

CSDN

首页 博客 学院 下载 GitChat TinyMind 论坛 问答 商城 ...

搜博主文章

写博客

发Ch...

雷霄骅 (leixiaohua1020)的专栏

一个广院工... 音视频技术笔记

目录

收藏

评论

微信

微博

QQ

HELLO WORLD! KEEP CODING!

RSS订阅

# 最简单的基于FFMPEG+SDL的音频播放器：拆分-解码器和播放器

2015年07月07日 09:31:10

阅读量: 12977

- 最简单的FFmpeg的音频播放器系列文章列表:
- 《最简单的基于FFMPEG+SDL的音频播放器》
  - 《最简单的基于FFMPEG+SDL的音频播放器 ver2 （采用SDL2.0）》
  - 《最简单的基于FFMPEG+SDL的音频播放器：拆分-解码器和播放器》

本文补充记录《最简单的基于FFMPEG+SDL的音频播放器》中的两个例子：FFmpeg音频解码器和SDL音频采样数据播放器。这两个部分是从音频播放器中拆分出来的两个例子。FFmpeg音频解码器实现了视频数据到PCM采样数据的解码，而SDL音频采样数据播放器实现了PCM数据到音频设备的播放。简而言之，原先的FFmpeg+SDL音频播放器实现了：

音频数据->PCM->音频设备

FFmpeg音频解码器实现了：

音频数据->PCM

SDL音频采样数据播放器实现了：

PCM->音频设备

## FFmpeg音频解码器

### 源代码

```
[cpp]
1.  /**
2.   * 最简单的基于FFmpeg的音频解码器
3.   *  Simplest FFmpeg Audio Decoder
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   *  leixiaohua1020@126.com
7.   *  中国传媒大学/数字电视技术
8.   *  Communication University of China / Digital TV Technology
9.   *  http://blog.csdn.net/leixiaohua1020
10.  *
11.  * 本程序可以将音频码流（mp3，AAC等）解码为PCM采样数据。
12.  * 是最简单的FFmpeg音频解码方面的教程。
13.  * 通过学习本例子可以了解FFmpeg的解码流程。
14.  *
15.  * This software decode audio streams (AAC,MP3 ...) to PCM data.
16.  * Suitable for beginner of FFmpeg.
17.  *
18.  */
19. #include <stdio.h>
20. #include <stdlib.h>
21. #include <string.h>
22.
23. #define __STDC_CONSTANT_MACROS
24.
25. #ifdef _WIN32
26. //Windows
```

```
29. #include "libavcodec/avcodec.h"
30. #include "libavformat/avformat.h"
31. ... lude "libswresample/swresample.h"
32.
33. 7 e
34. //Linux...
35. ef __cplusplus
36. rn "C"
37. 目录
38. if
39. lude <libavcodec/avcodec.h>
40. 收藏 lude <libavformat/avformat.h>
41. ... lude <libswresample/swresample.h>
42. ef __cplusplus
43. 评论
44. #ifdef
45. if
46. 微信
47. ine MAX_AUDIO_FRAME_SIZE 192000 // 1 second of 48khz 32bit audio
48.
49. main(int argc, char* argv[])
50. 微博
51. AVFormatContext *pFormatCtx;
52. int i, audioStream;
53. QQ AVCodecContext *pCodecCtx;
54. AVCodec *pCodec;
55. AVPacket *packet;
56. uint8_t *out_buffer;
57. AVFrame *pFrame;
58. int ret;
59. uint32_t len = 0;
60. int got_picture;
61. int index = 0;
62. int64_t in_channel_layout;
63. struct SwrContext *au_convert_ctx;
64.
65. FILE *pFile=fopen("output.pcm", "wb");
66. char url[]="skycity1.mp3";
67.
68. av_register_all();
69. avformat_network_init();
70. pFormatCtx = avformat_alloc_context();
71. //Open
72. if(avformat_open_input(&pFormatCtx,url,NULL,NULL)!=0){
73. printf("Couldn't open input stream.\n");
74. return -1;
75. }
76. // Retrieve stream information
77. if(avformat_find_stream_info(pFormatCtx,NULL)<0){
78. printf("Couldn't find stream information.\n");
79. return -1;
80. }
81. // Dump valid information onto standard error
82. av_dump_format(pFormatCtx, 0, url, false);
83.
84. // Find the first audio stream
85. audioStream=-1;
86. for(i=0; i < pFormatCtx->nb_streams; i++)
87. if(pFormatCtx->streams[i]->codec->codec_type==AVMEDIA_TYPE_AUDIO){
88. audioStream=i;
89. break;
90. }
91.
92. if(audioStream==-1){
93. printf("Didn't find a audio stream.\n");
94. return -1;
95. }
96.
97. // Get a pointer to the codec context for the audio stream
98. pCodecCtx=pFormatCtx->streams[audioStream]->codec;
99.
100. // Find the decoder for the audio stream
101. pCodec=avcodec_find_decoder(pCodecCtx->codec_id);
102. if(pCodec==NULL){
103. printf("Codec not found.\n");
104. return -1;
105. }
106.
107. // Open codec
108. if(avcodec_open2(pCodecCtx, pCodec,NULL)<0){
109. printf("Could not open codec.\n");
110. return -1;
111. }
112.
113. packet=(AVPacket *)av_malloc(sizeof(AVPacket));
114.
115.
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

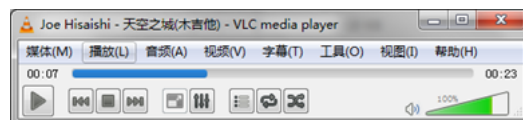
登录

注册

