# NDK学习及踏入c++想法

## NDK初次见面

1. NDK简介

**总体来说**：

AndroidNDK是运行于Android平台上的NativeDevelopmentKit的缩写。

Android应用开发者可以通过NDK调用C或C++本地代码。

**两种误解**：

误解一：NDK发布之前，Android不支持进行C开发

在Google中搜索“NDK”，很多“Android终于可以使用C++开发”之类的标题，这是一种对 Android平台编程方式的误解。其实，Android平台从诞生起，就已经支持C、C++开发。众所周知，Android的SDK基于Java实现，这意味着基于AndroidSDK进行开发的第三方应用都必须使用Java语言。但这并不等同于“第三方应用只能使用Java”。在AndroidSDK首次发布时，Google就宣称其虚拟机Dalvik支持JNI编程方式，也就是第三方应用完全可以通过JNI调用自己的C动态库，即在Android平台上，“Java+C”的编程方式是一直都可以实现的。

在NDK出来之前，我们将“Java+C”的开发模式称之为灰色模式，即官方既不声明“支持这种方式”，也不声明“不支持这种方式”。

误解二：有了NDK，我们可以使用纯C开发Android应用。AndroidSDK采用Java语言发布，把众多的C开发人员排除在第三方应用开发外（注意：我们所有讨论都是基于“第三方应用开发”，Android系统基于Linux，系统级别的开发肯定是支持C 语言的。）。NDK的发布，许多人会误以为，类似于Symbian、WM，在Android平台上终于可以使用纯C、C++开发第三方应用了！其实不然，NDK文档明确说明：it is not a goodway。因为 NDK并没有提供各种系统事件处理支持，也没有提供应用程序生命周期维护。此外，在本次发布NDK中，应用程序UI方面的API也没有提供。至少目前来说，使用纯C、C++开发一个完整应用的条件还不完备。

**定义**：

NDK提供了一系列的工具，帮助开发者快速开发C（或C++）的动态库，并能自动将so和 java应用一起打包成apk。这些工具对开发者的帮助是巨大的。NDK集成了交叉编译器，并提供了相应的mk文件隔离CPU、平台、ABI等差异，开发人员

只需要简单修改mk文件（指出“哪些文件需要编译”、“编译特性要求”等），就可以创建出so。

NDK可以自动地将so和Java应用一起打包，极大地减轻了开发人员的打包工作。

1. Android NDK内容

A:AndroidNDK是什么？

AndroidNDK是编译嵌入在Android应用中的原生代码的工具。

Android应用运行在Dalvik虚拟机上。NDK允许开发者用原生代码（C或C++）实现应用的一部分。这将给某些应用带来好处，这种方式可重用代码，而且在某些情况下可加快运行速度。

B:NDK提供了什么？

提供了将C和C++源代码生成原生代码库的工具和文件 提供了将原生库嵌入apk文件的方法提供了兼容1.5版本以上的原生系统头文件和库 提供了文档，示例和指引。

Android1.5NDK支持ARMv5TE机器指令集，提供稳定的C库头文件（Clibrary），JNI接口 和其他的库。

NDK并不适用于大部分应用。作为开发者，你应该衡量它的优缺点；很明显，用原生代码并不能自动提升性能，却增加了应用的复杂度。NDK适合用于独立的、占用内存少、占用较多CPU

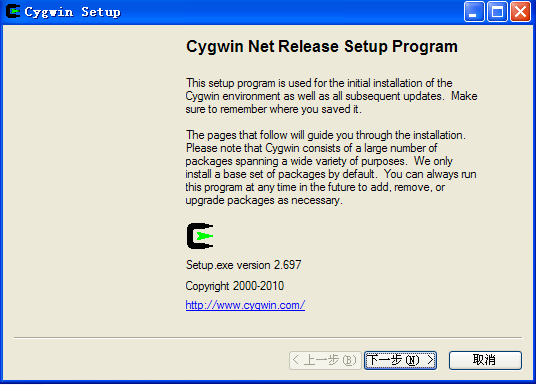
资源的处理，例如，信号处理、物理仿真等等。简单的用C重写代码一般不会带来性能的大幅提升。不过，NDK提供了重用现有C/C++的有效途径。

