


```
14     /
15     for (int i = 0; i < currentFiles.length; i++) {
16         if (currentFiles[i].getName().contains(DuduUtil.SoFile.JKFFMPEG)) {
17             hasJkffmpeg = true;
18         }
19     }
20     return hasJkffmpeg;
21 }
```

先判断app_lib下有没有我们需要加载的so文件,如果没有的话复制到指定目录,其中formPath是下载到sdcard上so文件的路径。

```
1 public static int copy(String fromFile, String toFile) {
2     //要复制的文件夹
3     File[] currentFiles;
4     File root = new File(fromFile);
5     //如同判断SD卡是否存在或者文件是否存在
6     //如果不存在则 return出去
7     if (!root.exists()) {
8         return -1;
9     }
10    //如果存在则获取当前目录下的全部文件 填充数组
11    currentFiles = root.listFiles();
12
13    //目标目录
14    File targetDir = new File(toFile);
15    //创建目录
16    if (!targetDir.exists()) {
17        targetDir.mkdirs();
18    }
19    //遍历要复制该目录下的全部文件
20    for (int i = 0; i < currentFiles.length; i++) {
21        if (currentFiles[i].isDirectory()) {
22            //如果当前项为子目录 进行递归
23            copy(currentFiles[i].getPath() + "/", toFile + currentFiles[i].getName() + "/");
24        } else {
25            //如果当前项为文件则进行文件拷贝
26            Log.e(TAG, "path:" + currentFiles[i].getPath());
27            Log.e(TAG, "name:" + currentFiles[i].getName());
28            if (currentFiles[i].getName().contains(".so")) {
29                int id = copySdcardFile(currentFiles[i].getPath(), toFile + File.separator + currentFiles[i].getName());
30                Log.e(TAG, "id:" + id);
31            }
32        }
33    }
34 }
35 return 0;
```



0



```

35         return 0;
36     }
37
38
39     //文件拷贝
40     //要复制的目录下的所有非子目录(文件夹)文件拷贝
41     public static int copySdcardFile(String fromFile, String toFile) {
42
43         try {
44             FileInputStream fosfrom = new FileInputStream(fromFile);
45             FileOutputStream fosto = new FileOutputStream(toFile);
46             ByteArrayOutputStream baos = new ByteArrayOutputStream();
47             byte[] buffer = new byte[1024];
48             int len = -1;
49             while ((len = fosfrom.read(buffer)) != -1) {
50                 baos.write(buffer, 0, len);
51             }
52             // 从内存写入到具体文件
53             fosto.write(baos.toByteArray());
54             // 关闭文件流
55             baos.close();
56             fosto.close();
57             fosfrom.close();
58             return 0;
59
60         } catch (Exception ex) {
61             return -1;
62         }
63     }

```

Load进工程：

```

1  File dir = context.getDir("libs", Context.MODE_PRIVATE);
2  File[] currentFiles;
3  currentFiles = dir.listFiles();
4  for (int i = 0; i < currentFiles.length; i++) {
5      Log.e(TAG, "#:" + currentFiles[i].getAbsolutePath());
6      System.load(currentFiles[i].getAbsolutePath());
7
8  }

```

出现的问题：

1. java.lang.UnsatisfiedLinkError: dlopen failed: “XXXX.so” is 32-bit instead of 64-bit

1. 大体原因是手机cpu为64位，并且在app中你配置了加载64位（arm64-v8a），所有系统会去加载64位路径下的so库，但是你load的so库却是32位，解决方法有两种：1、将32为库换成64位，2、只加载32位库，在build.gradle中配置,armeabi, armeabi-v7a, x86位32位, arm64-v8a, x86_64是64位。




```
1 defaultConfig {
2     applicationId "xxx"
3     minSdkVersion 10
4     targetSdkVersion 21
5     versionCode 1
6     versionName "1.0"
7
8     ndk {
9         abiFilters "armeabi","armeabi-v7a","x86"
10    }
11 }
```

2.java.lang.UnsatisfiedLinkError: Couldn't load XXX from loader dalvik.system.DexClassLoader[DexPathList[]: findLibrary returned null

加载时注意加载so文件的顺序，遇到的问题是load ijkffmpeg.so文件时需要先load另外一个so文件，但是我先load了ffmpeg，所以报了 couldn't load异常

版权声明：本文为博主原创文章，未经博主允许不得转载。


 发表你的评论

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

相关文章推荐


Android JNI调用的二种方式 (http://blog.csdn.net/csdn49532/article/details/46605813)

Android JNI调用的二种方式 首先创建一个专门用于JNI调用的单例MainNative,并且在里面添加静态块加载so库。 publicclass MainNative ...

 csdn49532 (http://blog.csdn.net/csdn49532) 2015年06月23日 14:49 537

在 Android 系统中直接调用 SO文件（包含System.loadLibrary加载so的路径解释） (http://bl...

