

原 最简单的基于FFmpeg的移动端例子：Android 视频转码器

2015年07月25日 17:10:54 阅读数：68402

=====

最简单的基于FFmpeg的移动端例子系列文章列表：

[最简单的基于FFmpeg的移动端例子：Android HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：Android 视频解码器](#)

[最简单的基于FFmpeg的移动端例子：Android 视频解码器-单个库版](#)

[最简单的基于FFmpeg的移动端例子：Android 推流器](#)

[最简单的基于FFmpeg的移动端例子：Android 视频转码器](#)

[最简单的基于FFmpeg的移动端例子附件：Android 自带播放器](#)

[最简单的基于FFmpeg的移动端例子附件：SDL Android HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：IOS HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：IOS 视频解码器](#)

[最简单的基于FFmpeg的移动端例子：IOS 推流器](#)

[最简单的基于FFmpeg的移动端例子：IOS 视频转码器](#)

[最简单的基于FFmpeg的移动端例子附件：IOS自带播放器](#)

[最简单的基于FFmpeg的移动端例子：Windows Phone HelloWorld](#)

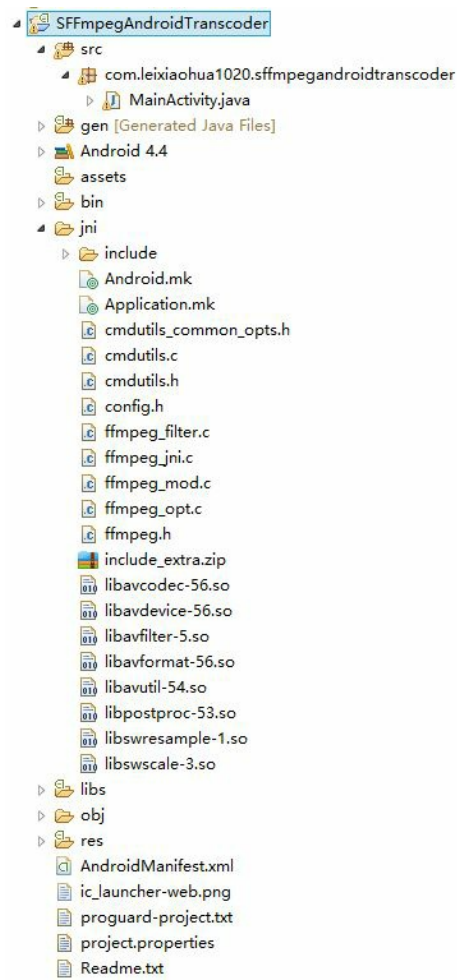
=====

本文记录一个安卓平台下基于FFmpeg的视频转码器。该转码器实际上移植自ffmpeg工程中的ffmpeg.c源代码。有关ffmpeg.c的源代码可以参考文章《[ffmpeg.c函数结构简单分析（画图）](#)》，在这里就不重复记录了。



