g++编译过程

1. 预处理 生成.i文件

g++ -E test.cpp -o test.i

1. 编译 生成.s文件

g++ -s test.i -o tset.s

1. 汇编 生成.o文件

g++ -c test.s -o test.o

1. 链接 生成bin文件(可执行)

g++ tset.o -o test

可以直接

g++ test.cpp -o test

-0[n] 优化源代码指令

g++ -o2 test.cpp #使用 -o2优化源代码,并输出可执行文件

-l 和 -L指定库文件 | 指定库文件路径

# -l参数(小写)就是用来指定程序要链接的库, -l参数紧接着就是库名

g++ -lglog test.cpp

VS 中打开终端的快捷键 Ctrl + `(Tab键上方那个键)

F11键全屏;

“ctrl” + “+” 实现放大字体;

“ctrl” + “-” 实现缩小字体;

“ctrl” + “shift” + ”enter” 向上加一行;

“ctrl” + ”enter” 向下加一行;

右键 format document格式化代码

编译指令

g++ main.cpp src/swap.cpp -Iinclude -o main

(在目录下生成main的可执行文件,-I(大写)Include)

ctrl + shfit + i 快速格式化代码

前言：

CMake是一个跨平台的安装编译工具，可以用简单的语句来描述所有平台的安装(编译过程)。

CMake可以说已经成为大部分C++开源项目标配

pwd查看当前目录路径

通过通过cmake编译的话,在修改了其中一个.cpp,只需要make即可!