# DoubleLi

更多

文章分类 (2) SilverLight(1) sql server(1) 整体结构大致如下:

gg: 517712484 wx: ldbgliet 博客园:首页:博问:闪存:新随笔:联系:订阅 🚾 :管理 :3867 随笔:2 文章:473 评论:1286万 阅读 2021年10月 ffmpeg 从内存中读取数据(或将数据输出到内存) 日一二三四五大 26 27 28 29 30 **1 2** 更新记录(2014.7.24): 3 4 5 6 7 8 9 10 11 12 13 14 15 16 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。 17 18 19 20 21 22 23 2.增加了将数据输出到内存的方法。 24 25 26 27 28 29 30 **31** 1 2 3 4 5 6 从内存中读取数据 公告 ffmpeg一般情况下支持打开一个本地文件,例如 "C:\test.avi" 昵称 · Doublel i 园龄: 11年9个月 或者是一个流媒体协议的URL,例如 "rtmp://222.31.64.208/vod/test.flv" 粉丝: 2080 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。 关注: 29 +加关注 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。 搜索 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 找找看 谷歌搜索 AVFormatContext \*ic = NULL; 常用链接 2. ic = avformat\_alloc\_context(); 我的随笔 我的评论 我的参与 最新评论 unsigned char \* iobuffer=(unsigned char \*)av\_malloc(32768); 我的标签 AVIOContext \*avio =avio\_alloc\_context(iobuffer, 32768,0,NULL,fill\_iobuffer,NULL,NULL); 2. **随笔分类** (5164) 3. ic->pb=avio: android(2) 4. err = avformat\_open\_input(&ic, "nothing", NULL, NULL); ASP.NET(30) ASP.NET MVC(11) Boost(118) 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 c#(10) 作为AVIOContext的缓存。 C++/C(778) c++11(15) cmake/autotool(66) com/ATL/Activex(75) 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。 Css(16) CxImage(12) darwin stream server(3) DataBase(32) 1. //读取数据的回调函数-----DirectX(16) //AVIOContext使用的回调函数! Extjs(13) //注意:返回值是读取的字节数 3. 更多 4. //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 5. //回调函数,功能就是:把buf\_size字节数据送入buf即可 随笔档案 (3864) 6. //第一个参数(void \*opaque)一般情况下可以不用 2021年10月(33) 7. int fill\_iobuffer(void \* opaque,uint8\_t \*buf, int bufsize){ 2021年9月(4) if(!feof(fp\_open)){ 2021年8月(10) 8. 2021年7月(43) 9. int true\_size=fread(buf,1,buf\_size,fp\_open); 2021年6月(1) 10. return true\_size; 2021年5月(29) 11. }else{ 2021年4月(15) 12. return -1: 2021年3月(13) 13. 2021年2月(96) 14. } 2021年1月(47) 2020年12月(2) 2020年11月(27) 2020年10月(44) 2020年9月(14) 2020年8月(4)

```
1.
      FILE *fp_open;
  2.
  3.
      int fill_iobuffer(void *opaque, uint8_t *buf, int buf_size){
 4.
  5.
  6.
  7.
       int main(){
  8.
  9.
 10.
 11.
 12.
 13.
 14.
 15.
 16.
 17. }
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。
      //写文件的回调函数
  1.
  2.
      int write_buffer(void *opaque, uint8_t *buf, int buf_size){
  3.
  4.
  5.
```

6.

7.

8.

9. }

主函数如下所示。

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

fp open=fopen("test.h264","rb+");

unsigned char \* iobuffer=(unsigned char \*)av\_malloc(32768);

err = avformat\_open\_input(&ic, "nothing", NULL, NULL);

int true\_size=fwrite(buf,1,buf\_size,fp\_write);

int write\_buffer(void \*opaque, uint8\_t \*buf, int buf\_size){

fp\_write=fopen("src01.h264","wb+"); //输出文件

avformat\_alloc\_output\_context2(&ofmt\_ctx, NULL, "h264", NULL);

AVIOContext \*avio\_out =avio\_alloc\_context(outbuffer, 32768,0,NULL,NULL,write\_buffer,NULL);

unsigned char\* outbuffer=(unsigned char\*)av\_malloc(32768);

AVFormatContext\* ofmt\_ctx=NULL;

ofmt\_ctx->pb=avio\_out;

from:http://blog.csdn.net/leixiaohua1020/article/details/12980423

AVIOContext \*avio =avio\_alloc\_context(iobuffer, 32768,0,NULL,fill\_iobuffer,NULL,NULL);