源代码

项目的目录结构如图所示。Java源代码位于src目录，而C代码位于jni目录。



Android程序Java端代码位于src\com\leixiaohua1020\sffmpegandroidtranscoder\MainActivity.java，如下所示。

```

1.  /**
2.   * 最简单的基于FFmpeg的转码器-安卓
3.   * Simplest FFmpeg Android Transcoder
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   * leixiaohua1020@126.com
7.   * 马小雨 Ma Xiaoyu
8.   * maxiaoyucuc@163.com
9.   * 中国传媒大学/数字电视技术
10.  * Communication University of China / Digital TV Technology
11.  * http://blog.csdn.net/leixiaohua1020
12.  *
13.  *
14.  * 本程序是安卓平台下的转码器。它移植于ffmpeg.c命令行工具。
15.  *
16.  * This software is a Transcoder in Android.
17.  * It is transplanted from ffmpeg.c command line tools.
18.  *
19.  */
20. package com.leixiaohua1020.sffmpegandroidtranscoder;
21.
22. import android.os.Bundle;
23. import android.os.Environment;
24. import android.app.Activity;
25. import android.util.Log;
26. import android.view.Menu;
27. import android.view.View;
28. import android.view.View.OnClickListener;
29. import android.widget.Button;
30. import android.widget.EditText;
31.
32. public class MainActivity extends Activity {
33.
34.     @Override
35.     protected void onCreate(Bundle savedInstanceState) {
36.         super.onCreate(savedInstanceState);
37.         setContentView(R.layout.activity_main);
38.
39.         final EditText cmdEditText= (EditText) this.findViewById(R.id.editText_cmd);
40.         Button startButton= (Button) this.findViewById(R.id.button_start);
41.
42.         startButton.setOnClickListener(new OnClickListener() {
43.             public void onClick(View arg0){
44.                 String cmdline=cmdEditText.getText().toString();
45.                 String[] argv=cmdline.split(" ");
46.                 Integer argc=argv.length;
47.                 ffmpegcore(argc,argv);
48.             }
49.         });
50.     }
51.
52.     public native int ffmpegcore(int argc,String[] argv);
53.     static{
54.         System.loadLibrary("avutil-54");
55.         System.loadLibrary("swresample-1");
56.         System.loadLibrary("avcodec-56");
57.         System.loadLibrary("avformat-56");
58.         System.loadLibrary("swscale-3");
59.         System.loadLibrary("postproc-53");
60.         System.loadLibrary("avfilter-5");
61.         System.loadLibrary("avdevice-56");
62.         System.loadLibrary("sfffttranscoder");
63.     }
64.
65.     @Override
66.     public boolean onCreateOptionsMenu(Menu menu) {
67.         // Inflate the menu; this adds items to the action bar if it is present.
68.         getMenuInflater().inflate(R.menu.main, menu);
69.         return true;
70.     }
71.
72. }

```

C语言端源代码包含多个文件。其中下面文件拷贝自FFmpeg源代码：

```

cmdutils.c
cmdutils.h
cmdutils_common_opts.h
config.h
ffmpeg.h
ffmpeg_filter.c
ffmpeg_opt.c

```

在编译ffmpeg.c的时候，除了需要FFmpeg SDK中的头文件之外，还需要从源代码中拷贝下面头文件。

```

libavformat/ffmpeg.h

```

libavformat/network.h

libavformat/os_support.h

libavformat/url.h

libavutil/libm.h

ffmpeg_mod.c是修改后的ffmpeg.c文件。ffmpeg.c本身是用于命令行使用的程序，因此在改为类库调用的时候需要对源代码做一些修改。例如去掉一些exit(0)，释放一些变量，main()函数改名为ffmpegmain()等。

ffmpeg_jni.c是ffmpeg.c和Android系统之间的接口。该部分的源代码如下所示。

```

1.  /**
2.   * 最简单的基于FFmpeg的转码器-安卓
3.   * Simplest FFmpeg Android Transcoder
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   * leixiaohua1020@126.com
7.   * 马小雨 Ma Xiaoyu
8.   * maxiaoyucuc@163.com
9.   * 中国传媒大学/数字电视技术
10.  * Communication University of China / Digital TV Technology
11.  * http://blog.csdn.net/leixiaohua1020
12.  *
13.  *
14.  * 本程序是安卓平台下的转码器。它移植于ffmpeg.c命令行工具。
15.  *
16.  * This software is a Transcoder in Android.
17.  * It is transplanted from ffmpeg.c command line tools.
18.  *
19.  */
20.
21. #include <string.h>
22. #include <jni.h>
23. #include <ffmpeg.h>
24.
25. #ifdef ANDROID
26. #include <jni.h>
27. #include <android/log.h>
28. #define LOGE(format, ...) __android_log_print(ANDROID_LOG_ERROR, ">_<", format, ##_VA_ARGS_)
29. #define LOGI(format, ...) __android_log_print(ANDROID_LOG_INFO, "^_^", format, ##_VA_ARGS_)
30. #else
31. #define LOGE(format, ...) printf(">_< " format "\n", ##_VA_ARGS_)
32. #define LOGI(format, ...) printf("^_^ " format "\n", ##_VA_ARGS_)
33. #endif
34.
35.
36. int ffmpegmain(int argc, char **argv);
37.
38. //Output Ffmpeg's av_log()
39. void custom_log(void *ptr, int level, const char* fmt, va_list vl){
40.
41.     //To TXT file
42.
43.     FILE *fp=fopen("/storage/emulated/0/av_log.txt","a+");
44.     if(fp){
45.         vfprintf(fp,fmt,vl);
46.         fflush(fp);
47.         fclose(fp);
48.     }
49.
50.
51.     //To Logcat
52.     //LOGE(fmt, vl);
53. }
54.
55. JNIEXPORT jint JNICALL Java_com_leixiaohua1020_sffmpegandroidtranscoder_MainActivity_ffmpegcore( JNIEnv * env, jobject this, jint cmd
, jobjectArray cmdline)
56. {
57.
58.     //FFmpeg av_log() callback
59.     av_log_set_callback(custom_log);
60.
61.     int argc=cmdnum;
62.     char** argv=(char**)malloc(sizeof(char*)*argc);
63.
64.     int i=0;
65.     for(i=0;i<argc;i++)
66.     {
67.         jstring string=(*env)->GetObjectArrayElement(env,cmdline,i);
68.         const char* tmp=(*env)->GetStringUTFChars(env,string,0);
69.         argv[i]=(char*)malloc(sizeof(char)*1024);
70.         strcpy(argv[i],tmp);
71.     }
72.
73.     ffmpegmain(argc,argv);
74.
75.     for(i=0;i<argc;i++){
76.         free(argv[i]);
77.     }
78.     free(argv);
79.     return 0;
80.
81. }

