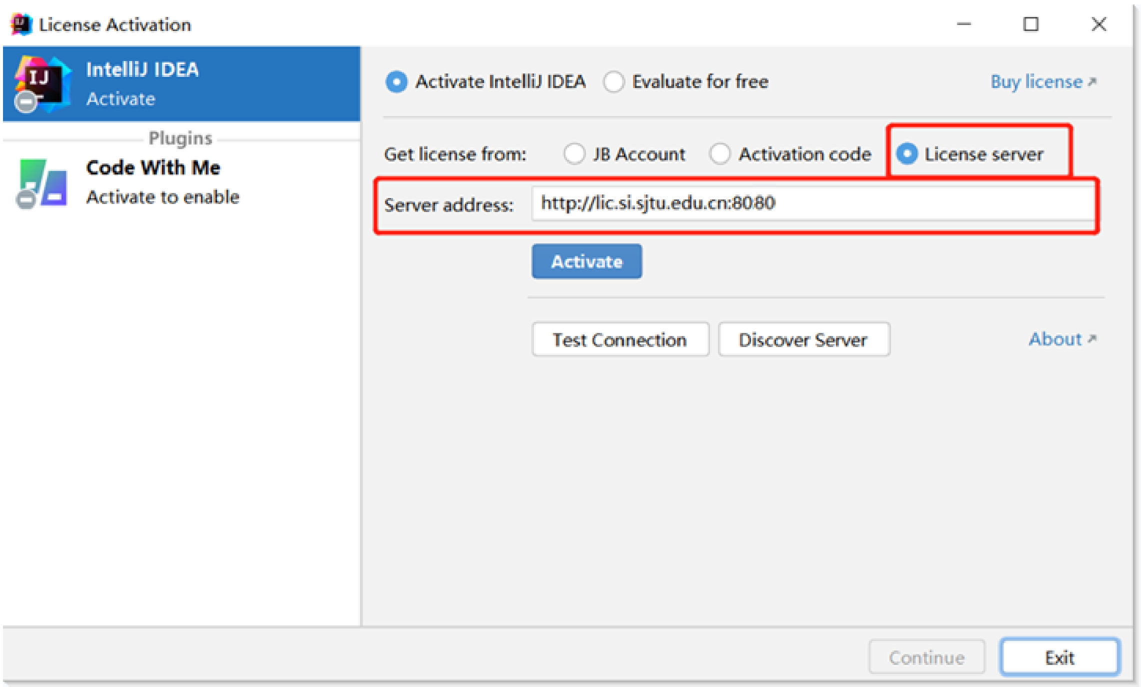
FAQ about C/C++ environment setup on Windows

VG101 FA2021 TA Group

1. My SJTU e-mail cannot be used to activate the CLion, what should I do?

For SJTU students, usually there are two ways to activate the CLion,

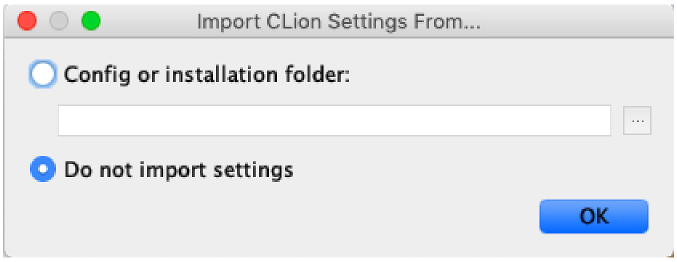
- Apply in: https://www.jetbrains.com/shop/eform/students **(Official Document)**, with your CHSI certification from https://my.chsi.com.cn/archive/gdjy/xj/show.action, or the photo of your student card.
- Or download **JetBrains All in One** from this page, and then follow the steps on this page, notice that the **server address** is http://lic.si.sjtu.edu.cn:8080, the address directly copied from the page maybe wrong.
    - 

      ● Activate IntelliJ IDEA ○ Evaluate for free

      Get license from: ○ JB Account ○ Activation code ● License server

      Server address: http://lic.si.sjtu.edu.cn:8080

2. What should I choose for "Import Clion Settings"?



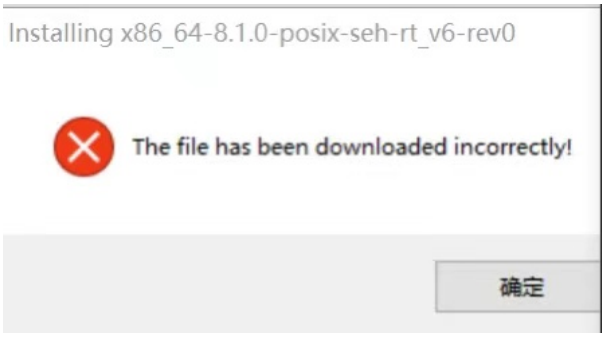
Import CLion Settings From...

○ Config or installation folder:

● Do not import settings

Since this is the first JetBrains product you install on your computer, there is no existing config, so just choose **"Do not import settings"**.

