







常用软件

FFMpeg | 实用的视频转换工具

朗读本文 

 五仁 · 4月18日 · 2019年

 工具 (<https://www.wurend.com/?tag=%e5%b7%a5%e5%85%b7>) · 软件 (<https://www.wurend.com/?tag=%e8%bd%af%e4%bb%b6>)

 2179次已读



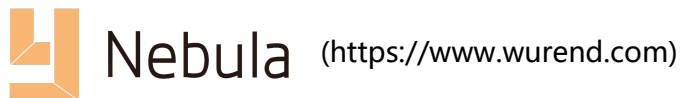


FFmpeg的下载与安装



FFmpeg是开源的。但我们不必去下载它的源代码。下载已经编译好的exe/dll即可。可以到 <http://www.ffmpeg.org/download.html> (<http://www.ffmpeg.org/download.html>)，然后点击“Windows Builds”，再根据你当前的机器系统，然后看是32位还是64位来决定下载哪个包。我用的windows 10 x64位的系统，因此我下载的是windos

版本，如下图所示：



FFmpeg的

关于

新闻

下载

文档

社区

邮件列表

IRC

论坛

错误报告

维基

开发

源代码

有助于

命运

代码覆盖率

更多

捐 ❤

聘请开发人员

联系

安全

法律

下载FFmpeg

下载

ffmpeg-4.0.tar.bz2

更多发布

更多的下载选项

如果您发现FFmpeg有用，欢迎您捐赠。

获取软件包

Windows软件包

视窗构建

获取来源

下载快照

您可以使用以下命令通过Git检索源代码：

```
git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg
```

无法访问Git或希望加快克隆速度并减少带宽使用量？

git 快照 浏览

FFmpeg一直是一个非常实验性和开发人员驱动的项目。它是许多多媒体项目的关键组件，并且不断添加新功能。开发分支快照在99%的时间内工作得非常好，所以人们不害怕使用它们。