```
116. //Out Audio Param
117. uint64_t out_channel_layout=AV_CH_LAYOUT_STEREO;
118. //nb_samples: AAC-1024 MP3-1152
119. int out_nb_samples=pCodecCtx->frame_size;
120. AVSampleFormat out_sample_fmt=AV_SAMPLE_FMT_S16;
121. int out_sample_rate=44100;
122. int out_channels=av_get_channel_layout_nb_channels(out_channel_layout);
123. //Out Buffer Size
124. int out_buffer_size=av_samples_get_buffer_size(NULL,out_channels ,out_nb_samples,out_sample_fmt, 1);
125.
126. out_buffer=(uint8_t *)av_malloc(MAX_AUDIO_FRAME_SIZE*2);
127. pFrame=av_frame_alloc();
128.
129. //FIX:Some Codec's Context Information is missing
130. in_channel_layout=av_get_default_channel_layout(pCodecCtx->channels);
131. //Swr
132. au_convert_ctx = swr_alloc();
133. au_convert_ctx=swr_alloc_set_opts(au_convert_ctx,out_channel_layout, out_sample_fmt, out_sample_rate,
134. in_channel_layout,pCodecCtx->sample_fmt , pCodecCtx->sample_rate,0, NULL);
135. swr_init(au_convert_ctx);
136.
137. while(av_read_frame(pFormatCtx, packet)>=0){
138.     if(packet->stream_index==audioStream){
139.
140.         ret = avcodec_decode_audio4( pCodecCtx, pFrame,&got_picture, packet);
141.         if ( ret < 0 ) {
142.             printf("Error in decoding audio frame.\n");
143.             return -1;
144.         }
145.         if ( got_picture > 0 ){
146.             swr_convert(au_convert_ctx,&out_buffer, MAX_AUDIO_FRAME_SIZE,(const uint8_t **)pFrame->data , pFrame->nb_sam
147. les);
148.
149.             printf("index:%5d\t pts:%lld\t packet size:%d\n",index,packet->pts,packet->size);
150.             //Write PCM
151.             fwrite(out_buffer, 1, out_buffer_size, pFile);
152.             index++;
153.         }
154.         av_free_packet(packet);
155.     }
156.
157.     swr_free(&au_convert_ctx);
158.
159.     fclose(pFile);
160.
161.     av_free(out_buffer);