Android虚拟机不能直接调用底层设备，如果要对底层设备进行调用，就需要用到so使用 C语言或C++编写完成，利用NDK进行编译，直接运行在Linux内核中，按jni调用 so时， 基本类型可...

 wangjie5540 (http://blog.csdn.net/wangjie5540) 2014年05月23日 11:38 7123



月薪40k的前端程序员都避开了哪些坑？

程序员薪水有高有低，同样工作5年的程序员，有的人月薪30K、50K，有的人可能只有5K、8K。是什么因素导致了这种差异？

(http://www.baidu.com/cb.php?c=laF pvfanHmkni0dP1f0lZ0anfK9uiYzP1nYPH0k0Aw-



- 🔗
- 👍
- 0
- ☰
- 🔖
- 💬
- 🔗

5Hc3rHnYnHb0TAq15HfLPWRznb0T1dBpVfHPAfkn1Nbuhf1mHN-0AwY5HDdnHc4njD4PHf0IgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfElAqspynElvNBnHqdlAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqnHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqn1nLrHR)

JNI调用时so链接库文件的路径设置 (http://blog.csdn.net/erliang20088/article/details/45790...

最近在做JNI程序时，用到了相当于windows环境下的dll--动态链接库的so库，即linux环境下，程序执行时所需要函数链接库。一般来说的so库路径寻找包括编译时和运行时，因为前边提到的是J...
erliang20088 (http://blog.csdn.net/erliang20088) 2015年05月17日 16:52 1930

Android JNI学习笔记——so文件动态加载 (http://blog.csdn.net/hp910315/article/details/51...

Android中对于so的加载提供了两个方法。System.loadLibrary("libName"); System.load("pathName");/** * See {@link Runt...
hp910315 (http://blog.csdn.net/hp910315) 2016年06月22日 15:36 1923

第二十期 在Android中修改GPS定位数据的完整方案《手机就是开发板》 (http://blog.csdn.ne...

现在很多的应用都是基于位置服务的，而且把位置服务作为重要的参考标准，其实这是很不安全的，位置信息的数据未经过任何加密，而且是从我们的手机中发送出去的，所以是可以修改的。这一期我们来探讨一下如何修改手机...
aggresss (http://blog.csdn.net/aggresss) 2017年01月11日 00:36 7121



一学就会的 WordPress 实战课

学习完本课程可以掌握基本的 WordPress 的开发能力，后续可以根据需要开发适合自己的主题、插件，打造最个性的 WordPress 站点。

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjfvP1m0IZ0qnfK9ujYzP1f4Pjnz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1d-n1c1PywhPycvPW7BuWnL0AwY5HDdnHc4njD4PHf0IgF_5y9YIZ0IQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1N9nHmvnj7hnHPWnjFhPAD1Pyn4uW99ujqdlAdxTvqdThP-5HDknWw9mhkEusKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnWnzPjn)



android WebView加载H5去地址栏的两种方式 (http://download.csdn.net/d...


(http://download... 2017年09月27日 15:47 15.9MB 下载

Android源码分析实战之JNI so库加载System.loadLibrary流程分析 (http://blog.csdn.net/zha...

JNI so库加载流程之System.loadLibrary流程分析最近在学习jni相关的知识，很多博客都说，jni**动态注册*时可以调用System.loadLibrary或者System.lo...
zhangmingbao2016 (http://blog.csdn.net/zhangmingbao2016) 2017年06月21日 17:50 1587

Android Jni调用so库，加载库失败分析 (http://blog.csdn.net/yang_hui1986527/article/deta...

