首页 (/) > 文章 (/blog/index.html) > android studio ndk-build 编译C生成.so文件 (ndk基础篇)

android studio ndk-build 编译C生成.so文件(ndk基础篇)

作者: android 发布时间: 2017-08-29 浏览: 611 次

一、概要

最近项目需要,要把代码中加密的部分打包成so文件,刚开始接触的时候真是痛苦呀,网上好多资料,都不是很详细,步骤也不清晰,所以我整理了一下,希望大家喜欢。

现在android studio打包so文件有两种方式,第一种是ndk-build编译项目,还有一种用CMake脚本构建项目,今天主要介绍ndk-build的编译过程

二、ndk-build 构建过程

1,下载NDK和构建工具

正所谓:"工欲善其事必先利其器",我们今天的主要工具是Android Studio2.3版本,至于 Android Studio环境搭建,sdk,jdk什么的,你们自己去弄,这里主要是讲解ndk编译c语言的配置。

为您的应用编译和调试原生代码,您需要以下组件:

- Android 原生开发工具包 (NDK):这套工具集允许您为 Android 使用 C 和 C++ 代码,并提供众多平台库,让您可以管理原生 Activity 和访问物理设备组件,例如传感器和触摸输入。
- CMake:一款外部构建工具,可与 Gradle 搭配使用来构建原生库。如果您只计划使用 ndk-build,则不需要此组件。
- LLDB:一种调试程序, Android Studio 使用它来调试原生代码。

您可以使用 SDK 管理器安装这些组件:

写文章 (/blog/create.html)



(/blog/user/101.html)

android (/blog/user/101.html)

移动开发工程师

38 篇博客 (/blo g/user/101.htm l) 5934 次访问 (/bl og/user/101.htm l)

热文排行榜

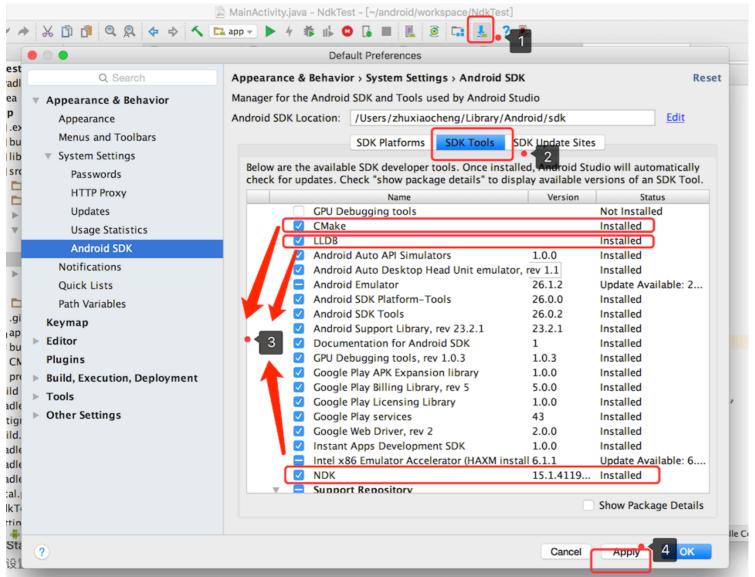
▶ Filezilla- 更多 (/blog/index.html?sort=hot)
-服务器发

回了不可路由的地址,使用服务器地址代替-解决办法 (/blog/linux/article/136.html)

浏览数:1242 关注数:297

▶ android studio ndk-build 编译C生成.so 文件 (ndk基础篇) (/blog/android/article/246.html)

- 1. 在打开的项目中,从菜单栏选择 Tools > Android > SDK Manager(或者在顶部工具栏中直接点击SDK Manager, 下图的标记就是从这里点击的)。
- 2. 点击 SDK Tools 标签。
- 3. 选中 LLDB、CMake 和 NDK 旁的复选框,如图 1 所



- 4. 点击 Apply, 然后在弹出式对话框中点击 OK。
- 5. 安装完成后,点击 Finish,然后点击 OK。

浏览数:612 关注数:297

► Mac上很多破解软件 爱情守望者 (/blog/littleGG/article/154.html)