162.     // Close the codec
163.     avcodec_close(pCodecCtx);
164.     // Close the video file
165.     avformat_close_input(&pFormatCtx);
166.
167.     return 0;
168. }
```

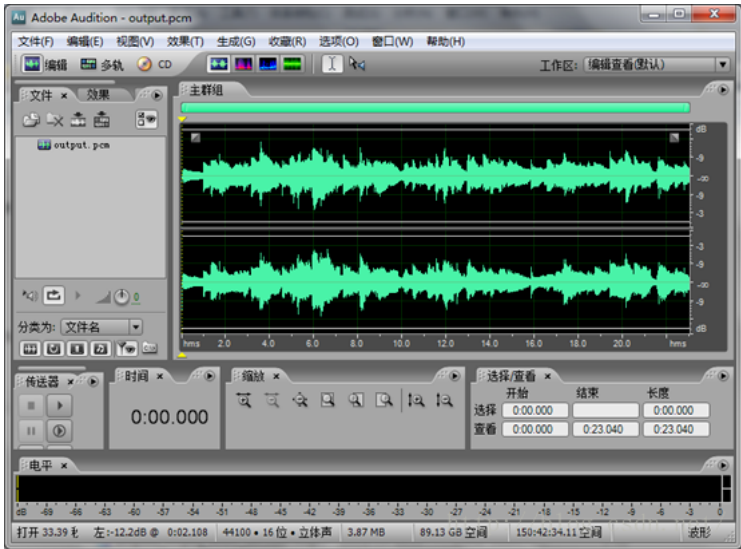
## 运行结果

程序运行后，会解码下面的音频文件。



解码后的PCM采样数据被保存成了一个文件。使用Adobe Audition设置采样率等信息后可以查看PCM的内容。

- 7
- 目录
- 收藏
- 评论
- 微信
- 微博
- QQ



SDL 音频采样数据播放器

源代码

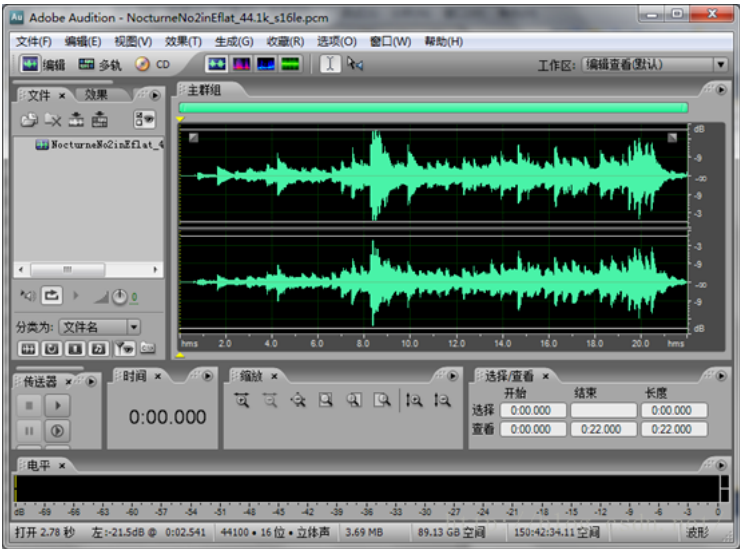
```
[cpp]
1.  /**
2.   * 最简单的SDL2播放音频的例子（SDL2播放PCM）
3.   * Simplest Audio Play SDL2 (SDL2 play PCM)
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   * leixiaohua1020@126.com
7.   * 中国传媒大学/数字电视技术
8.   * Communication University of China / Digital TV Technology
9.   * http://blog.csdn.net/leixiaohua1020
10.  *
11.  * 本程序使用SDL2播放PCM音频采样数据。SDL实际上是对底层绘图
12.  * API（Direct3D, OpenGL）的封装，使用起来明显简单于直接调用底层
13.  * API。
14.  *
15.  * 函数调用步骤如下：
16.  *
17.  * [初始化]
18.  * SDL_Init(): 初始化SDL。
19.  * SDL_OpenAudio(): 根据参数（存储于SDL_AudioSpec）打开音频设备。
20.  * SDL_PauseAudio(): 播放音频数据。
21.  *
22.  * [循环播放数据]
23.  * SDL_Delay(): 延时等待播放完成。
24.  *
25.  * This software plays PCM raw audio data using SDL2.
26.  * SDL is a wrapper of low-level API (DirectSound).
27.  * Use SDL is much easier than directly call these low-level API.
28.  *
29.  * The process is shown as follows:
30.  *
31.  * [Init]
32.  * SDL_Init(): Init SDL.
33.  * SDL_OpenAudio(): Opens the audio device with the desired
34.  * parameters (In SDL_AudioSpec).
35.  * SDL_PauseAudio(): Play Audio.
36.  *
37.  * [Loop to play data]
38.  * SDL_Delay(): Wait for completion of playback.
39.  */
40.
41. #include <stdio.h>
42. #include <tchar.h>
43.
44. extern "C"
45. {
46.     #include "sdl/SDL.h"
47. };
48.
49. //Buffer:
50. //|-----|-----|
51. //chunk-----pos---len----|
52. static Uint8 *audio_chunk;
53. static Uint8 *audio_pos;
54. static Uint8 *audio_pos;
```