现有一个项目，通过JNI接口调用底层库，例如：lib***.so 如下所示，总是加载不成功。调试发现，每次加载so库，会跳到cat ch异常处理部分，然后打印异常log信息。 static{ tr...
yang_hui1986527 (http://blog.csdn.net/yang_hui1986527) 2011年08月12日 00:27 3794




Android studio CMake方式引入jni .so文件 (<http://download.csdn.net/dow...>)

<http://download.csdn.net/dow...> 2017年11月16日 15:14 11.45MB

下载

android JNI 使用的两种形式 --自己实现c和调用第三方so库 (<http://blog.csdn.net/pengpeng3...>)


1. 在调用JNI的工程中实现JNI代码 JNI调用，一般书上都会给你讲第一种形式，自己写JNI代码，，然后编译使用，在eclipse中 --项目--右键- Android tools --添加...

 pengpeng375 (<http://blog.csdn.net/pengpeng375>) 2015年01月24日 10:53

1133

(9) hibernate加载持久化对象的两种方式——get、load (<http://blog.csdn.net/jianfpeng24124...>)


一.get与load对比 在hibernate中get和load方法是根据id取得持久化对象的两种方法，但在实际使用的过程中总会把两者混淆，不知道什么情况下使用get好，什么时候使用loa...

 jianfpeng241241 (<http://blog.csdn.net/jianfpeng241241>) 2016年11月20日 14:27

167

android jni方式调用c++代码并在其它app中使用生成的.so文件 (<http://blog.csdn.net/lishaob...>)


搞了一下午终于明白了一点机制，网上的代码太乱了，很多不经过验证就转载的，强烈谴责这种行为，以下所写都是android studio测试运行过的，欢迎交流指正。运行前提是在project structu...

 lishaobo211985 (<http://blog.csdn.net/lishaobo211985>) 2016年10月18日 21:46

1040

Android Studio cmake方式jni开发，在c层使用其它动态链接库*.so (<http://blog.csdn.net/qq...>)

最近在研究Google推荐的cmake方式的ndk开发，遍寻网络居然很难找到如何在c层调用其它动态链接库函数（cmake方式），经过不懈努力，终于找到了可行的方法；本人初次涉及ndk开发，如有失误之处...

 qq372848728 (<http://blog.csdn.net/qq372848728>) 2016年11月16日 01:30

384




内存加载exe的两种方式（支持win7） (<http://download.csdn.net/download...>)

<http://download.csdn.net/download...> 2016年11月24日 19:11 139KB

下载

Android 应用开发 之通过AsyncTask与ThreadPool（线程池）两种方式异步加载大量数据的分...


在加载大量数据的时候，经常会用到异步加载，所谓异步加载，就是把耗时的工作放到子线程里执行，当数据加载完毕的时候再到主线程进行UI刷新。在数据量非常大的情况下，我们通常会使用两种技术来进行异步加载，一是...

 nameyuxiang (<http://blog.csdn.net/nameyuxiang>) 2014年10月20日 16:40

670

Xamarin Android Fragment的两种加载方式 (<http://blog.csdn.net/kebi007/article/details/52...>)



xamarin android fragment两种加载方法的简单用法

 kebi007 (<http://blog.csdn.net/kebi007>) 2016年10月05日 21:29

1804



Android 应用开发 之通过AsyncTask与ThreadPool（线程池）两种方式异步加载大量数据的分...

Android 应用开发 之通过AsyncTask与ThreadPool（线程池）两种方式异步加载大量数据的分析与对比 标签： AndroidAsyncTaskThreadPool异步加载v...

 w690333243 (<http://blog.csdn.net/w690333243>) 2017年05月26日 17:15  162



Android 应用开发 之通过AsyncTask与ThreadPool（线程池）两种方式异步加载大量数据的分...

在加载大量数据的时候，经常会用到异步加载，所谓异步加载，就是把耗时的工作放到子线程里执行，当数据加载完毕的时候再到主线程进行UI刷新。在数据量非常大的情况下，我们通常会使用两种技术来进行异步加载，一是...

 carrey1989 (<http://blog.csdn.net/carrey1989>) 2013年09月25日 11:53  14892



Android实战简易教程-第四十九枪（两种方式实现网络图片异步加载）(<http://blog.csdn.net/yayun0516>...

加载图片属于比较耗时的工作，我们需要异步进行加载，异步加载有两种方式：1.通过AsyncTask类进行；2.通过Handler来实现，下面我们就来看一下如何通过这两种方式实现网络图片的异步加载。一、A...

 yayun0516 (<http://blog.csdn.net/yayun0516>) 2015年09月01日 16:38  1869

Dalvik虚拟机中so的加载和JNI调用 (<http://blog.csdn.net/dingjikerbo/article/details/50529516>...

本文主要目的是搞清楚so加载和调用的整体脉络，以便遇到so相关的问题时做到心中有数，能迅速把握解决问题的大方向。先抛出两个问题：1. 退出插件时如何卸载so，以免内存占用越来越大？2. 热修复...

 dingjikerbo (<http://blog.csdn.net/dingjikerbo>) 2016年01月25日 15:59  1224



0

