http://note.youdao.com/noteshare?id=ab688109ab0abde4f4d7bb5825e9ac97&sub=58DE0398325B459D9BCBD41B1C1830D6

说下我的编译环境

建议大家与我的环境保持一致，这样你遇到的问题我才方便给你解答

操作系统：Ubuntu16

boot jdk：jdk7

编译的jdk：jdk8

看源码工具：Clion、NetBeans

链接：https://pan.baidu.com/s/1tR3gMy_59YRekqX915AjsQ 提取码：ziya

编译

1、安装依赖

```
sudo apt-get install libx11-dev libxext-dev libxrender-dev libxtst-dev libxt-dev libcups2-dev libfreetype6-dev libasound2-dev ccache
```

2、boot jdk

因为openjdk源码中有些功能是用Java代码实现的，比如调试工具：jps、jstat等，还有一些核心jar包，如rt.jar、tools.jar等，都需要相应版本的JDK。boot jdk需要比你编译的jdk版本低，我只测试了低一个版本，低多个版本有没有问题，喜欢探索的同学可自行测试。

直接解压到Documents目录下，然后配置下环境

vi ~/.bashrc

文件底部加入下面三句话(ziya改成你自己的用户名)

```
export JAVA_HOME=/home/ziya/Documents/jdk1.7.0_80
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH
```

修改的配置文件载入系统

```
source ~/.bashrc
```

3、配置openjdk

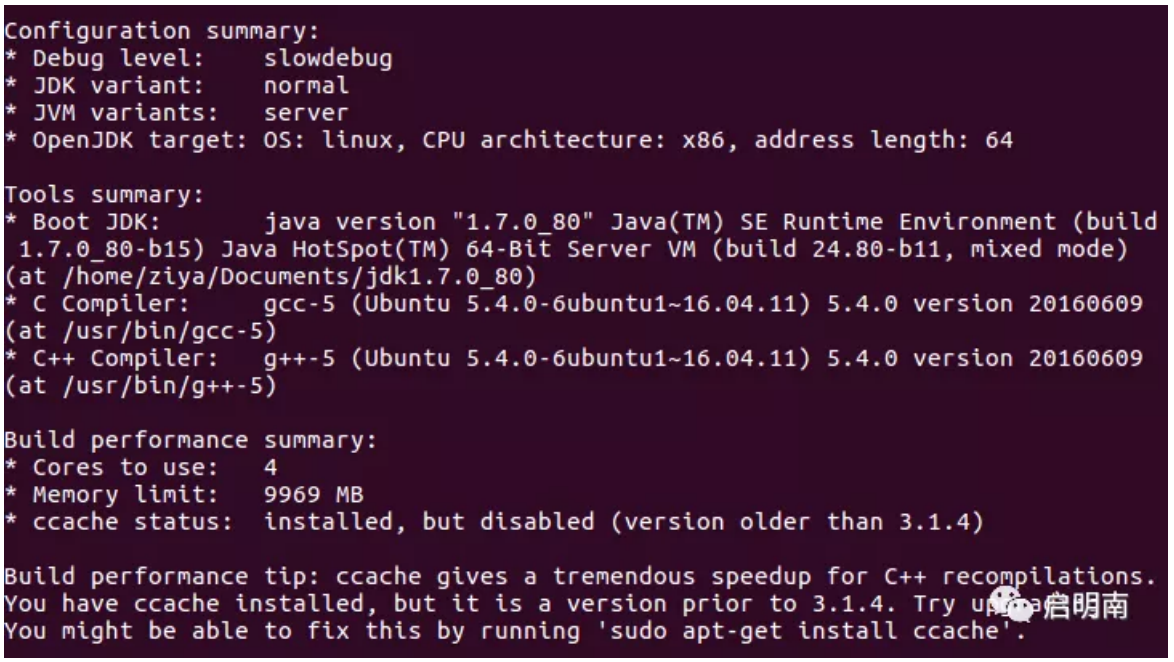
默认情况下configure不是可执行文件，执行命令

```
chmod u+x configure
```

执行配置命令

```
sudo ./configure --with-target-bits=64 --with-boot-jdk=/home/ziya/Documents/jdk1.7.0_80 --with-debug-level=slowdebug --enable-debug-symbols ZIP_DEBUGINFO_FILES=0
```

