

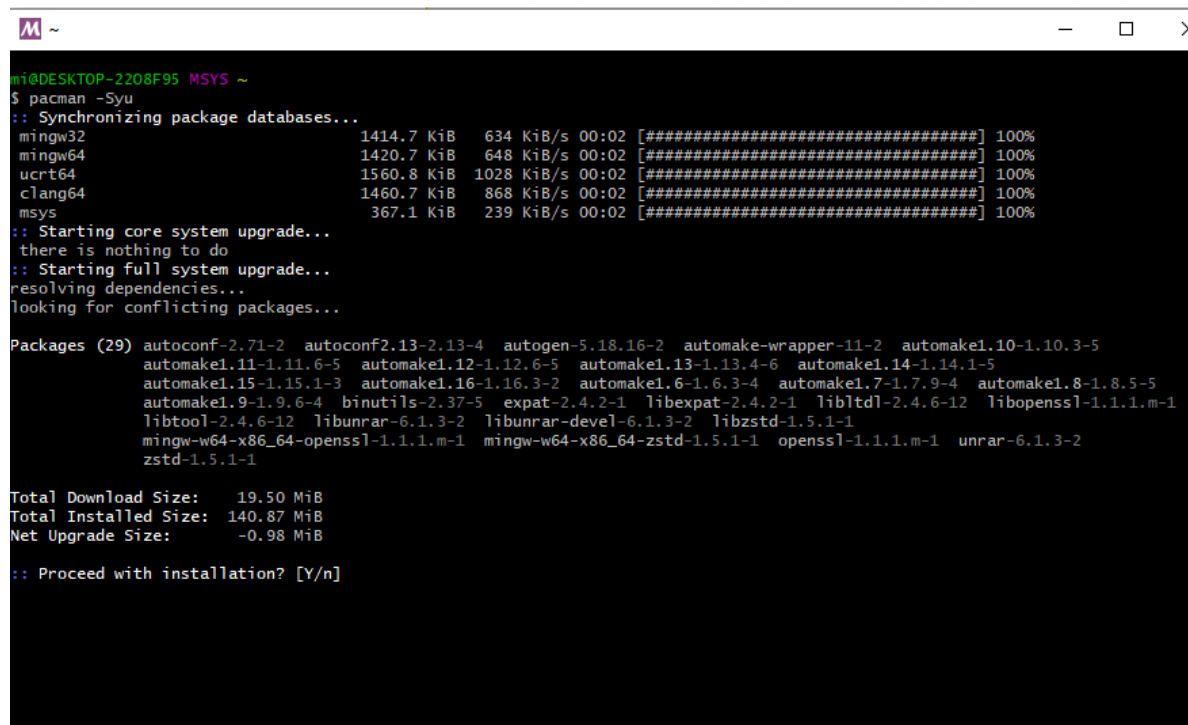
# Windows 环境下编译c++

## 1. 使用MSYS2安装MinGW 和 boost库

### MSYS2安装MinGw gcc编译器

[msys2官方教程](#)

1. 首先下载[msys2](#)，安装之。注意路径不要有中文，空格之类的。
2. 打开 MSYS2 MSYS，更新： `pacman -Syu`



```
mi@DESKTOP-2208F95 MSYS ~
$ pacman -Syu
:: Synchronizing package databases...
mingw32                  1414.7 KiB   634 KiB/s  00:02 [#####] 100%
mingw64                  1420.7 KiB   648 KiB/s  00:02 [#####] 100%
ucrt64                   1560.8 KiB  1028 KiB/s  00:02 [#####] 100%
clang64                  1460.7 KiB   868 KiB/s  00:02 [#####] 100%
msys                     367.1 KiB   239 KiB/s  00:02 [#####] 100%
:: Starting core system upgrade...
there is nothing to do
:: Starting full system upgrade...
resolving dependencies...
looking for conflicting packages...

Packages (29) autoconf-2.71-2  autoconf2.13-2.13-4  autogen-5.18.16-2  automake-wrapper-11-2  automake1.10-1.10.3-5
               automake1.11-1.11.6-5  automake1.12-1.12.6-5  automake1.13-1.13.4-6  automake1.14-1.14.1-5
               automake1.15-1.15.1-3  automake1.16-1.16.3-2  automake1.6-1.6.3-4  automake1.7-1.7.9-4  automake1.8-1.8.5-5
               automake1.9-1.9.6-4  binutils-2.37-5  expat-2.4.2-1  libexpat-2.4.2-1  libltdl-2.4.6-12  libopenssl-1.1.1.m-1
               libtool-2.4.6-12  libunrar-6.1.3-2  libunrar-devel-6.1.3-2  libzstd-1.5.1-1
               mingw-w64-x86_64-openssl-1.1.1.m-1  mingw-w64-x86_64-zstd-1.5.1-1  openssl-1.1.1.m-1  unrar-6.1.3-2
               zstd-1.5.1-1

Total Download Size:    19.50 MiB
Total Installed Size:  140.87 MiB
Net Upgrade Size:       -0.98 MiB

:: Proceed with installation? [Y/n]
```