Git存储库



下载完成之后，解压到硬盘上，假设路径为D:\me。启动一个控制台程序，然后执行DOS命令进入FFmpeg工具的bin目录，如下：

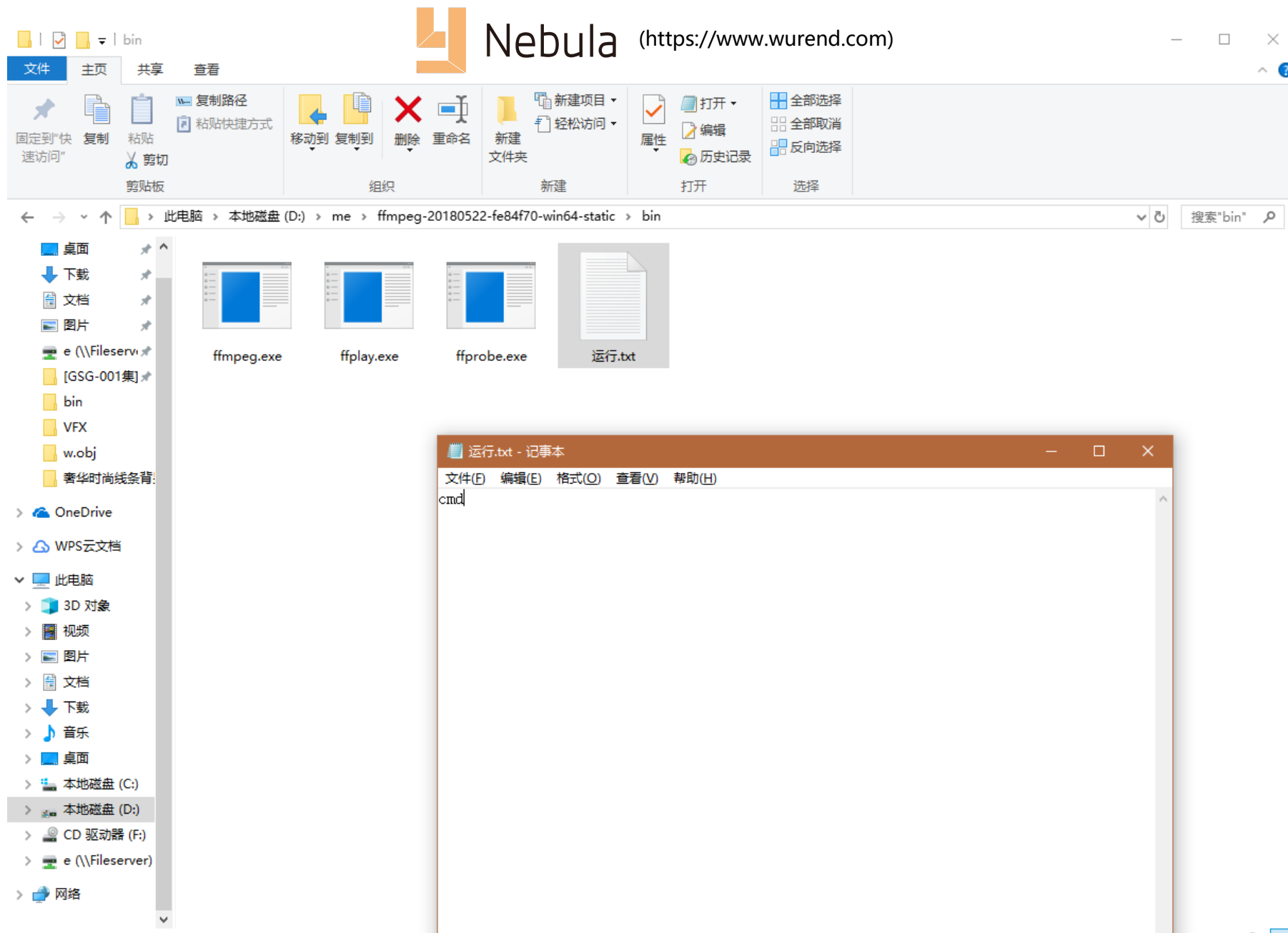


