

# Xu' Blog

mobile game developer interest in haXe,nme,cocos2d-x.

5月 20 2013 • ANDROID ► NDK

## [原创]编写Android.mk中的LOCAL\_SRC\_FILES的终极技巧

已废弃, 请参考[Update: Android.mk 中的 LOCAL\\_SRC\\_FILES, LOCAL\\_C\\_INCLUDES](#)

### 问题的引入

在使用NDK编译C/C++项目的过程中,免不了要编写Android.mk文件,其中最重要的就是 LOCAL\_SRC\_FILES 源文件列表.

考虑有如下源文件分布的情况:

cpp文件全部位于android项目下的jni文件夹下,结构如下

```
jni
|---1.cpp
|---2.cpp
|---Android.mk
|---Application.mk
|---ndk_test.cpp
|---src
|   |---core
|   |   |---core1.cpp
```

```
| | | ---core2.cpp
| | ---src1.cpp
| | ---src2.cpp
```

按照通常的写法,在 `android.mk` 中,应该写入

```
LOCAL_SRC_FILES := ndk_test.cpp \
                    1.cpp \
                    2.cpp \
                    src/src1.cpp \
                    src/src2.cpp \
                    src/core/core1.cpp \
                    src/core/core2.cpp
```

繁琐不堪!

初步解法:一句话引入单个目录(不包括子目录)下的所有**cpp**源文件

继续上面的情况为例,我可以这样写

```
MY_CPP_LIST := $(wildcard $(LOCAL_PATH)/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/src/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/src/core/*.cpp)

LOCAL_SRC_FILES := $(MY_CPP_LIST:$(LOCAL_PATH)/%=%)
```

问题解决.

简单解释一下上面的几句话

1. `MY_CPP_LIST := $(wildcard $(LOCAL_PATH)/*.cpp)` ,这句话的意思是使用`wildcard`函数获取 `$(LOCAL_PATH)`目录也就是jni目录下的所有后缀名为cpp的文件,并把结果放到变量 `MY_CPP_LIST` 里.我们知道`$(LOCAL_PATH)`指的是当前 `Android.mk` 文件所在目录,所以通过这句话, `MY_CPP_LIST` 中的值应该是 `jni/1.cpp jni/2.cpp jni/ndk_test.cpp` .
2. `MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/src/*.cpp)` , 获取 `jni/src` 目录下的源文件,并追加到变量 `MY_CPP_LIST` 里
3. `MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/src/core/*.cpp)` ,同上,获取 `jni/src/core` 目录下的源文件
4. 通过以上几步,得到 `MY_CPP_LIST` 中内容是 `jni/1.cpp jni/2.cpp jni/ndk_test.cpp jni/src/src1.cpp jni/src/src2.cpp jni/src/core/core1.cpp jni/src/core/core2.cpp`

5. `LOCAL_SRC_FILES := $(MY_CPP_LIST:$(LOCAL_PATH)/%=%)` ,前面我们获取的文件都是以 `jni` 开头的,而真正编译所需要的文件都应该是直接从 `jni` 目录开始的,所以我们使用 模式替换 把所有文件名前面的 `jni/` 去掉.

这里我解释一下 `$(MY_CPP_LIST:$(LOCAL_PATH)/%=%)` 的语法含义,它的意思是对 `MY_CPP_LIST` 中每一项,应用冒号后面的规则,规则是什么呢?规则是 `$(LOCAL_PATH)/%=%` ,意思是,查找所有 `$(LOCAL_PATH)/` 开头的项,并截取后面部分

最后一句话也可以使用`subst`函数写成:

```
#替换每一项中的 "$(LOCAL_PATH)/" 为 ""(空)
LOCAL_SRC_FILES := $(subst $(LOCAL_PATH)/, , $(MY_CPP_LIST))
```

或使用`patsubst`函数写成

```
#同模式替换,这里使用patsubst函数
LOCAL_SRC_FILES := $(patsubst $(LOCAL_PATH)/%, %, $(MY_CPP_LIST))
```

具体语法请参考:[Functions for String Substitution and Analysis](#)

实际使用中,可以把代码放在`jni`目录以外的目录里,这时只要修改`wildcard`函数里的相对路径就可以了,甚至也可以使用绝对路径,只要你愿意.

以上代码已经足以应付大多数情况了,不过人的懒惰是无极限的,像上面的情况我的所有源文件都在`jni`目录下,为什么还要把每个子目录都写一行呢,不太优雅呀,最好能写一句话把`jni`目录下的所有源文件都引入.