看到这个画面就证明配置成功了，就可以编译了。如果不是，就得一步步把问题解决了再重新配置



4、编译

```
sudo make all DISABLE_HOTSPOT_OS_VERSION_CHECK=OK ZIP_DEBUGINFO_FILES=0
```

看到下面这个画面就证明编译成功，就可以用了

```
----- Build times -------
Start 2020-07-05 21:20:07
End   2020-07-05 21:31:11
00:00:26 corba
00:00:17 demos
00:02:40 docs
00:03:24 hotspot
00:00:26 images
00:00:15 jaxp
00:00:20 jaxws
00:02:25 jdk
00:00:31 langtools
00:00:18 nashorn
00:11:04 TOTAL
-------------------------
Finished building OpenJDK for target 'all'
ziya@ubuntu:~/Documents/openjdk$
```


5、验证

注：验证之前将第二步配置的JAVA_HOME改成你编译好的

```
hiziya@ubuntu:~/IdeaProjects/java-research/target/classes$ cd ~/Documents/openjdbuild/linux-x86_64-no
ziya@ubuntu:~/Documents/openjdk/build/linux-x86_64-normal-server-slowdebug/jdk/bin$ java -version
openjdk version "1.8.0_252"
OpenJDK Runtime Environment (build 1.8.0_252-8u252-b09-1~16.04-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
ziya@ubuntu:~/Documents/openjdk/build/linux-x86_64-normal-server-slowdebug/jdk/bin$
```