注意，NDK并不能让你开发纯原生应用。Android的主要运行时仍然是Dalvik虚拟机。

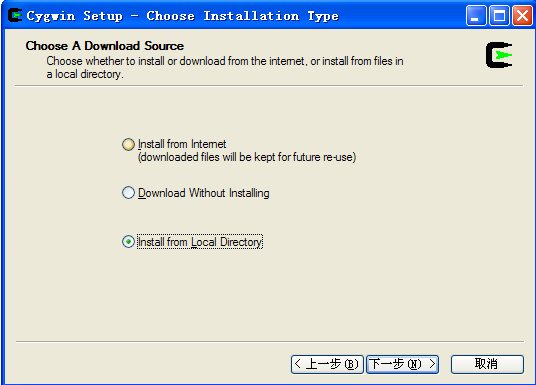
1. NDK环境配置

A: 安装Cygwin

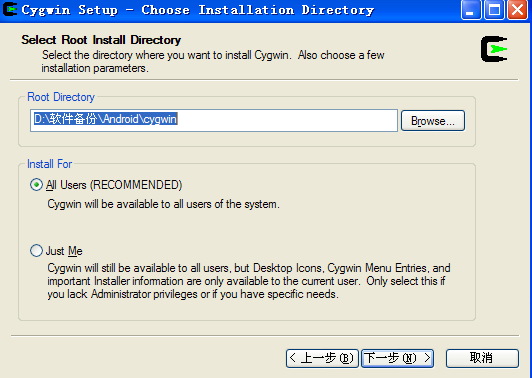
NDK编译需要用到Cygwin中的make和gcc,所以先来下载并安装Cygwin。在[\\192.168.20.224\sirshare\九九装机软件\Develop\Android下面有个cygwin.rar](file:///\\192.168.20.224\sirshare\九九装机软件\Develop\Android下面有个cygwin.rar)的压缩包，运行里面的setup.exe则