AVFormatContext \*ic = NULL;

ic->pb=avio;

if(!feof(fp\_write)){

return -1;

FILE \*fp\_write;

main(){

}else{

return true\_size;

...//解码

ic = avformat\_alloc\_context();

## 5. RTSP协议详解(8) 推荐排行榜

参考博客

Sloan

linux驱动

回忆未来-向东

大坡3D软件开发

Dean Chen的专栏

音视频FFmpeg等

opencv教程 个人开发历程知识库

关注DirectX

ffmpeg参考

更多

阅读排行榜

240898)

) (102153)

细)(94264)

评论排行榜

37)

19)

webrtc参考一

chenyujing1234

morewindows

雷雪骅(leixiaohua1020)的专栏

1. Nginx之location 匹配规则详解(

2. cmake使用方法详解(178488) 3. MinGW安装和使用(103448) 4. RTMP、RTSP、HTTP视频协议

详解(附:直播流地址、播放软件

5. C语言字符串操作总结大全(超详

1. 非IE内核浏览器支持activex插件(

2. Nginx之location 匹配规则详解(

3. Javascript中定义类(15)

4. C++中的头文件和源文件(9)

1. C++中的头文件和源文件(25)

Nginx模块开发与原理剖析

```
2. Nginx之location 匹配规则详解(
22)
3. Javascript中定义类(12)
4. JavaScript中typeof知多少?(11
5. MinGW安装和使用(9)
```

#### 最新评论 1. Re:windows下搭建nginx-rtmp 服务器

```
configuration-nginx.bat执行报错
啊 'auto' 不是内部或外部命令, 也
```

```
不是可运行的程序 或批处理文件。
'--conf-path' 不是内部或外部命
令,也不是可运行的程序或批..
```

```
--猫爷
2. Re:深入理解linux系统下proc文
件系统内容
```

怎么联系作者

```
3. Re:go mod模式下引用本地包/
```

```
视频流解决方案
目前市面上已经有很成熟且商用
```

内嵌播放多路RTSP的实时..

VLC的ActiveX控件可实现在网页中

5. Re:如何使用UDP进行跨网段广播

主机A: 192.168.3.100 子网掩码 255.255.0.0 ( 手动临时修改 ) 主机

```
Chrome播放海康威视大华的H.264
或H.265的RTSP视频流解决方案
了,就是猿大师中间件,底层调用
```

```
go mod用法,不错
```

```
模块(module)的方法
```

```
--立志做一个好的程序员
4. Re:谷歌浏览器Chrome播放rtsp
```

--喵大侠

```
分类: ffmpeg、ffplay
```

ofmt\_ctx->flags=AVFMT\_FLAG\_CUSTOM\_IO;

B: 192.168.120.100 子网掩码 255.255.255.0 主机A 广播 192.168.255...

--zzhilling



+加关注

DoubleLi 关注 - 29 粉丝 - 2080









0

0 印反对

刷新评论 刷新页面 返回顶部

€推荐

« 上一篇: FFMPEG基于内存的转码实例——输入输出视频均在内存 » 下一篇: 安装PostgreSQL数据库(Linux篇)

posted on 2017-08-11 15:59 DoubleLi 阅读(606) 评论(0) 编辑 收藏 举报

── 登录后才能查看或发表评论,立即登录或者
查证 博客园首页



#### 编辑推荐:

- ·聊聊我在微软外服的工作经历及一些个人见解
- · 死磕 NIO Reactor 模式就一定意味着高性能吗?
- ·消息队列那么多,为什么建议深入了解下RabbitMQ?
- · 技术管理进阶——管人还是管事?
- · 以终为始:如何让你的开发符合预期

### 最新新闻:

- ·何小鹏: 争取2024年实现飞行汽车量产 价格100万以内(2021-10-2423:35)
- ·供应链危机提振美国在线二手市场 全年销售额预计超650亿美元 (2021-10-2422:00)
- · CityTree: 一款利用苔藓和机器学习来捕捉空气污染的设备(2021-10-24 20:53)
- · 1024程序员节各家怎么过: 送霸王洗发水、集体穿格子衫、盲人按摩 ( 2021-10-24 20:00 )
- ·新卫星图展示泰国季风洪水所带来的巨大影响(2021-10-24 19:00)
- » 更多新闻...

Powered by: 博客园

Copyright © 2021 DoubleLi Powered by .NET 6 on Kubernetes