**FFMPEG功能和参数**

1．视频音频格式转换

Ffmpeg能使用任何支持的格式和协议作为输入：

\*比如你可以输入YUV文件：ffmpeg -i /tmp/test%d.Y /tmp/out.mpg

它将要使用如下文件：

      /tmp/test0.Y, /tmp/test0.U, /tmp/test0.V,

/tmp/test1.Y, /tmp/test1.U, /tmp/test1.V,等等…

   \*你能输入原始的YUV420P文件：ffmpeg -i /tmp/test.yuv /tmp/out.avi

   原始的YUV420P文件包含原始的YUV极性，每帧以Y平面开始，跟随U和V平面，它们是Y平面水平垂直的一半分辨率

   \*你能输出原始的YUV420P文件

   ffmpeg -i mydivx.avi -o hugefile.yuv

      \*你能设置几个输入文件和输出文件

      ffmpeg -i /tmp/a.wav -s 640x480 -i /tmp/a.yuv /tmp/a.mpg      上面的命令行转换音频文件a.wav和原始的YUV 视频文件 a.yuv到mpeg文件a.mpeg

      \*你也能同时转换音频和视频

ffmpeg -i /tmp/a.wav -ar 22050 /tmp/a.mp2

上面的命令行转换a.wav的采样率到22050HZ并编码为mpeg音频

\*你也能同时编码到几种格式并且在输入流和输出流之间建立映射

      ffmpeg -i /tmp/a.wav -ab 64 /tmp/a.mp2 -ab 128 /tmp/b.mp2 -map 0:0 -map 0:0

      上面的命令行转换一个64Kbits 的a.wav到128kbits的a.mp2 ‘-map file:index’在输出流的顺序上定义了那一路输入流是用于每一个输出流的，

      转码解密的VOB:

      ffmpeg -i snatch\_1.vob -f avi -vcodec mpeg4 -b 800 -g 300 -bf 2 -acodec mp3 -ab 128 snatch.avi

      上面的命令行将vob的文件转化成avi文件，mpeg4的视频和mp3的音频。注意命令中使用了B帧，所以mpeg4流是divx5兼容的。GOP大小是300意味着29.97帧频下每10秒就有INTRA帧。该映射在音频语言的DVD转码时候尤其有用

2．  Ffmpeg使用语法

       ffmpeg [[options][`-i' input\_file]]... {[options] output\_file}...

     如果没有输入文件，那么视音频捕捉就会起作用。

     作为通用的规则，选项一般用于下一个特定的文件。如果你给 –b 64选项，改选会设置下一个视频速率。对于原始输入文件，格式选项可能是需要的。

     缺省情况下，ffmpeg试图尽可能的无损转换，采用与输入同样的音频视频参数来输出。

3．选项

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。。。

     -f fmt 强迫采用格式fmt

     -I 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 设置横纵比 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 bitreate 设置最小视频码率容忍度

              -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(VBR)

             -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,该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

      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 打印特定调试信息

             -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

先从ffmpeg开始。  
http://ffmpeg.sourceforge.net上有说明，音视频的分离，转换，解码的完全解决方案。  
其中最重要的就是libavcodec库。它被mplayer或者xine使用作为解码器。还有，国内比较流行的播放器影音风暴或MyMPC的后端ffdshow也是使用ffmpeg的解码库的。  
  
ffmpeg包括一组软件，ffmpeg用于对媒体文件进行处理，ffserver是一个http的流媒体服务器，ffplay是一个基于SDL的简单播放器。两个库文件libavcodec和libavformat。  
  
ffmpeg作为媒体文件处理软件，基本用法如下：  
ffmpeg -i INPUTfile [OPTIONS] OUTPUTfile  
输入输出文件通常就是待处理的多媒体文件了。可以是纯粹的音频文件，纯粹的视频文件，或者混合的。  
大部分常见的格式都能够“通杀”。象常见的各种mpeg，AVI封装的DIVX和Xvid等等  
具体的格式支持列表可以使用ffmpeg -formats查看或直接查阅文档。  
  
另：由于Linux把设备视为文件，因此－i选项后可以跟设备名。比如DV，视频卡，光驱或者其它的各类设备。  
输出的内容通过Options调整。列出几个主要的选项  
-vcodec 视频流编码方式  
-b       视频流码率（默认只有200k，一般都需要手动设置，具体的数值视codec选择而定）  
-r        视频流帧数（一般说来PAL制式通常用25，NTSC制式通常用29）  
-s        视频解析度（分辨率，也要视codec和你的需要而定。通常改变某个视频流的解析度是很耗费CPU的事情。另：具体写法使用“数字x数字”的形式。中间是小写字母“x”，这个用过mplayer的应该都知道）  
-t        处理持续时间。  
-acodec  音频流编码方式  
-ab         音频流码率（默认是同源文件码率，也需要视codec而定）  
-ar          音频流采样率（大多数情况下使用44100和48000，分别对应PAL制式和NTSC制式，根据需要选择）  
  
还有些可能需要用到的选项如  
-vn和-an分别是屏蔽视频流和屏蔽音频流，分别对源文件处理一次即可得到分离的音频和视频  
-author -title分别是设置媒体文件的作者和title  
-f选项是强制使用某种格式  
－target type是使用预置的格式转换（可以转成dvd，vcd或svcd）  
  
除此之外还有些更高级的选项，如设定vbr，或设定high quality，或者设定vbr的buff和max/min码率，象一般我们自用的dvd抓轨啦，DV转vcd或dvd啦，网上下载的电影转成vcd或dvd都不一定需要用到它们。  
具体的使用方法在man里面有介绍。简单明了。  
  
少许使用经验：  
1：ffmpeg对于rm的处理能力实在不敢恭维。也许是因为我主要使用二进制包安装的缘故，对于Real媒体格式只能处理老式的RV8编码的格式。而且效果不佳。  
2：格式转换是一件很耗费CPU资源的事情。虽说ffmpeg已经比WinAVI啦，TmpgEnc这些win下的非专业级视频处理软件做的好些了。毕竟我们可以把ffmpeg运行的时候放到后台。  
3：ffmpeg不是万能的，虽说支持的格式很多，但是如果你不是用的最新CVS出来的版本，可能碰上某些古怪的媒体文件就要郁闷。  
4：ffmpeg全部是命令行操作。哪位达人写个GUI前端出来就可以让不少菜鸟脱离苦海了。还有就是不能批量处理，但是这个可以用shell帮忙解决。

[编辑此页面（如果您有权限）](http://docs.google.com/Doc?tab=edit&dr=true&id=ajf3xsjz9vhb_bdht5mzm2vsx4) | [报告滥用行为](http://docs.google.com/abuse?id=0AVS9AseQYmWPYWpmM3hzano5dmhiX2JkaHQ1bXptMnZzeDQ)

[Google 文档 － 网络文字处理、演示文稿和电子表格。](http://docs.google.com/)