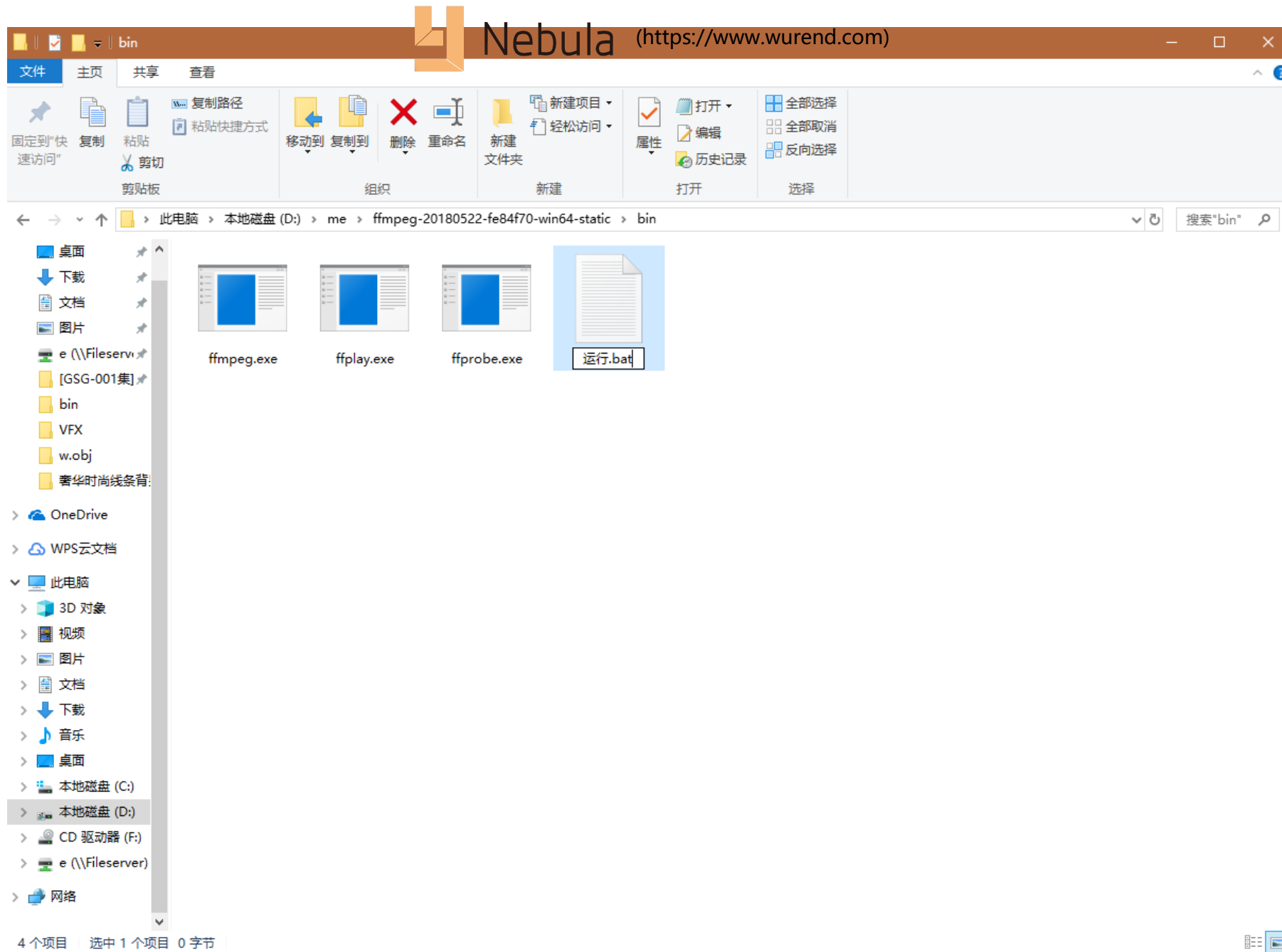
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.16299.431]
(c) 2017 Microsoft Corporation。保留所有权利。