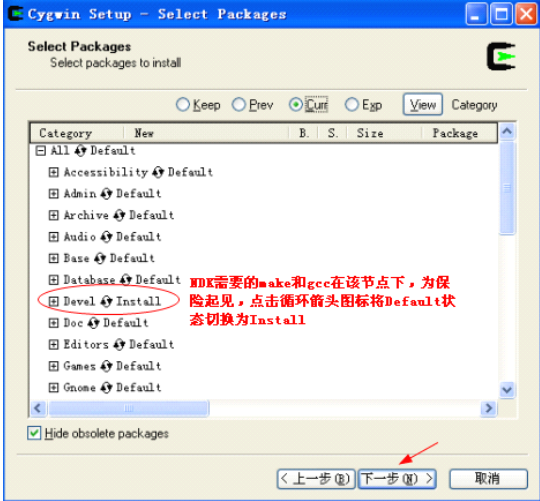
下一步：



选择从本地安装，下一步

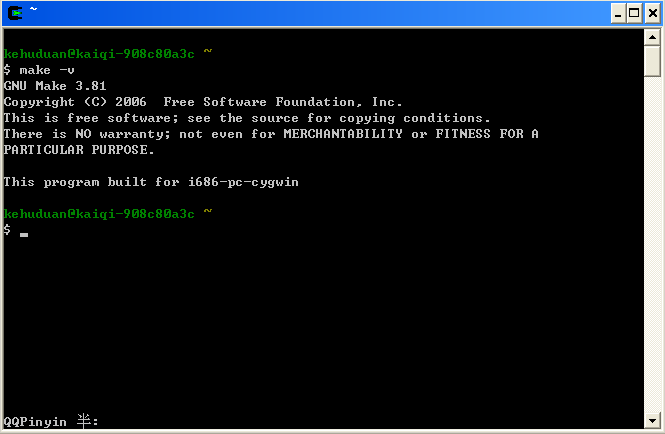


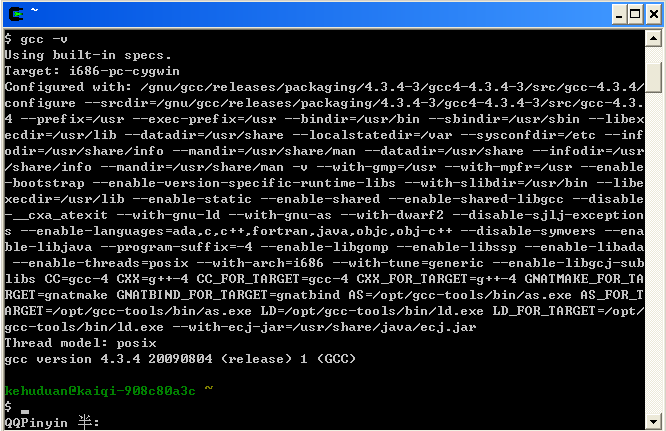
选择压缩包解压的目录。一路下一步到了



然后一路下一步就完成安装了。

在安装完成后运行Cygwin，测试安装





显示出版本号则说明安装完成。

B:安装NDK。下载[\\192.168.20.224\sirshare\九九装机软件\Develop\Android\android-ndk-r4-windows.zip](file:///\\\\192.168.20.224\\sirshare\\九九装机软件\\Develop\\Android\\android-ndk-r4-windows.zip) 解压到你想安装的目录比如我：E:\Android

C:配置环境变量

找到安装目录下面的.bash\_profile文件如我的目录E:\cygwin\home\kehuduan，用文本工具打开在最后面添加下面两句

ANDROID\_NDK\_ROOT=/cygdrive/E/Android/android-ndk-r4

export ANDROID\_NDK\_ROOT

保存后重启CygWin即可，如果有错则说明配置有问题。这时候你就可以编译E:\Android\android-ndk-r4\samples 下面的例子了。

1. NDK实现流程

A:写成native java代码

可以在eclipse里面编辑java文件，(见D:\test例子),编译生成class文件

B:生成c++头文件

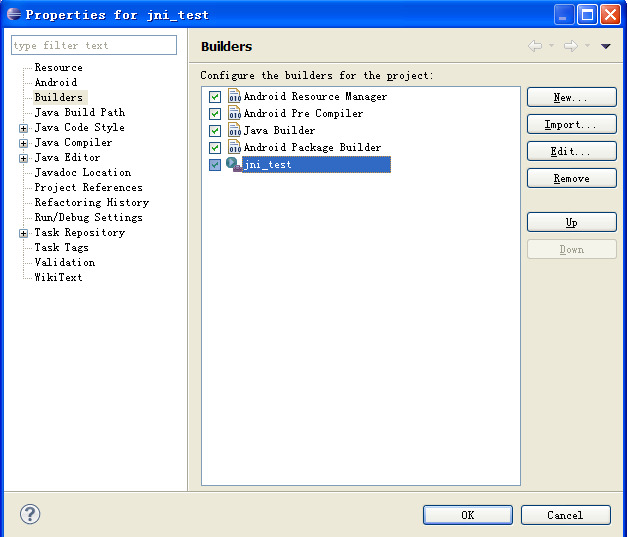
手动写或者是用jdk专门为jni提供的javah命令生成头文件,注意文件名还有函数名字都是带包名字的.