浏览数:602 关注数:297

▶ Android studio 2.2 2.3 2.4 如何使用 kotlin插件 (/blog/android/article/142.html)

浏览数:526 关注数:297

▶ 在Android studio3.0 用kotlin语言开发 Android app demo (/blog/android/article/149.html)

浏览数:471 关注数:297

▶ 在vue中获取dom元素 (/blog/web/article/307.html)

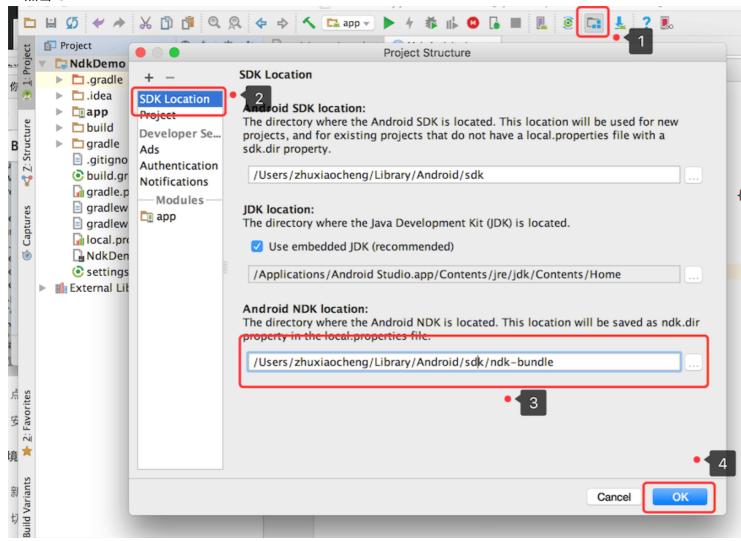
浏览数:385 关注数:297

SDK环境配置:

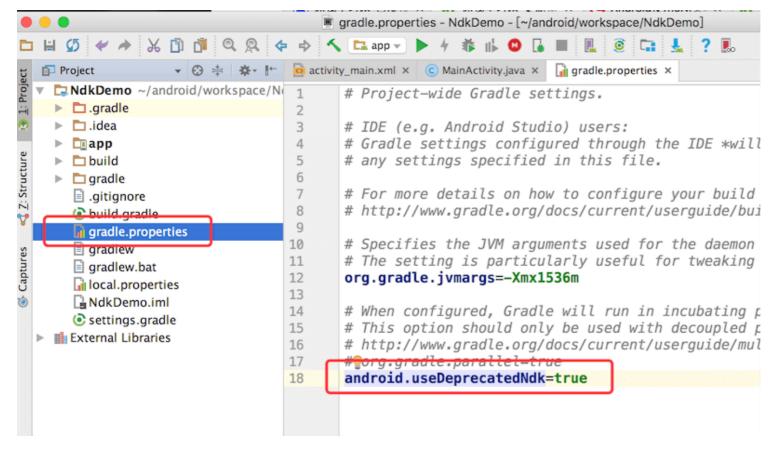
提前新建一个测试用的项目 NdkDemo

- 1. 切换到Project工程目录,打开Project Structure (方法一,直接在工具栏打开,如下图所示,方法二, 右键工程目录 -> Open Module Settings)
- 2. 选择左边栏 SDK Location
- 3. 在Android NDK Location 位置,选择 ndk安装包的路径,一般放在sdk目录下(下图是我mac系统放置ndk的路径,windows系统也类似,大家自己选择)

4. 点击 OK



5. 为最外层工程目录下的gradle.properties的文件末尾加上android.useDeprecatedNdk=true这段代码(如果没有这个目录,自己新建一个),如下图所示:



Ok,上面我们把环境给配置好了,下面我们就可以写Java和C代码了

2, Java代码和C代码的编写过程

1,首先新建一个java类JNIUtils.java,代码如下

```
public class JNIUtils {
    // 加载native-jni
    static {
        System.loadLibrary("native-jni");
    }
    //java调C中的方法都需要用native声明且方法名必须和c的方法名一样
    public native String stringFromJNI();
}
```

2,重新Make Project一下工程如下图2-1,完成后会在工程目录...