- 一路回车确认：

```
~
Total (29/29)                               19.5 MiB  4.22 MiB/s  00:05 [#####] 100%
(29/29) checking keys in keyring [#####] 100%
(29/29) checking package integrity [#####] 100%
(29/29) loading package files [#####] 100%
(29/29) checking for file conflicts [#####] 100%
(29/29) checking available disk space [#####] 100%
:: Processing package changes...
( 1/29) upgrading autoconf [#####] 100%
( 2/29) upgrading autoconf2.13 [#####] 100%
( 3/29) upgrading libltdl [#####] 100%
( 4/29) upgrading autogen [#####] 100%
( 5/29) upgrading automake1.6 [#####] 100%
( 6/29) upgrading automake1.7 [#####] 100%
( 7/29) upgrading automake1.8 [#####] 100%
( 8/29) upgrading automake1.9 [#####] 100%
( 9/29) upgrading automake1.10 [#####] 100%
(10/29) upgrading automake1.11 [#####] 100%
(11/29) upgrading automake1.12 [#####] 100%
(12/29) upgrading automake1.13 [#####] 100%
(13/29) upgrading automake1.14 [#####] 100%
(14/29) upgrading automake1.15 [#####] 100%
(15/29) upgrading automake1.16 [#####] 100%
(16/29) upgrading automake-wrapper [#####] 100%
(17/29) upgrading binutils [#####] 100%
(18/29) upgrading expat [#####] 100%
(19/29) upgrading libexpat [#####] 100%
(20/29) upgrading libopenssl [#####] 100%
(21/29) upgrading libtool [#####] 100%
(22/29) upgrading libunrar [#####] 100%
(23/29) upgrading libunrar-devel [#####] 100%
(24/29) upgrading libzstd [#####] 100%
(25/29) upgrading mingw-w64-x86_64-openssl [#####] 100%
(26/29) upgrading mingw-w64-x86_64-zstd [#####] 100%
(27/29) upgrading openssl [#####] 100%
(28/29) upgrading unrar [#####] 100%
(29/29) upgrading zstd [#####] 100%
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

mi@DESKTOP-2208F95 MSYS ~
$
```