进阶:引入单个目录(包括子目录)下的所有`cpp`源文件

为了达到引入目录下的所有源文件,包括子目录这个目标,我在 `android.mk` 中这样写

```
#声明一个变量MY_CPP_PATH表示源码目录
MY_CPP_PATH := $(LOCAL_PATH)/

#获取目录下的所有文件
My_All_Files := $(shell find $(MY_CPP_PATH)/.)
My_All_Files := $(My_All_Files:$(MY_CPP_PATH)/./*=$(MY_CPP_PATH)%)

#从My_All_Files中再次提取所有的cpp文件,这里也可以使用filter函数
MY_CPP_LIST := $(foreach c_file,$(My_All_Files), $(wildcard $(c_file)/*.cpp) )
MY_CPP_LIST := $(MY_CPP_LIST:$(LOCAL_PATH)/%=%)
```

```
LOCAL_SRC_FILES := $(MY_CPP_LIST)
```

通过以上几行,成功得到了jni目录包含它的子目录下的所有cpp源文件,并正确编译.实际使用中,代码不一定存放在jni目录下,修改 MY\_CPP\_PATH 就可以了,注意: MY\_CPP\_PATH 最好使用以 \$(LOCAL\_PATH) 开头的相对目录

这种写法极大的方便了项目的开发,以前在源码目录下新建cpp源文件,新建目录都不需要再来修改 android.mk 文件了.

还有一个问题,上面代码里只是引入cpp文件,如果源码文件夹下还有c文件呢,怎么办?再多写几行?

## 进阶2.0:引入单个目录(包括子目录)下的所有\*.cpp和\*.c源文件

这里,我直接给出代码

```
MY_CPP_PATH := $(LOCAL_PATH)/
My_All_Files := $(shell find $(MY_CPP_PATH)/.)
My_All_Files := $(My_All_Files:$(MY_CPP_PATH)/./*=$(MY_CPP_PATH)%)
MY_CPP_LIST := $(filter %.cpp %.c,$(My_All_Files))
MY_CPP_LIST := $(MY_CPP_LIST:$(LOCAL_PATH)/%=%)

LOCAL_SRC_FILES := $(MY_CPP_LIST)
```

代码中用到了filter函数.

还不满足?如果项目的源码有多个目录放在不同的地方,而且有多个后缀,怎么办?

## 终极进阶:引入多个目录(包括子目录)下的多个后缀名的源文件

上代码(2013年10月9日修正):

```
# 扫描目录下的所有源文件
MY_FILES_PATH := $(LOCAL_PATH) \
                $(LOCAL_PATH)/../..../Classes

MY_FILES_SUFFIX := %.cpp %.c %.cc

My_All_Files := $(foreach src_path,$(MY_FILES_PATH), $(shell find $(src_path) -type f) )
My_All_Files := $(My_All_Files:$(MY_CPP_PATH)/./*=$(MY_CPP_PATH)%)
MY_SRC_LIST := $(filter $(MY_FILES_SUFFIX),$(My_All_Files))
```

```
MY_SRC_LIST := $(MY_SRC_LIST:$(LOCAL_PATH)/%=%)  
LOCAL_SRC_FILES := $(MY_SRC_LIST)
```

以上代码中,变量 MY\_FILES\_PATH 保存源文件所在目录, MY\_FILES\_SUFFIX 保存源文件的后缀名

[原创文章](#),转载请注明,谢谢!

## PS:如何debug 一个android.mk文件

有一个办法,那就是在编译过程输出 android.mk 文件中变量的值,就可以观察分析问题所在了,使用代码

```
$(warning $(LOCAL_SRC_FILES))
```

就可以在编译过程中从终端窗口中观察到变量 LOCAL\_SRC\_FILES 的值

 Comments  分享到

### NEWER

[update SeaBIOS on Google Chromebook Pixel](#)

### OLDER

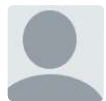
[使用Clover 修正被黑苹果识别为 HD 6xxx 的 HD 6450显卡](#)

9 Comments ready4go

 1 Login ▾

 Recommend  Share

Sort by Best ▾



Join the discussion...