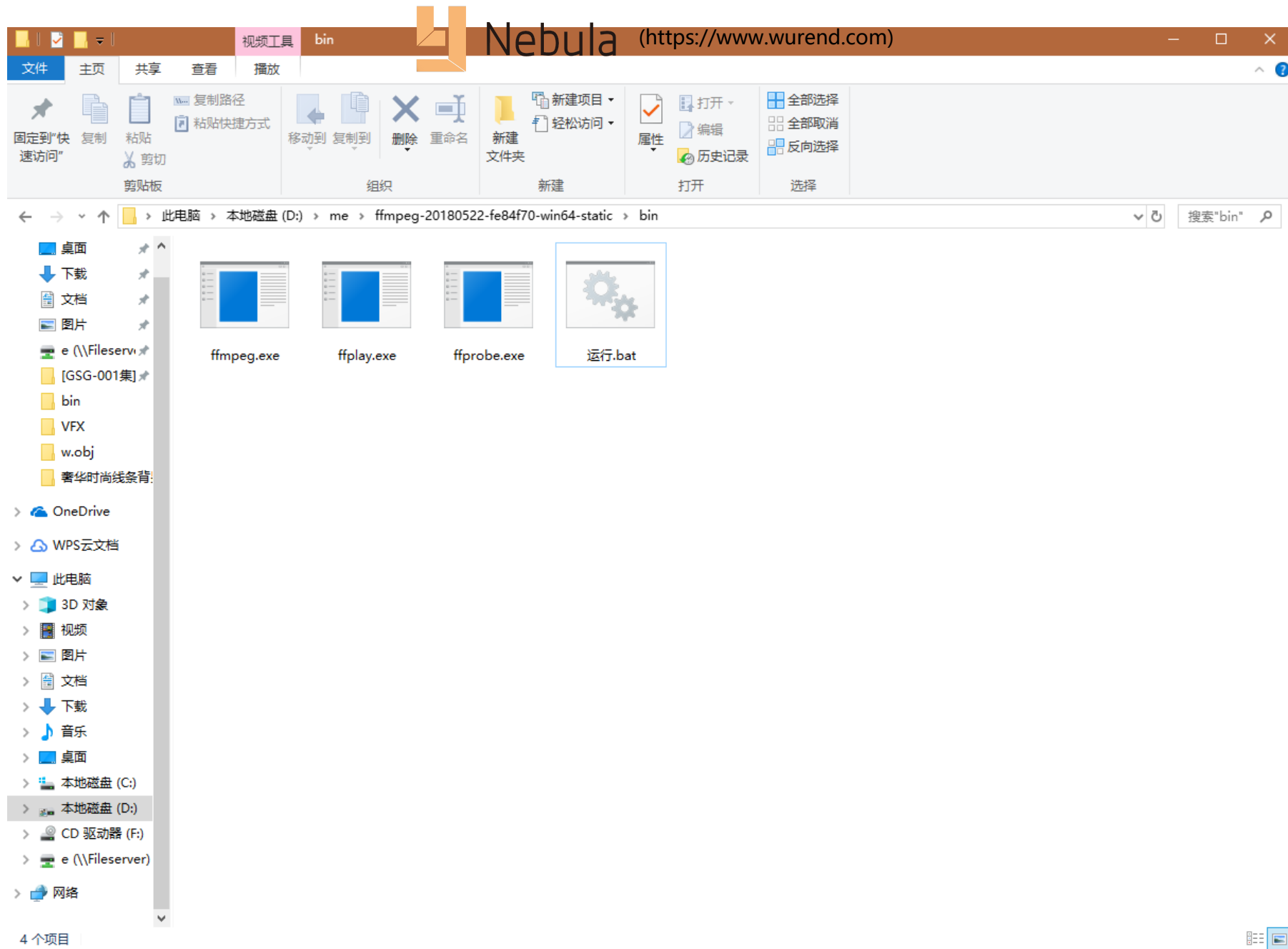
C:\Users\yt>d:

