

# 计算机程序设计基础开发环境配置说明

本教程主要面向于没有编程经验的同学，帮助大家快速上手使用 Visual Studio Code（简称 VS Code）编辑器和 g++ 命令行编译器来编写 C++ 程序。

## 编译器推荐

编译器是将代码转化为计算机可以执行的机器码的工具，是程序设计中最基础的工具。本教程将推荐使用 g++（GNU 的 C++ 编译器），请根据你的操作系统来选择合适的安装教程。

- 由于 Linux 系统对于编写程序的原生支持较为完善，对于 Windows 系统，推荐使用系统自带的 WSL 2 来模拟 Linux 环境（后面的课程项目会以 Linux 环境为准，注意一定要用 WSL 2，而不是 WSL 1，后者对 Linux 的支持不够完善）；如果编写自己的程序，尤其是希望得到 Windows 原生的可执行文件，可以考虑使用 MinGW 或者 Visual Studio（不是 Visual Studio Code，后者只是一个编辑器）。
- 在苹果的 macOS 系统上，系统自带 Xcode 以及相关编译器等工具，使用体验与 Linux 几乎相同。

## Windows 系统 + WSL 2

WSL (Windows Subsystem for Linux) 是适用于 Linux 的 Windows 子系统，可以让开发人员按原样运行 GNU/Linux 环境——包括大多数命令行工具、实用工具和应用程序，且不会产生传统虚拟机或双启动的设置开销。对 WSL 感兴趣的同学可以查阅【[官网](#)】。

推荐环境为 Windows 11 的同学使用 WSL 2，以享受 Linux 系统功能强大的命令行带来的便利。Windows 10 的最新版本下也可以使用。

## WSL 2 的安装与用户配置

- 打开 PowerShell，输入命令 `wsl --list --online` 可以看到所有可下载的 Linux 发行版本。
- 选择 Ubuntu 20.04 版本，输入命令 `wsl --install -d Ubuntu-20.04` 进行安装。
- 等待即可，如下图：

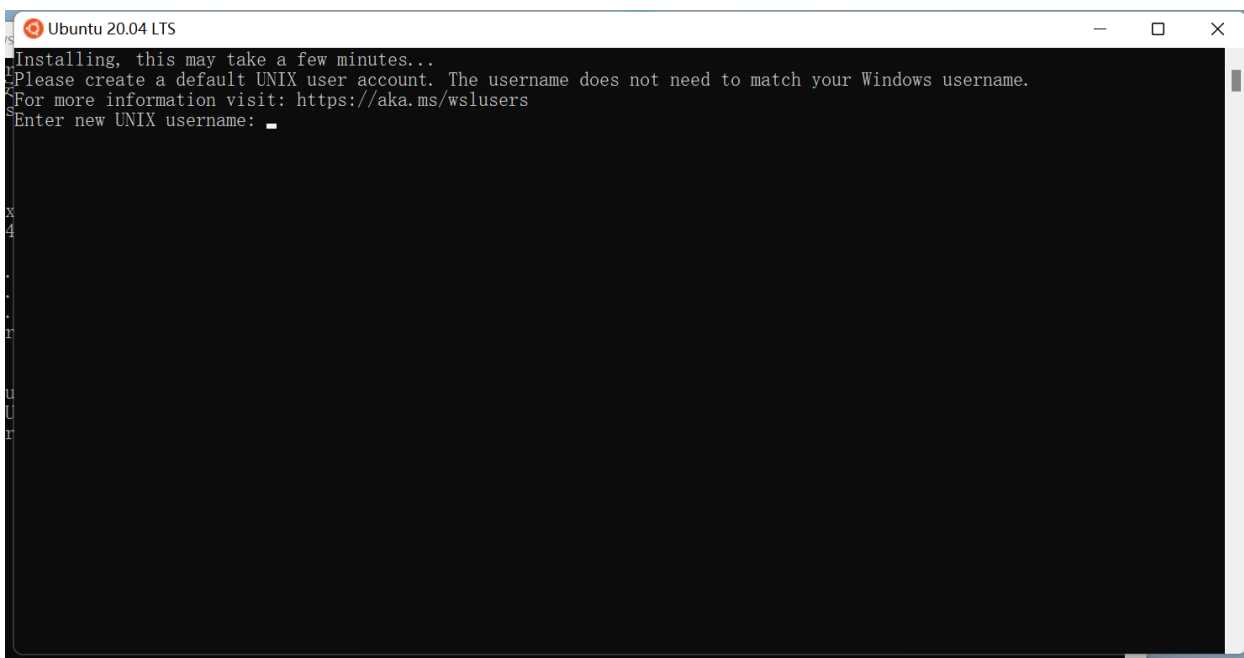


```

PS C:\Users\Yanzheng Cai> wsl --list --online
以下是可安装的有效分发的列表。
请使用“wsl --install -d <分发>”安装。

NAME                FRIENDLY NAME
Ubuntu              Ubuntu
Debian              Debian GNU/Linux
kali-linux           Kali Linux Rolling
openSUSE-42          openSUSE Leap 42
SLES-12              SUSE Linux Enterprise Server v12
Ubuntu-16.04         Ubuntu 16.04 LTS
Ubuntu-18.04         Ubuntu 18.04 LTS
Ubuntu-20.04         Ubuntu 20.04 LTS
PS C:\Users\Yanzheng Cai> wsl --install -d Ubuntu-20.04
正在下载: Ubuntu 20.04 LTS
===== 43.5%
    
```