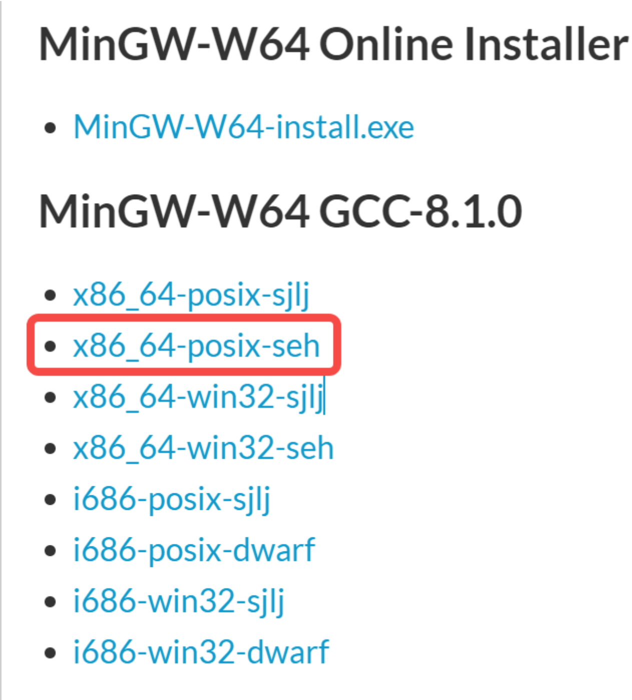
3. When installing MinGW-w64, it shows "The file has been downloaded incorrectly"



Installing x86_64-8.1.0-posix-seh-rt_v6-rev0

The file has been downloaded incorrectly!

The default installer of MinGW-w64 is an online installer, due to some network issues (especially without VPN), errors may occur in the downloading process. If the problem keeps happening, you can download the entire package of MinGW-w64. On https://sourceforge.net/projects/mingw-w64/files/, choose **x86_64-posix-seh**.



MinGW-W64 Online Installer

- MinGW-W64-install.exe

MinGW-W64 GCC-8.1.0

- x86_64-posix-sjlj
- x86_64-posix-seh
- x86_64-win32-sjlj
- x86_64-win32-seh
- i686-posix-sjlj
- i686-posix-dwarf
- i686-win32-sjlj
- i686-win32-dwarf

And then unzip the downloaded `.7z` file, if you do not have corresponding unzip software, you may install "7-Zip". Notice that in this case, the unzipped file is already the final form of MinGW-w64. You can move the folder to an ideal place (without Chinese characters), copy the path of the folder, and then follow the steps in next question.

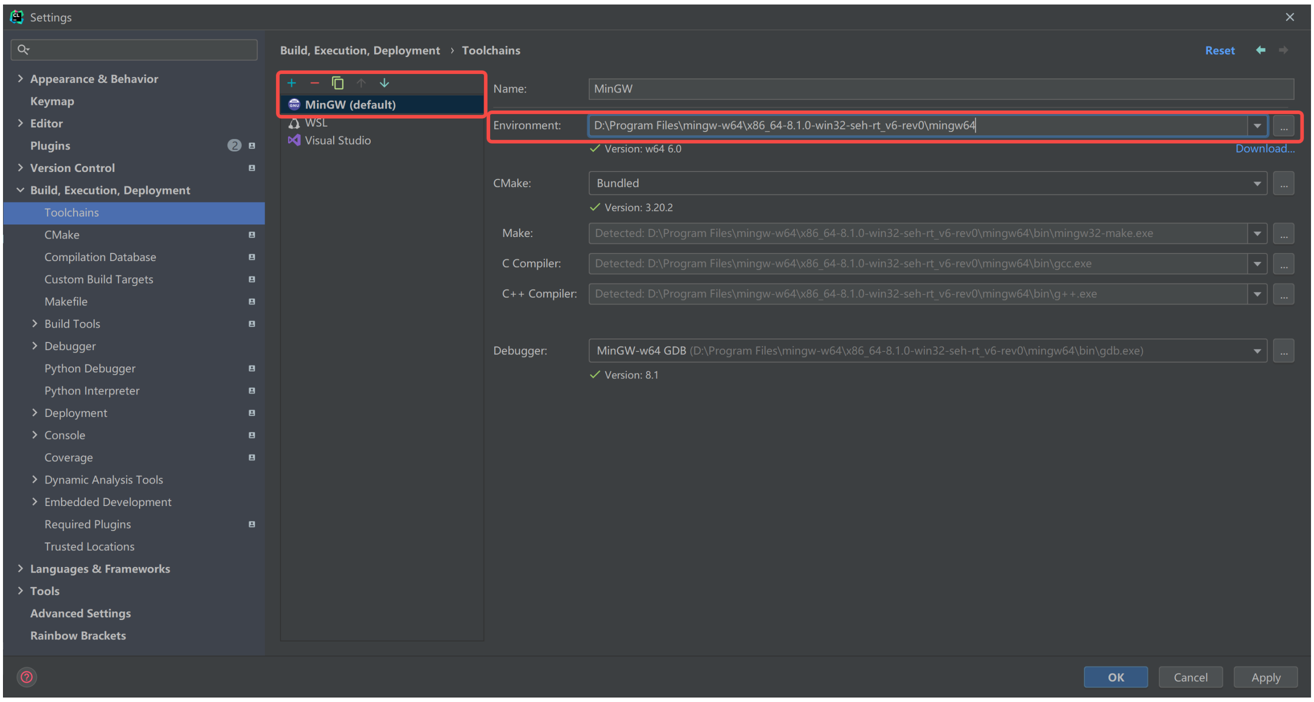
4. I have just installed MinGW32/MinGW-w64, where and how can I make use of it?

Usually there are two approaches,

In CLion (Required)

On the left corner of the interface of CLion, click **"File - Settings - Build, Execution, Deployment - Toolchains - MinGW"**, first use "Up" to set MinGW as the **default/first** toolchain, and then type the **installation path** of your MinGW32 or MinGW-w64 into the **"Environment"** blank.