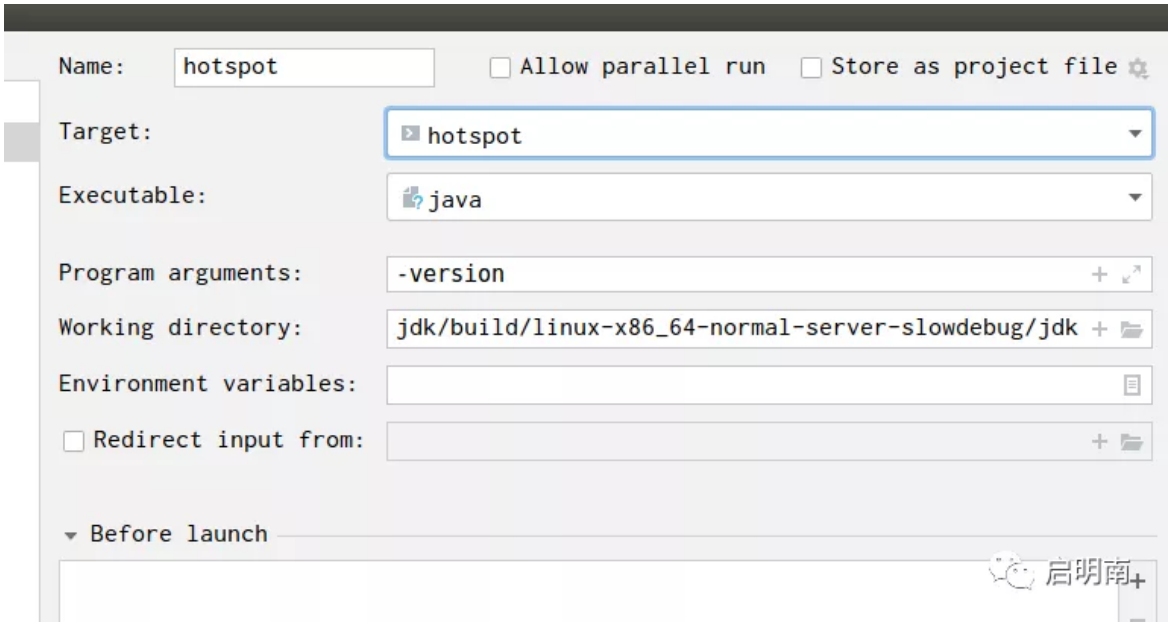

单步调试

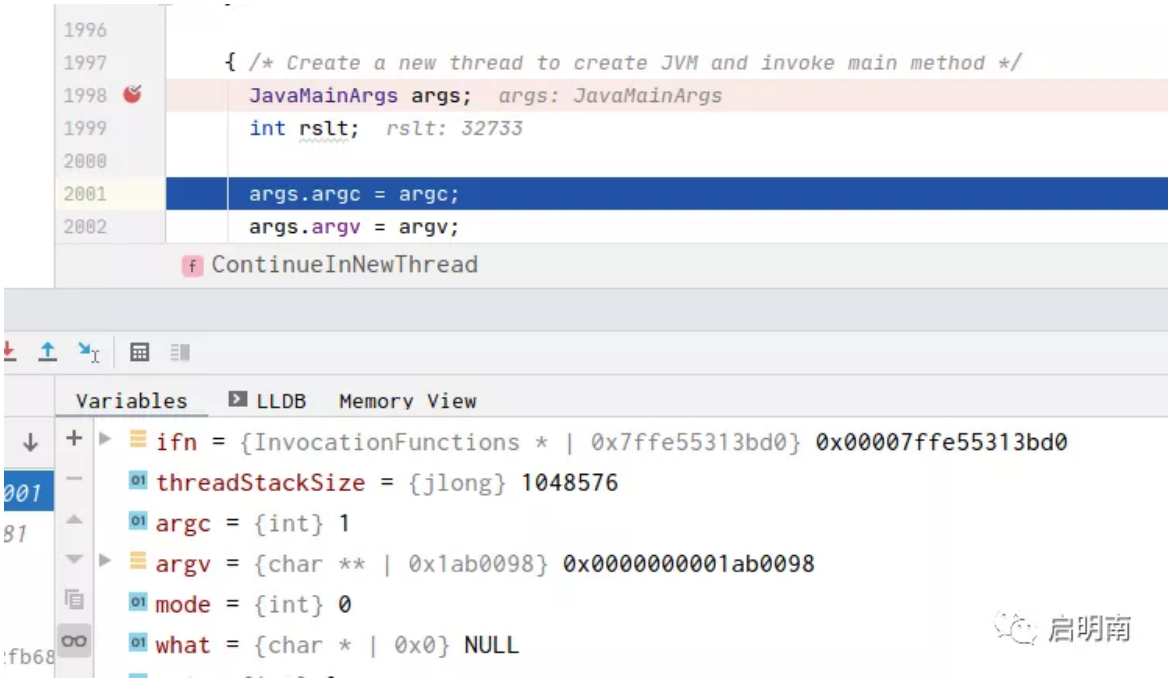
我一般是clion+NetBeans+lldb结合起来用

1、clion

这个IDE不支持Makefile，仅支持cmake，比较麻烦。这里大家也不用花时间去学习cmake了，我写好的cmakefile分享给大家

来看下单步调试效果

