# JNI基础

## 1、生成JNI头文件

Java代码

public class MyThread {

public native void run0();

public void run() {

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

System.out.println(Thread.currentThread().getName() + " say: " + i);

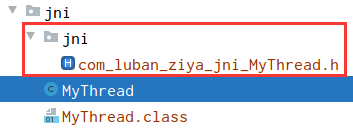
}

}

}

进入.java文件所在目录执行

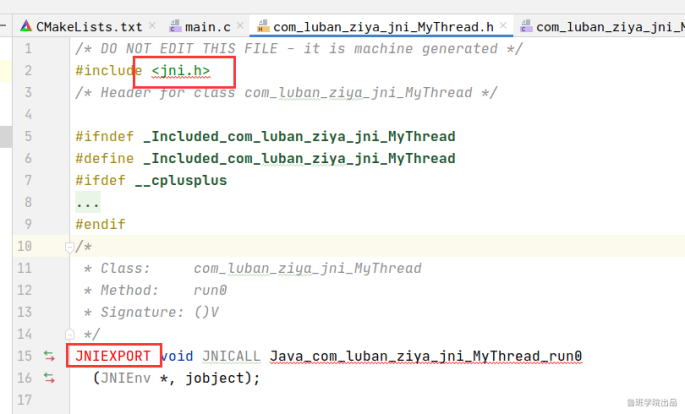
image.png



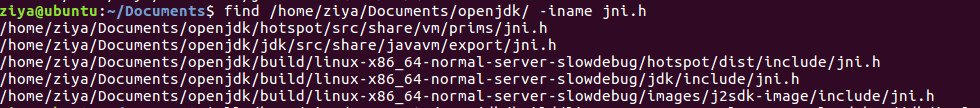
## 2、配置头文件

在Clion中创建一个项目，将这个.h文件拷贝过去（这块有点复杂，仔细看）

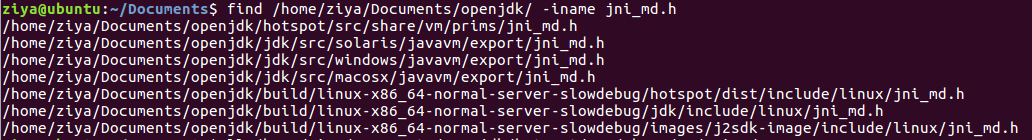
.h文件拷过来后会发现报红，那是因为clion找不到这个头文件需要我们配置



看哪个目录下有jni.h文件



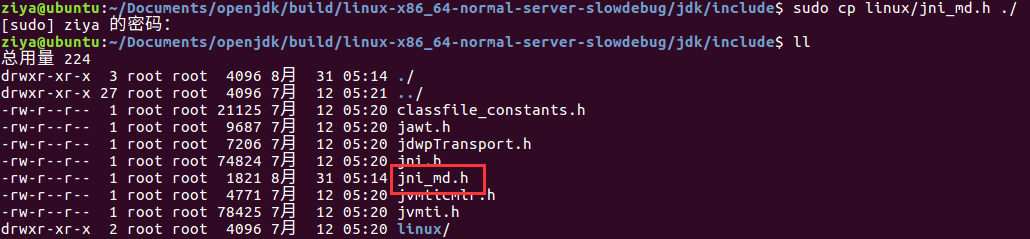
因为jni.h文件依赖jni\_md.h文件，再看哪个目录下有jni\_md.h文件



因为JNI依赖JVM的其他头文件，所以一般用目录

/home/ziya/Documents/openjdk/build/linux-x86\_64-normal-server-slowdebug/jdk/include

这个目录中没有jni\_md.h文件，从它的下级目录拷贝一份



然后修改CMakeLists.txt，加include\_directories，这样你写JNI程序，clion的自动提示就可以用了

cmake\_minimum\_required(VERSION 3.16)

project(c\_mythread C)

set(CMAKE\_C\_STANDARD 99)

include\_directories("/home/ziya/Documents/openjdk/build/linux-x86\_64-normal-server-slowdebug/jdk/include")

add\_executable(c\_mythread main.c com\_luban\_ziya\_jni\_MyThread.c)

## 3、生成动态链接库

/usr/bin/gcc -shared -fpic -I/home/ziya/Documents/openjdk/build/linux-x86\_64-normal-server-slowdebug/jdk/include com\_luban\_ziya\_jni\_MyThread.c -o /lib/libjni.so