- 下载完成后会弹窗出现 WSL 2 系统的用户名与密码设定的界面，如下图。用户名不需要与 Windows 保持一致，自由设置即可。



```

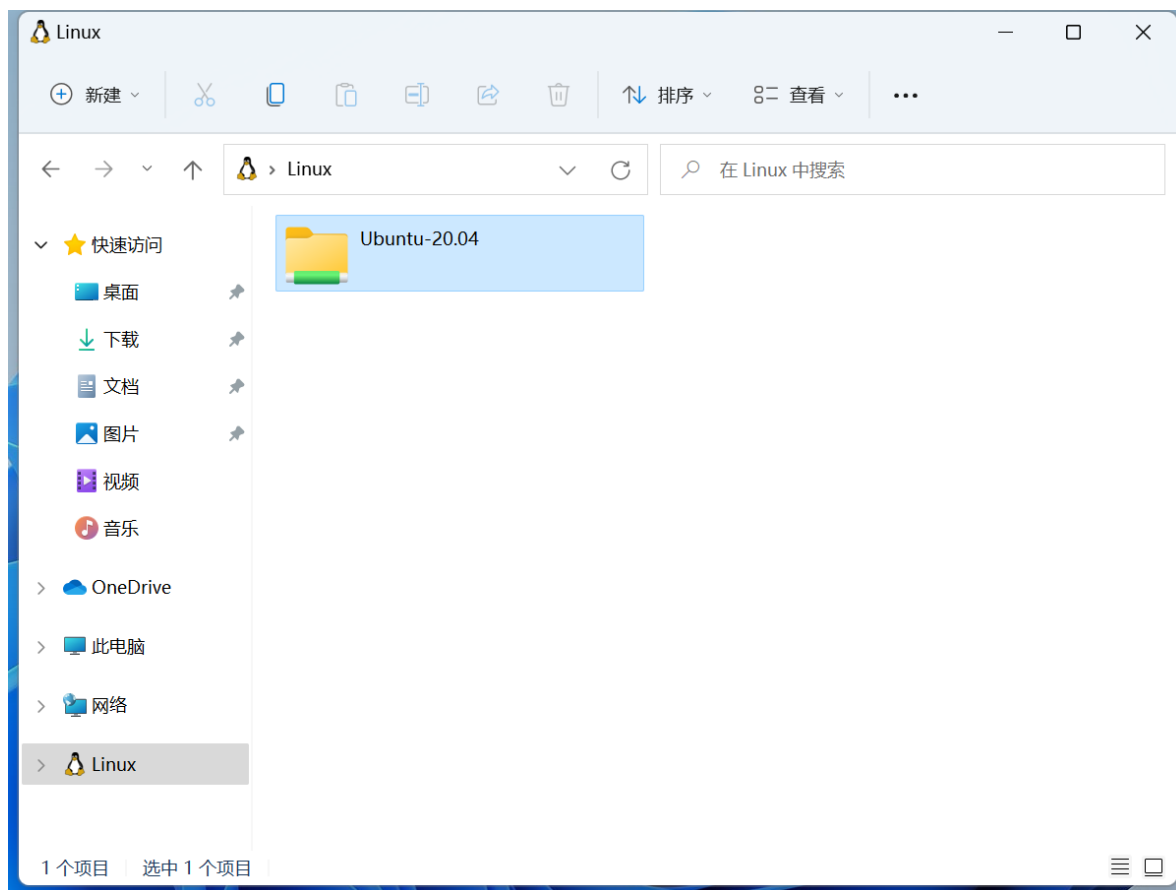
Ubuntu 20.04 LTS
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: 
    
```