/NdkDemo/app/build/intermediates/classes/debug/com/niwoxuexi/ndkdemo 看到自己编译后的classes文件JNIUtils.class如下图2-2所示:

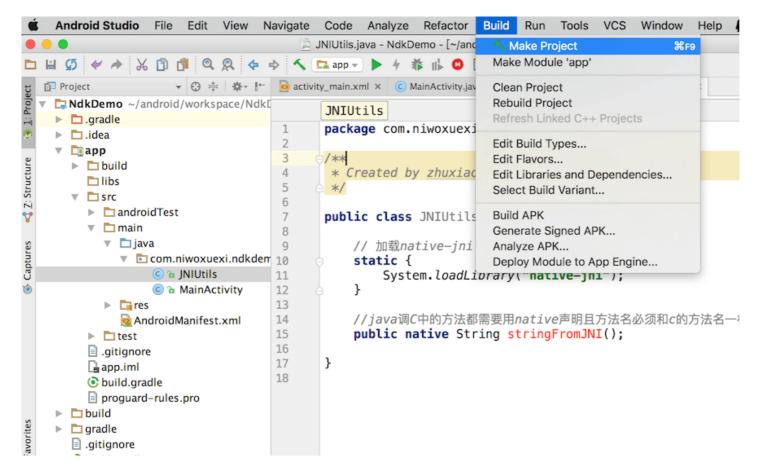


图:2-1

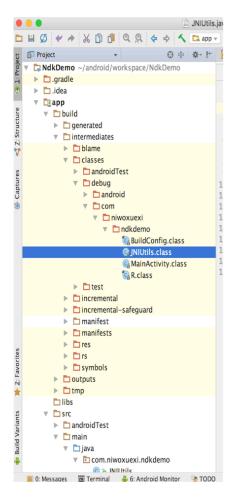


图:2-2

- 3,用javah工具生成头文件
 - 1) 首先新建一个java类JNIUtils.java,
 - 2) 代码在studio打开Terminal命令行工具,打开步骤是View->Tool Windows->Terminal (或者在下边的工具栏中直接打开或直接按Alt+F12)

在命令行中先进入到工程的main目录下

3) 输入命令: javah -d jni -classpath 自己编译后的class文件的绝对路径

例如:

javah -d jni -classpath /Users/zhuxiaocheng/android/workspace/NdkDemo/app/build/intermediates/classes/debug com.niwoxuexi.ndkkemo.JNIUtils

注意: 1, debug后的空格

2, windows 系统路径中的文件的分割线是 '\' 而不是mac系统的 '/'



4)按回车之后就会在main目录下生成jni文件夹,同时生成.h文件 如下图所示

```
h com_niwoxuexi_nakaemo_anioalis.n - nakbemo - [~/anaroia/workspace/nakbemo]
🗅 🗎 Ø 🐓 🥕 🔏 🖺 🐧 🔍 Q 💠 💠 🔨 📭 app v ▶ 4 🐞 🖟 🔼 🖟 🔲 👢 ⑧ 🗔 👢 🥍 🥷
                                  🕃 🛊 🗱 🏴 MainActivity.java × 📓 gradle.properties × 🌀 JNIUtils.java × 📵 native-lib.c × 📵 com_niwoxuexi_ndkaemo.
                                                      /* DO NOT EDIT THIS FILE - it is machine generated */
   gradle
                                                     #include <jni.h>
   idea idea
                                                     /* Header for class com_niwoxuexi_ndkdemo_JNIUtils */
   ▼ inapp
     build
                                                     #ifndef _Included_com_niwoxuexi_ndkdemo_JNIUtils
       🗀 libs
                                                     #define _Included_com_niwoxuexi_ndkdemo_JNIUtils
     ▼ □ src
                                                     #ifdef __cplusplus
extern "C" {
        androidTest
        main
                                                     #endif
          ▶ □ java
                                                     -/×
                                                                    com_niwoxuexi_ndkdemo_JNIUtils
                                                      * Class:
               ndkdemo JNIUtils.h
                                                      * Method: stringFromJNI
          res
                                                      * Signature: ()Ljava/lang/String;
            AndroidManifest.xml
                                             14
        ▶ ☐ test
                                             15 5 JNIEXPORT jstring JNICALL Java_com_niwoxuexi_ndkdemo_JNIUtils_stringFromJNI
        gitignore.
                                             16
                                                       (JNIEnv *, jobject);
                                             17
        app.iml
                                             18
                                                      #ifdef __cplusplus
        build.gradle
                                             19
        proguard-rules.pro
                                             20
                                                      #endif
   build
                                             21
                                                     #endif
   ▶ ☐ gradle
                                             22
     gitignore
     build.gradle
      gradle.properties
```