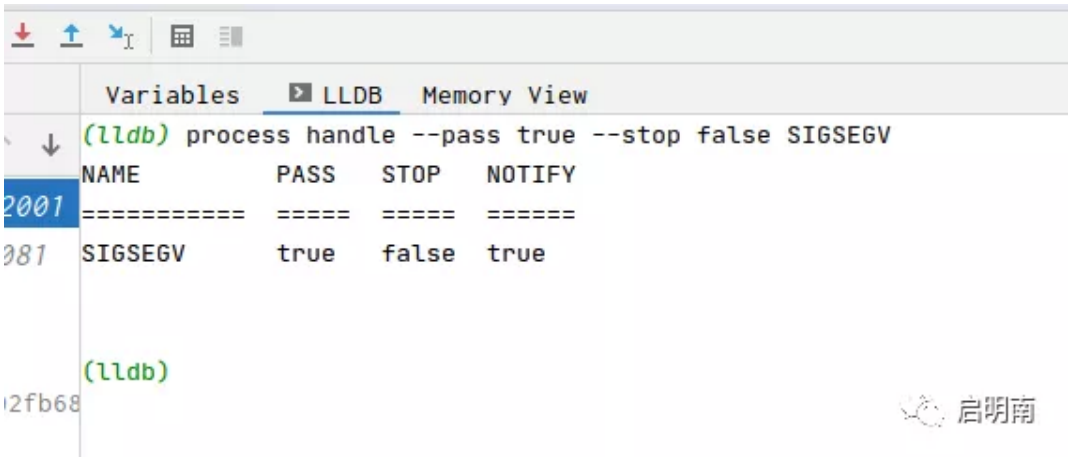




调试过程中会出现段错误导致程序终止，加上这句话忽略SIGSEGV信号

process handle --pass true --stop false SIGSEGV

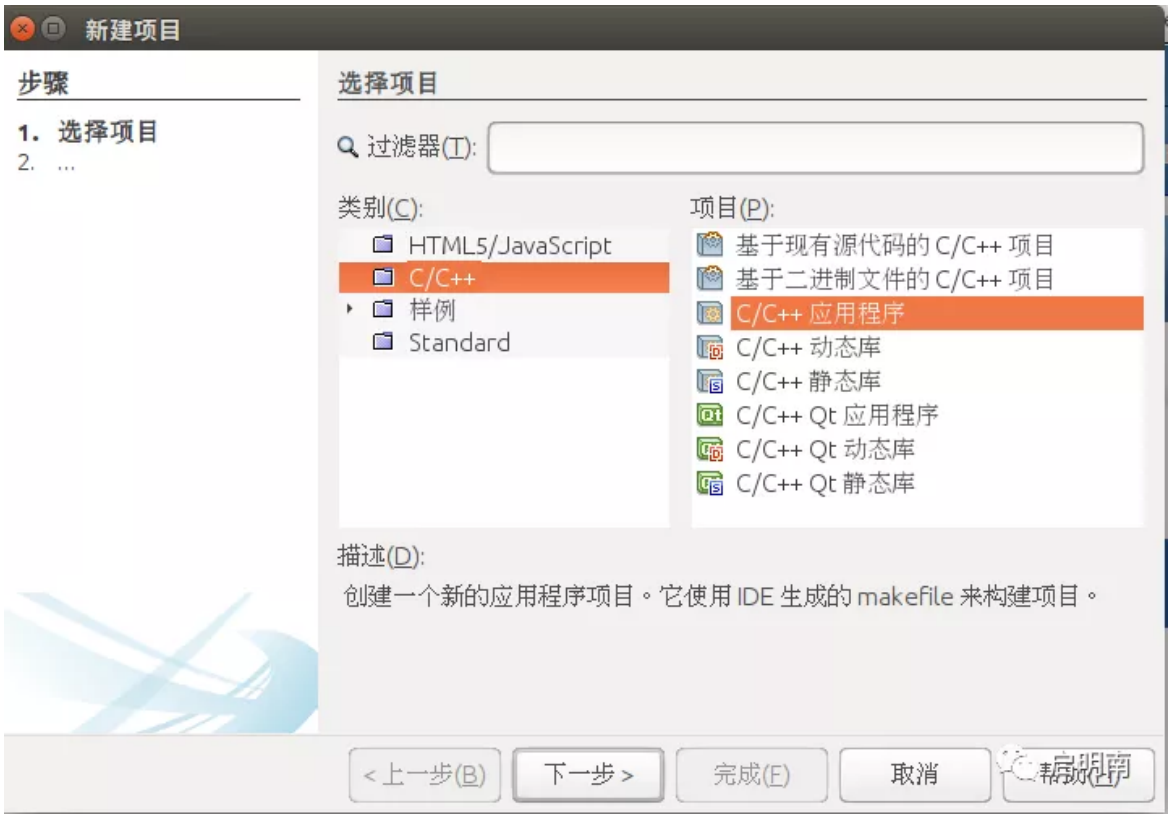
时机：在发生段错误之前都行，我一般是进入了调试程序就马上加上。目录下加.gdbinit文件也行



