5b27fabdb44.mp4
```

保持视频流时长不变的720格式GPU转码指令

```
/root/bin/ffmpeg -hwaccel cuvid -hwaccel_output_format cuda -noautorotate  
-threads 8 -i test_ori_trans_195266893067e6734829d205b449a18c.mp4 -r 24 -  
cq 35 -b:v 800k -maxrate 1200k -bf 2 -g 24 -c:v h264_nvenc -vsync cfr -  
strict -2 -max_muxing_queue_size 99999 -ar 32000 -copytb 1 -filter:v  
setpts=PTS,scale_npp=-1:720 -f mp4 -movflags faststart -y  
195266893067e6734829d205b449a18c_720.mp4
```

```
/root/bin/ffmpeg -hwaccel cuvid -hwaccel_output_format cuda -noautorotate -threads 8 -i  
/data/zt_transcode/download/test_195266893067e6734829d205b449a18c.mp4 -r 24 -cq 35 -b:v 800k -  
maxrate 1200k -bf 2 -g 24 -c:v h264_nvenc -vsync cfr -strict -2 -max_muxing_queue_size 99999 -ar 32000 -  
copytb 1 -vf setpts=PTS,scale_npp=-1:720 -f mp4 -movflags faststart -y  
/data/zt_transcode/download/720_195266893067e6734829d205b449a18c.mp4
```