D:\>cd d:\me\ffmpeg-20180522-fe84f70-win64-static\bin
d:\me\ffmpeg-20180522-fe84f70-win64-static\bin>
```

或者你也可以在解压路径下新建一个txt文档，在文档里编辑“cmd”（引号里面内容），保存，然后把txt后缀修改为bat，如下图所示：







应用场景



FFmpeg将mkv带字幕转mp4

软字幕

MP4 格式支持流文字格式字幕，播放时可在播放器中选择对应的字幕，但该软字幕功能可能在有些播放器或者设备上不支持。

相关参数:

```
1 | ffmpeg -i input.mkv -map 0:0 -map 0:1 -map 0:s:34 -c:v copy -c:a copy -c:s mov_text -metadata:s:s:0 language=c
```

- `-map` 选项可将你想选择的原mkv文件中的视频流 (v) / 音频流 (a) / 字幕流 (s) 复制到输出中; : 后的数字代表原mkv文件中的第几个流 (mkv 里有多个音频流/字幕流)。
 - 本例中选择了第0个视频流 (默认), 第1个音频流, 第34个字幕流 (简体中文)。
- `-c` 选项则代编解码器名称, : 后的 v/a/s 意义同上;
 - 在本例中视频和音频的编解码器直接复制的原视频音频流的编码器, 所以编码器设置的为 `copy`。
 - 而 `mov_text` 则为此类软字幕编码器
- `-metadata` 选项可设置视频的元数据信息, 之后的 `:s:s:0` 为可选项, 空格之后接一个要设置元信息的键值对。
 - 本例中的 `:s:0` 表示对**输出文件**第0个字幕流进行元信息的设置, 将字幕标识为简体中文。
 - 如果要设置第0个视频流的元信息, 则在 `-metadata` 之后加 `:s:a:0`
- 如不需要字幕, 直接将 `-c:s mov_text -metadata:s:s:0 language=chs` 该条命令删除即可, 命令如下:

```
1 | ffmpeg -i input.mkv -map 0:0 -map 0:1 -qscale 2 -vcodec libx264 output.mp4
```



外挂字幕



Nebula (<https://www.wurend.com>)



尴尬的是 iOS 版的暴风影音不支持 `mov_text` 编码的软字幕，那就只能尝试能不能用外挂字幕了。

直接通过 `-map` 选项将简体中文字幕流导出到 `output.srt` 文件

```
1 | ffmpeg -i input.mkv -map 0:s:34 output.srt
```

在暴风影音中选择这个 `output.srt` 文件，显示出来却是乱码，然而在电脑上的播放器上却一切正常，这暴风影音真的是不行啊。

硬字幕

字幕文件嵌入视频

将刚才导出的 `srt` 文件“烧入”到先前生成的那个 `output.mp4` 文件中，因为此处要对有字幕的视频帧进行压制，所以会比较慢。

```
1 | ffmpeg -i output.mp4 -vf subtitles=output.srt output-Subtitles.mp4
```

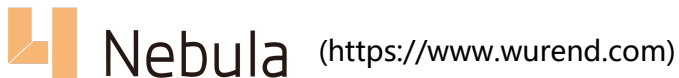
`-vf` 选项代表使用视频过滤器 (Video Filters)

- `subtitles` 就代表设置字幕为 `=` 之后接的字幕文件 / 流
- 如果需要嵌入的是 `ass` 格式的字幕，只需要将 `subtitles` 替换为 `ass`，然后 `=` 之后接上 `ass` 字幕文件即可，例如 `ffmpeg -i output.mp4 -vf ass=subtitle.ass output-Subtitles.mp4`



这次生成的 `mp4` 文件终于能在暴风影音里播放了，真是够折腾的.....

视频字幕流嵌入视频



但我不想生成中间的 srt 字幕文件，我们可以直接将 mkv 的字幕流在转成 mp4 的时候直接压制进视频文件中

