# jni中arm64-v8a，armeabi-v7a,armeabi文件夹的意义和用法

原创 2016年05月26日 11:15:01

* 标签：
* [jni](http://so.csdn.net/so/search/s.do?q=jni&t=blog) /
* [arm](http://so.csdn.net/so/search/s.do?q=arm&t=blog) /
* [android](http://so.csdn.net/so/search/s.do?q=android&t=blog)
* 12872

# jni中arm64-v8a，armeabi-v7a,armeabi文件夹的意义和用法

# 起因

之前并没有关注这块，直到：**您的应用被拒绝，原因：xplay5sQ心里点击笑值点击拍照显示停止运行**，查看发过来的日志，发现是**java.lang.UnsatisfiedLinkError: dlopen failed: library “/libfacesdk.so” not found** ，明明放了该so文件，为什么报错呢？

查xplay5s的cpu，发现是arm64架构的处理器，查看so文件目录，如下：

弄懂这几个文件夹到底是怎么一个顺序先。

# 兼容和文件读取顺序

arm64-v8a是可以向下兼容的，其下有armeabi-v7a，armeabi   
armeabi-v7a向下兼容armeabi

兼容得不够智能：   
对于一个cpu是arm64-v8a架构的手机，它运行app时，进入jnilibs去读取库文件时，先看有没有arm64-v8a文件夹：

如果没有该文件夹，去找armeabi-v7a文件夹，如果没有，再去找armeabi文件夹，如果连这个文件夹也没有，就抛出异常   
如果有arm64-v8a文件夹，那么就去找特定名称的.so文件，**注意：如果没有找到，不会再往下（armeabi-v7a文件夹）找了，而是直接抛出异常**

# armeabi与armeabi-v7a的区别

armeabi就是针对普通的或旧的arm cpu，armeabi-v7a是针对有浮点运算或高级扩展功能的arm cpu。   
android从2.2开始支持armeabi-v7a

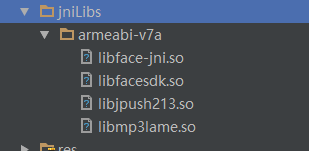
早期的Cortex-A系列处理器（A5、A7、A8、A9、A12、A15和A17）基于 ARMv7-A架构。

高通820 四核心ARMv8   
三星Exynos 8890 ARM Cortex-A53   
麒麟950 四核A72+四核A53

从cpu规格表可以看出，目前市面上主流手机cpu多属于armeabi-v7a。   
cpu天梯图和规格表：   
<http://www.mydrivers.com/zhuanti/tianti/01/>   
<http://www.mydrivers.com/zhuanti/tianti/01/index_other.html>

# 最终修正

所以，从兼容性和性能上考虑，上面的应该改成：



同时，为了避免引用的第三方库中有arm64-v8a的so文件导致64位机器仍然从arm64-v8a文件夹读取so文件，将非armeabi-v7a的so文件全部过滤掉:加入ndk .abiFilters

defaultConfig {

applicationId "xx.xx.x.xx"

minSdkVersion 15

targetSdkVersion 22

versionCode 15

versionName "1.2.7"

multiDexEnabled true

ndk {

abiFilters "armeabi-v7a" // 指定要ndk需要兼容的架构(这样其他依赖包里mips,x86,armeabi,arm-v8之类的so会被过滤掉)

}

}

版权声明：本文为博主原创文章，未经博主允许不得转载。