一个使用FFmpeg库读取3gp视频的例子-Android中使用FFmpeg媒体库(三)

转载 2011年11月02日 21:37:30

标签：android (http://so.csdn.net/so/search/s.do?q=android&t=blog) / jni (http://so.csdn.net/so/search/s.do?q=jni&t=blog) / button (http://so.csdn.net/so/search/s.do?q=button&t=blog) / null (http://so.csdn.net/so/search/s.do?q=null&t=blog) / video (http://so.csdn.net/so/search/s.do?q=video&t=blog) / codec (http://so.csdn.net/so/search/s.do?q=codec&t=blog)

7068

本文中的代码来自于https://github.com/churnlabs/android-ffmpeg-sample，更多的可以参考这个项目代码。我会在代码中加一些自己的注释。感谢作者churnlabs给我们提供这么好的例子以供我们学习。
在Android的一些系统层应用开发大多数是采用jni的方式调用，另外对于一些比较吃CPU或者处理逻辑比较复杂的程序，也可以考虑使用jni方式来封装。可以提高程序的执行效率。

本文涉及到以下几个方面：
1 将3gp文件push到模拟机器的sdcard中
2 写jni代码，内部调用ffmpeg库的方法，编译jni库
3 loadLibrary生成的库，然后撰写相应的java代码
4 执行程序，并查看最终运行结果。
最终程序的显示效果如下：

(http://doandroid.info/wp-content/uploads/2011/10/Screenshot3.png)

1 使用eclipse的DDMS工具，将vid.3pg push到sdcard中

1 使用eclipse的DDMS工具，将vid.3pg push到sdcard中

(http://doandroid.info/wp-content/uploads/2011/10/Screenshot-12.png)