```
1 | ffmpeg -i input.mkv -filter_complex "[0:v:0]subtitles=input.mkv:si=34[v]" -map "[v]" -map 0:a:0 -c:a copy outp
```

此处使用复杂图像过滤器 (Complex filtergraphs) 选项 `-filter_complex` 来处理视频，将第0个视频流加上 `input.mkv` 中的第34个字幕流一起压制到视频流 `v` 中，同时通过 `-map` 选项将第0个音频流直接导出到 `output.mp4` 中。

格式转换

我想把用iPhone拍的.MOV文件转成.avi文件。最简单了，可以执行下面的命令：

```
ffmpeg -i D:\Media\IMG_0873.MOV D:\Media\output.avi
```

意思是，把D:\Media目录下的源文件IMG_0873.MOV（视频：h.264，音频：aac）转换成output.avi（编码格式自动选择为：视频mpeg4，音频mp3），目标文件仍然保存到D:\Media目录下。问题来了：我想自己指定编码格式，怎么办呢？一种方法是，通过目标文件的扩展名（.flv、.mpg、.mp4、.wmv等）来控制，比如：

```
ffmpeg -i D:\Media\IMG_0873.MOV D:\Media\output2.flv
```

另一种方法是通过`-c:v`参数来控制，比如我想输出的视频格式是H.265（警告：编码时间会比较长哦）。命令行如下：

```
ffmpeg -i D:\Media\IMG_0873.MOV -c:v libx265 D:\Media\output265.avi
```

注：可以先用`ffmpeg -encoders`命令查看一下所有可选的编码格式。



不再深究了，我们继续。我发现源文件的帧尺寸是1920x1080，我不需要这么大——能有720 x 480就够了。于是，就要用上-s参数了。为了保证图像缩放后的质量，最好加上码流参数-b:v。如下：

```
ffmpeg -i D:\Media\IMG_0873.MOV -s 720x480 -b:v 1500k D:\Media\output2.avi
```

还可以更简单一点，使用-target参数匹配行业标准，参数值可以是vcd、svcd、dvd、dv、dv50等，可能还需要加上电视制式作为前缀（pal-、ntsc-或film-）。如下：

```
ffmpeg -i D:\Media\IMG_0873.MOV -target pal-dvd D:\Media\output2dvd.avi
```

又来一个问题：我发现用手机拍的视频中，有些是颠倒的，我想让它顺时针旋转90度。这时候，可以使用-vf参数加入一个过滤器，如下：

```
ffmpeg -i D:\Media\IMG_0873.MOV -vf "rotate=90*PI/180" D:\Media\output3.avi
```

注：如果想逆时针旋转90度，90前面加个负号就可以了。

如果我只需要从源视频里截取一小段，怎么办呢？比如从第2秒的地方开始，往后截取10秒钟。命令行可以这样：

```
ffmpeg -ss 2 -t 10 -i D:\Media\IMG_0873.MOV D:\Media\output4.avi
```

注：这种情况下，-ss和-t参数必须放在-i前面，表示是限定后面跟着的输入文件的。

视频合成

我发现，用手机拍的视频有时候背景噪音比较大。怎么把噪音去掉，换成一段美妙的音乐呢？使用FFmpeg也能轻易做到。

第一步：把源文件里的音频去掉，生成一个临时文件tmp.mov

ffmpeg -i D:\Media\IMG_0873.MOV -c:v copy -an D:\Media\tmp.mov (https://www.wurend.com)



注：-vcodeccopy的意思是对源视频不解码，直接拷贝到目标文件；-an的意思是将源文件里的音频丢弃。

第二步：把这个无声的视频文件（tmp.mov）与一个音乐文件（music.mp3）合成，最终生成output.mov

ffmpeg -i D:\Media\tmp.mov -ss 30 -t 52 -i D:\Media\music.mp3 -vcodec copy D:\Media\output5.avi

为了保证良好的合成效果，音乐时长必须匹配视频时长。这里我们事先知道视频时长为52秒，于是截取music.mp3文件的第30秒往后的52秒与视频合成。另外，为了保证音频时长截取的准确性，我们这里没有使用-acodec copy，而是让音频重新转码。

还有一种情况：我们希望在一段视频上叠加一张图片。可以简单实现如下：

ffmpeg -i D:\Media\IMG_0873.MOV -i D:\Media\logo.png -filter_complex 'overlay' D:\Media\output6.avi

视频播放

格式转换或合成之后，我们需要试着播放一下。播放器的选择很多。这里顺手用ffplay工具也行：

ffplay -i D:\Media\output6.avi

获取视频信息

有时候，我只是想看看这个视频文件的格式信息。可以用ffprobe工具：

ffprobe -i D:\Media\IMG_0873.MOV



其他应用

FFmpeg的功能非常强大。关键是要理解每个参数的意义,并且巧妙搭配。必要的话,就把在线文档完整读一遍吧: 官方文档 (<http://www.ffmpeg.org/ffmpeg.html>)



参数说明

转换文件test.avi到test.flv