Here are two samples,



Name: MinGW

Environment: D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64

✓ Version: w64 6.0

CMake: Bundled

✓ Version: 3.20.2

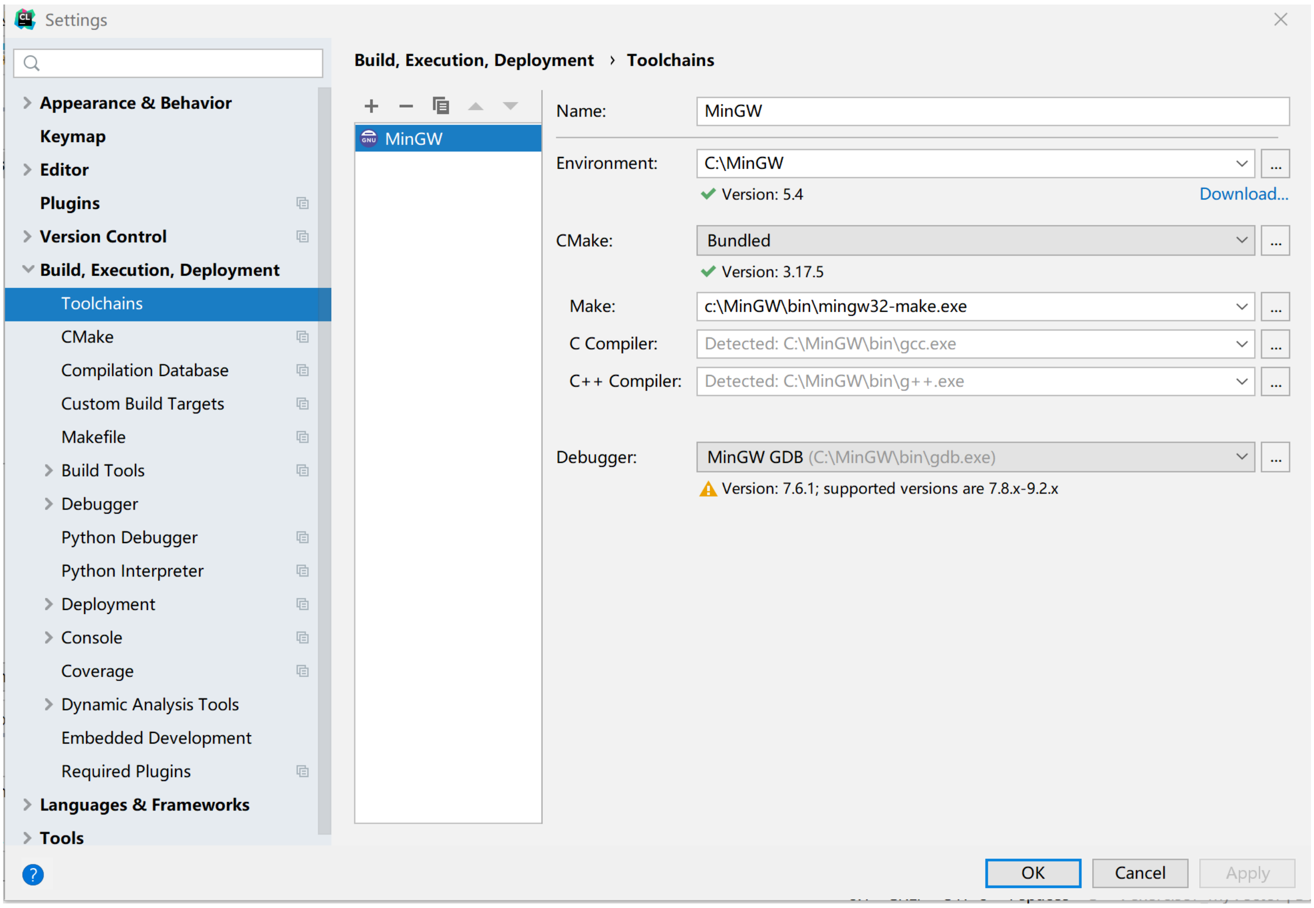
Make: Detected: D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin\mingw32-make.exe

C Compiler: Detected: D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin\gcc.exe

C++ Compiler: Detected: D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin\g++.exe

Debugger: MinGW-w64 GDB (D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin\gdb.exe)

✓ Version: 8.1



Name: MinGW

Environment: C:\MinGW

✓ Version: 5.4

CMake: Bundled

✓ Version: 3.17.5

Make: c:\MinGW\bin\mingw32-make.exe

C Compiler: Detected: C:\MinGW\bin\gcc.exe

C++ Compiler: Detected: C:\MinGW\bin\g++.exe

Debugger: MinGW GDB (C:\MinGW\bin\gdb.exe)

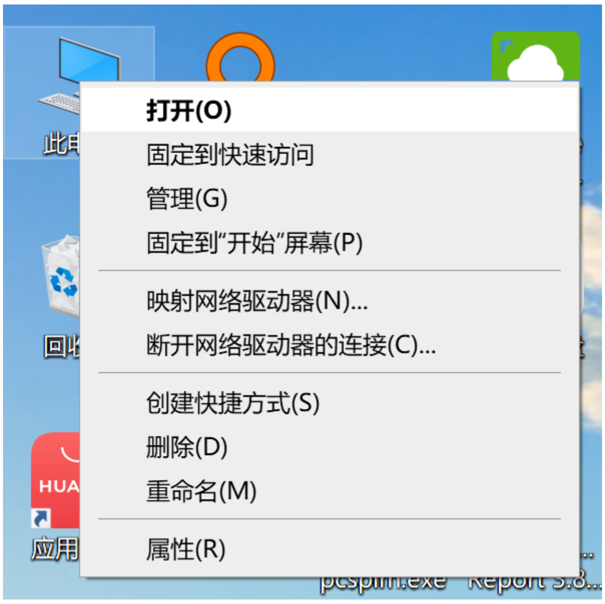
⚠ Version: 7.6.1; supported versions are 7.8.x-9.2.x

Then CLion will automatically detect the Make, C Compiler, C++ Compiler and Debugger. If you see some warnings about **CMake**, check the Question 5.