2 撰写相应的jni文件

```
/*
 * Copyright 2011 - Churn Labs, LLC
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0 (http://www.apache.org/
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implie
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/*
 * This is mostly based off of the FFMPEG tutorial:
 * http://dranger.com/ffmpeg/ (http://dranger.com/ffmpeg/)
 * With a few updates to support Android output mechanisms and to update
 * places where the APIs have shifted.
 */

#include <jni.h>
#include <string.h>
#include <stdio.h>
#include <android/log.h>
#include <android/bitmap.h>

   //包含ffmpeg库头文件，这些文件都直接方案jni目录下
```

```
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libswscale/swscale.h>

#define  LOG_TAG    "FFMPEGSample"
#define  LOGI(...)  __android_log_print(ANDROID_LOG_INFO,l
#define  LOGE(...)  __android_log_print(ANDROID_LOG_ERROR,

   /* Cheat to keep things simple and just use some globals. */
   //全局对象
AVFormatContext *pFormatCtx;
AVCodecContext *pCodecCtx;

AVFrame *pFrame;
AVFrame *pFrameRGB;
int
videoStream;

   /*
    * Write a frame worth of video (in pFrame) into the Android bitmap
    * described by info using the raw pixel buffer.  It's a very inefficient
    * draw routine, but it's easy to read. Relies on the format of the
    * bitmap being 8bits per color component plus an 8bit alpha channel.
    */

   //定义的静态方法，将某帧AVFrame在Android的Bitmap中绘制
static
void  fill_bitmap(AndroidBitmapInfo*  info, void
*pixels, AVFrame *pFrame)
{
    uint8_t *frameLine;

    int
yy;
    for
(yy = 0; yy < info->height; yy++) {
        uint8_t*  line = (uint8_t*)pixels;
        frameLine = (uint8_t *)pFrame->data[0] + (yy * pFr
```

```
        int
xx;
        for
(xx = 0; xx < info->width; xx++) {
            int
out_offset = xx * 4;
            int
in_offset = xx * 3;

            line[out_offset] = frameLine[in_offset];
            line[out_offset+1] = frameLine[in_offset+1];
            line[out_offset+2] = frameLine[in_offset+2];
            line[out_offset+3] = 0;
        }
        pixels = (char*)pixels + info->stride;
    }
}

  //定义java回调函数，相当与 com.churnlabs中的ffmpegsample中的MainActivity类中的o
void
Java_com_churnlabs_ffmpegsample_MainActivity_openFile(JNIE
this)
{
    int
ret;
    int
err;
    int
i;
    AVCodec *pCodec;
    uint8_t *buffer;
    int
numBytes;
      //注册所有的函数
    av_register_all();
    LOGE("Registered formats");
      //打开sdcard中的vid.3gp文件

    err = av_open_input_file(&pFormatCtx, "file:/sdcard/vi
    LOGE("Called open file");
    if(err!=0) {
```

```
102         LOGE("Couldn't open file");
103         return;
104     }
105     LOGE("Opened file");
106 
107     if(av_find_stream_info(pFormatCtx)<0) {
108         LOGE("Unable to get stream info");
109         return;
110     }
111 
112     videoStream = -1;
113       //定义设置videoStream
114     for
(i=0; i<pFormatCtx->nb_streams; i++) {
115         if(pFormatCtx->streams[i]->codec->codec_type==CODE
116             videoStream = i;
117             break;
118         }
119     }
120     if(videoStream==-1) {
121         LOGE("Unable to find video stream");
122         return;
123     }
124 
125     LOGI("Video stream is [%d]", videoStream);
126 
127       //定义编码类型
128     pCodecCtx=pFormatCtx->streams[videoStream]->codec;
129       //获取解码器
130     pCodec=avcodec_find_decoder(pCodecCtx->codec_id);
131     if(pCodec==NULL) {
132         LOGE("Unsupported codec");
133         return;
134     }
135       //使用特定的解码器打开
136     if(avcodec_open(pCodecCtx, pCodec)<0) {
137         LOGE("Unable to open codec");
```

```
138             return;
139         }
140           //分配帧空间
141         pFrame=avcodec_alloc_frame();
142           //分配RGB帧空间
143         pFrameRGB=avcodec_alloc_frame();
144         LOGI("Video size is [%d x %d]", pCodecCtx->width, pCoc
145           //获取大小
146         numBytes=avpicture_get_size(PIX_FMT_RGB24, pCodecCtx->
147         分配空间
148         buffer=(uint8_t *)av_malloc(numBytes*sizeof(uint8_t));
149
150         avpicture_fill((AVPicture *)pFrameRGB, buffer, PIX_FMT
151                                 pCodecCtx->width, pCodecCtx->h
152     }
153       //定义java回调函数，相当与 com.churnlabs中的ffmpegsample中的MainActivity类中的dr
154     void
155     Java_com_churnlabs_ffmpegsample_MainActivity_drawFrame(JNI
        this, jstring bitmap)
156     {
157         AndroidBitmapInfo  info;
158         void*              pixels;
159         int
        ret;
160
161
162         int
        err;
163         int
        i;
164
165         int
        frameFinished = 0;
166         AVPacket packet;
167         static
        struct  SwsContext *img_convert_ctx;
168         int64_t seek_target;
169
170         if
        ((ret = AndroidBitmap_getInfo(env, bitmap, &info)) < 0) {
171             LOGE("AndroidBitmap_getInfo() failed ! error=%d",
172             return;
173
```

```
    }
    LOGE("Checked on the bitmap");

    if ((ret = AndroidBitmap_lockPixels(env, bitmap, &pixels)) <
        LOGE("AndroidBitmap_lockPixels() failed ! error=%d
    }
    LOGE("Grabbed the pixels");


    i = 0;
    while((i==0) && (av_read_frame(pFormatCtx, &packet)>=0
        if(packet.stream_index==videoStream) {
            avcodec_decode_video2(pCodecCtx, pFrame, &fram

            if(frameFinished) {
                LOGE("packet pts %llu", packet.pts);
                  // This is much different than the tutorial, sws_sca
                  // replaces img_convert, but it's not a complete dro
                  // This version keeps the image the same size but sw
                  // RGB24 format, which works perfect for PPM output.
                int target_width = 320;
                int target_height = 240;
                img_convert_ctx = sws_getContext(pCodecCtx
                       pCodecCtx->pix_fmt,
                       target_width, target_height, PIX_FM
                       NULL, NULL, NULL);
                if(img_convert_ctx == NULL) {
                    LOGE("could not initialize conversion
                    return;
                }
                sws_scale(img_convert_ctx, (const uint8_t* const*)pFrame->data, pFrame->linesize, 0, pCodecC

                  // save_frame(pFrameRGB, target_width, target_height,
                fill_bitmap(&info, pixels, pFrameRGB);
                i = 1;
```

```
                i = 1;

            }
        }
        av_free_packet(&packet);
    }

    AndroidBitmap_unlockPixels(env, bitmap);
}

  //内部调用函数，不对外，用来查找帧
int
seek_frame(int
tsms)
{
    int64_t frame;

    frame = av_rescale(tsms,pFormatCtx->streams[videoStrea
    frame/=1000;

    if(avformat_seek_file(pFormatCtx,videoStream,0,frame,f
        return
0;
    }

    avcodec_flush_buffers(pCodecCtx);

    return
1;
}
  //定义java回调函数，相当与 com.churnlabs中的ffmpegsample中的MainActivity类中的d
void
Java_com_churnlabs_ffmpegsample_MainActivity_drawFrameAt(J
this, jstring bitmap, jint secs)
{
    AndroidBitmapInfo  info;
    void*              pixels;
    int
ret;
```

```
    int
err;
    int
i;
    int
frameFinished = 0;
    AVPacket packet;
    static
struct  SwsContext *img_convert_ctx;
    int64_t seek_target;


    if
((ret = AndroidBitmap_getInfo(env, bitmap, &info)) < 0) {
        LOGE("AndroidBitmap_getInfo() failed ! error=%d",
        return;
    }
    LOGE("Checked on the bitmap");


    if
((ret = AndroidBitmap_lockPixels(env, bitmap, &pixels)) <
        LOGE("AndroidBitmap_lockPixels() failed ! error=%d
    }
    LOGE("Grabbed the pixels");



    seek_frame(secs * 1000);


    i = 0;
    while
((i== 0) && (av_read_frame(pFormatCtx, &packet)>=0)) {
        if(packet.stream_index==videoStream) {
            avcodec_decode_video2(pCodecCtx, pFrame, &fram


            if(frameFinished) {
                // This is much different than the tutorial, sws_sca
                // replaces img_convert, but it's not a complete dro
                // This version keeps the image the same size but sw
                // RGB24 format, which works perfect for PPM output.
               int
target_width = 320;
               int
```