```
ffmpeg -i test.avi -ab 56 -ar 22050 -b 500 -r 29.97 -s 320×240 test.flv
```

对文件抓缩微图:

```
ffmpeg -i "test.avi" -y -f image2 -ss 4 -t 0.001 -s 350×240 test.jpg
```

对已有flv文件抓图:

```
ffmpeg -i "test.flv" -y -f image2 -ss 8 -t 0.001 -s 350×240 test.jpg
```

Ffmpeg转换命令(测试未通过)

```
ffmpeg -y -i test.avi -bitexact -vcodec h263 -b 128 -r 15 -s 176×144 -acodec aac -ac 2 -ar 22500 -ab 24 -f 3gp test.3gp
```

```
ffmpeg -y -i test.avi -ac 1 -acodec amr_nb -ar 8000 -s 176×144 -b 128 -r 15 test.3gp
```

-y (覆盖输出文件, 即如果1.***文件已经存在的话, 不经提示就覆盖掉了)

-i "1.avi" (输入文件是和ffmpeg在同一目录下的1.avi文件, 可以自己加路径, 改名字)

-title "Test" (在PSP中显示的影片的标题)



-vcodec xvid (使用XVID编码压缩视频, 不能改的) (https://www.wurend.com)



-s 368×208 (输出的分辨率为368×208, 注意片源一定要是16:9的不然会变形)

-r 29.97 (帧数, 一般就用这个吧)

-b 1500 (视频数据流量, 用-b xxxx的指令则使用固定码率, 数字随便改, 1500以上没效果; 还可以用动态码率如: -qscale 4 和-qscale 6, 4的质量比6高)

-acodec aac (音频编码用AAC)

-ac 2 (声道数1或2)

-ar 24000 (声音的采样频率, 好像PSP只能支持24000Hz)

-ab 128 (音频数据流量, 一般选择32、64、96、128)

-vol 200 (200%的音量, 自己改)

-f psp (输出psp专用格式)

-muxvb 768 (好像是给PSP机器识别的码率, 一般选择384、512和768, 我改成1500, PSP就说文件损坏了)

"1.***" (输出文件名, 也可以加路径改文件名)



-ab bitrate 设置音频码率

-ar freq 设置音频采样率



-ss 指定时间点开始转换任务,(time_offset time_start time_offset) -ss后跟的时间单位为秒 .
(<https://www.wurend.com/>)



-s 320×240 指定分辨率

-r 29.97 帧速率

-bitexact 使用标准比特率

-vcodec xvid 使用xvid压缩

例子: ffmpeg -y -i "1.avi" -title "Test" -vcodec xvid -s 368×208 -r 29.97 -b 1500 -acodec aac -ac 2 -ar 24000 -ab 128 -vol 200 -f psp -muxvb 768 "1.***"

解释: 以上命令可以在Dos命令行中输入, 也可以创建到批处理文件中运行。不过, 前提是: 要在ffmpeg所在的目录中执行 (转换君所在目录下面的cores子目录)。

参数:

-y (覆盖输出文件, 即如果1.***文件已经存在的话, 不经提示就覆盖掉了)

-i "1.avi" (输入文件是和ffmpeg在同一目录下的1.avi文件, 可以自己加路径, 改名字)

-title "Test" (在PSP中显示的影片的标题)

-vcodec xvid (使用XVID编码压缩视频, 不能改的)

-s 368×208 (输出的分辨率为368×208, 注意片源一定要是16:9的不然会变形)



-r 29.97 (帧数, 一般就用这个吧)



Nebula (<https://www.wurend.com>)



-b 1500 (视频数据流量, 用-b xxxx的指令则使用固定码率, 数字随便改, 1500以上没效果; 还可以用动态码率如: -qscale 4 和-qscale 6, 4的质量比6高)

-acodec aac (音频编码用AAC)

-ac 2 (声道数1或2)

-ar 24000 (声音的采样频率, 好像PSP只能支持24000Hz)

-ab 128 (音频数据流量, 一般选择32、64、96、128)

-vol 200 (200%的音量, 自己改)

-f psp (输出psp专用格式)

-muxvb 768 (好像是给PSP机器识别的码率, 一般选择384、512和768, 我改成1500, PSP就说文件损坏了)

"1.***" (输出文件名, 也可以加路径改文件名)

a) 通用选项

-L license



-h 帮助

-fromats 显示可用的格式，编解码的，协议的。Nebula (https://www.wurend.com)



-f fmt 强迫采用格式fmt

-l filename 输入文件

-y 覆盖输出文件

-t duration 设置纪录时间 hh:mm:ss[.xxx]格式的记录时间也支持

-ss position 搜索到指定的时间 [-]hh:mm:ss[.xxx]的格式也支持

-title string 设置标题

-author string 设置作者

-copyright string 设置版权

-comment string 设置评论

-target type 设置目标文件类型(vcd,svcd,dvd) 所有的格式选项（比特率，编解码以及缓冲区大小）自动设置，只需要输入如下的就可以了：

ffmpeg -i myfile.avi -target vcd /tmp/vcd.mpg

-hq 激活高质量设置



-itsoffset offset 设置以秒为基准的时间偏移。该选项影响所有后面的输入文件。该偏移被加到输入文件的时戳，定义一个正偏移意味着相应的流被延迟了 offset秒。[-hh:mm:ss[.xxx]]的格式也支持

b) 视频选项

-b bitrate 设置比特率, 缺省200kb/s

-r fps 设置帧频 缺省25

-s size 设置帧大小 格式为WXH 缺省160X128.下面的简写也可以直接使用:

Sqcif 128X96 qcif 176X144 cif 252X288 4cif 704X576

-aspect aspect 设置纵横比 4:3 16:9 或 1.3333 1.7777

-croptop size 设置顶部切除带大小 像素单位

-cropbottom size -cropleft size -cropright size

-padtop size 设置顶部补齐的大小 像素单位

