

# Android NDK开发扫盲及最新CMake的编译使用



Tsy远 (/u/21716b19302d) [+ 关注](#)

2017.07.06 10:46\* 字数 2971 阅读 11424 评论 19 喜欢 79 赞赏 2

(/u/21716b19302d)

本篇文章旨在简介 Android 中 NDK 是什么以及重点讲解最新 Android Studio 编译工具 CMake 的使用

## 1 NDK 简介

在介绍 NDK 之前还是首推 Android 官方 NDK 文档。传送门 (<https://link.jianshu.com?t=https://developer.android.com/ndk/guides/index.html>)

官方文档分别从以下几个方面介绍了 NDK

1. NDK 的基础概念
2. 如何编译 NDK 项目
3. ABI 是什么以及不同 CPU 指令集支持哪些 ABI
4. 如何使用您自己及其他预建的库

本节将会对文档进行总结和补充。所以建议先浏览一遍文档，或者看完本篇文章再回头看一遍文档。

### 1.1 NDK 基础概念

首先先用简单的话分别解释下 JNI、NDK，以及分别和 Android 开发、c/c++ 开发的配合。在解释过程中会对 Android.mk、Application.mk、ndk-build、CMake、CMakeList 这些常见名词进行扫盲。

**JNI** (Java Native Interface)：Java本地接口。是为了方便Java调用c、c++等本地代码所封装的一层接口（也是一个标准）。大家都知道，Java的优点是跨平台，但是作为优点的同时，其在本地交互的时候就编程了缺点。Java的跨平台特性导致其本地交互的能力不够强大，一些和操作系统相关的特性Java无法完成，于是Java提供了jni专门用于和本地代码交互，这样就增强了Java语言的本地交互能力。上述部分文字摘自任玉刚的 Java JNI 介绍 (<https://link.jianshu.com?t=http://blog.csdn.net/singwhatiwanna/article/details/9061545>)



**NDK** ( Native Development Kit ) : 原生开发工具包, 即帮助开发原生代码的一系列工具, 包括但不限于编译工具、一些公共库、开发IDE等。

NDK 工具包中提供了完整的一套将 C/C++ 代码编译成静态/动态库的工具, 而 `Android.mk` 和 `Application.mk` 你可以认为是描述编译参数和一些配置的文件。比如指定使用C++11还是C++14编译, 会引用哪些共享库, 并描述关系等, 还会指定编译的 `abi`。只有有了这些 NDK 中的编译工具才能准确的编译 C/C++ 代码。

`ndk-build` 文件是 Android NDK r4 中引入的一个 shell 脚本。其用途是调用正确的 NDK 构建脚本。其实最终还是会去调用 NDK 自己的编译工具。

那 CMake 又是什么呢。脱离 Android 开发来看, C/C++ 的编译文件在不同平台是不一样的。Unix 下会使用 `makefile` 文件编译, Windows 下会使用 `project` 文件编译。而 CMake 则是一个跨平台的编译工具, 它并不会直接编译出对象, 而是根据自定义的语言规则 ( `CMakeLists.txt` ) 生成 对应 `makefile` 或 `project` 文件, 然后再调用底层的编译。

在Android Studio 2.2 之后, 工具中增加了 CMake 的支持, 你可以这么认为, 在 Android Studio 2.2 之后你有2种选择来编译你写的 C/C++ 代码。一个是 `ndk-build` + `Android.mk` + `Application.mk` 组合, 另一个是 `CMake` + `CMakeLists.txt` 组合。这2个组合与Android 代码和C/C++代码无关, 只是不同的构建脚本和构建命令。本篇文章主要会描述后者的组合。(也是Android现在主推的)

## 1.2 ABI 是什么

ABI ( Application binary interface ) 应用程序二进制接口。不同的CPU 与指令集的每种组合都有定义的 ABI (应用程序二进制接口), 一段程序只有遵循这个接口规范才能在该 CPU 上运行, 所以同样的程序代码为了兼容多个不同的CPU, 需要为不同的 ABI 构建不同的库文件。当然对于CPU来说, 不同的架构并不意味着一定互不兼容。

- armeabi设备只兼容armeabi ;
- armeabi-v7a设备兼容armeabi-v7a、armeabi ;
- arm64-v8a设备兼容arm64-v8a、armeabi-v7a、armeabi ;
- X86设备兼容X86、armeabi ;
- X86\_64设备兼容X86\_64、X86、armeabi ;
- mips64设备兼容mips64、mips ;
- mips只兼容mips ;

具体的兼容问题可以参见这篇文章。Android SO文件的兼容和适配 (<https://link.jianshu.com?t=http://blog.coderclock.com/2017/05/07/android/Android-so-files-compatibility-and-adaptation/>)