内容举报

返回顶部

0

```
int
target_height = 240;
                img_convert_ctx = sws_getContext(pCodecCtx
                        pCodecCtx->pix_fmt,
                        target_width, target_height, PIX_FM
                        NULL, NULL, NULL);
                if(img_convert_ctx == NULL) {
                    LOGE("could not initialize conversion
                    return;
                }
                sws_scale(img_convert_ctx, (const
uint8_t* const*)pFrame->data, pFrame->linesize, 0, pCodecC


                   // save_frame(pFrameRGB, target_width, target_height,
                fill_bitmap(&info, pixels, pFrameRGB);

                i = 1;
            }
        }
        av_free_packet(&packet);
    }

    AndroidBitmap_unlockPixels(env, bitmap);
}
```

3 撰写相应的Android.mk文件

```
1  LOCAL_PATH := $(call my-dir)
2
3  include $(CLEAR_VARS)
4
5  LOCAL_MODULE    := ffmpegutils
6  LOCAL_SRC_FILES := native.c
7
8  LOCAL_C_INCLUDES := $(LOCAL_PATH)/include
9  LOCAL_LDLIBS := -L$(NDK_PLATFORMS_ROOT)/$(TARGET_PLATFORM)/
10
11 include $(BUILD_SHARED_LIBRARY)
```

这里需要注意一下文件的目录情况，我截图说明一下。

(http://doandroid.info/wp-content/uploads/2011/10/Screenshot-22.png)

在Android.mk中有意个LOCAL_C_INCLUDES :=$(LOCAL_PATH)/include指明了相应的FFmpeg的头文件路径。故在代码中包含

```
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libswscale/swscale.h>
```

就可以。

4 调用ndk-build，生成libffmpegutils.so文件，将这个文件拷贝到/root/develop/android-ndk-r6/platforms/android-8/arch-arm/usr/lib目录，使得我们在下面使用Android AVD2.2的时候，可以加载到这个so文件。