C:在vs中实现相应c++代码，并且调试透过

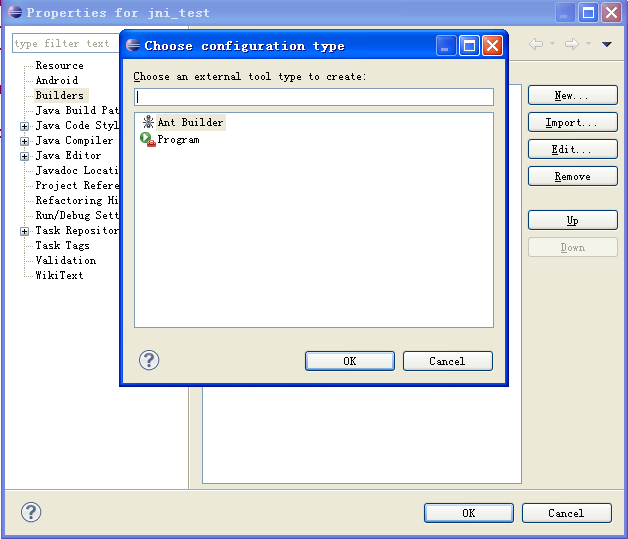
在vs中导入c++代码,导入jdk下面的jni需要的头文件就可以写jni的代码了.

E:编写NDK配置文件（生成.so文件）

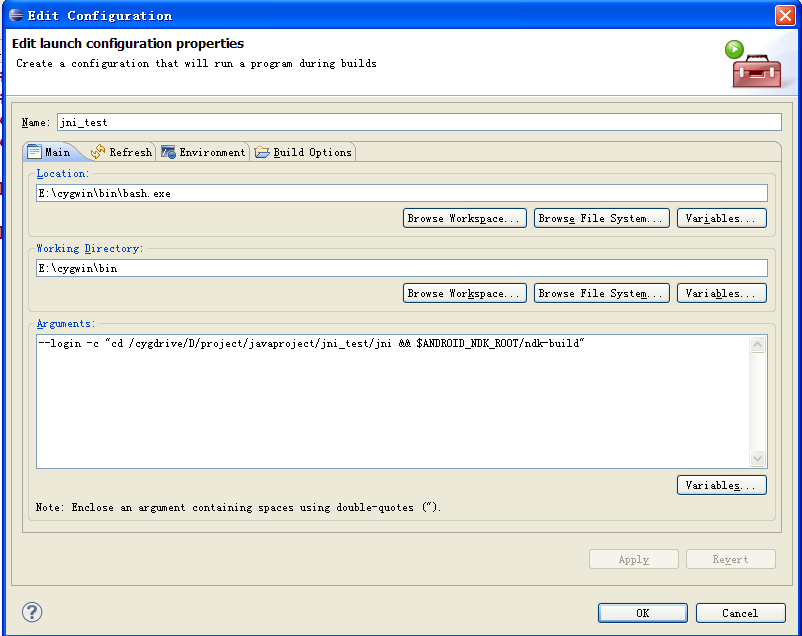
1. 在eclipse中配置NDK编译环境.
2. 在eclipse工程目录下面新建一个文件夹,名字自定,比如我的叫jni.
3. 打开工程的属性页面选到”Builders”选项，



选择New

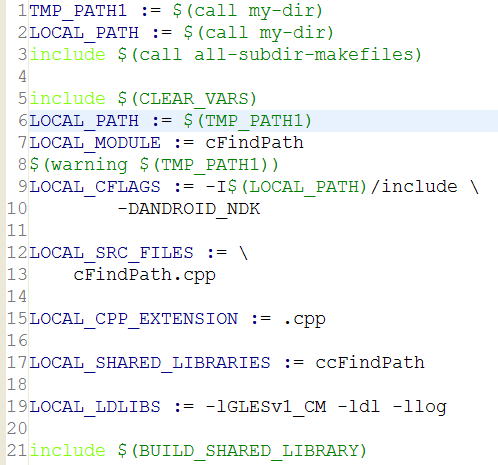


双击Program，填写下图信息，Name是标识Program的，Location选择你的Cygwin的bash.exe，Working Directory是你的Cygwin的工具bin目录，Arguments 需要填写你要编译的NDK代码的目录和NDK环境变量。