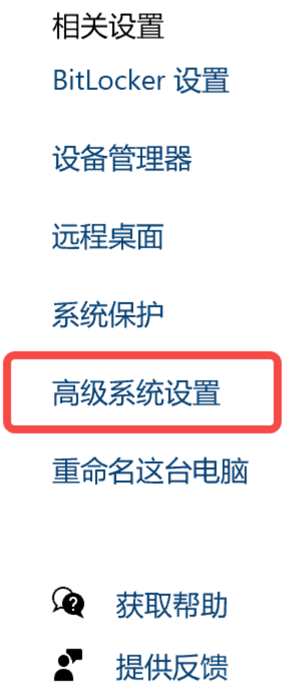
In Command-Line Interfaces (Cmd/PowerShell) by modifying the environment path

Note: Since I cannot change my Windows language to English, the following part will be partly in Chinese.

1. On your desktop, right click "此电脑 - 属性"



2. On the right side, click "高级系统设置"



相关设置

BitLocker 设置

设备管理器

远程桌面

系统保护

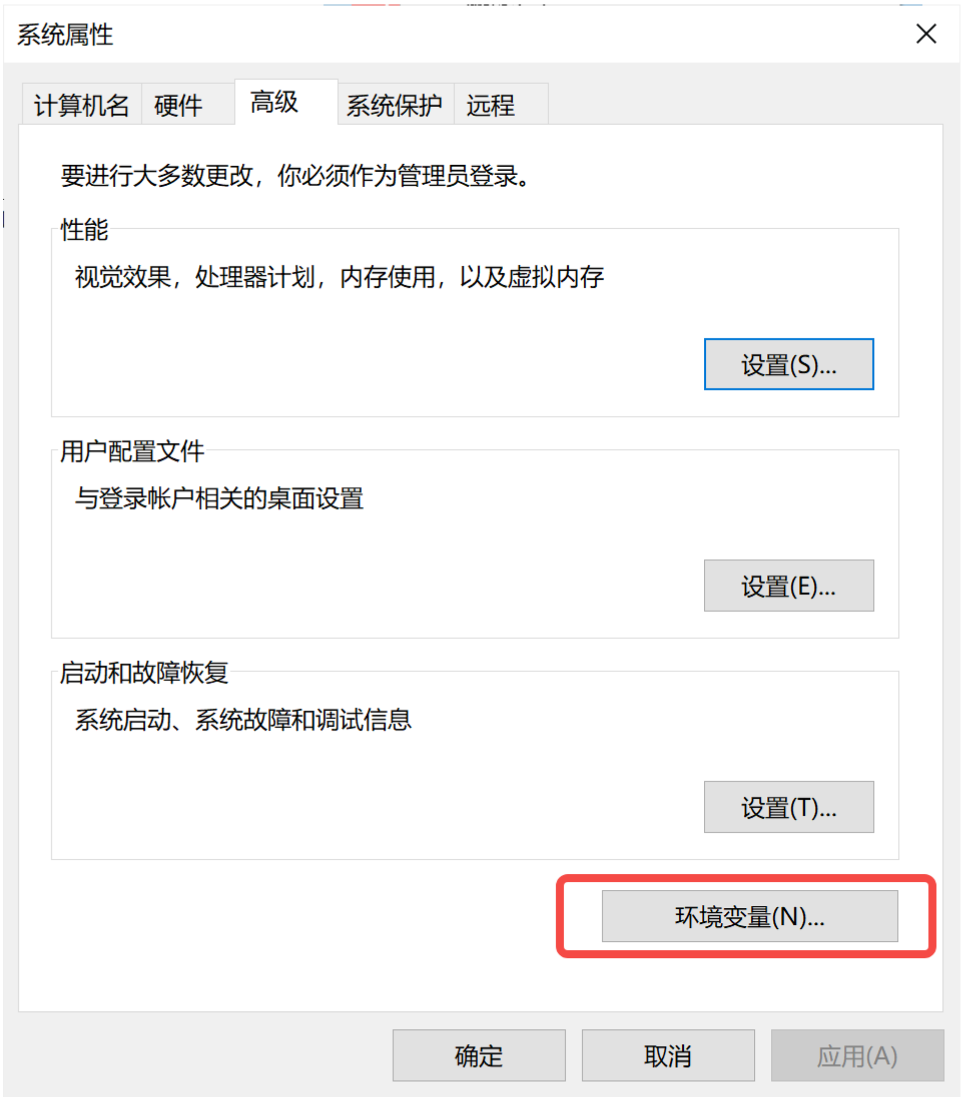
高级系统设置

重命名这台电脑

获取帮助

提供反馈

3. Click "环境变量"



系统属性

计算机名 | 硬件 | 高级 | 系统保护 | 远程

要进行大多数更改，你必须作为管理员登录。

性能

视觉效果，处理器计划，内存使用，以及虚拟内存

设置(S)...

用户配置文件

与登录帐户相关的桌面设置

设置(E)...