**文件访问问题** 安装完成后，电脑上存在两个终端：Windows 系统的 PowerShell 与 Linux 发行版的 Ubuntu 20.04 LTS（都可以在 Windows 的开始中搜索到）。

在终端中输入命令 `cd ~` 可以返回用户的家目录。在 Windows 中，这个目录形为 `C:\Users\user`。在 Linux 中，这个目录形为 `/home/user`。需要注意 Windows 和 Linux 下路径中目录之间的分隔符是不同的，Windows 的目录以反斜杠 `\\` 相连，Linux 的目录以斜杠 `/` 相连。接下来以家目

录为案例，介绍 Windows 系统与 WSL 2 子系统的文件与目录访问。

- 从 Windows 访问 Windows 项目
  - 略。
- 从 WSL 2 访问 WSL 2 项目
  - 【推荐】打开 Ubuntu 20.04 LTS 终端。
  - 【不推荐】打开 PowerShell，输入命令 `wsl` 后输入命令 `cd ~`。
  - 上面两种操作都会停留在 Linux 系统的家目录。
- 从 Windows 访问 WSL 2 项目
  1. 打开 PowerShell。
  2. 输入命令 `explorer.exe .`，在左侧栏中点击 Linux，如下图：



3. 依次点击 Ubuntu-20.04（你下载的 linux 发行版），home，zhangsan（你设置的用户名），即可从 Windows 进入 Linux 系统的家目录。
- 从 WSL 2 访问 Windows 项目

1. 打开 Ubuntu 20.04 LTS 终端。
2. 假设你 Windows 系统的用户名为 user，输入命令 `cd '/mnt/c/Users/user'`，即可从 WSL 2 进入 Windows 系统的家目录。