```

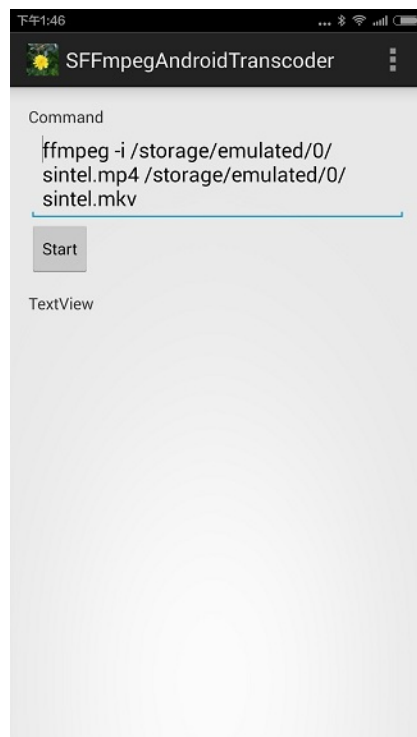
Android.mk文件位于jni/Android.mk，如下所示。

[plain]  

```
1.  # Android.mk for FFmpeg
2.  #
3.  # Lei Xiaohua 雷霄骅
4.  # leixiaohua1020@126.com
5.  # http://blog.csdn.net/leixiaohua1020
6.  #
7.
8.  LOCAL_PATH := $(call my-dir)
9.
10. # FFmpeg library
11. include $(CLEAR_VARS)
12. LOCAL_MODULE := avcodec
13. LOCAL_SRC_FILES := libavcodec-56.so
14. include $(PREBUILT_SHARED_LIBRARY)
15.
16. include $(CLEAR_VARS)
17. LOCAL_MODULE := avdevice
18. LOCAL_SRC_FILES := libavdevice-56.so
19. include $(PREBUILT_SHARED_LIBRARY)
20.
21. include $(CLEAR_VARS)
22. LOCAL_MODULE := avfilter
23. LOCAL_SRC_FILES := libavfilter-5.so
24. include $(PREBUILT_SHARED_LIBRARY)
25.
26. include $(CLEAR_VARS)
27. LOCAL_MODULE := avformat
28. LOCAL_SRC_FILES := libavformat-56.so
29. include $(PREBUILT_SHARED_LIBRARY)
30.
31. include $(CLEAR_VARS)
32. LOCAL_MODULE := avutil
33. LOCAL_SRC_FILES := libavutil-54.so
34. include $(PREBUILT_SHARED_LIBRARY)
35.
36. include $(CLEAR_VARS)
37. LOCAL_MODULE := postproc
38. LOCAL_SRC_FILES := libpostproc-53.so
39. include $(PREBUILT_SHARED_LIBRARY)
40.
41. include $(CLEAR_VARS)
42. LOCAL_MODULE := swresample
43. LOCAL_SRC_FILES := libswresample-1.so
44. include $(PREBUILT_SHARED_LIBRARY)
45.
46. include $(CLEAR_VARS)
47. LOCAL_MODULE := swscale
48. LOCAL_SRC_FILES := libswscale-3.so
49. include $(PREBUILT_SHARED_LIBRARY)
50.
51. # Program
52. include $(CLEAR_VARS)
53. LOCAL_MODULE := sfftranscoder
54. LOCAL_SRC_FILES := ffmpeg_jni.c ffmpeg_mod.c ffmpeg_opt.c ffmpeg_filter.c cmdutils.c
55. LOCAL_C_INCLUDES += $(LOCAL_PATH)/include
56. LOCAL_LDLIBS := -llog -lz
57. LOCAL_SHARED_LIBRARIES := avcodec avdevice avfilter avformat avutil postproc swresample swscale
58. include $(BUILD_SHARED_LIBRARY)
```