当我们开发 Android 应用的时候, 由于 Java 代码运行在虚拟机上, 所以我们从来没有关心过这方面的问题。但是当我们开发或者使用原生代码时就需要了解不同 ABI 以及为自己的程序选择接入不同 ABI 的库。(库越多, 包越大, 所以要有选择)



下面我们来看下一共有哪些 ABI 以及对应的指令集

支持的 ABI

每个 ABI 支持一个或多个指令集。表 1 提供每个 ABI 支持的指令集概览。

表 1. ABI 和支持的指令集。

ABI	支持的指令集	说明
armeabi	<ul style="list-style-type: none"><li>ARMV5TE 和更高级版本</li><li>Thumb-1</li></ul>	无硬浮点。
armeabi-v7a	<ul style="list-style-type: none"><li>armeabi</li><li>Thumb-2</li><li>VFPv3-D16</li><li>其他 (可选)</li></ul>	与 ARMv5、v6 设备不兼容。
arm64-v8a	<ul style="list-style-type: none"><li>AArch-64</li></ul>	
x86	<ul style="list-style-type: none"><li>x86 (IA-32)</li><li>MMX</li><li>SSE/2/3</li><li>SSSE3</li></ul>	不支持 MOVBE 或 SSE4。
x86_64	<ul style="list-style-type: none"><li>x86-64</li><li>MMX</li><li>SSE/2/3</li><li>SSSE3</li><li>SSE4.1、4.2</li><li>POPCNT</li></ul>	
mips	<ul style="list-style-type: none"><li>MIPS32r1 及更高级版本</li></ul>	使用硬浮点。并且假设 CPU/FPU 频率比为 2:1 以获取最大兼容性。不提供 microMIPS 或 MIPS16。
mips64	<ul style="list-style-type: none"><li>MIPS64r1</li></ul>	

ABI

2 CMake 的使用

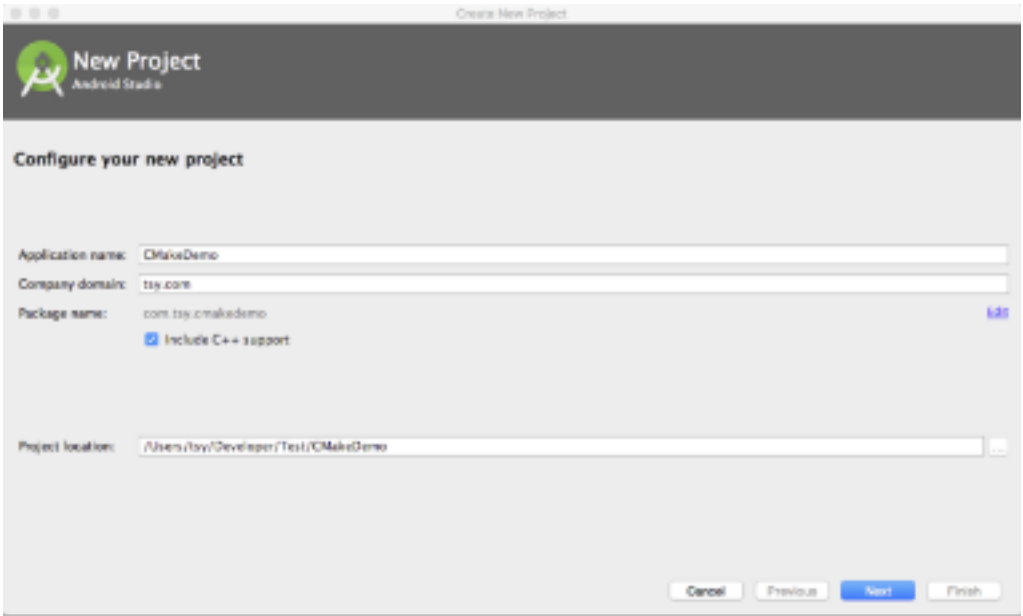
这一节将重点介绍 CMake 的规则和使用，以及如何使用 CMake 编译自己及其他预建的库。

2.1 Hello world

我们通过一个Hello World项目来理解 CMake

首先创建一个新的包含原生代码的项目。在 New Project 时，勾选 Include C++ support

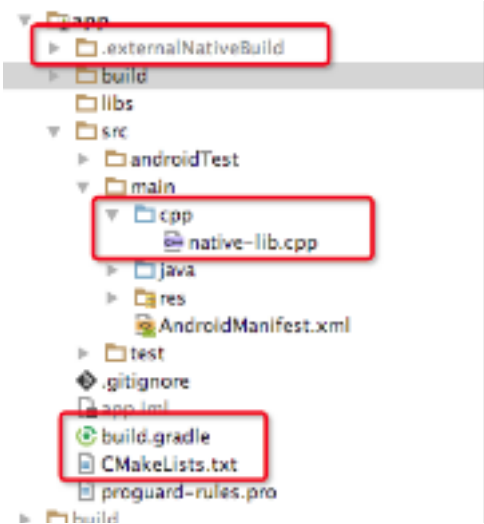




New Project

项目创建好以后我们可以看到和普通Android项目有以下4个不同。

- 1. main 下面增加了 cpp 目录，即放置 c/c++ 代码的地方
- 2. module-level 的 build.gradle 有修改
- 3. 增加了 CMakeLists.txt 文件
- 4. 多了一个 .externalNativeBuild 目录



