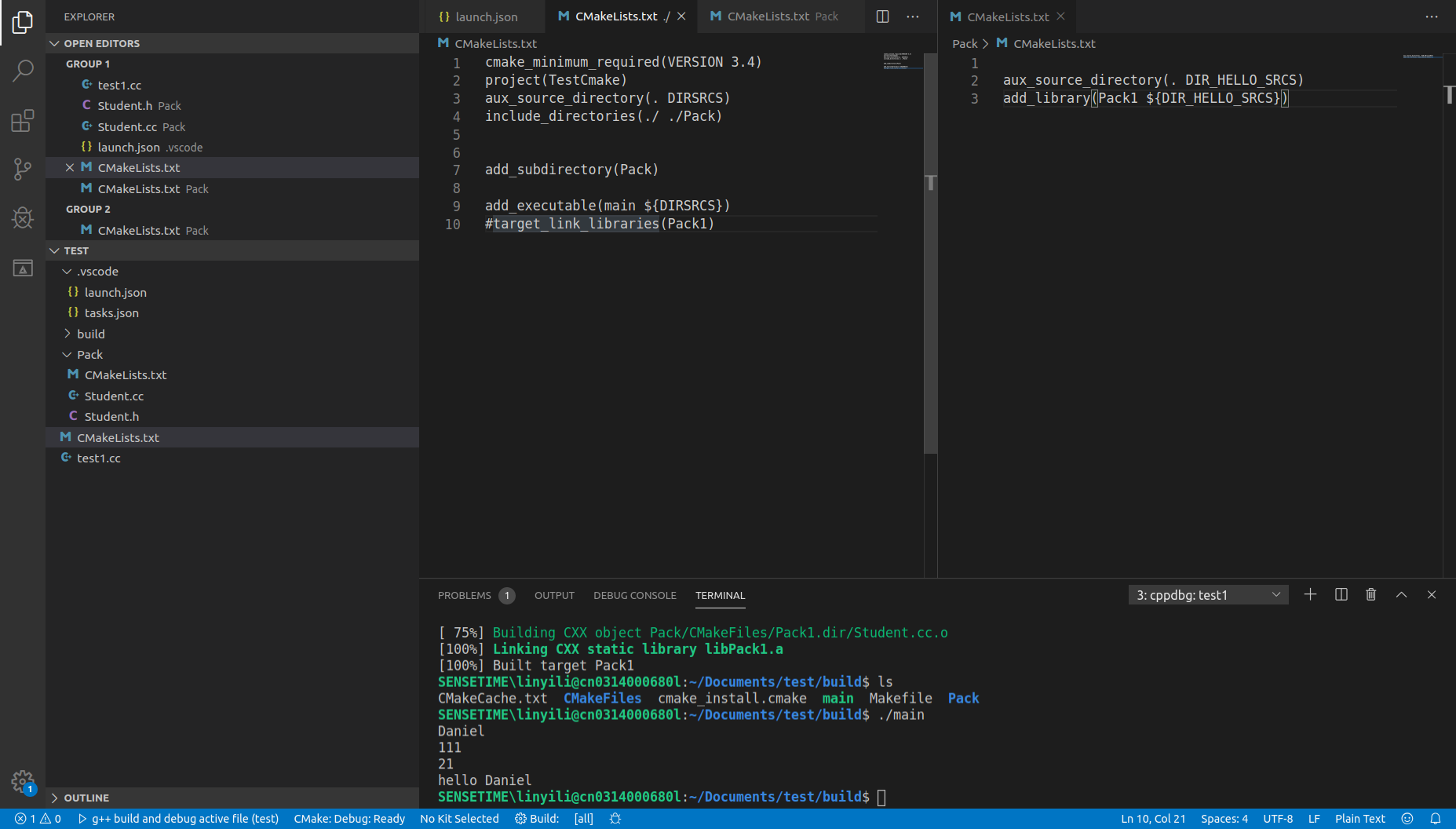
1. 全屏截图 ==>> ctrl+print screen实现ubuntu

窗口截图 ==>> alt+print screen实现ubuntu

区域截图==>> shift+print screen实现ubuntu



2. 上图是cmake多级的载入代码，可供借鉴。

3. 对git使用的教程：<https://www.liaoxuefeng.com/wiki/896043488029600/900003767775424>

git指令汇总pdf：<https://gitee.com/liaoxuefeng/learn-java/raw/master/teach/git-cheatsheet.pdf>

4. gcc/g++与clang/clang++区别

首先gcc编译c，g++是针对c++，同样的clang是做c，clang++是处理c++。然后gcc支持更多平台，clang是后来发展的，编译效率更高，占用内存更小，但是支持的平台少于gcc所支持的。gcc/g++是几乎所有平台的默认官方编译器。而clang/clang++目前是部分平台的默认编译器。mac os上是默认的clang/clang++。而ubuntu上是默认的gcc/g++。

***gcc/g++ 和 clang/clang++ 都是 Linux 下常用的 C/C++ 编译器。gcc 是 GNU 亲儿子，Ubuntu 等常用发行版标配。clang 是后起之秀，配合 llvm，以优秀的前端闻名于世，现在已经是 Mac（XCode） 的默认编译器，微软等大公司都在往上靠，前景一片光明。知乎上对几个主流编译器的评价：***