-padbottom size -padleft size -padright size -padcolor color 设置补齐条颜色(hex,6个16进制的数, 红:绿:兰排列, 比如000000代表黑色)

-vn 不做视频记录

-bt tolerance 设置视频码率容忍度kbit/s

-maxrate bitrate设置最大视频码率容忍度

-minrate bitrate 设置最小视频码率容忍度



Nebula (<https://www.wurend.com>)



-bufsize size 设置码率控制缓冲区大小

-vcodec codec 强制使用codec编解码方式。 如果用copy表示原始编解码数据必须被拷贝。

-sameq 使用同样视频质量作为源 (VBR)

-pass n 选择处理遍数 (1或者2) 。两遍编码非常有用。第一遍生成统计信息，第二遍生成精确的请求的码率

-passlogfile file 选择两遍的纪录文件名为file

c)高级视频选项

-g gop_size 设置图像组大小

-intra 仅适用帧内编码

-qscale q 使用固定的视频量化标度(VBR)

-qmin q 最小视频量化标度(VBR)

-qmax q 最大视频量化标度(VBR)

-qdiff q 量化标度间最大偏差 (VBR)

-qblur blur 视频量化标度柔化(VBR)



-qcomp compression 视频量化标度压缩 (https://www.wurend.com)

-rc_init_cplx complexity 一遍编码的初始复杂度

-b_qfactor factor 在p和b帧间的qp因子

-i_qfactor factor 在p和i帧间的qp因子

-b_qoffset offset 在p和b帧间的qp偏差

-i_qoffset offset 在p和i帧间的qp偏差

-rc_eq equation 设置码率控制方程 默认tex^qComp

-rc_override override 特定间隔下的速率控制重载

-me method 设置运动估计的方法 可用方法有 zero phods log x1 epzs(缺省) full

-dct_algo algo 设置dct的算法 可用的有 0 FF_DCT_AUTO 缺省的DCT 1 FF_DCT_FASTINT 2 FF_DCT_INT 3 FF_DCT_MMX 4 FF_DCT_MLIB 5 FF_DCT_ALTIVEC

-idct_algo algo 设置idct算法。可用的有 0 FF_IDCT_AUTO 缺省的IDCT 1 FF_IDCT_INT 2 FF_IDCT_SIMPLE 3 FF_IDCT_SIMPLEMMX 4 FF_IDCT_LIBMPEG2MMX 5 FF_IDCT_PS2 6 FF_IDCT_MLIB 7 FF_IDCT_ARM 8 FF_IDCT_ALTIVEC 9 FF_IDCT_SH4 10 FF_IDCT_SIMPLEARM

-er n 设置错误残留为n 1 FF_ER_CAREFULL 缺省 2 FF_ER_COMPLIANT 3 FF_ER_AGGRESSIVE 4 FF_ER_VERY_AGGRESSIVE

-ec bit_mask 设置错误掩蔽为bit_mask, 值为如下值的位掩码 1 FF_EC_GUESS_MVS (default=enabled) 2 FF_EC_DEBLOCK (default=enabled)

-bf frames 使用frames B 帧, 支持mpeg1,mpeg2,mpeg4

-mbd mode 宏块决策 0 FF_MB_DECISION_SIMPLE 使用mb_cmp 1 FF_MB_DECISION_BITS 2 FF_MB_DECISION_RD

-4mv 使用4个运动矢量 仅用于mpeg4

-part 使用数据划分 仅用于mpeg4

-bug param 绕过没有被自动监测到编码器的问题

-strict strictness 跟标准的严格性

-aic 使能高级帧内编码 h263+

-umv 使能无限运动矢量 h263+

-deinterlace 不采用交织方法

-interlace 强迫交织法编码 仅对mpeg2和mpeg4有效。当你的输入是交织的并且你想要保持交织以最小图像损失的时候采用该选项。可选的方法是不交织, 但是损失更大

-psnr 计算压缩帧的psnr

-vstats 输出视频编码统计到vstats_hhmmss.log

-vhook module 插入视频处理模块 module 包括了模块名和参数，用空格分开 (https://www.wurend.com)



D)音频选项

-ab bitrate 设置音频码率

-ar freq 设置音频采样率

-ac channels 设置通道 缺省为1

-an 不使能音频纪录

-acodec codec 使用codec编解码

E)音频/视频捕获选项

-vd device 设置视频捕获设备。比如/dev/video0

-vc channel 设置视频捕获通道 DV1394专用

-tvstd standard 设置电视标准 NTSC PAL(SECAM)

-dv1394 设置DV1394捕获

-av device 设置音频设备 比如/dev/dsp



F)高级选项

-map file:stream 设置输入流映射



-debug 打印特调试信息

-benchmark 为基准测试加入时间

-hex 倾倒每一个输入包

-bitexact 仅使用位精确算法 用于编解码测试

-ps size 设置包大小, 以bits为单位

-re 以本地帧频读数据, 主要用于模拟捕获设备

-loop 循环输入流。只工作于图像流, 用于ffserver测试

转视频格式

```
ffmpeg -i source.mp4 -c:v libx264 -crf 24 destination.flv
```

其中 -crf 很重要, 是控制转码后视频的质量, 质量越高, 文件也就越大。

此值的范围是 0 到 51: 0 表示高清无损; 23 是默认值 (如果没有指定此参数) ; 51 虽然文件最小, 但效果是最差的。

值越小, 质量越高, 但文件也越大, 建议的值范围是 18 到 28。而值 18 是视觉上看起来无损或接近无损的, 当然不代表是数据 (技术上) 的转码无损。





☆ 相关文章

用bat命令运行ffmpeg进行批量
转码

7月24日 · 2020年

(https://www.wurend.com/?p=1401)

又一款全能的开源下载神器：
Motrix

12月30日 · 2019年

(https://www.wurend.com/?p=1183)

AE插件 | 超强粒子插件

8月23日 · 2019年

(https://www.wurend.com/?p=976)

💬 0 条回应



如果你还有什么好的意见，请给我留言！



Nebula (<https://www.wurend.com>)



昵称

提交评论

Copyright ©2018-2021 五仁的记录

京ICP备18057649号 (<https://beian.miit.gov.cn>)