Difference

build.gradle



```
android {  
    ...  
    defaultConfig {  
        ...  
        externalNativeBuild {  
            cmake {  
                cppFlags "-frtti -fexceptions"  
                arguments "-DANDROID_ARM_NEON=TRUE"  
            }  
        }  
    }  
    buildTypes {  
        ...  
    }  
    externalNativeBuild {  
        cmake {  
            path "CMakeLists.txt"  
        }  
    }  
}
```

由于 CMake 的命令集成在了 gradle - externalNativeBuild 中，所以在 gradle 中有2个地方配置 CMake。

defaultConfig 外面的 externalNativeBuild - cmake，指明了 CMakeList.txt 的路径；  
defaultConfig 里面的 externalNativeBuild - cmake，主要填写 CMake 的命令参数。即由 arguments 中的参数最后转化成一个可执行的 CMake 的命令，可以在 .externalNativeBuild/cmake/debug/{abi}/cmake\_build\_command.txt 中查到。如下



```
1 Executable : /Users/tsty/Library/Android/sdk/cmake/3.6.3155568/bin/cmake  
2 arguments :  
3 -R/Users/tsty/Developer/Test/CMakeDemo/app  
4 -R/Users/tsty/Developer/Test/CMakeDemo/app/.externalNativeBuild/cmake/debug/armeabi  
5 -GAndroid Gradle - Ninja  
6 -DANDROID_ABI=armeabi  
7 -DANDROID_NDK=/Users/tsty/Library/Android/sdk/ndk-bundle  
8 -DANDROID_LIBRARY_OUTPUT_DIRECTORY=/Users/tsty/Developer/Test/CMakeDemo/app/build/intermediates/cmake/debug/obj/armeabi  
9 -DANDROID_PLATFORM=android-15  
10 -DANDROID_TOOLCHAIN_FILE=/Users/tsty/Library/Android/sdk/ndk-bundle/build/cmake/android.toolchain.cmake  
11 -DANDROID_PLATFORM=android-15  
12 -DANDROID_PLATFORM=android-15  
13 -DANDROID_PLATFORM=android-15  
14 -DANDROID_PLATFORM=android-15  
15 -DANDROID_PLATFORM=android-15  
16
```

cmake command

更多的可以填写的命令参数和含义可以参见Android NDK-CMake文档 (<https://link.jianshu.com?t=https://developer.android.com/ndk/guides/cmake.html>)

## CMakeLists.txt

CMakeLists.txt 中主要定义了哪些文件需要编译，以及和其他库的关系等。

看下新项目中的 CMakeLists.txt



```
cmake_minimum_required(VERSION 3.4.1)

# 编译出一个动态库 native-lib, 源文件只有 src/main/cpp/native-lib.cpp
add_library( # Sets the name of the library.
    native-lib
    # Sets the library as a shared library.
    SHARED
    # Provides a relative path to your source file(s).
    src/main/cpp/native-lib.cpp )

# 找到预编译库 log-lib 并link到我们的动态库 native-lib中
find_library( # Sets the name of the path variable.
    log-lib
    # Specifies the name of the NDK library that
    # you want CMake to locate.
    log )

target_link_libraries( # Specifies the target library.
    native-lib
    # Links the target library to the log library
    # included in the NDK.
    ${log-lib} )
```

这其实是一个最基本的 CMakeLists.txt , 其实 CMakeLists.txt 里面可以非常强大, 比如自定义命令、查找文件、头文件包含、设置变量等等。建议结合 CMake 的官方文档 (<https://link.jianshu.com?t=https://cmake.org/documentation/>)使用。同时在这推荐一个中文翻译的简易的CMake手册 (<https://link.jianshu.com?t=https://www.zybuluo.com/khan-lau/note/254724>)

