

原 最简单的基于FFmpeg的移动端例子附件：SDL Android HelloWorld

2015年07月25日 23:02:55 阅读数：17556

=====

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SDL的例子。该例子读取并显示了一张BMP图片，通过该例子可以了解SDL在安卓平台下的使用。本文中使用的SDL版本为2.0。



Android程序SDL2类库使用说明

简单记录一下Android程序使用SDL2类库的方法。这部分的信息可以参考SDL2源代码目录中的README-android.txt文件。SDL2使用的步骤可以分为以下步骤：

(1) 配置Android工程

a) 新建Android工程

可以直接使用SDL2源代码目录中的android-project工程，也可以自己新建工程。如果自己新建工程，则需要拷贝android-project工程中src目录下的SDLActivity.java文件。该文件中的SDLActivity定义比较长，目前还没有研究过。

b) 拷贝SDL2源代码

将SDL2源代码目录中下列内容拷贝至Android工程的jni目录下：

src目录

include目录

Android.mk

(2) 编译C语言代码

a) 编写C语言代码

在Android工程的jni目录下新建C语言文件并编写调用SDL2的代码。在这里需要注意C语言代码的主函数名称和命令执行程序是一样的，依然是“main(int argc, char *argv[])”。

b) 修改Android.mk文件

在Android.mk文件后面追加一段代码，用来编译一个libSDL2main.so的库。该库可以调用自己的C语言程序。在这里需要注意，“libSDL2main”这个名称是可以随意起的，但是要和SDLActivity中LoadLibrary()函数中的库名称相对应（默认名称应该是libmain.so）。代码如下所示。

```
[plain]
1. #libSDL2main=====
2. # Lei Xiaohua
3. include $(CLEAR_VARS)
4. LOCAL_MODULE := SDL2main
5. SDL_PATH := ./
6. LOCAL_C_INCLUDES := $(LOCAL_PATH)/$(SDL_PATH)/include
7. # Add your application source files here...
8. LOCAL_SRC_FILES := $(SDL_PATH)/src/main/android/SDL_android_main.c \
9.     $(SDL_PATH)/simplest_showbmp.c
10. LOCAL_SHARED_LIBRARIES := SDL2
11. LOCAL_LDLIBS := -lGLESv1_CM -lGLESv2 -llog
12. include $(BUILD_SHARED_LIBRARY)
```

PS：上述代码修改自SDL2源代码目录中android-project工程的jni\src目录下的Android.mk文件。

c) 编写Application.mk文件（可选）

Application.mk中的APP_ABI设定了编译后库文件支持的指令集，默认使用“armeabi”，这里使用默认值就可以了。

d) 运行ndk-build

编写完C语言代码和Android.mk之后就可以运行ndk-build命令进行编译了。ndk-build命令运行成功后，会在根目录下的“libs/armeabi”目录中生成2个库文件：

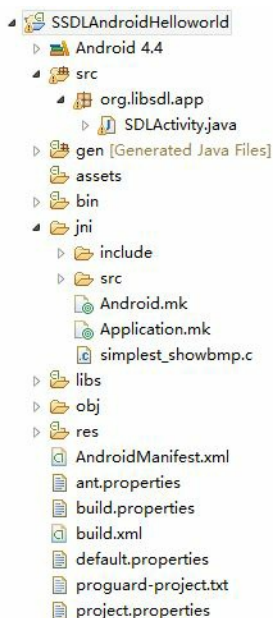
libSDL2.so

libSDL2main.so

接下来就可以在Android手机或者虚拟机上对整个Android工程进行测试了。

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\org\libsdl\app\SDLActivity.java。该Activity取自于SDL2源代码目录中的android-project工程，在这里不再记录。