1. 编写NDK编译脚本文件Andorid.mk文件，其语法可以在ndk目录下面的docs/ANDROID-MK.TXT中找到。简而言之，NDK通过mk文件组织源文件（c，c++代码）进入“modules”，每个模块可以是静态库或者是共享库，你可以定义一个模块一个mk文件，也可以多个模块在一个mk文件中。

稍微介绍下mk的内容：



LOCAL\_PATH:=$(call my-dir) ：

my-dir'宏的功能由编译器提供，被用来返回当前目录的地址。

TMP\_PATH1 := $(call my-dir)：

类似于批处理中的路径赋值。

include $(call all-subdir-makefiles)：

类似于批处理的递归调用。

include $(CLEAR\_VARS)：

类似于批处理中的clean。

LOCAL\_MODULE := cFindPath

是你要生成的so的文件名字，比如上面会生成：libcFindPath.so，lib是编译器自动加的。

LOCAL\_CFLAGS := -I$(LOCAL\_PATH)/include \

-DANDROID\_NDK

编译当前文件需要用的c++头文件。

LOCAL\_SRC\_FILES := \

cFindPath.cpp

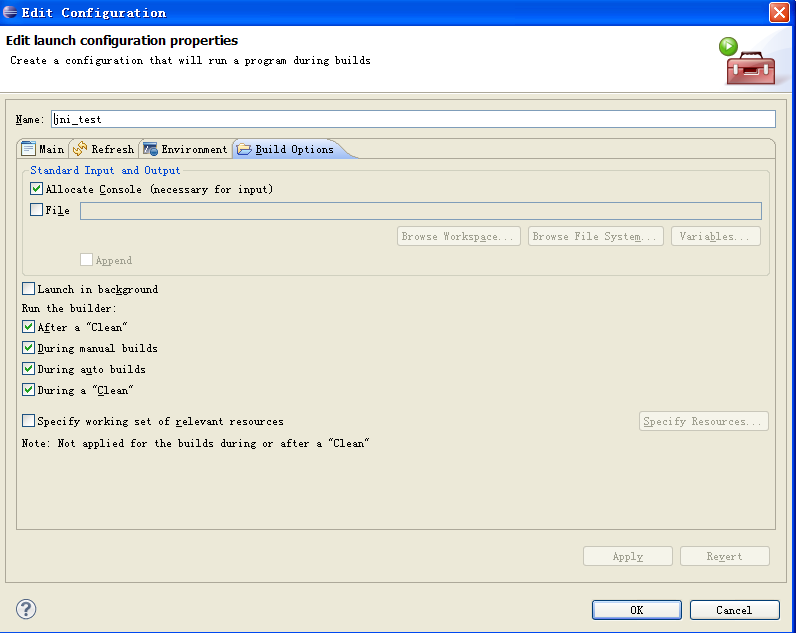
编译的c++的文件。

LOCAL\_CPP\_EXTENSION := .cpp

编译文件的扩展名