启动和故障恢复

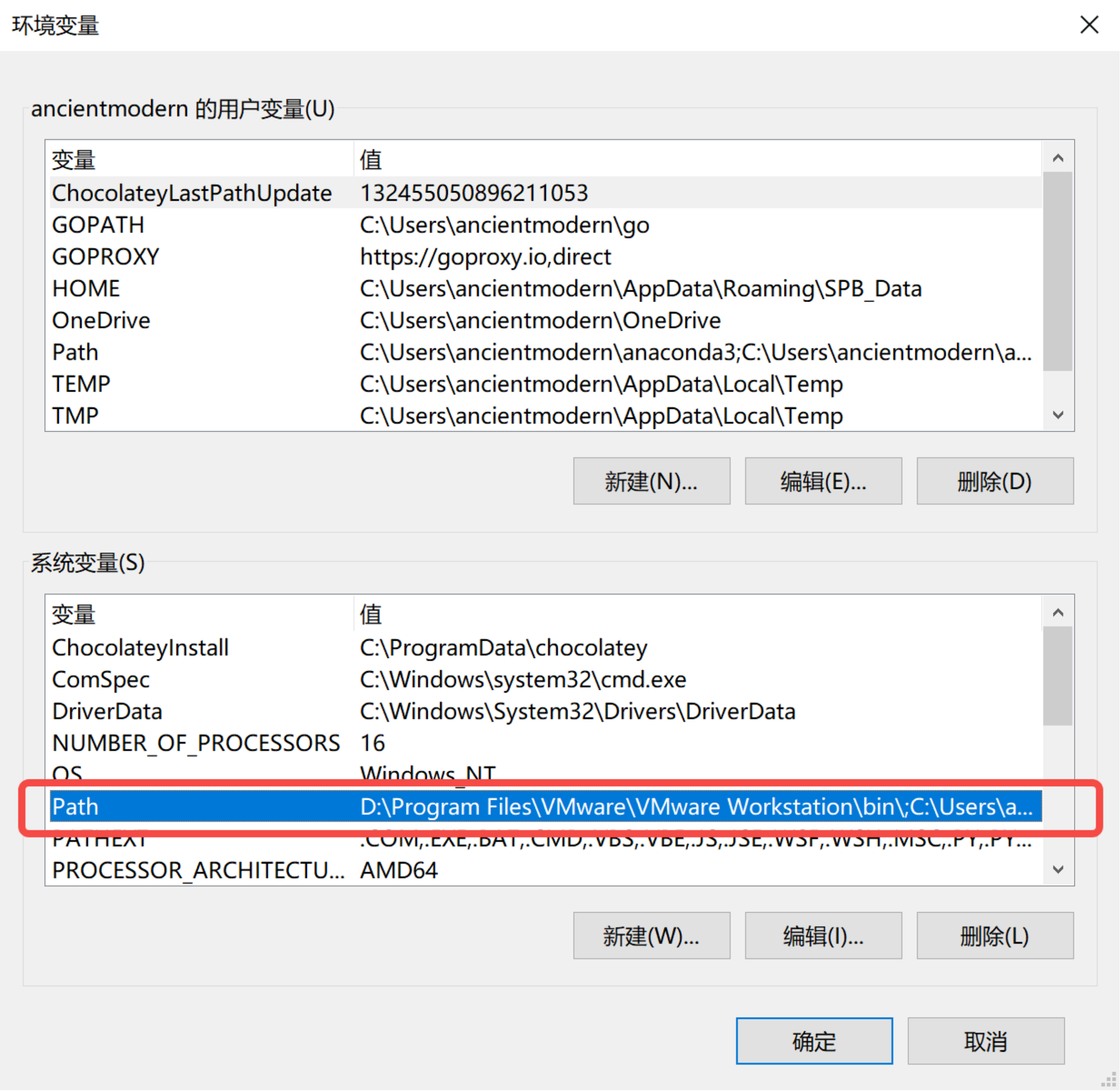
系统启动、系统故障和调试信息

设置(T)...

环境变量(N)...

确定 | 取消 | 应用(A)

4. Double click the variable "Path" in "系统变量"



环境变量

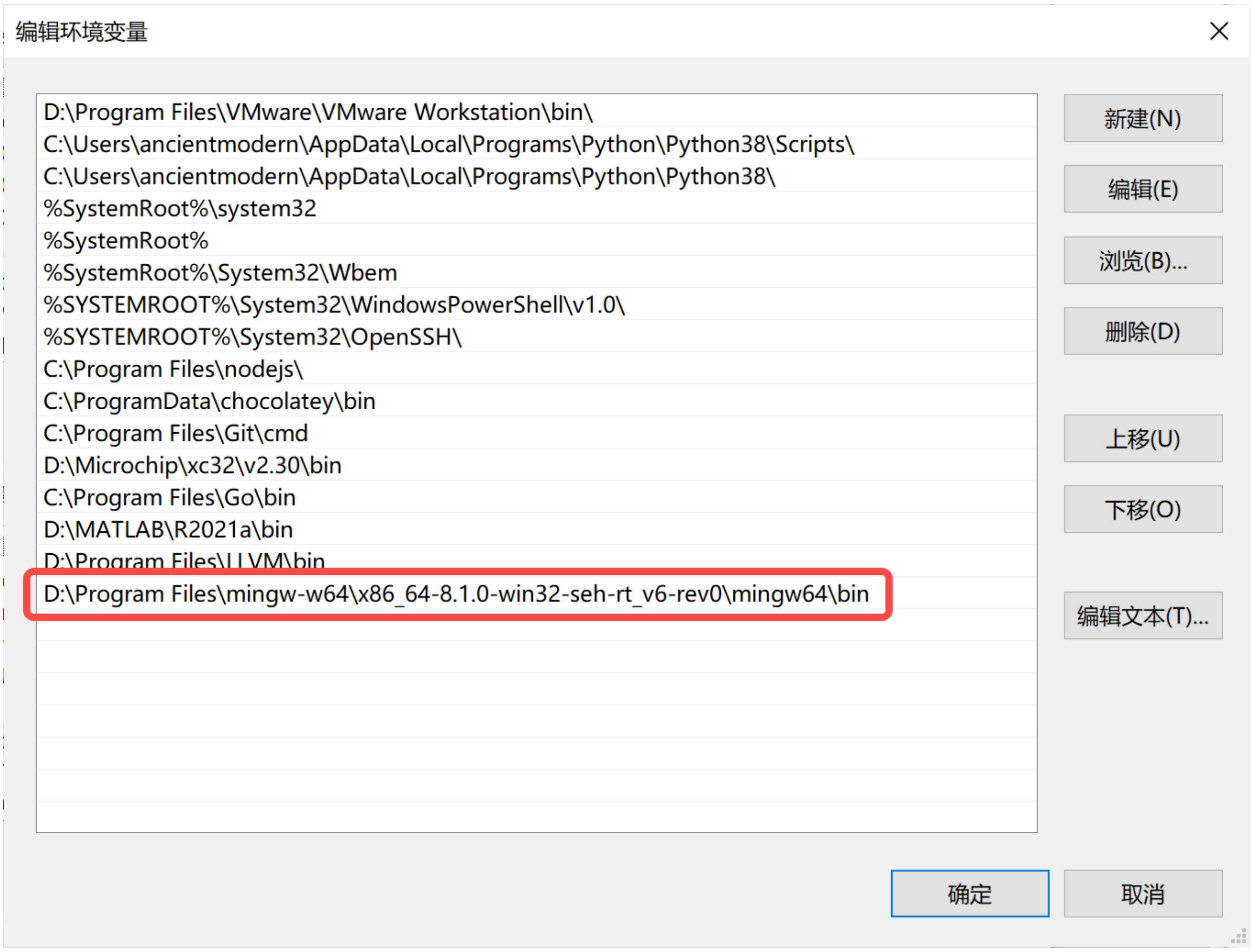
ancientmodern 的用户变量(U)