jni目录中的src和include文件夹分别是SDL2的源代码目录中的src文件夹和include文件夹。

C语言端源代码位于jni/simplest_showbmp.c，如下所示。该源代码读取了存储卡中的一张test.bmp文件，并显示出来。

```

1.  /**
2.   * 移植SDL到安卓平台的HelloWorld程序
3.   * Simplest SDL Android Helloworld
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.  *
12.  * 本程序是移植SDL到安卓平台的最简单程序。它可以读取并显示一张BMP图片。
13.  *
14.  * This software is the simplest program transplant SDL2 to Android platform.
15.  * It shows a BMP file on the screen.
16.  */
17.
18. #ifdef __ANDROID__
19.
20. #include <jni.h>
21. #include <android/log.h>
22. #define LOGI(...) __android_log_print(ANDROID_LOG_INFO, "(^_^)", __VA_ARGS__)
23. #define LOGE(...) __android_log_print(ANDROID_LOG_ERROR, "(^_^)", __VA_ARGS__)
24. #else
25. #define LOGE(format, ...) printf("(>_<) " format "\n", ##__VA_ARGS__)
26. #define LOGI(format, ...) printf("(^_^) " format "\n", ##__VA_ARGS__)
27. #endif
28.
29. #include "SDL.h"
30. #include "SDL_log.h"
31. #include "SDL_main.h"
32.
33. int main(int argc, char *argv[]) {
34.     struct SDL_Window *window = NULL;
35.     struct SDL_Renderer *render = NULL;
36.     struct SDL_Surface *bmp = NULL;
37.     struct SDL_Texture *texture = NULL;
38.
39.     char *filepath = "/storage/emulated/0/test.bmp";
40.
41.     if (SDL_Init(SDL_INIT_VIDEO | SDL_INIT_AUDIO | SDL_INIT_TIMER) == -1) {
42.         LOGE("SDL_Init failed %s", SDL_GetError());
43.     }
44.
45.     window = SDL_CreateWindow("SDL HelloWorld!", 100, 100, 640, 480,
46.                               SDL_WINDOW_SHOWN);
47.     if (window == NULL) {
48.         LOGE("SDL_CreateWindow failed %s", SDL_GetError());
49.     }
50.
51.     render = SDL_CreateRenderer(window, -1,
52.                                SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC);
53.     if (render == NULL) {
54.         LOGE("SDL_CreateRenderer failed %s", SDL_GetError());
55.     }
56.
57.     bmp = SDL_LoadBMP(filepath);
58.     if (bmp == NULL) {
59.         LOGE("SDL_LoadBMP failed: %s", SDL_GetError());
60.     }
61.
62.     texture = SDL_CreateTextureFromSurface(render, bmp);
63.     SDL_FreeSurface(bmp);
64.
65.     SDL_RenderClear(render);
66.     SDL_RenderCopy(render, texture, NULL, NULL);
67.     SDL_RenderPresent(render);
68.
69.     SDL_Delay(10000);
70.
71.     SDL_DestroyTexture(texture);
72.     SDL_DestroyRenderer(render);
73.     SDL_DestroyWindow(window);
74.
75.     //Quit SDL
76.     SDL_Quit();
77.     return 0;
78. }

