**Android.mk文件**

Android.mk文件用于向系统描述你要编译的代码，模块。

在一个完整的安卓源码工程中，一般性根目录下的Android.mk中只有一句话： include build/core/main.mk

这个文件暂时不看，进入到每一个android的suite中，就可以查看一个基本的Android.mk文件的内容。

其中指定的几个关键的变量：

LOCAL\_PATH：用于在开发树中查找源文件，一般指定为宏函数my-dir，由编译系统提供，返回当前路径，即包含Android.mk文件的路径。

Include $(CLEAR\_VARS)：CLEAR\_VARS同样是由编译系统提供的宏，指定由GNU\_MAKEFILE为你清除许多LOCAL\_XXX变量比如说LOCAL\_MODULE，LOCAL\_SRC\_FILES，LOCAL\_STATIC\_LIBRARIES等。清除这些变量是必要的，因为所有的编译控制文件（Android.mk，Makefile等）都必须在同一个GNU MAKE环境中执行，所有的变量必须是GLOBAL的，而不能够是LOCAL的。

如果是安卓NDK的编译的Android.mk文件，则还必须要指定LOCAL\_MODULE。这个变量用于标识唯一的模块名，由这个名字添加相应的前缀和后缀名，命名规则中不允许包含任何空格。例如，指定LOCAL\_MODULE = hello，编译系统就会最终生成libhello.so文件。

LOCAL\_SRC\_FILES：加入需要编译的源文件的路径，必须包含要构建到模块中的C/C++文件列表。多个文件之间用 \ 隔开。

LOCAL\_C\_INCLUDES：指定所需包含的头文件的路径。

**安卓NDK**

NDK是安卓原生的开发工具，能够在安卓代码中加入和管理C/C++程序。

<https://developer.android.com/ndk/guides/android_mk.html?hl=zh-cn>

**静态库和动态库**

Linux下的动态库和静态库的区别在于，代码被载入的时刻不同。

静态库：在编译过程中被载入可执行程序，体积较大。

动态库：在可执行程序运行过程中被载入，在编译过程中仅仅是简单的引用，代码体积较小。

因此，在程序运行过程中，已经不再需要静态库，因为其代码已经被载入可执行程序中，而动态库却是需要的，因为它是在运行过程中被载入的。

静态库名字：一般为\*.a

动态库名字：一般为\*.so

**编译命令**

mm: 编译当前路径下的所有模块

m: 从最上层树开始编译全部的模块

mmm:编译指定路径下的模块，指定路径下必须有Android.mk文件