注意：动态链接库需要生成在jvm能加载到的地方，如果不知道放哪，可通过打印参数看jvm默认从哪些位置加载

System.out.println(System.getProperty("java.library.path"));

## 4、使用动态链接库

public class App {

public static void main(String[] args) {

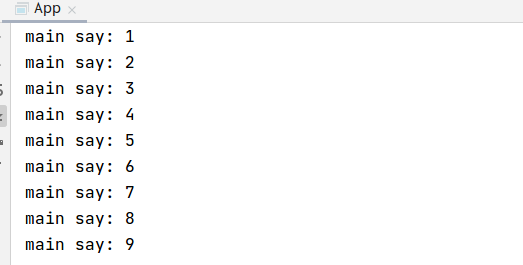
System.loadLibrary("jni");

new MyThread().run0();

}

}

## 5、结果



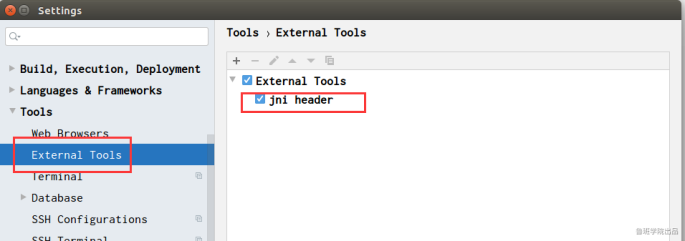
# 配置Idea、Clion

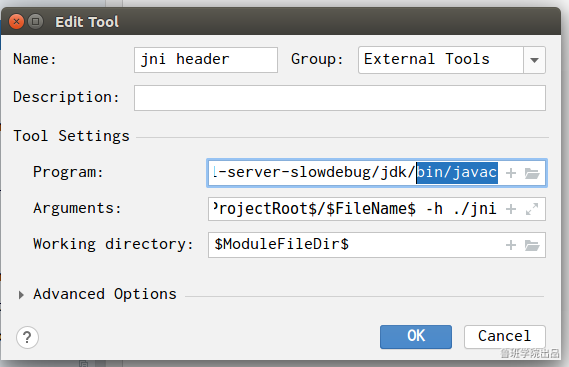
配置了每次生成JNI头文件、动态链接库就不需要敲命令了

image.png

image.png

## 1、Idea

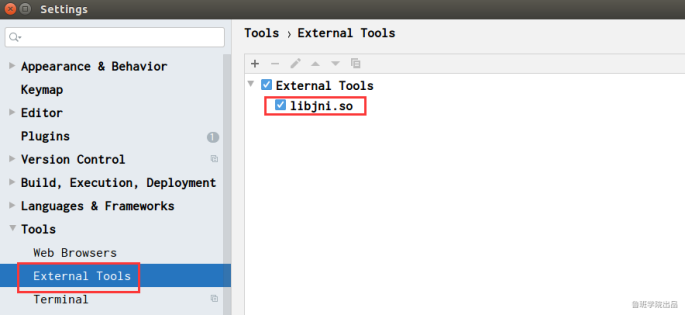


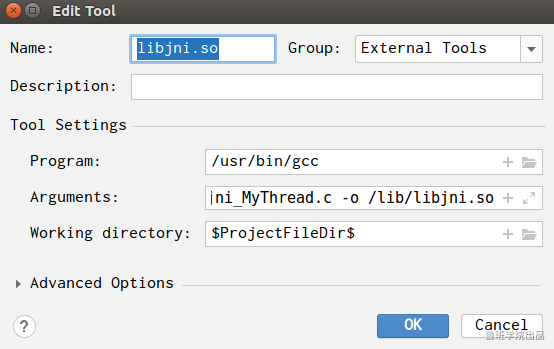


Program：/home/ziya/Documents/openjdk/build/linux-x86\_64-normal-server-slowdebug/jdk/bin/javac

Arguments：$FileDirRelativeToProjectRoot$/$FileName$ -h ./jni

## 2、配置Clion





Arguments这样填

-shared

-fpic

-I/home/ziya/Documents/openjdk/build/linux-x86\_64-normal-server-slowdebug/jdk/include

com\_luban\_ziya\_jni\_MyThread.c

-o

/lib/libjni.so