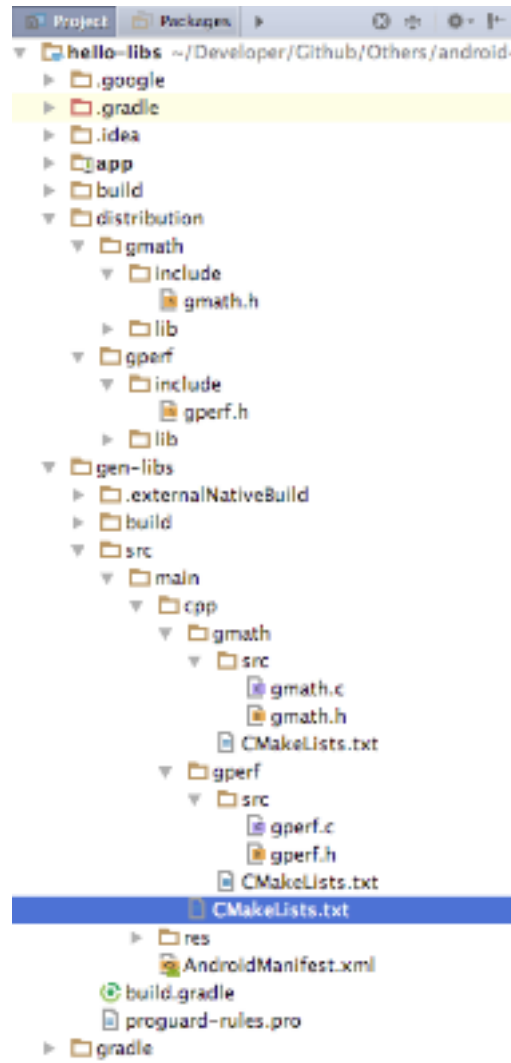
## 2.2 CMake 使用自己及其他预建的库

当你需要引入已有的静态库/动态库 (FFmpeg) 或者自己编译核心部分并提供出去时就需要考虑如何在 CMake 中使用自己及其他预建的库。

Android NDK 官网的使用现有库 (<https://link.jianshu.com?t=https://developer.android.com/ndk/guides/libs.html>)的文档中还是使用 ndk-build + Android.mk + Application.mk 组合的说明文档。(其实官方文档中大部分都是的, 并没有使用 CMake)

幸运的是, Github上的官方示例 (<https://link.jianshu.com?t=https://github.com/googleamples/android-ndk>) 里面有个项目 hello-libs (<https://link.jianshu.com?t=https://github.com/googleamples/android-ndk/tree/master/hello-libs>) 实现了如何创建出静态库/动态库, 并引用它。现在我们把代码拉下来看下具体是如何实现的。





hello-libs

我们先看下Github上的README介绍：

- app - 从 `$project/distribution/` 中使用一个静态库和一个动态库
- gen-libs - 生成一个动态库和一个静态库并复制到 `$project/distribution/` 目录，你不需要再编译这个库，二进制文件已经保存在了项目中。当然，如果有需要你也可以编译自己的源码，只需要去掉 `setting.gradle` 和 `app/build.gradle` 中的注释，然后执行一次，接着注释回去，防止在 `build` 的过程中不受影响。

我们采用自底向上的方式分析模块，先看下 `gen-libs` 模块。

**gen-libs/build.gradle**



```
android {
    ...
    defaultConfig {
        ...
        externalNativeBuild {
            cmake {
                arguments '-DANDROID_PLATFORM=android-9',
                        '-DANDROID_TOOLCHAIN=clang'
                // explicitly build libs
                targets 'gmath', 'gperf'
            }
        }
    }
    ...
}
```

查询文档 (<https://link.jianshu.com?t=https://developer.android.com/ndk/guides/cmake.html>)可以知道 arguments 中 -DANDROID\_PLATFORM 代表编译的 android 平台，文档建议直接设置 minSdkVersion 就行了，所以这个参数可忽略。另一个参数 -DANDROID\_TOOLCHAIN=clang，CMake 一共有2种编译工具链 - clang 和 gcc，gcc 已经废弃，clang 是默认的。

targets 'gmath', 'gperf' 代表编译哪些项目。（不填就是都编译）

### cpp/CMakeLists.txt

```
cmake_minimum_required(VERSION 3.4.1)

set(CMAKE_VERBOSE_MAKEFILE on)

set(lib_src_DIR ${CMAKE_CURRENT_SOURCE_DIR})

set(lib_build_DIR $ENV{HOME}/tmp)
file(MAKE_DIRECTORY ${lib_build_DIR})

add_subdirectory(${lib_src_DIR}/gmath ${lib_build_DIR}/gmath)
add_subdirectory(${lib_src_DIR}/gperf ${lib_build_DIR}/gperf)
```

外层的 CMakeLists 里面核心就是 add\_subdirectory，查询CMake 官方文档 (<https://link.jianshu.com?t=https://cmake.org/documentation/>) 可以知道这条命令的作用是为构建添加一个子路径。子路径中的 CMakeLists.txt 也会被执行。即会去分别执行 gmath 和 gperf 中的 CMakeLists.txt