5) 现在我们在jni目录下新建一个 native-lib.c 的 c 文件,内容如下

6)会后在app的build.gradle配置文件中添加如下代码:

```
//ndk编译生成.so文件
ndk {
    moduleName "native-lib" //生成的so名字
    abiFilters "armeabi", "armeabi-v7a", "x86" //输出指定三种abi体系结构下的so库。
}
```

如图所示:

```
app - NdkDemo - [~/android/workspace/NdkDemo]
□ H Ø ♥ → H □ □ □ Q R ← → ヘ □ app > ▶ + # II □ □ ■ ■ ■ ◎ □ L 2 ? ■
                     🔻 🔀 🛊 🗱 - 👫 📴 activity_main.xml × (C) MainActivity.java × (C) app × 🔓 gradle.properties × (C) JNIUtils.java × (C) app ×
  Project
      Liapp
                                         apply plugin: 'com.android.application'
      build
        libs
                                        android {
      ▼ 🗀 src
                                             compileSdkVersion 26
        androidTest
                                             buildToolsVersion "25.0.3"
        main
                                             defaultConfig {
          java
                                                 applicationId "com.niwoxuexi.ndkdemo"
           ▼ 🛅 jni
                                                 minSdkVersion 14
               com_niwoxuexi_ndkder g
                                                 targetSdkVersion 26
               native-lib.c
                                                 versionCode 1
                                  10
           ▼ □ res
                                  11
                                                 versionName "1.0"
                                                 testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
               drawable
                                  12
                                  13
             layout
                                                 //ndk编译生成。so文件
                                  14
             mipmap-hdpi
                                  15
                                                 ndk {
             mipmap-mdpi
                                  16
                                                     moduleName "native-lib"
                                                                                     //生成的so名字
             mipmap-xhdpi
                                                     abiFilters "armeabi", "armeabi-v7a", "x86" //输出指定三种abi体系结构下的so库。
                                  17
             mipmap-xxhdpi
                                                 }
                                  18
             ▶ imipmap-xxxhdpi
                                  19
             values
                                  20
                                             buildTypes {
             AndroidManifest.xml
                                  21
                                                 release {
         ▶ test
                                                     minifyEnabled false
                                  22
         gitignore
                                                     proquardFiles getDefaultProquardFile('proquard-android.txt'), 'proquard-rul
                                  23
         app.iml
                                  24
         build.gradle
                                  25
         proguard-rules.pro
                                  26
    build
                                  27
  Android Monitor
                                                                                                                            - IÇE-
      Xiaomi MI NOTE LTE Android 6.0.1, API 23
                                          com,niwoxuexi,ndkdemo (4660)
```

7) 最后在我们来测试一下,只需要在MainActivity中调用一下C就可以了,代码如下:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView textView = (TextView) findViewById(R.id.text);
        textView.setText(new JNIUtils().stringFromJNI());
    }
}
```

8) 直接运行项目,结果如下所示:



ok, 没问题, 可以调用, 其实也没有想象中的那么难。

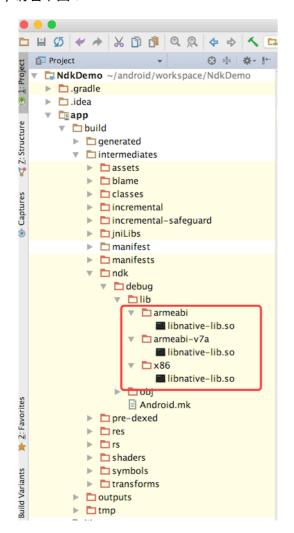
代码地址:https://image.niwoxuexi.com/blog/download/NdkDemo.zip

(https://image.niwoxuexi.com/blog/download/NdkDemo.zip)

三、调用编译过的 .so 库

上边编译完成了,有人会问:我要的是编译后的.so库,别人用的时候直接拿来用就可以了,那编译后的.so库在哪呢?不要着

急,请看下图:



根据这个路径就可以找到指定输出的三种体系结构下的.so库文件,然后把.so文件复制出来,如下图所示的放到相应libs的文件夹,

1,把复制的so包,放到项目的libs目录下

2, 在app module 下的buide.gradle 中添加下面代码:

```
//放在libs目录中
sourceSets {
    main {
        jniLibs.srcDirs = ['libs']
    }
}
```

如图所示:

```
□ ₩ Ø # A □ □ □ Q R + → < □ app - ▶ 4 # ₩ □ □ ■ ■ □ © □ ±
                                🔾 🛊 🔅 - 🏗 😼 activity_main.xml × (c) MainActivity.java × (c) app × 🔝 gradle.properties × (c) JNIUtils.java ×
    NdkDemo ~/android/workspace/NdkDemo
                                                   android {
    gradle
                                                        compileSdkVersion 26
    ▶ □ .idea
                                                        buildToolsVersion "25.0.3"
     ▼ inapp
                                             6
                                                        defaultConfig {
😋 Z: Structure
                                                            applicationId "com.niwoxuexi.ndkdemo"
       ► □ build
                                             8
                                                            minSdkVersion 14
       ▼ 🛅 libs
                                                            targetSdkVersion 26
                                             9
         versionCode 1
                                            10
              ■ libnative-lib.so
                                            11
                                                            versionName "1.0"

▼ armeabi-v7a

                                            12
                                                            testInstrumentationRunner "android.support.test.runne
Captures
              ■ libnative-lib.so
                                            13
                                                                                  注意把这个去掉
                                                            //ndk编译生成,so文件
                                            14
              ■ libnative-lib.so
                                            15
                                                            /*ndk {
       ▶ ☐ src
                                                                                                  //生成的so名字
                                            16
                                                                moduleName "native-lib"
         gitignore ...
                                                                abiFilters "armeabi", "armeabi-v7a", "x86" //輸
                                            17
         app.iml
                                            18
         build.gradle
                                            19
         proguard-rules.pro
                                                        //放在libs目录中
                                            20
     ▶ □ build
                                            21
                                                        sourceSets {
     gradle
                                            22
                                                            main {
       gitignore ...
                                            23
                                                                jniLibs.srcDirs = ['libs']
       build.gradle
                                            24
                                            25
       a gradle.properties
                                            26
                                                        buildTypes {
       gradlew
                                            27
                                                            release {
       gradlew.bat
                                                                minifyEnabled false
                                            28
       local.properties
                                                                proguardFiles getDefaultProguardFile('proguard-a)
                                            29
       NdkDemo.iml
                                            30
       @ settings.gradle
                                            31
    III External Libraries
                                            32
Android Monitor
```

四、总结

也没什么好总结的,直接按照上面步骤一步一步来,就可以了,jni调用过程有什么问题,欢迎留言!

最后是贴上代码地址:https://image.niwoxuexi.com/blog/download/NdkDemo%202.zip

(https://image.niwoxuexi.com/blog/download/NdkDemo%202.zip)

(https://image.niwoxuexi.com/blog/download/NdkDemo%202.zip)

标签: Android studio ndk ndk-build java调用c

上一篇: Android 中向您的项目中添加ndk代码(C和C++)--翻译(/blog/android/article/245.html)

下一篇: android studio cMake脚本添加ndk,构建C/C++程序(基础使用篇)(/blog/android/article/260.html)

本文地址: https://www.niwoxuexi.com/blog/android/article/246.html (/blog/android/article/246.html)

版权声明:本篇文章,如非注明,皆为博主 android (/blog/user/101.html) 原创。转载需注明本文链接并保证链接可用。



请先登录后评论...

6

全部评论: 1条



android00 1F

写得不错,按照你的步骤来,成功了

2017-09-02 03:20

已赞同 1

回复

Copyright © 2017 niwoxuexi.com (https://niwoxuexi.com) All Rights Reserved | 皖 ICP备 17011993号-1

[一] 阿里亞 (http://www.aliyun.com/?ref=3) 提供计算与安全服务