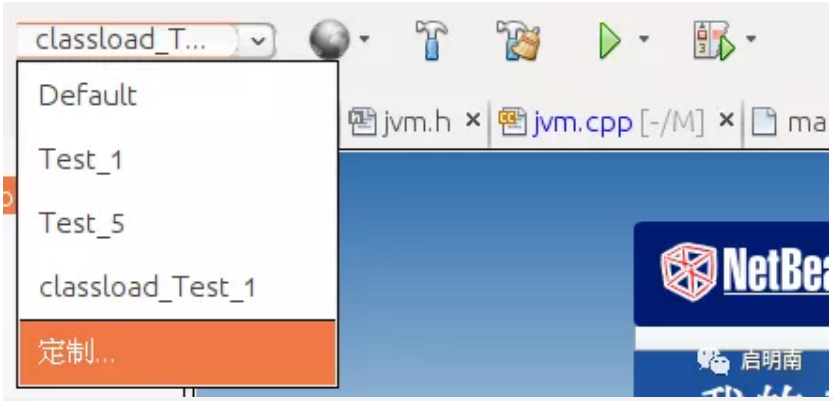
2、NetBeans8.2

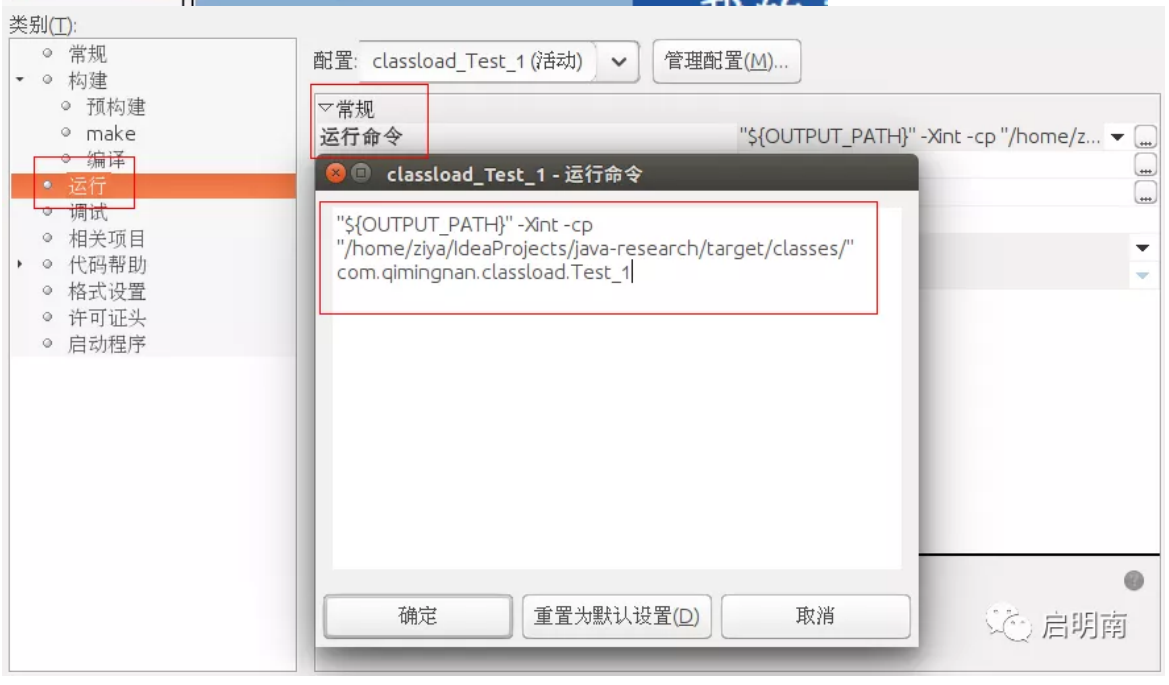
这个IDE支持Makefile，但是高版本就不支持了，所以只能用8.2版本

注意是新建项目，它会根据Makefile将项目构建好。后面就是一直下一步、下一步.....