### cpp/gmath/CMakeLists.txt





```
cmake_minimum_required(VERSION 3.4.1)

set(CMAKE_VERBOSE_MAKEFILE on)

add_library(gmath STATIC src/gmath.c)

# copy out the lib binary... need to leave the static lib around to pass gradle check
set(distribution_DIR ${CMAKE_CURRENT_SOURCE_DIR}/../../../../../distribution)
set_target_properties(gmath
    PROPERTIES
    ARCHIVE_OUTPUT_DIRECTORY
    "${distribution_DIR}/gmath/lib/${ANDROID_ABI}")

# copy out lib header file...
add_custom_command(TARGET gmath POST_BUILD
    COMMAND "${CMAKE_COMMAND}" -E
    copy "${CMAKE_CURRENT_SOURCE_DIR}/src/gmath.h"
    "${distribution_DIR}/gmath/include/gmath.h"
    # **** the following 2 lines are for potential future debug purposes
    # COMMAND "${CMAKE_COMMAND}" -E
    # remove_directory "${CMAKE_CURRENT_BINARY_DIR}"
    COMMENT "Copying gmath to output directory")
```

这个是其中一个静态库的 `CMakeLists.txt`，另一个跟他很像。只是把 `STATIC` 改成了 `SHARED`（动态库）。

`add_library(gmath STATIC src/gmath.c)` 之前用到过，编译出一个静态库，源文件是 `src/gmath.c`

`set_target_properties` 命令的意思是设置目标的一些属性来改变它们构建的方式。这个命令中设置了 `gmath` 的 `ARCHIVE_OUTPUT_DIRECTORY` 属性。也就是改变了输出路径。

`add_custom_command` 命令是自定义命令。命令中把头文件也复制到了 `distribution_DIR` 中。

以上就是一个静态库/动态库的编译过程。总结以下3点

1. 编译静态库/动态库
2. 修改输出路径
3. 复制暴露的头文件

接着，我们看下 `app` 模块是如何使用预建好的静态库/动态库的。

**app/src/main/cpp/CMakeLists.txt**



```
cmake_minimum_required(VERSION 3.4.1)

# configure import libs
set(distribution_DIR ${CMAKE_SOURCE_DIR}/../../../../distribution)

# 创建一个静态库 lib_gmath 直接引用libgmath.a
add_library(lib_gmath STATIC IMPORTED)
set_target_properties(lib_gmath PROPERTIES IMPORTED_LOCATION
    ${distribution_DIR}/gmath/lib/${ANDROID_ABI}/libgmath.a)

# 创建一个动态库 lib_gperf 直接引用libgperf.so
add_library(lib_gperf SHARED IMPORTED)
set_target_properties(lib_gperf PROPERTIES IMPORTED_LOCATION
    ${distribution_DIR}/gperf/lib/${ANDROID_ABI}/libgperf.so)

# build application's shared lib
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=gnu++11")

# 创建库 hello-libs
add_library(hello-libs SHARED
    hello-libs.cpp)

# 加入头文件
target_include_directories(hello-libs PRIVATE
    ${distribution_DIR}/gmath/include
    ${distribution_DIR}/gperf/include)

# hello-libs库链接上 lib_gmath 和 lib_gperf
target_link_libraries(hello-libs
    android
    lib_gmath
    lib_gperf
    log)
```

我将解释放在了注释中。可以看下基本上分成了4个步骤引入：

1. 分别创建静态库/动态库，直接引用已经有的 .a 文件 或者 .so 文件
2. 创建自己应用的库 hello-libs
3. 加入之前暴露头文件
4. 链接上静态库/动态库

还是很好理解的。编辑好并 Sync 后，你就可以发现 hello-libs 中的C/C++代码可以引用暴露的头文件调用内部方法了。

### 3 资料文献

首推 Android NDK 官方文档 (<https://link.jianshu.com?t=https://developer.android.com/ndk/guides/index.html>)，虽然很多都不完整，但是绝对是必须看一遍的东西。

当初次接触 NDK 开发又觉得新建的 Hello World 项目过于简单时。建议把 googlesamples - android-ndk (<https://link.jianshu.com?t=https://github.com/googlesamples/android-ndk>) 项目拉下来。里面有多个实例参考，比官方文档完整很多。





Google Samples

当你发现示例里的一些NDK配置满足不了你的需求后，你就需要到 CMake 官方文档 (<https://link.jianshu.com?t=https://cmake.org/documentation/>) 去查询完整的支持的函数，同时这里也提供一个中文翻译的简易的CMake手册 (<https://link.jianshu.com?t=https://www.zybuluo.com/khan-lau/note/254724>)。

以上文档资料仅为了解决 NDK 开发过程中编译配置问题，具体 c/c++ 的逻辑编写、jni等不在此范畴。

### 彩蛋

文末献上一组彩蛋，将 CMake 或者 NDK 开发过程中遇到的坑和小技巧以 Q&A 的方式列出。持续更新



**Q1：怎么指定 C++标准？**

A：在 build.gradle 中，配置 cppFlags -std

```
externalNativeBuild {
    cmake {
        cppFlags "-frtti -fexceptions -std=c++14"
        arguments '-DANDROID_STL=c++_shared'
    }
}
```