```
55.
56. /* Audio Callback
57.  * The audio function callback takes the following parameters:
58.  * tream: A pointer to the audio buffer to be filled
59.  * en: The length (in bytes) of the audio buffer
60.
61.
62.     fill_audio(void *udata, Uint8 *stream, int len){
63. //SDL 2.0
64.     SDL_memset(stream, 0, len);
65.     if(audio_len==0) /* Only play if we have data left */
66.         return;
67.     len=(len>audio_len?audio_len:len); /* Mix as much data as possible */
68.
69.     SDL_MixAudio(stream, audio_pos, len, SDL_MIX_MAXVOLUME);
70.     audio_pos += len;
71.     audio_len -= len;
72.
73.
74.     main(int argc, char* argv[])
75.
76.     //Init
77.     if(SDL_Init(SDL_INIT_AUDIO | SDL_INIT_TIMER)) {
78.         printf("Could not initialize SDL - %s\n", SDL_GetError());
79.         return -1;
80.     }
81.     //SDL_AudioSpec
82.     SDL_AudioSpec wanted_spec;
83.     wanted_spec.freq = 44100;
84.     wanted_spec.format = AUDIO_S16SYS;
85.     wanted_spec.channels = 2;
86.     wanted_spec.silence = 0;
87.     wanted_spec.samples = 1024;
88.     wanted_spec.callback = fill_audio;
89.
90.     if (SDL_OpenAudio(&wanted_spec, NULL)<0){
91.         printf("can't open audio.\n");
92.         return -1;
93.     }
94.
95.     FILE *fp=fopen("NocturneNo2inEflat_44.1k_s16le.pcm", "rb+");
96.     if(fp==NULL){
97.         printf("cannot open this file\n");
98.         return -1;
99.     }
100.
101.     int pcm_buffer_size=4096;
102.     char *pcm_buffer=(char *)malloc(pcm_buffer_size);
103.     int data_count=0;
104.
105.     //Play
106.     SDL_PauseAudio(0);
107.
108.     while(1){
109.         if (fread(pcm_buffer, 1, pcm_buffer_size, fp) != pcm_buffer_size){
110.             // Loop
111.             fseek(fp, 0, SEEK_SET);
112.             fread(pcm_buffer, 1, pcm_buffer_size, fp);
113.             data_count=0;
114.         }
115.         printf("Now Playing %10d Bytes data.\n", data_count);
116.         data_count+=pcm_buffer_size;
117.         //Set audio buffer (PCM data)
118.         audio_chunk = (Uint8 *) pcm_buffer;
119.         //Audio buffer length
120.         audio_len = pcm_buffer_size;
121.         audio_pos = audio_chunk;
122.
123.         while(audio_len>0)//Wait until finish
124.             SDL_Delay(1);
125.     }
126.     free(pcm_buffer);
127.     SDL_Quit();
128.     return 0;
129. }
```

## 运行结果

程序运行后，会读取程序文件夹下的一个PCM采样数据文件，内容如下所示。

- 7
- 目录
- 收藏
- 评论
- 微信
- 微博



接下来QQ:1043966288:M数据输出到系统的音频设备上（音响、耳机）。

下载

Simplest FFmpeg Audio Player

SourceForge: <https://sourceforge.net/projects/simplestffmpegaudioplayer/>  
Github: [https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_audio\\_player](https://github.com/leixiaohua1020/simplest_ffmpeg_audio_player)  
开源中国: [http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_audio\\_player](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_audio_player)

CSDN下载地址: <http://download.csdn.net/detail/leixiaohua1020/8924329>

本程序实现了音频的解码和播放。是最简单的FFmpeg音频解码方面的教程。  
通过学习本例子可以了解FFmpeg的解码流程。  
项目包含3个工程：

- simplest\_ffmpeg\_audio\_player：基于FFmpeg+SDL的音频解码器
- simplest\_ffmpeg\_audio\_decoder：音频解码器。使用了libavcodec和libavformat。
- simplest\_audio\_play\_sdl2：使用SDL2播放PCM采样数据的例子。

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/46890259>

文章标签： FFmpeg 音频 SDL PCM 解码  
个人分类： FFMPEG  
所属专栏： FFmpeg



【CSDN学院】零基础在线学Python，一个视频，从入门到精通，学完就能上手！


想对作者说点什么？ 我来说一句

N3verL4nd 2016-08-31 14:22:30 #5楼

[查看回复\(2\)](#)


wanted\_spec\_sample\_rate = 0, //设置静音的值 wanted\_spec\_samples = 1024; //音频缓冲区中的采样个数（要求必须是2的n次方）这两个参数是做什么的啊？  
加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册

 N3verL4nd 2016-08-27 09:46:54 #4楼