- 继续更新: `pacman -Su` (我这里是已经更新完的, 显示不需要更新)

```
mi@DESKTOP-2208F95 MSYS ~
$ pacman -Su
:: Starting core system upgrade...
there is nothing to do
:: Starting full system upgrade...
there is nothing to do
```

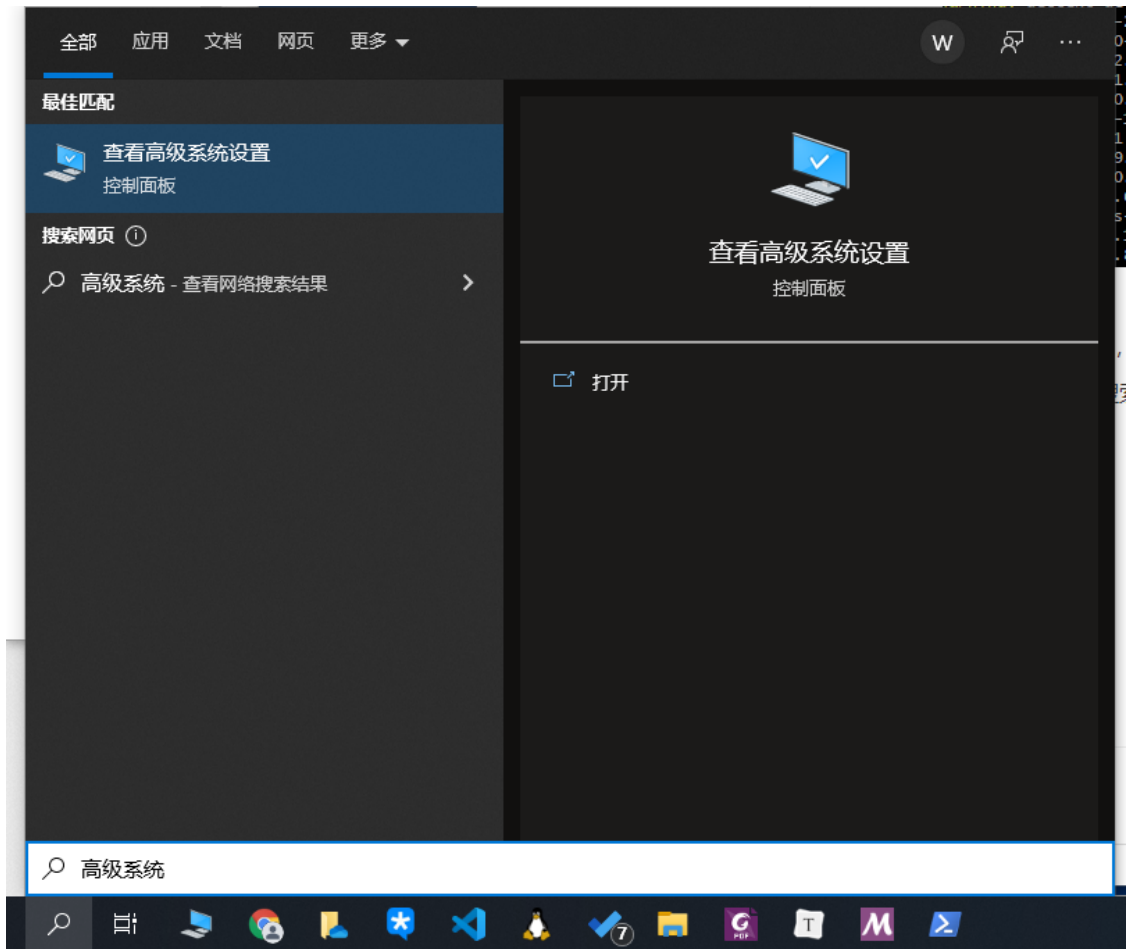
3. 安装gcc: `pacman -S --needed base-devel mingw-w64-x86_64-toolchain`, 我这里也是之前就安装好了, 所以显示跳过。

```
pacman -S --needed base-devel mingw-w64-x86_64-toolchain
warning: asciidoc-9.1.1-1 is up to date -- skipping
warning: binutils-2.37-5 is up to date -- skipping
warning: bison-3.8.2-1 is up to date -- skipping
warning: byacc-20210619-1 is up to date -- skipping
warning: diffstat-1.64-1 is up to date -- skipping
warning: diffutils-3.8-1 is up to date -- skipping
warning: dos2unix-7.4.2-1 is up to date -- skipping
warning: file-5.41-1 is up to date -- skipping
warning: flex-2.6.4-2 is up to date -- skipping
warning: gawk-5.1.0-2 is up to date -- skipping
warning: gdb-10.2-3 is up to date -- skipping
warning: gettext-0.19.8.1-1 is up to date -- skipping
warning: gettext-devel-0.19.8.1-1 is up to date -- skipping
warning: gperf-3.1-2 is up to date -- skipping
warning: grep-1~3.0-3 is up to date -- skipping
warning: groff-1.22.4-1 is up to date -- skipping
warning: help2man-1.48.5-1 is up to date -- skipping
warning: intltool-0.51.0-2 is up to date -- skipping
warning: m4-1.4.19-1 is up to date -- skipping
warning: make-4.3-1 is up to date -- skipping
warning: man-db-2.9.4-1 is up to date -- skipping
warning: pacman-6.0.1-8 is up to date -- skipping
warning: patch-2.7.6-1 is up to date -- skipping
warning: patchutils-0.4.2-2 is up to date -- skipping
warning: perl-5.32.1-1 is up to date -- skipping
warning: pkgconf-1.8.0-1 is up to date -- skipping
```