变量	值
ChocolateyLastPathUpdate	132455050896211053
GOPATH	C:\Users\ancientmodern\go
GOPROXY	https://goproxy.io,direct
HOME	C:\Users\ancientmodern\AppData\Roaming\SPB_Data
OneDrive	C:\Users\ancientmodern\OneDrive
Path	C:\Users\ancientmodern\anaconda3;C:\Users\ancientmodern\a...
TEMP	C:\Users\ancientmodern\AppData\Local\Temp
TMP	C:\Users\ancientmodern\AppData\Local\Temp

系统变量(S)

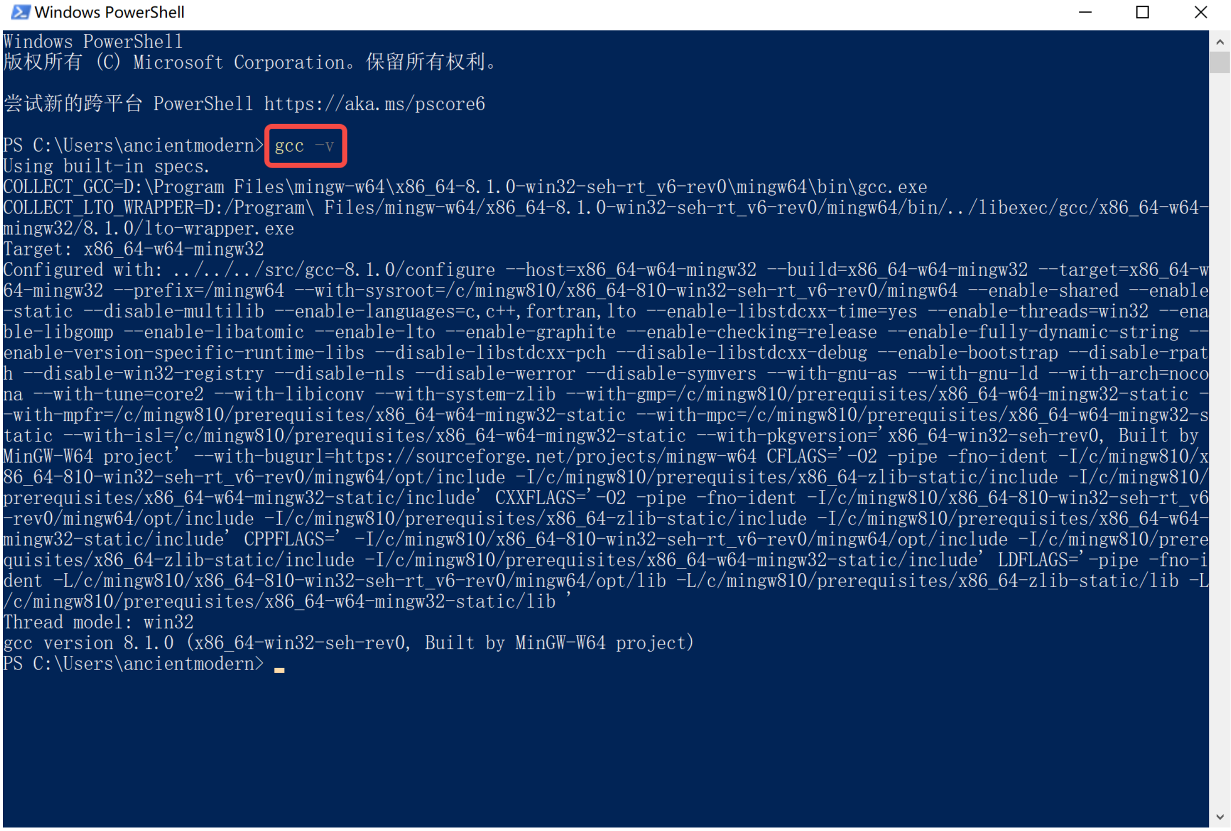
变量	值
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	D:\Program Files\VMware\VMware Workstation\bin\;C:\Users\a...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PY...
PROCESSOR_ARCHITECTU...	AMD64