```
[cpp]
7 swr_convert(au_convert_ctx,&out_buffer, MAX_AUDIO_FRAME_SIZE,(const uint8_t **)pFrame->data , pFrame->nb_samples);
fwrite(out_buffer, 1, out_buffer_size, pFile);
```

M 目录 IDIO\_FRAME\_SIZE换为out\_buffer\_size?

 左收藏 2016-04-03 15:08:58 #3楼 [查看回复\(1\)](#)

雷 好! 我是河海大学的大四学生，之前有在网上搜到您的ffmpeg教学视频很是有帮助，很是感谢。可我在自己实践的时候遇到了一个问题还想请教老师

上搭建ffmpeg环境时输入#define \_\_STDC\_CONSTANT\_MACROS这条指令时，define后面的依旧是黑色，之后#include " libavcodec /

评论 h " 也显示打开不了源文件，系统是win10，还请老师多多指教。

 Zi 微信 2015-12-28 15:57:21 #2楼 [查看回复\(1\)](#)

查看 12 条热评

微博

## FFmpeg MP3解码器的研究与优化

FFmpeg是如今多媒体领域用途非常广泛的一个开源免费跨平台的视频和音频流方案，MP3格式是一种常用的音频格式。但是ffmpeg对MP3v2.5规格的MP3音频文件解码支持不是很好，在播放时会有明显失...

 qvfovkcf 2014-08-15 22:41:52 阅读数：482

## [总结]FFMPEG视音频编解码零基础学习方法

在CSDN上的这一段日子，接触到了很多同行业的人，尤其是使用FFMPEG进行视音频编解码的人，有的已经是有多年经验的“大神”，有的是刚开始学习的初学者。在和大家探讨的过程中，我忽然发现了一个问题：在“...

 leixiaohua1020 2013-11-16 00:04:05 阅读数：505825

## 耳聋、耳鸣、耳朵嗡嗡响，试试这个小妙招，一招解决!

君旺生物 · 顶新



## ffmpeg MP3解码

2017年03月21日 7KB [下载](#)



## 最简单的基于FFMPEG+SDL的音频播放器

FFMPEG工程浩大，可以参考的书籍又不是很多，因此很多刚学习FFMPEG的人常常感觉到无从下手。在此我把自己做项目过程中实现的一个非常简单的音频播放器（大约100行代码）源代码传上来，以作备忘，同...

 leixiaohua1020 2013-08-29 17:11:09 阅读数：63608

## 我的开源视音频项目汇总

本文汇总一下自己视音频编解码学习方面的开源项目。这些开源项目大体上可以分成专业领域程序，FFmpeg示例程序，FFmpeg移植程序，多媒体项目示例程序，视音频编解码原理学习工程几个类别。...

 leixiaohua1020 2015-01-13 01:17:34 阅读数：72196

## FFmpeg 从零开始开发简单的音视频播放器（二）

简介和准备工作一、FFmpeg FFmpeg是一套开源的、业界领先的多媒体框架，可以用来记录、转换数字音频、视频，并能将其转化为流。很多的视音频方案都离不开FFmpeg，比如著名的VLC...

 HenryChow\_2015 2018-03-20 19:14:07 阅读数：73

## 老中医说：男人多吃它，性生活时间延长5倍

国瑞兴邦 · 顶新



## FFmpeg - 音频解码

一个语音项目，音频经过格式，简单记录下遇到的问题及解决办法

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

 mqdChalal 2018-09-07 14:43:09 阅读数：1375