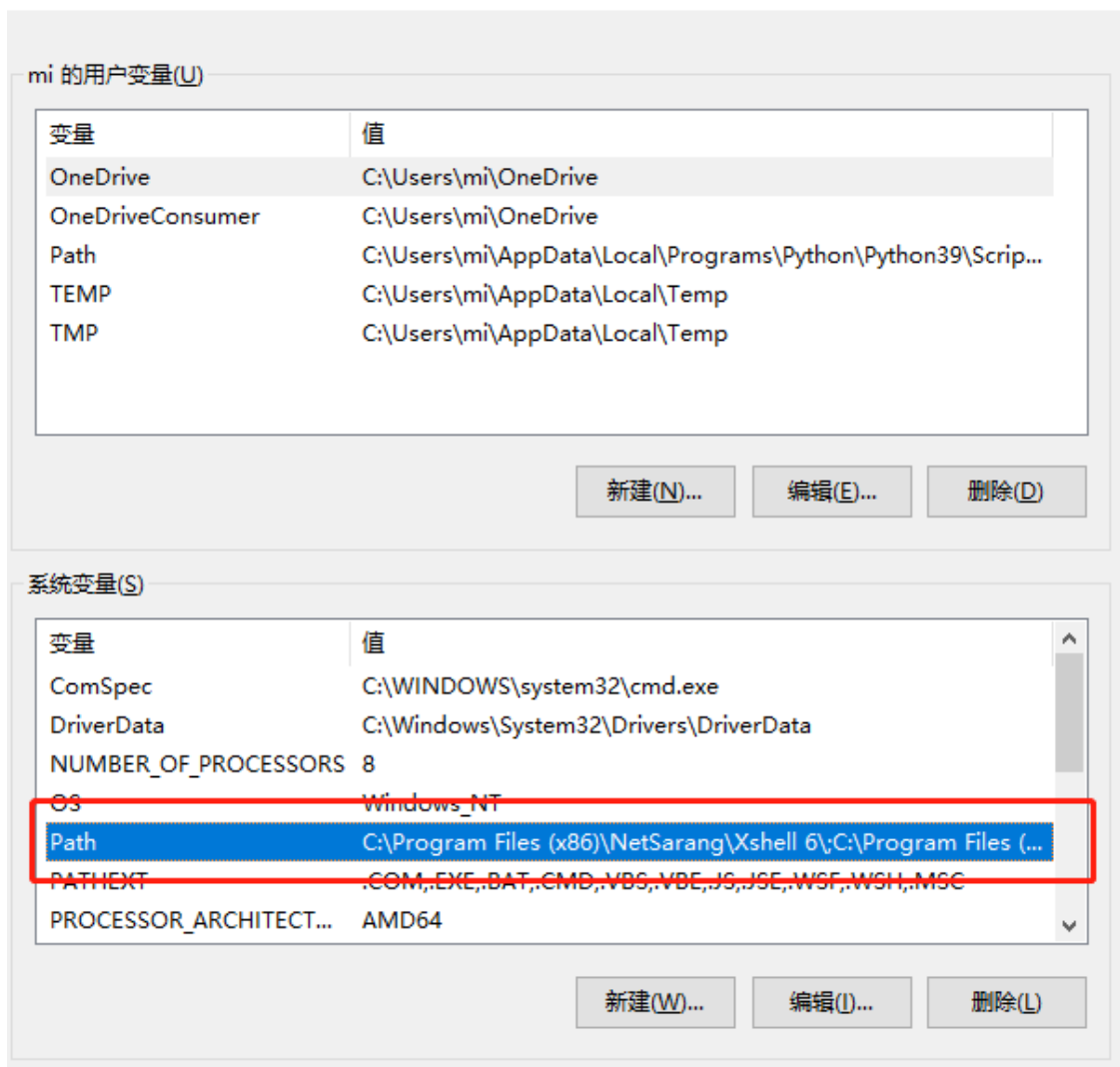
4. 环境变量

找到MinGW的bin目录，添加到系统环境变量中,我的目录是 `C:\msys64\mingw64\bin`

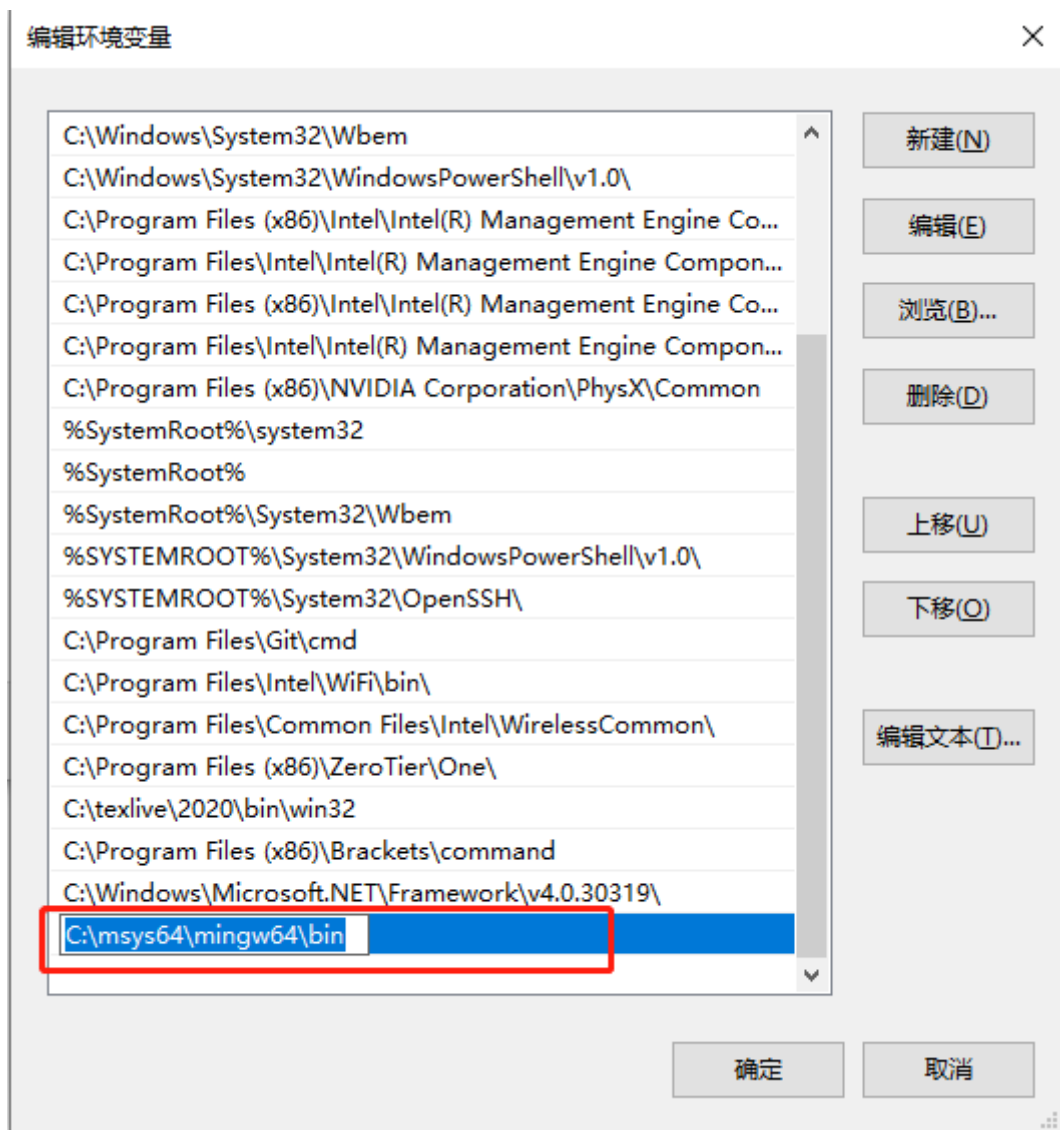
- Windows左下角搜索框中搜索 高级系统设置，点击它。



- 环境变量->找到系统变量的 `Path`，双击它，点击 新建

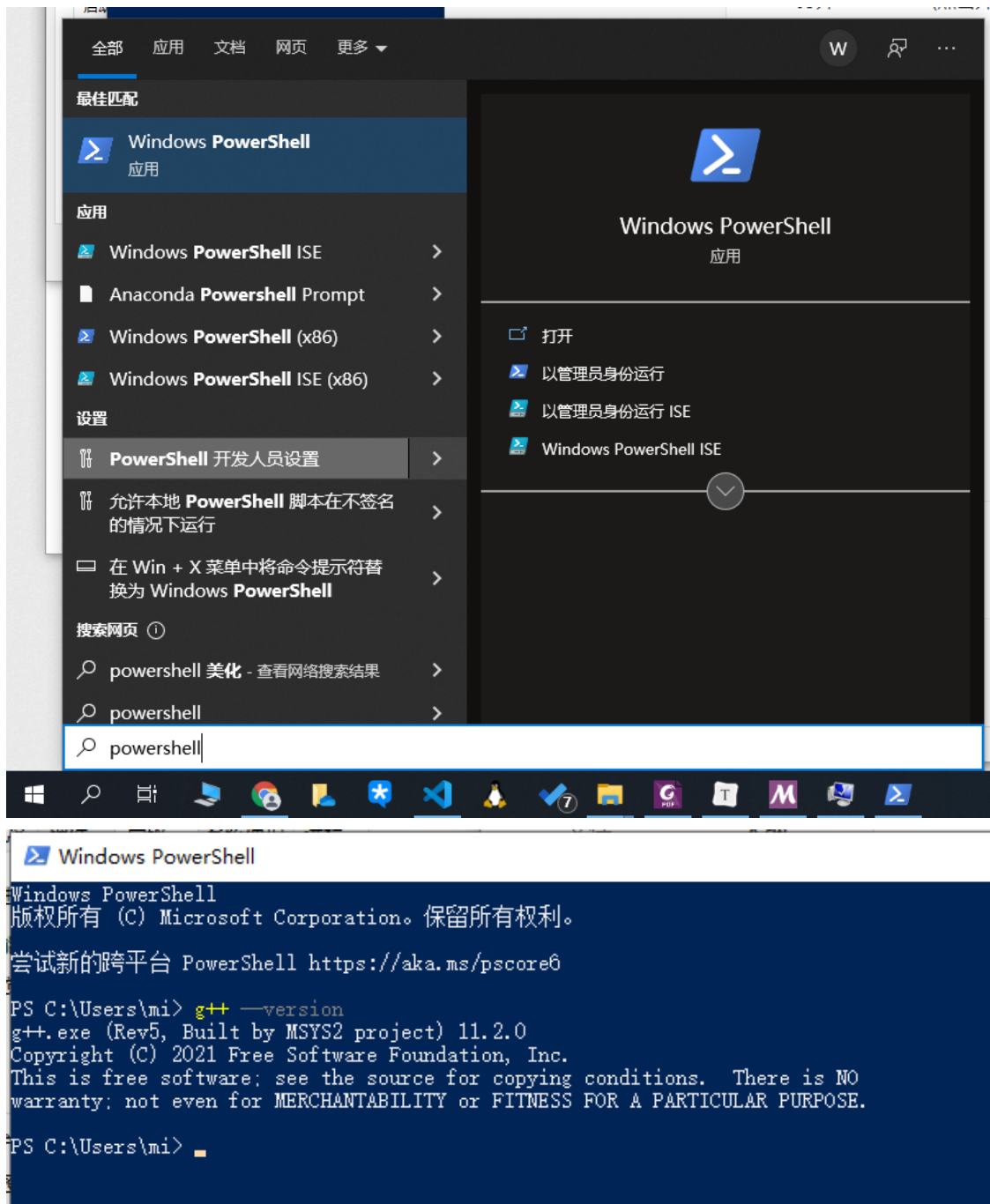


- 把目录填进去，确定。



##### 5. 确认安装成功

打开PowerShell(点击开始菜单可以搜索到), 输入 `g++ --version`, 出现版本信息, 说明gcc编译器安装成功



## boost 库安装

Boost是一个强大的C++程序库，我们的Lab S也会用到它。

[Boost 官方教程](#)

### 1. 使用MSYS2安装

MSYS2也有维护[Boost](#)，打开MSYS MinGW，使用 `pacman` 安装：`pacman -S mingw-w64-x86_64-boost`

```
mi@DESKTOP-2208F95 MSYS ~
$ pacman -S mingw-w64-x86_64-boost
warning: mingw-w64-x86_64-boost-1.78.0-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) mingw-w64-x86_64-boost-1.78.0-1

Total Installed Size: 323.88 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n]
```