5. Click "新建" and add the path of the "bin" directory (containing `gcc.exe`/`g++.exe`) of your MinGW



```
D:\Program Files\VMware\VMware Workstation\bin\
C:\Users\ancientmodern\AppData\Local\Programs\Python\Python38\Scripts\
C:\Users\ancientmodern\AppData\Local\Programs\Python\Python38\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\nodejs\
C:\ProgramData\chocolatey\bin
C:\Program Files\Git\cmd
D:\Microchip\xc32\v2.30\bin
C:\Program Files\Go\bin
D:\MATLAB\R2021a\bin
D:\Program Files\LLVM\bin
D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin
```

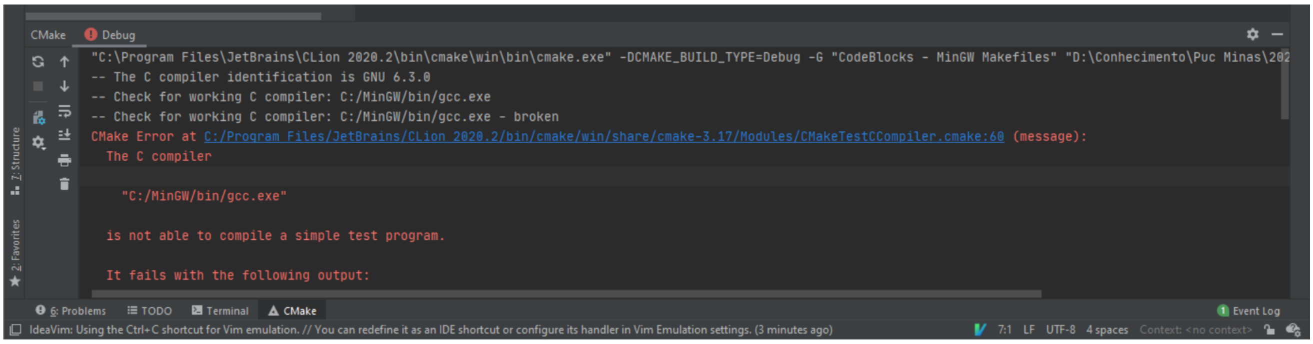
6. Press "Win + R" and enter "powershell", and enter the command `gcc -v` in the PowerShell, if you see message like this, it means your environment path is configured correctly. Note that you may restart or open a new PowerShell, since the PowerShell only loads the environment path when it launches.



```
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\ancientmodern> gcc -v
Using built-in specs.
COLLECT_GCC=D:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rev0\mingw64\bin\gcc.exe
COLLECT_LTO_WRAPPER=D:/Program\ Files/mingw-w64/x86_64-8.1.0-win32-seh-rt_v6-rev0/mingw64/bin/../libexec/gcc/x86_64-w64-
mingw32/8.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../../../src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w
64-mingw32 --prefix=/mingw64 --with-sysroot=/c/mingw810/x86_64-810-win32-seh-rt_v6-rev0/mingw64 --enable-shared --enable
-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-threads=win32 --ena
ble-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --
enable-version-specific-runtime-libs --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-rpat
h --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=noco
na --with-tune=core2 --with-libiconv --with-system-zlib --with-gmp=/c/mingw810/prerequisites/x86_64-w64-mingw32-static -
-with-mpfr=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpc=/c/mingw810/prerequisites/x86_64-w64-mingw32-s
tatic --with-isl=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-pkgversion='x86_64-win32-seh-rev0, Built by
MinGW-W64 project' --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -fno-ident -I/c/mingw810/x
86_64-810-win32-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/
prerequisites/x86_64-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -fno-ident -I/c/mingw810/x86_64-810-win32-seh-rt_v6
-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-
mingw32-static/include' CPPFLAGS=' -I/c/mingw810/x86_64-810-win32-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prere
quisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/include' LDFLAGS='-pipe -fno-i
dent -L/c/mingw810/x86_64-810-win32-seh-rt_v6-rev0/mingw64/opt/lib -L/c/mingw810/prerequisites/x86_64-zlib-static/lib -L
/c/mingw810/prerequisites/x86_64-w64-mingw32-static/lib '
Thread model: win32
gcc version 8.1.0 (x86_64-win32-seh-rev0, Built by MinGW-W64 project)
PS C:\Users\ancientmodern>
```

7. Enjoy using GCC/G++ via command line!

5. I have completed all the steps above, but in CLion the CMake shows "not able to compile a simple test program"

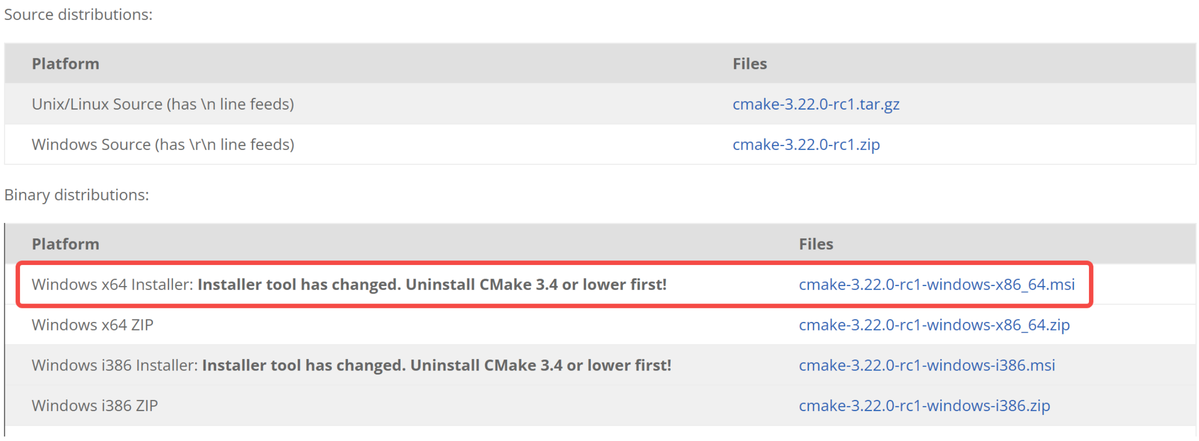


```
"C:\Program Files\JetBrains\CLion 2020.2\bin\cmake\win\bin\cmake.exe" -DCMAKE_BUILD_TYPE=Debug -G "CodeBlocks - MinGW Makefiles" "D:\Conhecimento\Puc Minas\202
-- The C compiler identification is GNU 6.3.0
-- Check for working C compiler: C:/MinGW/bin/gcc.exe
-- Check for working C compiler: C:/MinGW/bin/gcc.exe - broken
CMake Error at C:/Program Files/JetBrains/CLion 2020.2/bin/cmake/win/share/cmake-3.17/Modules/CMakeTestCCompiler.cmake:60 (message):
  The C compiler

    "C:/MinGW/bin/gcc.exe"

  is not able to compile a simple test program.

  It fails with the following output:
```

```
IdeaVim: Using the Ctrl+C shortcut for Vim emulation. // You can redefine it as an IDE shortcut or configure its handler in Vim Emulation settings. (3 minutes ago)    7:1  LF  UTF-8  4 spaces  Context: <no context>
```