## Q2：add\_library 如何编译一个目录中所有源文件？

A：使用 aux\_source\_directory 方法将路径列表全部放到一个变量中。

```
# 查找所有源码 并拼接到路径列表
aux_source_directory(${CMAKE_HOME_DIRECTORY}/src/api SRC_LIST)
aux_source_directory(${CMAKE_HOME_DIRECTORY}/src/core CORE_SRC_LIST)
list(APPEND SRC_LIST ${CORE_SRC_LIST})
add_library(native-lib SHARED ${SRC_LIST})
```

## Q3：怎么调试 CMakeLists.txt 中的代码？

A：使用 message 方法

```
cmake_minimum_required(VERSION 3.4.1)
message(STATUS "execute CMakeLists")
...
```

然后运行后在 .externalNativeBuild/cmake/debug/{abi}/cmake\_build\_output.txt 中查看 log。

## Q4：什么时候 CMakeLists.txt 里面会执行？

A：测试了下，好像在 sync 的时候会执行。执行一次后会生成 makefile 的文件缓存之类的东西放在 externalNativeBuild 中。所以如果 CMakeLists.txt 中没有修改的话再次同步好像是会重新执行的。（或者删除 .externalNativeBuild 目录）

真正编译的时候好像只是读取 .externalNativeBuild 目录中已经解析好的 makefile 去编译。不会再去执行 CMakeLists.txt

小礼物走一走，来简书关注我

赞赏支持



(/u/c0141a80040e962)







volatile unsigned val = gpower(exp);


I could see the return value correctly in the debugger. let me know what you find, I would try to replicate and see what the problem is.

feel free to file bug on [github.com/google/android-ndk](https://github.com/google/android-ndk)

2017.08.03 00:55  回复

mrzxm (/u/0ccc08ccec0b) : @guanghuafan (/users/8fd3fede7b0d) 看不懂英文呀😓好累呀，还是要谢谢你的回复，问题我已经解决了，是我的系统配置问题，看来盗版的系统不能乱用。😓

2017.08.04 17:08  回复

 添加新评论



hywenbinger (/u/55350d205fb6)

4楼 · 2017.08.02 09:04

(/u/55350d205fb6)  
学习了

赞  回复




makings (/u/f8e84fa44ecd)

5楼 · 2017.08.02 10:34


(/u/f8e84fa44ecd)  
我觉得我是个文盲

赞  回复


Tsyj远 (/u/21716b19302d) : 为嘛😓


2017.08.02 11:14  回复

makings (/u/f8e84fa44ecd) : @Tsyj远 (/users/21716b19302d) 我一直用java开发，不动c/c++说实话我一点都看不懂

2017.08.02 11:28  回复

Tsyj远 (/u/21716b19302d) : 以后有需要可以再回头看

2017.08.02 11:45  回复

 添加新评论



guanghuafan (/u/8fd3fede7b0d)

6楼 · 2017.08.03 00:36

(/u/8fd3fede7b0d)  
CMake's Android Studio port generates ninja script to build. Great documents, thanks.

赞  回复



最最最醉人 (/u/9398e088c29c)

7楼 · 2017.08.03 09:35

(/u/9398e088c29c)

请教一个问题，我设置了-DANDROID\_PLATFORM=android-16，但是引用的头文件，总是指向ndk-bundle-sysroot-user-include下的，我想要指向ndk-bundle-platforms-android16下的是还要做什么设置吗？

赞 回复



qluojieq (/u/45bedd902c76)

8楼 · 2017.08.04 21:41

(/u/45bedd902c76)  
最近弄Opencv，就环境都把我弄得晕头转向的，很有帮助，

赞 回复

烧烤摊前卖烧烤 (/u/439cbd149387)：通过jni的方式就不用装opencv manager了是吗？

2018.02.27 17:20 回复

添加新评论



风影\_638f (/u/c0141d800403)

9楼 · 2017.08.16 16:22

(/u/c0141d800403)  
营养价值很高

赞 回复



sp82 (/u/fdd25815d442)

10楼 · 2017.09.01 11:28

(/u/fdd25815d442)  
谢谢分享，请问cmake如何支持中文目录？

赞 回复



RaymondMonki (/u/204439967771)

11楼 · 2017.09.11 09:30

(/u/204439967771)  
mark

赞 回复





Android知识 (/c/3fde3b545a35?utm\_source=desktop&utm\_medium=notes-included-collection)



Android... (/c/5139d555c94d?utm\_source=desktop&utm\_medium=notes-included-collection)