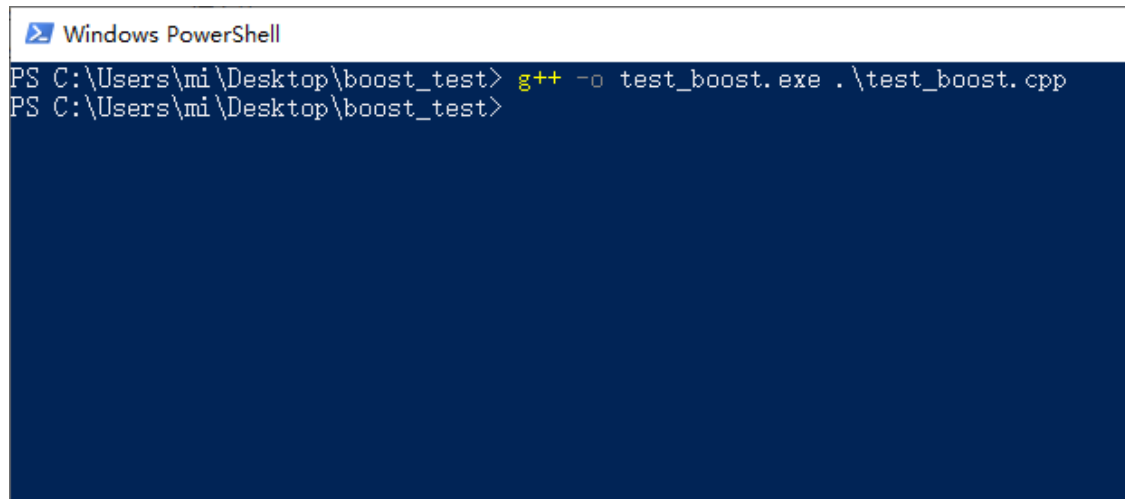
## 2. 测试安装成功

使用boost官网中一个计算 $x * 3$ 的例子，新建文件 `test_boost.cpp`，复制下面代码

```
1  #include <boost/lambda/lambda.hpp>
2  #include <iostream>
3  #include <iterator>
4  #include <algorithm>
5
6  int main()
7  {
8      using namespace boost::lambda;
9      typedef std::istream_iterator<int> in;
10
11     std::for_each(
12         in(std::cin), in(), std::cout << (_1 * 3) << " " );
13 }
```

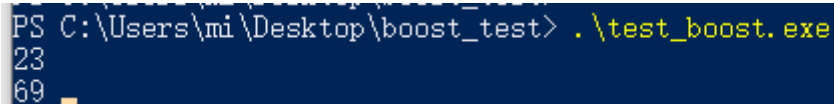
在文件同一文件夹下打开PowerShell（例如，我是在 `C:\Users\mi\Desktop\boost_test`），按住键盘上的 `shift` 在文件夹内右击鼠标->在此处打开Powershell窗口。

输入 `g++ -o test_boost.exe .\test_boost.cpp`



```
Windows PowerShell
PS C:\Users\mi\Desktop\boost_test> g++ -o test_boost.exe .\test_boost.cpp
PS C:\Users\mi\Desktop\boost_test>
```

测试 `.\test_boost.exe`

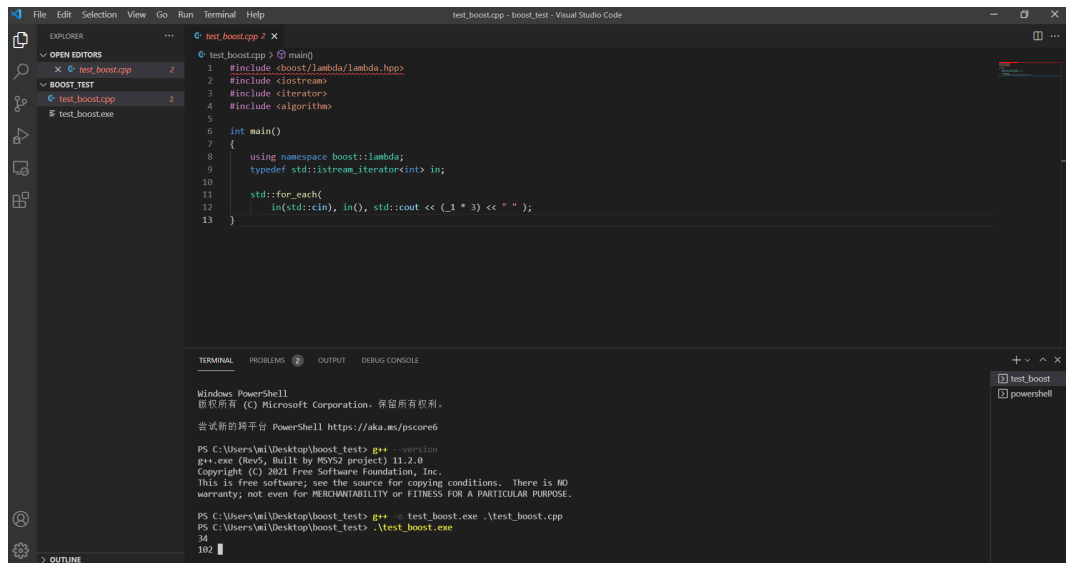


```
PS C:\Users\mi\Desktop\boost_test> .\test_boost.exe
23
69
```

- VSCode

也可以直接在vscode中打开终端：

1. 在文件夹空白中右击鼠标，通过Code打开。
2. 点击菜单栏的 Terminal -> New Terminal
3. 然后在打开的终端中输入命令进行编译和运行



## 2. cmake

我们可以使用命令行编译c++代码。但是当项目变大，文件很多时，使用命令行编译会变得又慢又复杂。makefile可以用来管理项目中这些文件的编译。

MinGW中也包含了make，因此可以直接在终端中使用 `mingw32-make.exe`。

但是，不同平台又有各自的make工具，如GNU的 `make`，Microsoft的 `nmake`。如果同一份代码想在不同平台上编译，还需要重新编写makefile文件。而 `cmake` 则可以实现使用同一份CMakefile为不同平台生成相对应的make文件。

