<https://blog.csdn.net/afunx/article/details/53447563>

### ****摘要****：本文介绍了如何使用命令行工具来编写Jni程序，以实现Java中调用C库函数的功能。

### 包含Windows 32位、Windows 64位、Linux、Mac OS三种环境。

**前提条件：**

1. 已经安装好JDK，并配置好环境变量

2. Windows 32位已安装并配置好Cygwin可正常使用gcc

3. Windows 64位尚未完成，完成后更新

4. Linux已安装并配置好gcc命令行工具

5. Mac OS在terminal下，可正常使用gcc命令行工具

6. 能编写简单Java代码

7. 能看懂简单C语言代码

**在Java中调用C库函数主要分为6个步骤：**

1. 编写Java代码

2. 生成C语言头文件

3. 编写C代码

4. 生成C共享库

4.1 Windows 32位生成C共享库

4.2 Windows 64位成成C共享库

4.2 Linux生成C共享库

4.3 Mac生成C共享库

5. 编译Java代码

6. 运行Java程序

    分步详解：

**1. 编写Java代码**

在当前目录下，创建HelloJni.java文件，代码如下：

**[java]** [view plain](https://blog.csdn.net/afunx/article/details/53447563) [copy](https://blog.csdn.net/afunx/article/details/53447563)

1. **public** **class** HelloJni {
3. **native** **void** hello();
5. **static** {
6. System.loadLibrary("HelloJni");
7. }
9. **public** **static** **void** main(String args[]) {
10. HelloJni helloJni = **new** HelloJni();
11. helloJni.hello();
12. }
13. }

**2. 生成C语言头文件**

在当前目录下，已可以看到HelloJni.java，在命令行工具中，执行javah HelloJni命令(**注意：不是HelloJni.java**)。

此时，可看到在当前目录下，有HelloJni.java和HelloJni.h,HelloJni.h，代码如下：

**[cpp]** [view plain](https://blog.csdn.net/afunx/article/details/53447563) [copy](https://blog.csdn.net/afunx/article/details/53447563)

1. /\* DO NOT EDIT THIS FILE - it is machine generated \*/
2. #include <jni.h>
3. /\* Header for class HelloJni \*/
5. #ifndef \_Included\_HelloJni
6. #define \_Included\_HelloJni
7. #ifdef \_\_cplusplus
8. **extern** "C" {
9. #endif
10. /\*
11. \* Class:     HelloJni
12. \* Method:    hello
13. \* Signature: ()V
14. \*/
15. JNIEXPORT **void** JNICALL Java\_HelloJni\_hello
16. (JNIEnv \*, jobject);
18. #ifdef \_\_cplusplus
19. }
20. #endif
21. #endif

**3.编写C代码**

在当前目录下，创建HelloJni.c（或HelloJni.cpp)文件，代码如下：

**[cpp]** [view plain](https://blog.csdn.net/afunx/article/details/53447563) [copy](https://blog.csdn.net/afunx/article/details/53447563)

1. #include "HelloJni.h"
3. JNIEXPORT **void** JNICALL Java\_HelloJni\_hello
4. (JNIEnv \*env, jobject obj)
5. {
6. printf("Hello Jni\n");
7. **return**;
8. }

**4.生成共享库**

**4.1 Windows 32位生成共享库**

这里需要特别注意一点，在Windows的Cmd命令行中，如果你的路径中带有空格，需要额外进行处理。

具体处理方法，可参考[文献[4]](http://blog.csdn.net/duanbeibei/article/details/45364407)，这里仅举个例子来简单说明。

echo %JAVA\_HOME

C:\Program Files\Java\jdk1.8.0\_102

那么，需要执行以下命令来生成32位dll：

gcc -I C:\PROGRA~1\Java\jdk1.8.0\_102\include -I C:\PROGRA~1\Java\jdk1.8.0\_102\include\win32 -shared -Wl,–kill-at -s -o Hello.dll HelloJni.c

**4.2 Windows 64位生成共享库**

注意：使用编译出来的32位dll库，会与之不兼容。

目前，只编译出来了32位dll库。之后，完成了64位dll库编译，会再更新博客。

**4.3 Linux生成共享库**

在当前目录下，执行echo $JAVA\_HOME，查看JDK路径，之后的命令都会基于JDK路径。

比如，echo $JAVA\_HOME，结果为/usr/local/jvm/openjdk-1.8.0-internal

4.3.1 编译并生成.o文件

g++ -I /usr/local/jvm/openjdk-1.8.0-internal/include -I /usr/local/jvm/openjdk-1.8.0-internal/include/linux/ -fPIC -c HelloJni.c

4.3.2 编译并生成.so文件

g++ -shared -Wl,-soname -o libHelloJni.so HelloJni.o

4.3.3 通知动态链接程序此共享文件的路径

export LD\_LIBRARY\_PATH=`pwd`:$LD\_LIBRARY\_PATH

**4.4 Mac生成共享库**

在当前目录下，命令行工具中，执行如下命令：

g++ -dynamiclib -I /System/Library/Frameworks/JavaVM.framework/Headers HelloJni.c -o libHelloJni.jnilib

成功后，可看到新增libHelloJni.jnilib。

(在Mac OS环境下，动态库的命名格式为libXXX.jnilib）

**5.编译java代码**

在当前目录下，已可以看到HelloJni.java、HelloJni.h、HelloJni.c、 libHelloJni.jnilib。

在命令行工具中，执行javacHelloJni.java后，可看到新增HelloJni.class

**6.运行java程序**

在当前目录下，已可以看到HelloJni.java、HelloJni.h、HelloJni.c、HelloJni.dll（Windows环境）、libHelloJni.so（Linux环境）或libHelloJni.jnilib（Mac OS环境）、HelloJni.class。

在命令行工具中，执行java HelloJni，可看到执行结果为：

**[plain]** [view plain](https://blog.csdn.net/afunx/article/details/53447563) [copy](https://blog.csdn.net/afunx/article/details/53447563)

1. Hello Jni

参考文献：

1.《Android框架揭秘--人民邮电出版社》

2. [博客：Mac OS 环境下编译JNI](http://blog.csdn.net/codigger/article/details/9296037)（<http://blog.csdn.net/codigger/article/details/9296037>）

3. [博客：在Linux平台下使用JNI](http://www.ibm.com/developerworks/cn/java/l-linux-jni/)（[http://www.ibm.com/developerworks/cn/java/l-linux-jni](http://www.ibm.com/developerworks/cn/java/l-linux-jni/)/）

4. [博客：命令行如何处理windows含空格的目录](http://blog.csdn.net/duanbeibei/article/details/45364407) （<http://blog.csdn.net/duanbeibei/article/details/45364407>）

5. [博客：在windows下使用gcc编译jni的简单教程](http://blog.csdn.net/duanbeibei/article/details/45364407)（<http://blog.csdn.net/duanbeibei/article/details/45364407>）