```

Android.mk文件位于jni/Android.mk，如下所示。

```

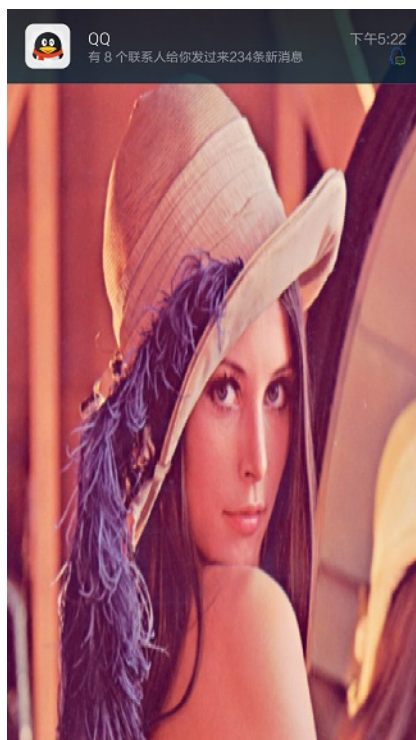
[plain]
1.  LOCAL_PATH := $(call my-dir)
2.
3.  #####
4.  #
5.  # SDL shared library
6.  #
7.  #####
8.
9.  include $(CLEAR_VARS)
10.
11.  LOCAL_MODULE := SDL2
12.
13.  LOCAL_C_INCLUDES := $(LOCAL_PATH)/include
14.
15.  LOCAL_EXPORT_C_INCLUDES := $(LOCAL_C_INCLUDES)
16.
17.  LOCAL_SRC_FILES := \
18.      $(subst $(LOCAL_PATH)/,, \
19.      $(wildcard $(LOCAL_PATH)/src/*.c) \
20.      $(wildcard $(LOCAL_PATH)/src/audio/*.c) \
21.      $(wildcard $(LOCAL_PATH)/src/audio/android/*.c) \
22.      $(wildcard $(LOCAL_PATH)/src/audio/dummy/*.c) \
23.      $(LOCAL_PATH)/src/atomic/SDL_atomic.c \
24.      $(LOCAL_PATH)/src/atomic/SDL_spinlock.c.arm \
25.      $(wildcard $(LOCAL_PATH)/src/core/android/*.c) \
26.      $(wildcard $(LOCAL_PATH)/src/cpuinfo/*.c) \
27.      $(wildcard $(LOCAL_PATH)/src/dynapi/*.c) \
28.      $(wildcard $(LOCAL_PATH)/src/events/*.c) \
29.      $(wildcard $(LOCAL_PATH)/src/file/*.c) \
30.      $(wildcard $(LOCAL_PATH)/src/haptic/*.c) \
31.      $(wildcard $(LOCAL_PATH)/src/haptic/dummy/*.c) \
32.      $(wildcard $(LOCAL_PATH)/src/joystick/*.c) \
33.      $(wildcard $(LOCAL_PATH)/src/joystick/android/*.c) \
34.      $(wildcard $(LOCAL_PATH)/src/loadso/dlopen/*.c) \
35.      $(wildcard $(LOCAL_PATH)/src/power/*.c) \
36.      $(wildcard $(LOCAL_PATH)/src/power/android/*.c) \
37.      $(wildcard $(LOCAL_PATH)/src/filesystem/dummy/*.c) \
38.      $(wildcard $(LOCAL_PATH)/src/render/*.c) \
39.      $(wildcard $(LOCAL_PATH)/src/render/*.c) \
40.      $(wildcard $(LOCAL_PATH)/src/stdlib/*.c) \
41.      $(wildcard $(LOCAL_PATH)/src/thread/*.c) \
42.      $(wildcard $(LOCAL_PATH)/src/thread/pthread/*.c) \
43.      $(wildcard $(LOCAL_PATH)/src/timer/*.c) \
44.      $(wildcard $(LOCAL_PATH)/src/timer/unix/*.c) \
45.      $(wildcard $(LOCAL_PATH)/src/video/*.c) \
46.      $(wildcard $(LOCAL_PATH)/src/video/android/*.c) \
47.      $(wildcard $(LOCAL_PATH)/src/test/*.c)
48.
49.  LOCAL_CFLAGS += -DGL_GLEXT_PROTOTYPES
50.  LOCAL_LDLIBS := -ldl -lGLESv1_CM -lGLESv2 -llog -landroid
51.
52.  include $(BUILD_SHARED_LIBRARY)
53.
54.  #####
55.  #
56.  # SDL static library
57.  #
58.  #####
59.
60.  LOCAL_MODULE := SDL2_static
61.
62.  LOCAL_MODULE_FILENAME := libSDL2
63.
64.  LOCAL_SRC_FILES += $(LOCAL_PATH)/src/main/android/SDL_android_main.c
65.
66.  LOCAL_LDLIBS :=
67.  LOCAL_EXPORT_LDLIBS := -Wl,--undefined=Java_org_libsdl_app_SDLActivity_nativeInit -ldl -lGLESv1_CM -lGLESv2 -llog -landroid
68.
69.  include $(BUILD_STATIC_LIBRARY)
70.
71.
72.  #libSDL2main=====
73.  # Lei Xiaohua
74.  include $(CLEAR_VARS)
75.  LOCAL_MODULE := SDL2main
76.  SDL_PATH := ./
77.  LOCAL_C_INCLUDES := $(LOCAL_PATH)/$(SDL_PATH)/include
78.  # Add your application source files here...
79.  LOCAL_SRC_FILES := $(SDL_PATH)/src/main/android/SDL_android_main.c \
80.      $(SDL_PATH)/simplest_showbmp.c
81.  LOCAL_SHARED_LIBRARIES := SDL2
82.  LOCAL_LDLIBS := -lGLESv1_CM -lGLESv2 -llog
83.  include $(BUILD_SHARED_LIBRARY)

```

运行结果

App在手机上运行后的结果如下图所示。可以看出App读取了test.bmp文件并且显示出来。

注意需要把BMP文件拷贝至存储卡相应的目录中。程序默认情况下会读取根目录下的“test.bmp”文件。



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/47059553>

文章标签：SDL Android BMP 视频

个人分类：FFMPEG SDL Android多媒体

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