### cmake 安装

可以直接在官网下载并安装，也可以使用MSYS2安装，这里使用后者。

- 打开MinGW-64，输入命令安装 `pacman -S mingw-w64-x86_64-cmake`



```
mi@DESKTOP-2208F95 MINGW64 ~
$ pacman -S mingw-w64-x86_64-cmake
resolving dependencies...
looking for conflicting packages...

Packages (18) mingw-w64-x86_64-brotli-1.0.9-4 mingw-w64-x86_64-c-ares-1.17.2-1
mingw-w64-x86_64-curl-7.80.0-1 mingw-w64-x86_64-jansson-2.14-1
mingw-w64-x86_64-jemalloc-5.2.1-2
mingw-w64-x86_64-jsoncpp-1.9.4-1
mingw-w64-x86_64-libarchive-3.5.2-1
mingw-w64-x86_64-libidn2-2.3.1-1
mingw-w64-x86_64-libssl-0.21.1-4
mingw-w64-x86_64-libssh2-1.10.0-1
mingw-w64-x86_64-libunistring-0.9.10-4
mingw-w64-x86_64-libuv-1.42.0-3
mingw-w64-x86_64-libxml2-2.9.12-3 mingw-w64-x86_64-lz4-1.9.3-1
mingw-w64-x86_64-nettle-3.7.3-3
mingw-w64-x86_64-nghttp2-1.45.1-1 mingw-w64-x86_64-rhash-1.4.2-1
mingw-w64-x86_64-cmake-3.22.1-1

Total Download Size: 17.00 MiB
Total Installed Size: 130.28 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
mingw-w64-x86_64... 929.5 KiB 672 KiB/s 00:01 [#####] 100%
mingw-w64-x86_64... 729.7 KiB 519 KiB/s 00:01 [#####] 100%
mingw-w64-x86_64... 1490.0 KiB 884 KiB/s 00:02 [#####] 100%
mingw-w64-x86_64... 720.2 KiB 1611 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 386.7 KiB 1026 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 1297.5 KiB 620 KiB/s 00:02 [#####] 100%
mingw-w64-x86_64... 571.0 KiB 784 KiB/s 00:01 [#####] 100%
mingw-w64-x86_64... 344.6 KiB 783 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 255.0 KiB 763 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 240.6 KiB 678 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 221.8 KiB 626 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 200.3 KiB 569 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 199.3 KiB 623 KiB/s 00:00 [#####] 100%
mingw-w64-x86_64... 169.3 KiB 494 KiB/s 00:00 [#####] 100%
```

- 安装之后，在Powershell中输入 `cmake.exe -h`，打印帮助信息，则安装成功。

```
PS C:\Users\mi\Desktop\boost_test> cmake.exe -h
Usage

cmake [options] <path-to-source>
cmake [options] <path-to-existing-build>
cmake [options] -S <path-to-source> -B <path-to-build>

Specify a source directory to (re-)generate a build system for it in the
current working directory. Specify an existing build directory to
re-generate its build system.

Options
-S <path-to-source>           = Explicitly specify a source directory.
-B <path-to-build>            = Explicitly specify a build directory.
-C <initial-cache>            = Pre-load a script to populate the cache.
-D <var>[:<type>]=<value>    = Create or update a cmake cache entry.
-U <globbing_expr>            = Remove matching entries from CMake cache.
-G <generator-name>          = Specify a build system generator.
-T <toolset-name>             = Specify toolset name if supported by
                              generator.
-A <platform-name>           = Specify platform name if supported by
                              generator.
--toolchain <file>            = Specify toolchain file
                              [CMAKE_TOOLCHAIN_FILE].
--install-prefix <directory> = Specify install directory
                              [CMAKE_INSTALL_PREFIX].
-Wdev                         = Enable developer warnings.
-Wno-dev                      = Suppress developer warnings.
-Werror=dev                   = Make developer warnings errors.
-Wno-error=dev                = Make developer warnings not errors.
-Wdeprecated                  = Enable deprecation warnings.
-Wno-deprecated               = Suppress deprecation warnings.
-Werror=deprecated            = Make deprecated macro and function warnings
                              errors.
-Wno-error=deprecated         = Make deprecated macro and function warnings
```

## 使用cmake为Lab S生成makefile

- 在Lab S文件夹下打开powershell，输入 `cmake.exe -G"MinGW Makefiles" .\CMakeLists.txt`

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\mi\Desktop\ISC21Lab-Design-main\simulator> cmake.exe -G"MinGW Makefiles" .\CMakeLists.txt
-- The C compiler identification is GNU 11.2.0
-- The CXX compiler identification is GNU 11.2.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/msys64/mingw64/bin/gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/msys64/mingw64/bin/g++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
CMake Warning at C:/msys64/mingw64/share/cmake-3.22/Modules/FindBoost.cmake:1369 (message):
  New Boost version may have incorrect or missing dependencies and imported
  targets
Call Stack (most recent call first):
  C:/msys64/mingw64/share/cmake-3.22/Modules/FindBoost.cmake:1492 (_Boost_COMPONENT_DEPENDENCIES)
  C:/msys64/mingw64/share/cmake-3.22/Modules/FindBoost.cmake:2102 (_Boost_MISSING_DEPENDENCIES)
  CMakeLists.txt:10 (find_package)

```

- 发现文件夹下多了一个 Makefile 文件，此时使用 `mingw32-make.exe` 就可以编译Lab S了。（这里假设Lab S中 `TO BE DONE` 已经完成，没完成会报错）

