[mplayer的执行过程分析（本地、网络）](http://blog.csdn.net/lqx4_3/article/details/7481807)

标签： [网络](http://www.csdn.net/tag/%e7%bd%91%e7%bb%9c)[stream](http://www.csdn.net/tag/stream)[buffer](http://www.csdn.net/tag/buffer)[audio](http://www.csdn.net/tag/audio)[video](http://www.csdn.net/tag/video)[codec](http://www.csdn.net/tag/codec)

2012-04-20 16:04 1667人阅读 [评论](http://blog.csdn.net/lqx4_3/article/details/7481807#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/lqx4_3/article/details/7481807#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

linux应用（12） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

main函数流程分析，分本地和网络两种情况   
  
1. 本地文件的播放，以mp3文件的播放为例，命令行输入：mplayer test.mp3   
   只给出关键步骤   
2. open\_stream，                                        打开文件流   
        open\_stream\_full                                找到合适的打开文件的方式，对于本地文件选择   
            open\_stream\_plugin                          stream\_info\_file，使用它的open\_f函数读取文件   
                open\_f                                  成功，返回文件句柄，文件访问函数   
   对于流媒体文件，第8行改成如下的第10行，表明流媒体文件选择stream\_info\_rtsp\_sip   
                open\_live\_rtsp\_sip   
                                    
3. mpctx->stream                                        2854行，记录文件流的信息，文件句柄，访问函数等   
4. demux\_open                                           打开分离器（音视频）   
        demux\_open\_stream                               从demuxer\_list中找到合适的demuxer描述符，此处为demuxer\_desc\_audio   
            new\_demuxer   
                stream\_seek   
                    cache\_stream\_seek\_long   
                        stream\_seek\_long                读取部分数据至内存   
                            stream\_fill\_buffer          while(stream\_fill\_buffer(s) > 0 && pos >= 0) {   
                                  
            demux\_audio\_open                            使用demuxer\_desc\_audio的打开函数，读取媒体数据的信息，头信息   
                                                        对于m4v视频demuxer\_desc\_lavf\_preferred，打开demux\_open\_lavf   
   对于流媒体文件，上面的第15行换成如下的第18行，使用demuxer\_desc\_rtp，使用sdp文件建立流媒体会话，   
            demux\_open\_rtp                              创建媒体会话，创建音视频数据接收会话，初始化音视频的codec   
                            
初始化视频解码器及输出设备   
5. reinit\_video\_chain   
        init\_best\_video\_out                             从video\_out\_drivers数组中找出视频输出设备video\_out\_gl   
            preinit(vo\_gl.c)   
        vf\_open\_filter                                  在众多驱动中过滤出需要的那种   
            vf\_open\_plugin   
                open(vf\_vo.c)                           vf\_info\_vo   
        init\_best\_video\_codec                             
            init\_video                                  初始化视频解码器，从mpcodecs\_vd\_drivers数组中选择合适的解码器   
                                                        选择mpcodecs\_vd\_ffmpeg解码m4v的视频文件，vd\_ffmpeg.c   
初始化音频解码器及输出设备                                                           
6. reinit\_audio\_chain   
        init\_best\_audio\_codec                           mpcodecs\_ad\_drivers(见LIBAD\_EXTERN(mp3lib))，ad\_internal.h   
            init\_audio                                  选择mpcodecs\_ad\_mp3lib解码mp3的音频文件   
        init\_best\_audio\_out                             从audio\_out\_drivers选择音频输出设备audio\_out\_win32   
            init                                        初始化   
开始播放   
音频，mp3   
7. fill\_audio\_out\_buffers   
        decode\_audio                                    dec\_audio.c   
            mpadec->decode\_audio   
            MP3\_DecodeFrame                             获取数据并解码   
                read\_frame   
                    stream\_read\_frame\_body   
                        mp3\_read   
                            mplayer\_audio\_read   
                                demux\_read\_data   
                                    ds\_fill\_buffer   
                                        demux\_fill\_buffer   
                                            demux\_audio\_fill\_buffer(demux\_audio.c)   
                                                stream\_read   
                                                    cache\_stream\_fill\_buffer   
                                                        stream\_fill\_buffer   
                                                            fill\_buffer（stream\_file.c）   
        如果是流媒体，上面的47行将变为:     demux\_rtp\_fill\_buffer(demux\_rtp.cpp)                                                       
        play   
            write\_waveOutBuffer   
视频   
8. update\_video   
        video\_read\_frame   
            read\_video\_packet   
                demux\_pattern\_3   
                    demux\_read\_data   
                        ds\_fill\_buffer   
                            demux\_fill\_buffer   
                                demux\_lavf\_fill\_buffer（m4v）   
        对于流媒体的视频，上面的一行将变为:   
                                demux\_rtp\_fill\_buffer(demux\_rtp.cpp)                           
        decode\_video   
            decode(vd\_ffmpeg.c)                         avctx->opaque=sh; 在   
        filter\_video   
            put\_image(vf\_vo.c)       
              
              
=========================================================================================   
另一种读取视频文件的过程：   
demux\_mpg\_es\_fill\_buffer   
    ds\_read\_packet   
        stream\_read   
            cache\_stream\_fill\_buffer   
                stream\_fill\_buffer                      read   
                default:s->fill\_buffer   
                    fill\_buffer                         stream\_file.c   
windows下的界面部分，由编译开关HAVE\_NEW\_GUI控制   
mplayer.c文件下的   
2661～2668行，界面初始化   
2692～2727行，界面的消息循环，获取用户的命令，如文件名，执行   
  
视频驱动如果是directx，相应的处理函数在vo\_directx.c文件中       
ffmpeg编解码器中的avcodec\_decode\_video函数可以解析出视频的规格，即高和宽   
ff\_h263\_decode\_frame   
ff\_mpeg4\_decode\_picture\_header解析MPEG4视频的头信息   
decode\_vol\_header解析出高和宽     
  
具体过程如下：   
  
avcodec\_decode\_video   
    ff\_h263\_decode\_frame   
        MPV\_frame\_start   
            alloc\_picture   
                ff\_mpeg4\_decode\_picture\_header          解析出高和宽   
                get\_buffer   
                    init\_vo   
                        mpcodecs\_config\_vo              打印到屏幕                                                                                                                       
  
网络流媒体的视频高和宽必须解码才能获取，过程如下：   
  
如何从一个音视频文件中分出音视频流？   
参考m4v\_demuxer，分离器使用宏REGISTER\_DEMUXER注册或REGISTER\_MUXDEMUX，见allformats.c文件   
lavf\_check\_file在打开之前运行，从而获取真正的demuxer   
对于.m4v文件，打开demux\_open\_lavf中av\_open\_input\_stream获取流信息，音频视频   
demux\_open\_stream时，new\_demuxer开辟音视频流的缓存   
文件扩展名是.mp4时，打开流mov\_read\_header，首先重置音视频部分：   
stream\_reset   
    s->control（demux\_mov\_control）   
      
      
      
对于mp4流媒体  
根据sdp初始化：  
    bih->biCompression = sh\_video->format  
      = mmioFOURCC('m','p','4','v');  
        
确定视频解码器：  
init\_video中find\_video\_codec，使用上面的biCompression到文件codecs.conf文件中找到ffodivx，对于的  
驱动为ffmpeg

转自：<http://hi.baidu.com/tiplip/blog/item/4ab23d04f3cce96103088155.html>