***（1）前景最好的应该是clang，错误提示/编译速度都是一流。另外现在有很多代码自动完成/提示工具是基于clang的。clang编译出的debug文件已经可以用gdb调试了哈，vs2015也内置了clang来提供对cross platfrom的支持。***

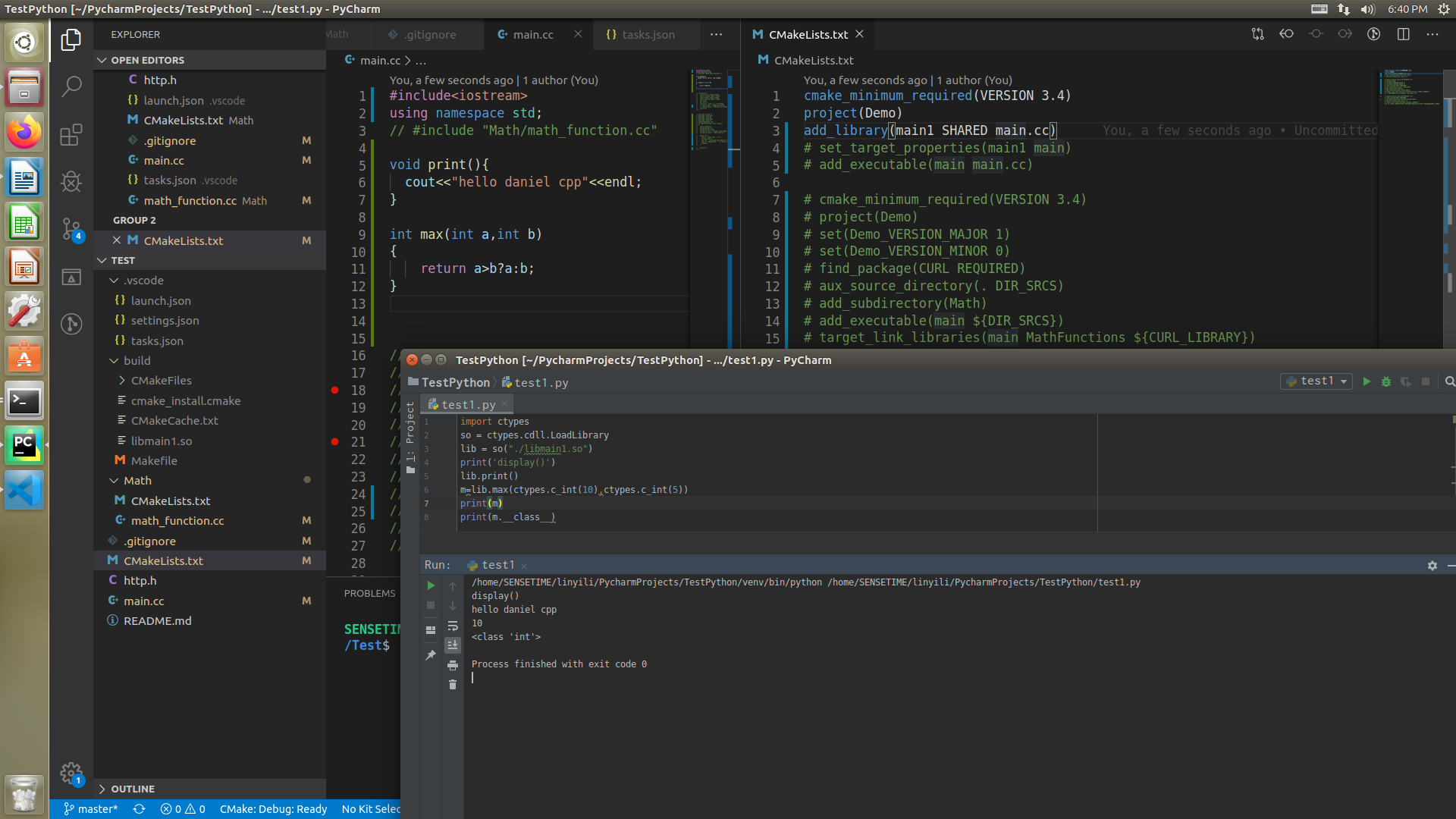
***（2）gcc/g++/msvc因为面向了具体的platform，所以老的项目里面可能用了他们自带的扩展/或者在彼此应用环境下有一些best practice，这个没有办法，还得接着用。***

5.cmake编译与cmake生成动态库

add\_library(main1 SHARED main.cc) // 生成动态库文件

add\_executable(main main.cc) // 生成可执行文件

6.cmake生成so动态库，使用python调用so文件



7.vscode简单项目的时候，配置launch.json需要判断有没有必要“preLaunchTask”字段。其中还需要注意的是"program": "${workspaceFolder}/build/test\_s"后面的test\_s是cmake生成的执行文件，而 **${workspaceFolder}** 是根目录地址。

8.vscode要想进行debug，则需要在cmake中设置debug类型：set(CMAKE\_BUILD\_TYPE Debug)

9.vscode中向前向后的快捷键名称是go forward和go back.

File=>Preferences=>Keyboard shortcuts可以设置为：

Go Back ===>>> alt+leftArrow

Go Forward ===>>> alt+rightArrow

10.VsCode中配置指定的python版本，在Preferences=>Setting=>搜索python.pythonPath。将python的版本路径更新在此处。