5 撰写相应的Eclipse项目代码，由于在native.c文件中指明了项目的工程名词以及类名词还有函数名词，故我们的项目为com.churnlabs.ffmpegsample下面的MainActivity.java文件

```
package
com.churnlabs.ffmpegsample;


import
android.app.Activity;
import
android.graphics.Bitmap;

import
android.os.Bundle;
import
```

```
import
android.view.View;

import
android.view.View.OnClickListener;

import
android.widget.Button;

import
android.widget.ImageView;


public
class  MainActivity extends
Activity {

    private
static  native  void  openFile();

    private
static  native  void  drawFrame(Bitmap bitmap);

    private
static  native  void  drawFrameAt(Bitmap bitmap, int
secs);


    private
Bitmap mBitmap;

    private
int  mSecs = 0;


    static
{
        System.loadLibrary("ffmpegutils");
    }

    /** Called when the activity is first created. */
    @Override
    public
void  onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
           //setContentView(new VideoView(this));
        setContentView(R.layout.main);

        mBitmap = Bitmap.createBitmap(320, 240, Bitmap.Con
        openFile();

        Button btn = (Button)findViewById(R.id.frame_adv);
        btn.setOnClickListener(new
OnClickListener() {
```

```
45              public
    void  onClick(View v) {
46
47                  drawFrame(mBitmap);
48
49                  ImageView i = (ImageView)findViewById(R.id.
50
51                  i.setImageBitmap(mBitmap);
52
53              }
54          });
55
56
57          Button btn_fwd = (Button)findViewById(R.id.frame_fw
58
59          btn_fwd.setOnClickListener(new
    OnClickListener() {
60              public
    void  onClick(View v) {
61
62                  mSecs += 5;
63
64                  drawFrameAt(mBitmap, mSecs);
65
66                  ImageView i = (ImageView)findViewById(R.id.
67
68                  i.setImageBitmap(mBitmap);
69
70              }
71          });
72
73
74
75          Button btn_back = (Button)findViewById(R.id.frame_b
76
77          btn_back.setOnClickListener(new
    OnClickListener() {
78              public
    void  onClick(View v) {
79
80                  mSecs -= 5;
81
82                  drawFrameAt(mBitmap, mSecs);
83
84                  ImageView i = (ImageView)findViewById(R.id.
85
86                  i.setImageBitmap(mBitmap);
87
88              }
89          });
90      }
91  }
```

6 编译运行即可，最终效果图

(http://doandroid.info/wp-

content/uploads/2011/10/Screenshot3.png)

7 项目代码下载：

https://github.com/churnlabs/android-ffmpeg-sample/zipball/master

参考：

1 https://github.com/churnlabs/android-ffmpeg-sample
2 http://www.360doc.com/content/10/1216/17/474846_78726683.shtml
3 https://github.com/prajnashi

内容举报

返回顶部

发表你的评论

(http://my.csdn.net/weixin_35068028)

tiandaozhang (/tiandaozhang) 2013-01-24 20:02 1楼

(/tiandaozhang)，生成JNI里面的静态库.a文件的ffmpeg是哪个版本的啊？我将0.8.1版本编译生成的静态库和头文件的放入到里面，提示错误啊！

回复

0

相关文章推荐

Delphi XE2 新增 System.Zip 单元，压缩和解压缩文件 (http://blog.csdn.net/sunylat/article/…

Delphi XE2 新增 System.Zip 单元, 可用一句话压缩整个文件夹了 单元内主要就是 TZipFile 类, 最方便使用的是它的类方法: TZipFile.ExtractZip...

sunylat (http://blog.csdn.net/sunylat) 2013年06月23日 09:47 3760

delphi压缩与解压_不需要特别的控件 (http://blog.csdn.net/hnbcjzj/article/details/7972607)

delphi压缩与解压，比较简单，也比较实用，对我来说已经足够，我只想让客户端下载一个压缩文件到本地后解压出来就行，看网上有用什么控件的，我的是delphi6没装那控件，在公司管的比较严，要装控件比较...

hnbcjzj (http://blog.csdn.net/hnbcjzj) 2012年09月12日 20:28 9117

内容举报

返回顶部

用编译好的FFmpeg库。 ...

Mirage520 (http://blog.csdn.net/Mirage520) 2012年07月23日 17:42 1323

在Android中通过jni方式使用编译好的FFmpeg库-Android中使用FFmpeg媒体库 (http://blog....