Android开发 (/c/d1591c322c89?utm\_source=desktop&utm\_medium=notes-included-collection)



Android... (/c/58b4c20abf2f?utm\_source=desktop&utm\_medium=notes-included-collection)



程序员 (/c/NEt52a?utm\_source=desktop&utm\_medium=notes-included-collection)



首页投稿 (/c/bDHhpK?utm\_source=desktop&utm\_medium=notes-included-collection)



Android... (/c/965715ea037b?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多 ▾

推荐阅读

更多精彩内容 > (/)

### Android开源项目-ASimpleCache (/p/25c107ed7348?utm\_campaign=ma...

ASimpleCache 是一个封装好的本地持久化存储的开源缓存框架 1 相关文档 Github地址：<https://github.com/yanfuhai/ASimpleCache> 2 封装了哪些常用类型的缓存 String类型 Bitmap JSONArray JSONObject Serializable 使用示例：Serializable的使用：Serializable适合于在使用Gson实例化实体类的适合保存自定义的数据类型。但是该实体类需要implements Serializable 接口。 例如：实体类 如何使用： 3 总结 ASimpleCache非常小，只有一个...

Tsy远 (/u/21716b19302d?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

### Android开源项目-Retrofit+Okhttp+Gson二次封装 (/p/283d1a7a0aff?utm...

Android网络请求开源项目组合：Retrofit+Okhttp+Gson，并二次封装，简化开发时的调用。 1 Retrofit+Okhttp+Gson是什么 Retrofit和Okhttp都是square公司开源的网络请求开源项目，也是当前最流行的网络请求组合。Gson可以将json反序列化为相应的数据model，相对于从json取数据，数据model的建立对于获取字段数据和了解接口定义更为清晰。 2 为什么选择Retrofit+Okhttp+Gson 对于网络框架的选择有volley，android-async-http等很多很多，这些其实都是基于android本身httpurl...

Tsy远 (/u/21716b19302d?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

我才二十多岁，就怕再也遇不到真爱了 (/p/d9cce2716...

(/p/d9cce2716bc2?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)



2018年1月27日 周六 大雪 文 # 阿呗 你可以不在乎我的感受，也可以不在乎我的热情，甚至不用理会我的沮丧与难过。但你要知道，每个人的付出都是有限的，我输的很惨，堕的也很惨，所以我不要了，都不要了。期末考完这两天，我躲在寝室里刷剧，想把这整整一个月，缺失的空闲都熬回来。同寝的人都踏上了回家的旅程，夜里就独剩我一人，北方的冬天，学校的夜，狂风和大雪，黑漆漆的，寂静里带着麻木，其余再无他物。而也在这个夜，我看到了这句话，也是这一刻，我突然觉得。不靠谱的从来都不是感情，是人。脑海里飘过曾印象深刻的一句话。我伪装成一个骗子，后来所有人都说我是个骗子，我阔绰，人人都以为我是...

阿呗有个帅阿呗 (/u/bfb1aa483a03?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

utm\_medium=pc\_all\_hots&utm\_source=recommendation)

## 大学生的我，是如何在旅行中省钱和赚钱的？ (/p/0f33...

经常会有人问我：“你一个大学生，去那么多地方，旅费都是家人给的吗？”其实每次出行，我都会给自己列一个表，这次旅行大概要花多少钱，有哪些是一定要花的，哪些是可以不花的。今天我写了一篇如何在旅行中省钱以及赚钱的文章，也是这些方法，让我走了云南、四川、贵州、浙江、西藏等十几个省市。我们出门旅行的开销无非就是：“交通、饮食、住宿、门票、购买纪念品”机票现在预定机票非常地方便，大家不再使用以前的打电话预定机票或者找旅行团预定机票，都是网上自己搞定的，但是不同的app，不同的网站价格又不太一样，究竟怎样才更划算呢。我们经常使用的来预定机票的无非就这几个app，再者就是各个航空公司的官网。 ...

(/p/0f33dd3347e6?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

流浪摄影师 (/u/df7a9d2e01e6?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)


## 茅侃侃自杀了，有人说他“不成熟” (/p/c7ed8150230e?utm\_campaign=ma...

曾经被视为80后创业先行者的茅侃侃去世了。根据媒体报道，因为创业失败，他选择了自杀。于是评论者们又出来了。我看到两种不同风格的评论：1) 死者为大，不要对人评头论足。2) 创业本身就有巨大风险，因为创业失败就去死，只能是因为他不成熟。1、首先，死者为大，并不客观，并不合理。我们都会死的，先死先大么？作为当事人的亲友，不希望听到不敬的评价，这合人情，却并不现实。任何一个人，生于世，逝于世，从头到尾都会被周围的人们包围着渡过，死后也无法避免。已成为历史的人和事，发挥最大价值的办法是变成一面镜子，让活着的人从中看见自己、看清自己、看到自己可能走或不可能走的路，并且更坚定自己的选择。 ...