This is usually caused by non-Latin symbols (for example, Chinese) in the installation path of CLion, as CMake and MinGW do not support non-Latin symbols. Hence, when you use the bundled CMake in CLion, there will be some problems.

Besides reinstall CLion in another path without Chinese character, a possible solution is to use a custom CMake. You can download CMake from https://cmake.org/download/,



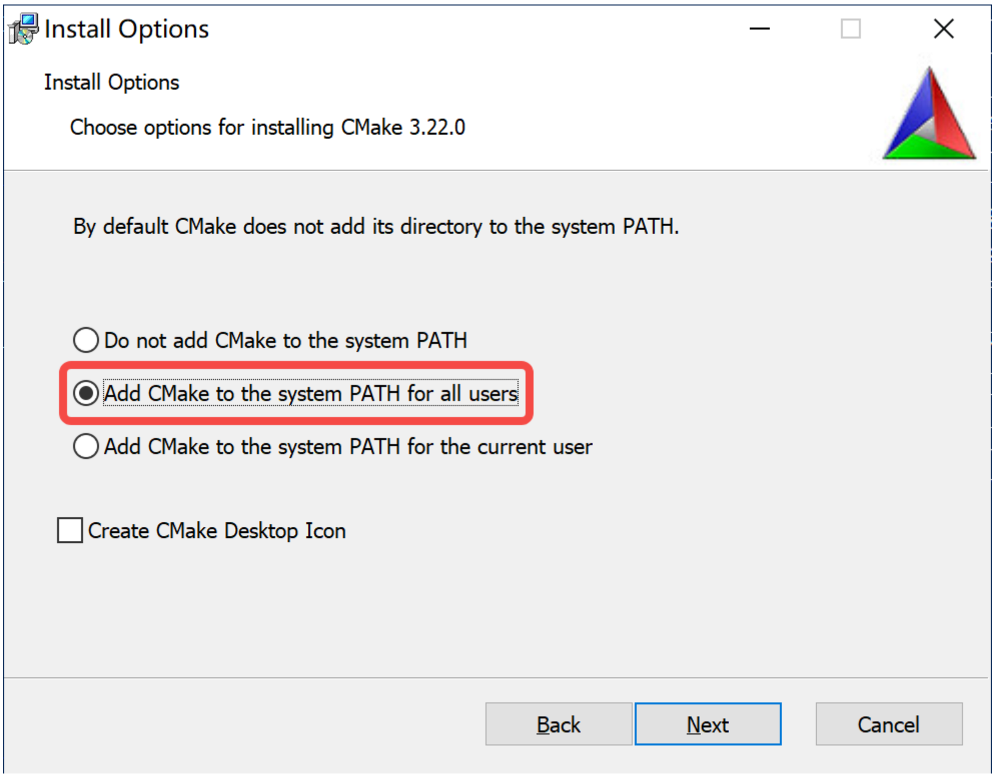
Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.22.0-rc1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.22.0-rc1.zip

Binary distributions:

Platform	Files
Windows x64 Installer: **Installer tool has changed. Uninstall CMake 3.4 or lower first!**	cmake-3.22.0-rc1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.22.0-rc1-windows-x86_64.zip
Windows i386 Installer: **Installer tool has changed. Uninstall CMake 3.4 or lower first!**	cmake-3.22.0-rc1-windows-i386.msi
Windows i386 ZIP	cmake-3.22.0-rc1-windows-i386.zip

and install it in the path without Chinese character. Then you can **follow similar steps in Question 4 to add the custom CMake into CLion's toolchain**. When installing CMake, you can select this option, and it will automatically add CMake into your environment path.



Install Options

Install Options

Choose options for installing CMake 3.22.0

By default CMake does not add its directory to the system PATH.

○ Do not add CMake to the system PATH

● Add CMake to the system PATH for all users

○ Add CMake to the system PATH for the current user

□ Create CMake Desktop Icon

6. My problem is neither listed above nor in the previous guidelines, what should I do?

First you may try to search your problems (for example, the error messages) through the search engines. Actually I got all of above solutions from Google.

If you do not find the solution, feel free to ask TAs for help. Don't worry that your question may be a little "stupid", in fact I think setting up C/C++ environment is really a difficult task on Windows, and it requires a certain amount of programming knowledge. Our three TAs are always willing to help you solve your problems, and assist you to build up the foundation for further C/C++ journey. Good luck!