登录

注册

FFmpeg 音频解码浅析

转自：  7 og.csdn.net/xiaozhu1100/article/details/16929181 结合各种资料和自己的理解，估计有些浅显。 FFMPEG...

 wei... 2017-03-10 15:12:38 阅读数：1473

ffmpeg 音频解码方法（附代码）+ffmpeg 音频解码播放速度快的问题（随手笔记，以供查阅）

最近在 收藏 取名为变速不变调播放器的时候，解码 音频遇到了些问题（ffmpeg 音频解码播放速度快的问题），网络上的方法对绝大多数的音视频文件有效，但是对 音频会有问题，比如某些ADPCM编码的WAV...

 xjb... 2018-01-05 14:14:11 阅读数：314

ffmpeg 码音频保存为PCM

微信 [TOC] 简单保存PCM的demo,注意保存那里对双声道的处理，因为pcm对双声道是交替存储的代码/\* 一个简单保存PCM的demo,注意保存那里对双声道的处理 pcm对双声道是交替存储...

 dar... ight 2016-11-08 23:40:54 阅读数：3247

FFmpeg 音频编解码处理

新版的ffmpeg对 音频编码处理已经有了很大的变化，记录在此，做个备忘。 早期ffmpeg编码 音频，输入数据一般都是S16格式，解码输出一般也是S16，也就是说PCM数据是存储在连续的buffer中...

 chinabinlang 2015-08-13 17:50:36 阅读数：6305

FFmpeg 3.2.2 音频解码

网上找了好多例子，因为是用的是最新的3.2.2（2017年1月23日 04:22:33）版本的FFmpeg，所以找的东西基本上都不可能拿过来就能跑起来，研究了许多，参考了诸多大神的博客，然后一句句代码专...

 qq\_33601179 2017-01-23 04:41:10 阅读数：272

ffmpeg关于音频的总结(一)

ffmpeg关于 音频的总结(一) 一：关于 音频分片的问题 enum AVSampleFormat { AV\_SAMPLE\_FMT\_NONE = -1, AV\_SAMPLE\_FMT\_U8, ...

 zhuweigangzwwg 2016-05-25 16:01:03 阅读数：11807


外语好课！特价1元你想学的都在这！抢课>

1元学外语！点击抢课>7天说流利小语种！



ffmpeg中音频格式与解码器的对应关系

Fmpeg中对于相应的 音频格式，只要配置了正确的protocol、demuxer、decoder、parser、filters就可以正确解码了。 其中：(1)、Protocol： Ffmpe...

 liufang200706 2017-09-30 15:43:05 阅读数：363

FFMPEG 解码音频

目的通过FFMPEG解码 音频的码流，得到PCM的 音频采样数据并用AudioTraker播放步骤1.注册所有组件av\_register\_all(); 2.拿到封装格式上下文AVFormatCont ext...

 itheimach 2017-01-03 21:51:13 阅读数：937

音视频开发——ffmpeg解码（四）

音视频最强大的开源库非ffmpeg莫属，网上对ffmpeg总结的最好的是雷神的博客（http://blog.csdn.net/leixiaohua1020/article/details/158119...

 a411358606 2016-08-10 16:18:40 阅读数：1448

新版FFmpeg 音频编解码处理

fmpeg版本更新近一年跟打了鸡血一样，刷刷刷的往上升（貌似现在很多软件版本升级都相当快，而且都是大版本升级），应用软件也就算了，升就升吧，最多新版可能运行速度稍慢些，可API的版本升级太快，对开发来...

