

原 FFMPEG类库打开流媒体的方法（需要传参数的时候）

2013年11月06日 20:07:23 阅读数：33268

使用ffmpeg类库进行开发的时候，打开流媒体（或本地文件）的函数是avformat_open_input()。

其中打开网络流的话，前面要加上函数avformat_network_init()。

一般情况下，只要传入流媒体的url就可以了。但是在打开某些流媒体的时候，可能需要附加一些参数。

例如在播放中央人民广播电台的声音信号的时候，其url为"rtsp://mms.cnr.cn/cnr003?MzE5MTg0IzEjIzI5NjgwOQ=="

如果直接进行打开是不会成功的，我们可以使用ffplay做一下实验：

```
[plain]
1. ffplay rtsp://mms.cnr.cn/cnr003?MzE5MTg0IzEjIzI5NjgwOQ==
```

会出现错误：

Invalid data found when processing input

这时候我们需要指定其传输方式为TCP，需要将命令改为如下形式：

```
[plain]
1. ffplay -rtsp_transport tcp rtsp://mms.cnr.cn/cnr003?MzE5MTg0IzEjIzI5NjgwOQ==
```

附加了参数以后，发现就可以正常播放了。

此外还可以附加一些参数，比如

```
[plain]
1. ffplay -rtsp_transport tcp -max_delay 5000000 rtsp://mms.cnr.cn/cnr003?MzE5MTg0IzEjIzI5NjgwOQ==
```

在使用FFMPEG类库进行编程的时候，如何将这些附加的参数传递给avformat_open_input()呢？经过研究后发现，可以通过AVDictionary把参数传给avformat_open_input()。

看一下avformat_open_input()的定义：

```
[cpp]
1. /**
2.  * Open an input stream and read the header. The codecs are not opened.
3.  * The stream must be closed with av_close_input_file().
4.  *
5.  * @param ps Pointer to user-supplied AVFormatContext (allocated by avformat_alloc_context).
6.  *           May be a pointer to NULL, in which case an AVFormatContext is allocated by this
7.  *           function and written into ps.
8.  *           Note that a user-supplied AVFormatContext will be freed on failure.
9.  * @param filename Name of the stream to open.
10. * @param fmt If non-NULL, this parameter forces a specific input format.
11. *            Otherwise the format is autodetected.
12. * @param options A dictionary filled with AVFormatContext and demuxer-private options.
13. *               On return this parameter will be destroyed and replaced with a dict containing
14. *               options that were not found. May be NULL.
15. *
16. * @return 0 on success, a negative AVERROR on failure.
17. *
18. * @note If you want to use custom IO, preallocate the format context and set its pb field.
19. */
20. int avformat_open_input(AVFormatContext **ps, const char *filename, AVInputFormat *fmt, AVDictionary **options);
```



可以看出avformat_open_input()的第4个参数是一个AVDictionary类型的参数。这个参数就是传入的附加参数。

设置AVDictionary的时候会用到av_dict_set()。

下面看看把命令

```
[plain]
1. ffplay -rtsp_transport tcp -max_delay 5000000 rtsp://mms.cnr.cn/cnr003?MzE5MTg0IzEjIzI5NjgwOQ==
```

转化为代码实现的方式：

```
[cpp]    
1. AVFormatContext *pFormatCtx;  
2. pFormatCtx = avformat_alloc_context();  
3. ...代码略  
4. AVDictionary *avdic=NULL;  
5. char option_key[]="rtsp_transport";  
6. char option_value[]="tcp";  
7. av_dict_set(&avdic,option_key,option_value,0);  
8. char option_key2[]="max_delay";  
9. char option_value2[]="5000000";  
10. av_dict_set(&avdic,option_key2,option_value2,0);  
11. char url[]="rtsp://mms.cnr.cn/cnr003?MzE5MTg0IzEjIzI5NjgwOQ==";  
12.  
13. avformat_open_input(&pFormatCtx,url,NULL,&avdic);
```

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/14215393>

文章标签：[ffmpeg](#) [流媒体](#) [AVDictionary](#) [avformat_open_input](#)

个人分类：[FFMPEG](#)

所属专栏：[FFmpeg](#)

此PDF由[spygg](#)生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com