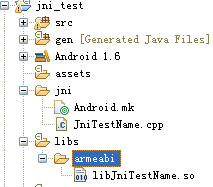
LOCAL\_SHARED\_LIBRARIES := ccFindPath

生成当前so要用到的库。



在刚刚新建的ndk的Program的Build Options里面可以设置编译的条件，比如要clean才会重新编译等等。

编译成功后会在工程目录下的。



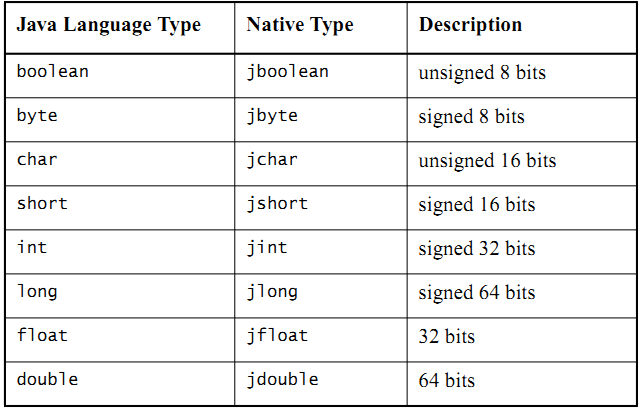
生成so文件。接下来运行就能调用库了。

## JNI介绍

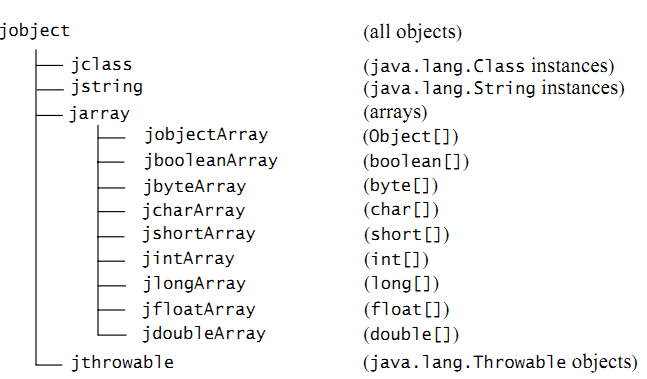
JNI = Java™ Native Interface 。就是java调用不同平台的底层代码的一种方式，在jdk的安装目录下面（Y:\jdk1.6.0\include）里面有jni的头文件。

1. JNI数据类型

基本数据类型：

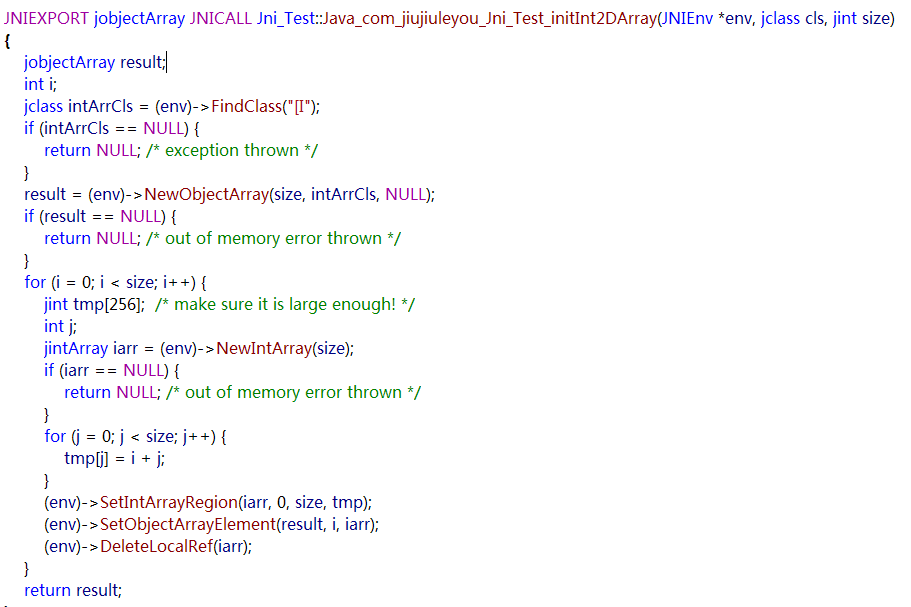


类数据类型：



1. JNI与c++数据转换

直接看个函数吧：



1. JNI 返回的内存回收

这个最近还不是很清楚，在了解中。

## 三．踏入c++想法

1. vs的使用

vs不想eclipse会自动找你加载的目录的文件，要自己添加。

1. 二维数组

提供的数据结构的使用跟java区别很大。int a[][]，会编译错误。

1. 效率问题

for(int i = 0; i < 10; i++) {

int a = i\*10;

}

C++中这样比较快。