原文：http://doandroid.info/?p=471 在继上篇在32位的Ubuntu 11.04中为Android NDK r6编译FFmpeg最新版0.8.1后，本人来给大家...

toddmi (http://blog.csdn.net/toddmi) 2012年07月26日 12:50 848

在Android中通过jni方式使用编译好的FFmpeg库-Android中使用FFmpeg媒体库(二) (http://bl...

原文：http://doandroid.info/?p=471 在继上篇在32位的Ubuntu 11.04中为Android NDK r6编译FFmpeg最新版0.8.1后，本人来给大家...

yang_hui1986527 (http://blog.csdn.net/yang_hui1986527) 2011年10月13日 02:52 6370

android媒体库使用demo及相关工具 (http://download.csdn.net/download/...

(http://download.c 2015年10月26日 16:16 26.63MB 下载

FFPlay是一个使用了 ffmpeg 和 sdl 库的、一个简单的可移植的媒体播放器。 (http://blog.csd...

视频播放器播放一个互联网上的视频文件，需要经过以下几个步骤：解协议，解封装，解码视音频，视音频同步。如果播放本地文件则不需要解协议，为以下几个步骤：解封装，解码视音频，视音频同步。他们的过程如图所示。...

wishfly (http://blog.csdn.net/wishfly) 2017年07月17日 17:48 902

Android 媒体库图片，音频，视频，文件的查询 (http://blog.csdn.net/consumer11/article/de...

项目开发中我们要使用到本地SD卡中的媒体文件，例如图片，音频，视频，压缩文件等，ContentResolver 可以很方便的帮助我们查询所有信息。...

consumer11 (http://blog.csdn.net/consumer11) 2016年08月01日 09:53 1165

Android添加新的视频格式 - - 媒体库扫描 (http://blog.csdn.net/deng0zhaotai/article/detail...

需求：在mediaprovider数据库中添加.mov后缀格式的视频文件 可以使用工具MediaInfo_GUI_0.7.67_Windows.3243836749.exe 查看mov文件编码格式类型...

deng0zhaotai (http://blog.csdn.net/deng0zhaotai) 2015年06月03日 14:01 1057

Android搜索视媒体库视频 列表显示选择 (http://blog.csdn.net/cybbz/article/details/4004847...

public class MainActivity extends Activity {public static List data = null;// 视频信息集合Cursor cursor;Bu...

cybbz (http://blog.csdn.net/cybbz) 2014年10月13日 17:29 729

android系统媒体库开发之视频 (http://blog.csdn.net/zj695469296/article/details/50586817)

Android系统为我们建立起多媒体数据库之后，便把多媒体常用的信息，比如歌曲名、文件大小、播放时长、专辑、歌手等常用信息保存在了数据库里，那我们可以直接用多媒体库中的数据，完成这个需求。虽然我们需要...

zj695469296 (http://blog.csdn.net/zj695469296) 2016年01月26日 13:54 302

最简单的基于FFmpeg的移动端例子：Android 视频解码器-单个库版 (http://blog.csdn.net/lei...

本文记录另一个安卓平台下基于FFmpeg的视频解码器。与前一篇文章记录的解码器不同，本文记录的解码器不再使用libavcodec.so、libavformat.so等类库，而只使用了一个类库——lib...

leixiaohua1020 (http://blog.csdn.net/leixiaohua1020) 2015年07月25日 11:42 17350

最简单的基于FFmpeg的移动端例子：Android 视频解码器-单个库版 (http://blog.csdn.net/zh...

本文记录另一个安卓平台下基于FFmpeg的视频解码器。与前一篇文章记录的解码器不同，本文记录的解码器不再使用libavcodec.so、libavformat.so等类库，而只使用了一个类库——lib...

zhangxin0414 (http://blog.csdn.net/zhangxin0414) 2016年10月09日 20:13 392

Android开发本地及网络Mp3音乐播放器(十九)通知媒体库更新刚下载的MP3 (http://blog.csdn...

转载请注明出处：http://blog.csdn.net/iwanghang/article/details/51427158 觉得博文有用，请点赞，请留言，请关注，谢谢！~ 实现功能： 通知媒体库...

iwanghang (http://blog.csdn.net/iwanghang) 2016年05月16日 18:44 6041

Android 拍照更新媒体库，相册选取图片显示 (http://download.csdn.net/do...

(http://download.csdn.net/download/vrix/6930095) 2016年05月27日 14:58 1.35MB 下载

内容举报

返回顶部

0