 wangxing1231 2015-06-04 16:39:32 阅读数：714

最简单的基于FFMPEG的音频编码器（PCM编码为AAC）  
加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



本文介绍一个最简单的基于FFMPEG的音频编码器。该编码器实现了PCM音频采样数据编码为AAC的压缩编码数据。编码器代码十分简单，但是每一行代码都很重要，适合好好研究一下。 本程序使用最新版的类库（编...

leix 1020 2014-05-11 00:14:28 阅读数：67098

7

深圳二手房成交均价一览!是涨是跌一目了然

深圳二手房成交均价一览!是涨是跌一目了然

百度广告

收藏



## FFmpeg 浅尝辄止（四）——音频的解码和编码

评论

音频和解码是一样的，在文件中寻找音频流，然后解压出来，得到音频帧的数据，同样也可以按照设定的编码格式进行压缩，我这里把音频的解码和编码做成了...

微信

yan 521 2012-06-28 16:45:46 阅读数：31020

微博

## ffmpeg 解码音频并保存PCM的简单demo

2016年 3月 19.33MB

下载

RAR

## FFmpeg音视频编解码示例

FFmpeg示例程序集：https://www3.nd.edu/~sjiang1/docio/ffmpeg\_doc/globals\_eval.htmlFFmpeg音视频编解码示例：https://www...

boonya 2018-01-11 09:42:32 阅读数：273

## FFmpeg音频解码实例

2012年06月28日 4.44MB

下载

ZIP

## FFmpeg+SDL2.0 音频解码播放（部分格式杂音处理）

还是参照雷大神的例子改写得，但是改写过程中碰到几个问题，纠结的比较久的问题是解码MP4，avi文件，播放出来的音频不仅很多杂音而且断断续续。因为是刚接触这块，没什么音频方面得知识，出了问题也找...

u012853439 2016-11-18 16:41:02 阅读数：1246

## 外语好课！特价1元你想学的都在这！抢课>

1元学外语！点击抢课>7天说流利小语种！



## FFMPEG1.2 音频解码的过程

FFMPEG输出WAV的audio：ffmpeg -i ~/test\_22m.audio.flv -acodec pcm\_s16le -ar 8000 -ac 1 -f wav -y ~/test\_2...

xiaojun111111 2016-10-14 14:01:04 阅读数：960

## ffmpeg的音频解码部分

总而言之是这样的 一个装音频packet的队列 这个队列是2个指针做的。然后一个回调函数 主线程放包，回调取包，解码在回调里面做。 关键代码： 回调： void...

hustbin 2013-11-25 21:22:07 阅读数：571

## 最简单的基于FFMPEG+SDL的音频播放器 ver2 （采用SDL2.0）

简介之前做过一个简单的音频播放器：《最简单的基于FFMPEG+SDL的音频播放器》，采用的是SDL1.2。前两天刚把原先做的《最简单的基于FFMPEG+SDL的视频播放器》更新采用了SDL2.0，于是...

leixiaohua1020 2014-09-01 14:26:55 阅读数：27969

## 最简单的基于FFMPEG+SDL的音视频播放器

一、概述 在《最简单的基于FFMPEG+SDL的音频播放器》记录一中，我们实现了音频的播放。更早前，我们在《最简单的基于FFMPEG+SDL的视频播放器》记录一和二中，实现了视频的播放...

