原 ffmpeg 从内存中读取数据（或将数据输出到内存）

2013年10月24日 00:03:25　阅读数：36162

更新记录（2014.7.24）：

1.为了使本文更通俗易懂，更新了部分内容，将例子改为从内存中打开。

2.增加了将数据输出到内存的方法。

从内存中读取数据

ffmpeg一般情况下支持打开一个本地文件，例如“C:\test.avi”

或者是一个流媒体协议的URL，例如“rtmp://222.31.64.208/vod/test.flv”

其打开文件的函数是avformat_open_input()，直接将文件路径或者流媒体URL的字符串传递给该函数就可以了。

但其是否支持从内存中读取数据呢？这个问题困扰了我很长时间。当时在做项目的时候，通过Winpcap抓取网络上的RTP包，打算直接送给ffmpeg进行解码。一直没能找到合适的方法。因为抓取的数据包是存在内存中的，所以无法传递给avformat_open_input()函数其路径（根本没有路径= =）。当然也可以将抓取的数据报存成文件，然后用ffmpeg打开这个文件，但是这样的话，程序的就太难控制了。

后来经过分析ffmpeg的源代码，发现其竟然是可以从内存中读取数据的，代码很简单，如下所示：

```cpp
AVFormatContext *ic = NULL;
ic = avformat_alloc_context();
```

```cpp
unsigned char * iobuffer=(unsigned char *)av_malloc(32768);
AVIOContext *avio =avio_alloc_context(iobuffer, 32768,0,NULL,fill_iobuffer,NULL,NULL);
ic->pb=avio;
err = avformat_open_input(&ic, "nothing", NULL, NULL);
```

关键要在avformat_open_input()之前初始化一个AVIOContext，而且将原本的AVFormatContext的指针pb（AVIOContext类型）指向这个自行初始化AVIOContext。当自行指定了AVIOContext之后，avformat_open_input()里面的URL参数就不起作用了。示例代码开辟了一块空间iobuffer作为AVIOContext的缓存。

fill_iobuffer则是将数据读取至iobuffer的回调函数。fill_iobuffer()形式（参数，返回值）是固定的，是一个回调函数，如下所示（只是个例子，具体怎么读取数据可以自行设计）。示例中回调函数将文件中的内容通过fread()读入内存。

```cpp
//读取数据的回调函数-----------------------
//AVIOContext使用的回调函数！
//注意：返回值是读取的字节数
//手动初始化AVIOContext只需要两个东西：内容来源的buffer，和读取这个Buffer到FFmpeg中的函数
//回调函数，功能就是：把buf_size字节数据送入buf即可
//第一个参数(void *opaque)一般情况下可以不用
int fill_iobuffer(void * opaque,uint8_t *buf, int bufsize){
    if(!feof(fp_open)){
        int true_size=fread(buf,1,buf_size,fp_open);
        return true_size;
    }else{
        return -1;
    }
}
```

整体结构大致如下：

```cpp
FILE *fp_open;

int fill_iobuffer(void *opaque, uint8_t *buf, int buf_size){
...
}

int main(){
    ...
    fp_open=fopen("test.h264","rb+");
    AVFormatContext *ic = NULL;
    ic = avformat_alloc_context();
    unsigned char * iobuffer=(unsigned char *)av_malloc(32768);
    AVIOContext *avio =avio_alloc_context(iobuffer, 32768,0,NULL,fill_iobuffer,NULL,NULL);
    ic->pb=avio;
    err = avformat_open_input(&ic, "nothing", NULL, NULL);
    ...//解码
}
```

将数据输出到内存

和从内存中读取数据类似，ffmpeg也可以将处理后的数据输出到内存。

回调函数如下示例，可以将输出到内存的数据写入到文件中。

```cpp
//写文件的回调函数
int write_buffer(void *opaque, uint8_t *buf, int buf_size){
    if(!feof(fp_write)){
        int true_size=fwrite(buf,1,buf_size,fp_write);
        return true_size;
    }else{
        return -1;
    }
}
```

主函数如下所示。

```cpp
FILE *fp_write;

int write_buffer(void *opaque, uint8_t *buf, int buf_size){
...
}

main(){
    ...
    fp_write=fopen("src01.h264","wb+"); //输出文件
    ...
    AVFormatContext* ofmt_ctx=NULL;
    avformat_alloc_output_context2(&ofmt_ctx, NULL, "h264", NULL);
    unsigned char* outbuffer=(unsigned char*)av_malloc(32768);

    AVIOContext *avio_out =avio_alloc_context(outbuffer, 32768,0,NULL,NULL,write_buffer,NULL);

    ofmt_ctx->pb=avio_out;
    ofmt_ctx->flags=AVFMT_FLAG_CUSTOM_IO;
    ...
}
```


文章标签： ffmpeg 内存 数据 avformat_open_input 回调

个人分类： FFMPEG

所属专栏： FFmpeg