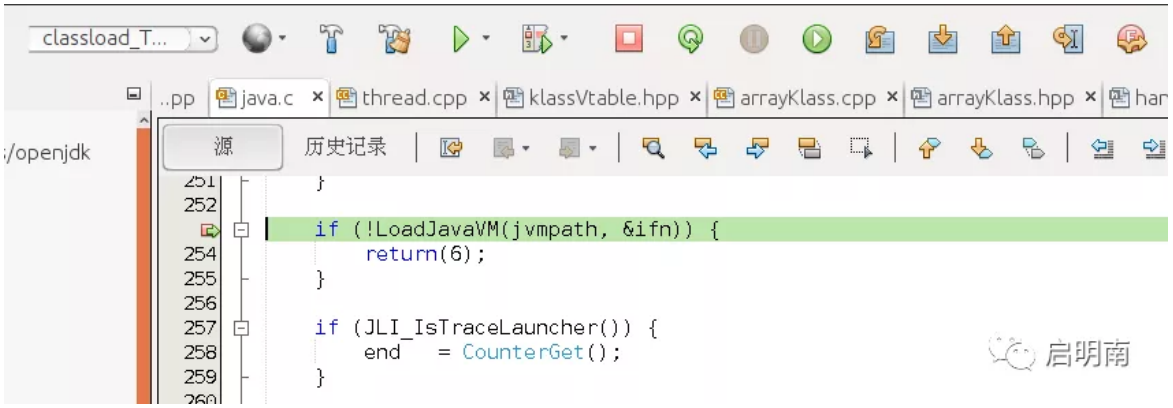


项目构建好后，点击【定制】配置调试环境





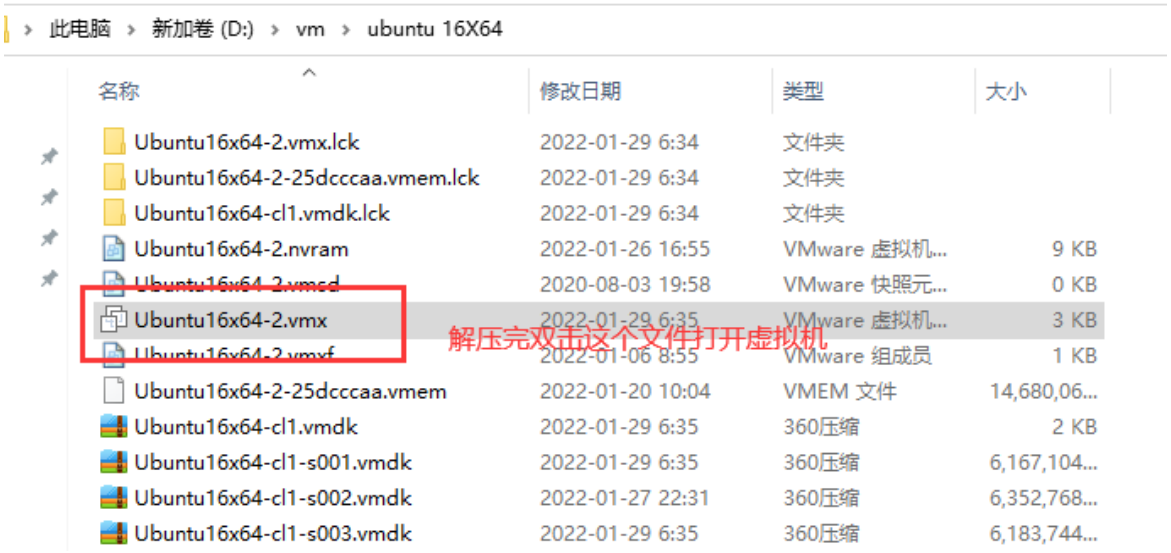
-cp即classpath，后面是要运行的class文件，这样调试环境就搭建完成。



环境介绍

这两个环境的差别：

1. 一条龙是我把我编译好的环境打包出来了，这里面有四个文件，都要下载，然后合并解压（一定要合并解压，不要一个一个解压，会造成文件损坏）（密码：123qwer）
2. 下面那个是编译openjdk源码需要用到的软件包（建议用我提供的，不要在外面随便下，有的版本编译报错）



如果你用一条龙，启动软件我都写好了脚本

```
ziya@ubuntu:~$ cat Makefile
clion:
	sh /home/ziya/Documents/clion-2020.1.2/bin/clion.sh
idea:
	sh /home/ziya/Documents/idea-IU-201.7846.76/bin/idea.sh
netbeans:
	cd /home/ziya/netbeans-8.2 && ./bin/netbeans --jdkhome /usr/lib/jvm/java
-8-openjdk-amd64
hsdb:
	sudo java -cp \
	/home/ziya/Documents/openjdk/build/linux-x86_64-normal-server-slowdebug/
jdk/lib/sa-jdi.jar \
	sun.jvm.hotspot.HSDB
sys-hsdb:
	sudo java -cp \
	/usr/lib/jvm/java-8-openjdk-amd64/lib/sa-jdi.jar \
	sun.jvm.hotspot.HSDB
visualvm:
	sh ~/Documents/visualvm_202/bin/visualvm
```

启动NetBeans：make netbeans

启动hsdb：make hsdb
启动idea：make idea
启动clion：make clion