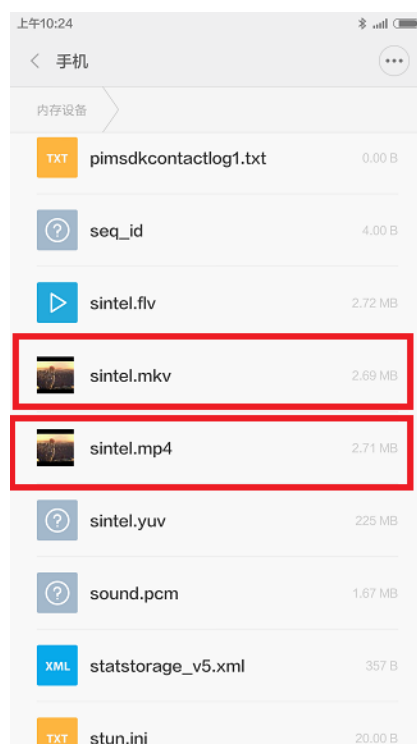
运行结果

App在手机上运行后的结果如下图所示。



注意需要把等待转码的视频文件拷贝至存储卡相应的目录中。例如对于上述截图的情况，需要将sintel.mp4拷贝至存储卡的根目录中。

单击“Start”按钮开始转码，将存储卡中的sintel.mp4转码为sintel.mkv。



下载

simplest ffmpeg mobile

项目主页

Github：https://github.com/leixiaohua1020/simplest_ffmpeg_mobile

开源中国：https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_mobile

SourceForge：<https://sourceforge.net/projects/simplestffmpegmobile/>

CSDN工程下载地址：<http://download.csdn.net/detail/leixiaohua1020/8924391>

本解决方案包含了使用FFmpeg在移动端处理多媒体的各种例子：

[Android]

simplest_android_player: 基于安卓接口的视频播放器

simplest_ffmpeg_android_helloworld: 安卓平台下基于FFmpeg的HelloWorld程序

simplest_ffmpeg_android_decoder: 安卓平台下最简单的基于FFmpeg的视频解码器

simplest_ffmpeg_android_decoder_onelib: 安卓平台下最简单的基于FFmpeg的视频解码器-单库版

simplest_ffmpeg_android_streamer: 安卓平台下最简单的基于FFmpeg的推流器

simplest_ffmpeg_android_transcoder: 安卓平台下移植的FFmpeg命令行工具

simplest_sdl_android_helloworld: 移植SDL到安卓平台的最简单程序

[IOS]

simplest_ios_player: 基于IOS接口的视频播放器

simplest_ffmpeg_ios_helloworld: IOS平台下基于FFmpeg的HelloWorld程序

simplest_ffmpeg_ios_decoder: IOS平台下最简单的基于FFmpeg的视频解码器

simplest_ffmpeg_ios_streamer: IOS平台下最简单的基于FFmpeg的推流器

simplest_ffmpeg_ios_transcoder: IOS平台下移植的ffmpeg.c命令行工具

simplest_sdl_ios_helloworld: 移植SDL到IOS平台的最简单程序

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/47056365>

文章标签： ffmpeg Android 转码 JNI 视频

个人分类： Android多媒体 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