tian • 3 years ago

嘿嘿 老徐威武

^ | v • Reply • Share ›



wangbin.jxust • 3 years ago



我这边也是，使用shell find，输出的日志显示：

FIND: ??????`?????`

Find后面全是乱码，请问这是为什么呢？编码的问题吗？

^ | v • Reply • Share ›



**Frank** • 3 years ago

shell find 貌似不能跨平台啊。

^ | v • Reply • Share ›



**xhcnb** Mod → Frank • 3 years ago

可能windows没有find命令吧, 我手边没有windows机器,没有测试过,Mac和Linux都没问题的

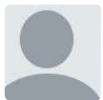
^ | v • Reply • Share ›



**Annie** → xhcnb • 3 years ago

windows下，FIND: 参数格式不正确

^ | v • Reply • Share ›



**mrjeye** • 4 years ago

你好,

```
app
|---Classes
| |---libjson
| | |---include
| | | |---json.h
| | | |---...
| | |---...
| |---...
| |---Common
| | |---CoreMacros.h
| | |---...
| |---...
| |---...
|---proj.android
| |---build_native.sh
| |---jni
| |---...
|---...
```

see more

^ | v • Reply • Share ›



**xhcnb** Mod → mrjeye • 4 years ago

头文件要这样引入的

```
LOCAL_C_INCLUDES := $(LOCAL_PATH)/../Classes/libjson/include
```

```
LOCAL_C_INCLUDES := $(LOCAL_PATH)/../../Classes/libjson/include
```

就是要修改变量LOCAL\_C\_INCLUDES,它表示的头文件的目录

^ | v • Reply • Share ›



**mrjeje** → xhcnb • 4 years ago

我知道了,LOCAL\_C\_INCLUDES指定的是目录-;

^ | v • Reply • Share ›



**mrjeje** → xhcnb • 4 years ago

但是我的.分布在多个不同的目录,现在我使用你上面枚举CPP的方式将所有.H枚举出来指定给LOCAL\_C\_INCLUDES(并且使用\$(warning \$(MY\_H\_LIST))来看所有.H都已包含在内)

可是目前还是会报json.h找不到

是否是.H出现的顺序对编译有影响?

^ | v • Reply • Share ›

#### ALSO ON READY4GO

查看[android](#)上程序的内存占用

1 comment • 3 years ago •



**tian** — nice

自定义[xcode](#)代码提示的风格 - Xu' Blog

1 comment • 3 years ago •



**NovemberEleven** — 搞了好久原来Xcode5.1位置变了, /Applications/Xcode.app/Contents/Frameworks/

Update: [Android.mk](#) 中的LOCAL\_SRC\_FILES, ...

7 comments • 3 years ago •



**biao chang** — 非常感谢

使用[eclipse](#)调试[cocos2d-x Native C++](#) 代码

12 comments • 4 years ago •



**Joker Lee** — NDK\_MODULE\_PATH的问题我也遇到了我的解决方法是把path设置放到Android.mk中, 比如\$(call ...

✉ Subscribe   Add Disqus to your site   Add Disqus   Add   Privacy

#### 分类

[Linux](#) (1)

- [Ubuntu](#) (1)
  - [Mac](#) (1)
    - [TimeMachine](#) (1)

[Mac](#) (1)

- [xcode](#) (1)

[NDK](#) (1)

- [xcode](#) (1)
- [android](#) (7)
- [NDK](#) (5)
  - [eclipse](#) (1)
- [chromebook\\_pixel](#) (1)
- [linux](#) (1)
- [cocos2d-x](#) (2)
- [android](#) (1)
    - [NDK](#) (1)
      - [eclipse](#) (1)
- [hackintosh](#) (1)
- [Clover](#) (1)
- [mac](#) (1)
- [python](#) (1)
- [ubuntu](#) (1)
- [xcode](#) (1)

## 归档

- [June 2014](#) (1)
- [November 2013](#) (1)
- [October 2013](#) (3)
- [September 2013](#) (1)
- [August 2013](#) (2)
- [July 2013](#) (2)
- [June 2013](#) (5)
- [May 2013](#) (5)

## 近期文章

- [xcode6\\_beta build fix: Undefined symbols \\_fwrite\\$UNIX2003](#)
- [解决 xcode 5 资源文件更新后不自动复制的问题](#)
- [Set ubuntu as Mac AFP&TimeMachine server with netatalk 3.05](#)
- [Update: Android.mk 中的 LOCAL\\_SRC\\_FILES, LOCAL\\_C\\_INCLUDES](#)
- [自定义xcode代码提示的风格](#)

## 友情链接

- [风间星魂](#)
- [Charles](#)



© 2014 Jason Xu

Powered by [Hexo](#) . Theme by [Landscape-plus](#)