```

PS C:\Users\mi\Desktop\ISC21Lab-Design-main\simulator> mingw32-make.exe
Consolidate compiler generated dependencies of target lc3simulator
[ 20%] Building CXX object CMakeFiles/lc3simulator.dir/src/main.cpp.obj
[ 40%] Building CXX object CMakeFiles/lc3simulator.dir/src/memory.cpp.obj
[ 60%] Building CXX object CMakeFiles/lc3simulator.dir/src/register.cpp.obj
[ 80%] Building CXX object CMakeFiles/lc3simulator.dir/src/simulator.cpp.obj
[100%] Linking CXX executable lc3simulator.exe
[100%] Built target lc3simulator
PS C:\Users\mi\Desktop\ISC21Lab-Design-main\simulator>

```

## 其他平台的生成器

- |    |                              |   |
|----|------------------------------|---|
| 1  | visual studio 17 2022        | = Generates Visual Studio 2022 project files.   |
| 2  |                              | Use -A option to specify architecture.  |
| 3  | visual studio 16 2019        | = Generates Visual Studio 2019 project files.   |
| 4  |                              | Use -A option to specify architecture.  |
| 5  | visual studio 15 2017 [arch] | = Generates Visual Studio 2017 project files.   |
| 6  |                              | Optional [arch] can be "win64" or "ARM".  |
| 7  | visual studio 14 2015 [arch] | = Generates Visual Studio 2015 project files.   |
| 8  |                              | Optional [arch] can be "win64" or "ARM".  |
| 9  | visual studio 12 2013 [arch] | = Generates Visual Studio 2013 project files.   |
| 10 |                              | Optional [arch] can be "win64" or "ARM".  |
| 11 | visual studio 11 2012 [arch] | = Generates Visual Studio 2012 project files.   |
| 12 |                              | Optional [arch] can be "win64" or "ARM".  |
| 13 | visual studio 10 2010 [arch] | = Deprecated. Generates visual studio 2010 project files. Optional [arch] can be "win64" or "IA64". |
| 14 |                              |   |
| 15 |                              |   |

```

16  visual Studio 9 2008 [arch] = Generates Visual Studio 2008 project
    files.
17                                     Optional [arch] can be "win64" or "IA64".
18  Borland Makefiles             = Generates Borland makefiles.
19  NMake Makefiles               = Generates NMake makefiles.
20  NMake Makefiles JOM           = Generates JOM makefiles.
21  MSYS Makefiles                = Generates MSYS makefiles.
22  MinGW Makefiles               = Generates a make file for use with
23                                mingw32-make.
24  Green Hills MULTI             = Generates Green Hills MULTI files
25                                (experimental, work-in-progress).
26  Unix Makefiles                = Generates standard UNIX makefiles.
27  Ninja                         = Generates build.ninja files.
28  Ninja Multi-Config             = Generates build-<Config>.ninja files.
29  Watcom WMake                  = Generates Watcom WMake makefiles.
30  CodeBlocks - MinGW Makefiles = Generates CodeBlocks project files.
31  CodeBlocks - NMake Makefiles = Generates CodeBlocks project files.
32  CodeBlocks - NMake Makefiles JOM
33                                = Generates CodeBlocks project files.
34  CodeBlocks - Ninja             = Generates CodeBlocks project files.
35  CodeBlocks - Unix Makefiles   = Generates CodeBlocks project files.
36  CodeLite - MinGW Makefiles    = Generates CodeLite project files.
37  CodeLite - NMake Makefiles    = Generates CodeLite project files.
38  CodeLite - Ninja             = Generates CodeLite project files.
39  CodeLite - Unix Makefiles     = Generates CodeLite project files.
40  Eclipse CDT4 - NMake Makefiles
41                                = Generates Eclipse CDT 4.0 project files.
42  Eclipse CDT4 - MinGW Makefiles
43                                = Generates Eclipse CDT 4.0 project files.
44  Eclipse CDT4 - Ninja          = Generates Eclipse CDT 4.0 project files.
45  Eclipse CDT4 - Unix Makefiles = Generates Eclipse CDT 4.0 project files.
46  Kate - MinGW Makefiles        = Generates Kate project files.
47  Kate - NMake Makefiles        = Generates Kate project files.
48  Kate - Ninja                  = Generates Kate project files.
49  Kate - Unix Makefiles         = Generates Kate project files.
50  Sublime Text 2 - MinGW Makefiles
51                                = Generates Sublime Text 2 project files.
52  Sublime Text 2 - NMake Makefiles
53                                = Generates Sublime Text 2 project files.
54  Sublime Text 2 - Ninja        = Generates Sublime Text 2 project files.
55  Sublime Text 2 - Unix Makefiles
56                                = Generates Sublime Text 2 project files.

```