The solution I discovered was that I was not specifying the directory where javac should place the header files.

Executing javac -h . NativeTest.java worked.

gcc -I/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.fc30.x86\_64/include

The JNI header "jni.h" provided by JDK is available under the "<JAVA\_HOME>\include" and "<JAVA\_HOME>\include\win32" (for Windows) or "<JAVA\_HOME>\include\linux" (for Ubuntu) [Check Mac OS X] directories, where <JAVA\_HOME> is your JDK installed directory (e.g., "c:\program files\java\jdk10.0.x" for Windows).

For 64-bit JDK, you need to find a compiler that produces target of 64-bit native Windows. This is provided by MinGW-W64. You can install MinGW-W64 under Cygwin, by selecting packages "mingw64-x86\_64-gcc-core" (C compiler) and "mingw64-x86\_64-gcc-g++" (C++ compiler). The executables are "x86\_64-w64-mingw32-gcc" (C Compiler) and "x86\_64-w64-mingw32-g++" (C++ Compiler), respectively.

Check if JAVA\_HOME is already set

Cygwin: Execute : echo $JAVA\_HOME

If output is a path , then your JAVA\_HOME is set , make sure the path is correct

Navigate a folder with spaces in their names: go to folder “graduate school”

cd graduate\ school/cs587/…

in Cygwin: assume in home directory and want to change to java.include:

cd c/program\ files/Java/jdk-10.0.2/include

this following one does not work for unknown reason:

x86\_64-w64-mingw32-gcc -I"%JAVA\_HOME%\include" -I"%JAVA\_HOME%\include\win32" -shared -o hello.dll HelloWorldImp.c

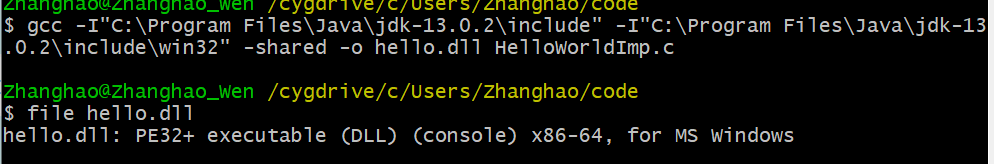
but this one eventually worked:

$ x86\_64-w64-mingw32-gcc -c -I"C:\Program Files\Java\jdk-13.0.2\include" -I"C:\Program Files\Java\jdk-13.0.2\include\win32" -shared -o hello.dll HelloWorldImp.c

This will leads to access denied!

Use this: gcc -I"C:\Program Files\Java\jdk-13.0.2\include" -I"C:\Program Files\Java\jdk-13.0.2\include\win32" -shared -o hello.dll HelloWorldImp.c

Can print out



* 51 = Java 7
* 52 = Java 8
* 53 = Java 9
* 54 = Java 10
* 55 = Java 11
* 56 = Java 12
* 57 = Java 13

When run

java -XshowSettings:properties -version (java -version)

results shows that java version is 1.8.0\_241 which is java 8

when echo %JAVA\_HOME%, it shows jdk-10.0.2 which is java 10

this leads to java.lang.UnsupportedClassVersionError

after solving this issues,

another issues is access denied

switch code location to user/Zhanghao and fixed this issue.

Final problem is segmentation fault.

Changed file jni\_md\_h:

typedef long jint;

typedef \_\_int64 jlong;

typedef signed char jbyte;

to:

typedef long jint;

#ifdef \_\_GNUC\_\_

typedef long long jlong;

#else

typedef \_\_int64 jlong;

#endif

typedef signed char jbyte;