GAMEloft9 2015-06-21 20:43:49 阅读数：1336

困扰我有一段日子的这个东东今晚终于搞好了。先声明一下，我用的ffmpeg的版本是ffmpeg-20140227-git-b5005de 版本升级以后，跟以前主要的区别是ffmpeg对音频...

htbl

7

2014-08-15 14:09:32    阅读数：666

现在：人买银行大额存单吗

大额存单

百度广告

收藏



最简单：基于FFMPEG+SDL的视频播放器 ver2 （采用SDL2.0）

评论

之前做：=FFMPEG+SDL的简单播放器：《100行代码实现最简单的基于FFMPEG+SDL的视频播放器》。最近有不少人反映SDL已经升级到2.0版本了，甚至官网上都只有SDL2.0的文...

微信

leixiaohua1020

2014-08-31 01:20:14    阅读数：78780

微博

最简单：基于ffmpeg+SDL的视频播放器

环境：12.04 Qt 5.5.0 ffmpeg 3.0.1 SDL 2.1流程图该播放器解码的流程用图的方式可以表示称如下形式： SDL显示YUV图像的流程图： simple...

QQ

MLEem

2016-04-22 18:14:02    阅读数：631

个人资料



雷霄骅

关注

原创	粉丝	喜欢	评论
373	1万+	453	7315

等级： 博客 9    访问： 1039万+

积分： 6万+    排名： 50

勋章：

摩托罗拉新上市



关于我

姓名：雷霄骅  
网名：leixiaohua1020  
本科：中国传媒大学-广播电视工程  
硕士：中国传媒大学-数字电视技术  
博士：中国传媒大学-数字视频技术  
Email：leixiaohua1020@126.com  
QQ：494085803

[注1：QQ消息较多，难以一一回复，见谅]  
[注2：CSDN私信功能使用很少，有问题可以直接在博客评论处留言]

- 7
- 目录
- 收藏
- 评论
- 微信
- 微博
- QQ

2014年度 - CSDN博客之星  
2015年度 - 微软MVP  
2015年度 - CSDN博客之星  
简介：  
主要从事与广播电视有关的视音频技术的研究。包括视音频质量评价，视音频编解码，流媒体，媒资检索等。

最新文章

- [投稿] Speex回声消除原理深度解析
- [投稿]房间声学原理与Schroeder混响算法实现
- [投稿]一个频域语音降噪算法实现及改进方法
- 最简单的基于FFmpeg的AVfilter的例子-纯净版
- 视音频数据处理入门：UDP-RTP协议解析

博主专栏

-  视频质量评价  
阅读量：433208 41 篇
-  开源多媒体项目源代码分析  
阅读量：1156747 91 篇
-  FFmpeg  
阅读量：4841230 135 篇

归档

2016年4月	1篇
2016年3月	2篇
2016年2月	1篇
2016年1月	7篇
2015年12月	1篇

展开

最新评论

- 视频编码器评测系统：VideoCo...  
tqs\_1220：天妒英才
- FFmpeg获取DirectSho...  
a137748099：[reply]liangqingzhi[reply] 大佬，有的电脑使用regsvr32注册不成...
- [总结]FFMPEG视音频编解码零...  
qq\_17276615：每次视频编解码都会搜到你的文章，感谢您给我们这些菜鸟一些指引。谢谢！
- [投稿] Speex回声消除原理深...  
jwq1220：厉害！
- 最简单的基于librtmp的示例： ...  
qq\_32245927：[reply]zzqgtt[reply] 你好，你的QQ多少