行者慎思 (/u/7624e38a895a?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

## JNI：为项目添加C和C++代码库 (/p/6b31c37e9fdc?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

developer.android.com/studio/projects/add-native-code.html 本文内容 下载 NDK 和构建工具 创建支持 C/C++ 的新项目 构建和运行示例应用 向现有项目添加 C/C++ 代码 创建新的原生源文件 创建 CMake...

 四季明 (/u/39ff43a96104?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

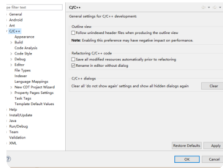



(/p/cb2f7e2da29f?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

utm\_medium=seo\_notes&utm\_source=recommendation)

Android游戏开发实践(1)之NDK与JNI开发02 (/p/cb2f...


Android游戏开发实践(1)之NDK与JNI开发02 承接上篇Android游戏开发实践(1)之NDK与JNI开发01分享完JNI的基础和简要开发流程之后，再来分享下在Android环境下的JNI的开发，以及涉及到的NDK相关的操作。当然，本篇仍是以Eclipse作为开...



 AlphaGL (/u/4cc00e2af10b?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


Android NDK 开发：CMake 使用 (/p/c71ec5d63f0d?utm\_campaign=mal...

注：首发地址 1. 前言 当在做 Android NDK 开发时，如果不熟悉用 CMake 来构建，读不懂 CMakeLists.txt 的配置脚本，很容易就会踩坑，遇到编译失败，一个很小的配置问题都会浪费很多时间。所谓工欲善其事必先利其器，学习 NDK 开发还是要大致了解...

 cfanr (/u/134f7552aa7e?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


Android - 收藏集 - 掘金 (/p/5ad013eb5364?utm\_campaign=maleskine&...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金Cover 有什么料？从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金相信有很多朋友...

 掘金官方 (/u/5fc9b6410f4f?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


我真的适合当教师吗？ (/p/09bc2c7e0156?utm\_campaign=maleskine&ut...

我是一名教师，越来越感觉自己不适合当教师了。我理想中的老师是博学的，有威严的。可，我这个人没有建立起一套给人威慑力的感觉，(自己找的借口是太年轻了，脾气不温不火的)。在学生班级管理上一头雾水，有时无从下手，固然学校的整体风气是很差的，抱着在其位谋其职的终旨，努力想教好一堂课...

 梦里梦外 (/u/2f66d3a6f1bc?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

同课异构展风采，教学相长促提升 (/p/9acf203fde6a?utm\_campaign=mal...

10月26日，我有幸跟随本部、领航语文老师一起赴新乡参加同课异构观课活动。这次活动由河南省初专委主办，不同地市多所学校一起参与，来自不同学校的五位教师同课异构，各展风采，省语文教研室教研员丁亚宏老师对本次赛课进行了精彩而深刻的点评。回来反思，我也收获颇多。特撰小文以...

 严峰 (/u/5dac7434119b?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/9b3a0fef3371?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

六岁小豪的成长痛 (/p/9b3a0fef3371?utm\_campaign=maleskine&utm\_co...



今天期中考试，小豪是哥哥、小月是妹妹都在我们班，下午试卷成绩出来。小豪考87，小月100.作为老师我边发试卷边读了分数，当读到小豪时，他小声说老师我有一题不会，我也小声给他说：不要紧，待会老师教你。…… 尽管这样小豪还是不出声的哭了。边流泪边擦，擦完了，眼泪又出来了，就这样...




 一株紫丁香 (/u/9f919929a86f?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/319bbf73a968?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



**2017.11.23 Angel 感恩奇迹分享 (/p/319bbf73a968?ut...**

奇迹1.【轻而易举的富足】第三册 第二十三天第八十三课 巴阿，我全然的信任你 我在这里只是为你服务，对我来说，今天是属于巴阿的一天.《梦境还是现实》《沉重的身体》在肉体里是和兄弟姐妹们后天的媒介 离开身体，是体验真正自由与爱的途径 今日功课：整天都跟巴阿保持连接...


 桉岐公主 (/u/2800c78a7dc0?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/aa55ddc17a45?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



**零爱你 (/p/aa55ddc17a45?utm\_campaign=maleskin...**

我认真地行走在 绝望的边缘上 你却站在原点 冷眼旁观着我的流离 有时停留在你的肩膀上 可是永远都不能依靠 所以 我倔强地学会了成长 祝福 是我送给你 最真挚的蓝天白云 我想 到了以后 你一定会感谢那个 不帮你逃避 不劝你放弃 不笑你笨拙 不留你一人 独自面对屋前蛮荒的 梦...

 柒忆 (/u/3ec0fb5e4f6d?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