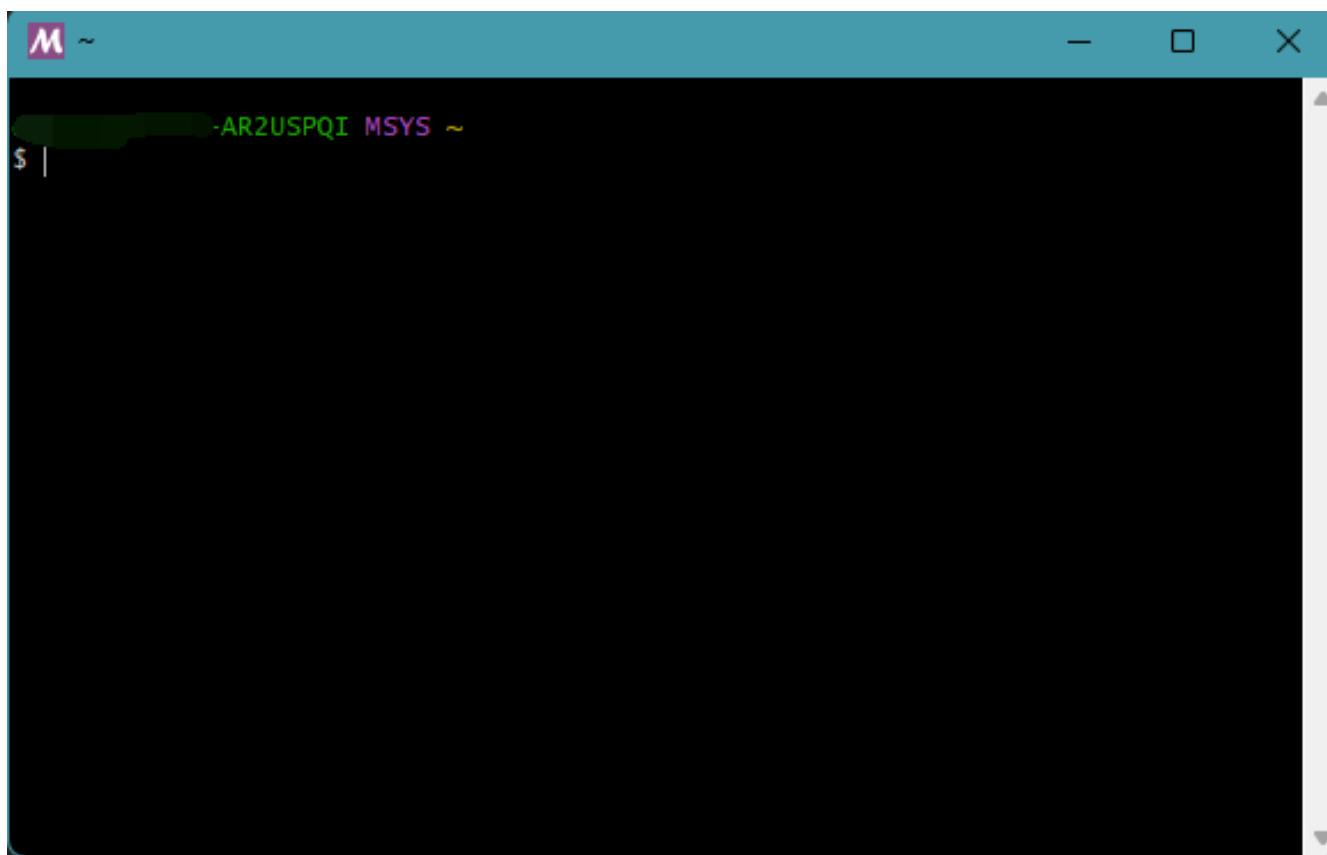
上文介绍了在 Windows 与 WSL 2 两个系统中互相访问对方文件的方法，但是跨操作系统访问文件可能显著降低性能，所以不推荐这么做。如果需要使用 Windows 软件（如 Word、PPT）或者 PowerShell 进行处理，请把文件存储在 Windows 系统中；如果需要使用 Linux 中的工具在 Ubuntu 命令行进行处理，请把文件存储在 WSL 2 文件系统中。

**WSL 2 环境配置** 打开 Ubuntu 20.04 LTS 终端，按照下文 Ubuntu 系统的配置方法进行操作

## Windows 系统 + MinGW

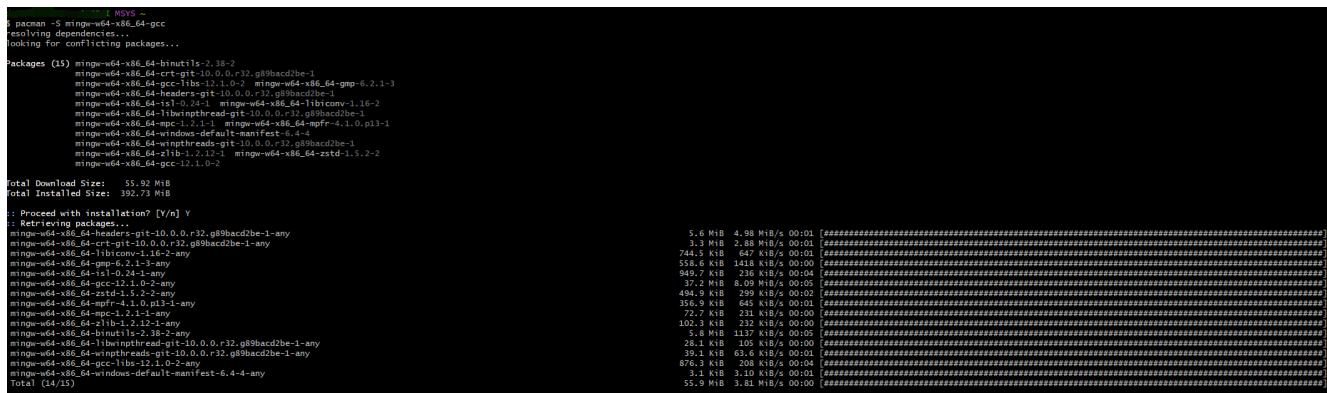
**MSYS2 安装** 我们推荐采用 MSYS2 包管理软件来进行 MinGW 工具的安装。对 MSYS2 感兴趣的同学可以查阅其【官网】。

- 首先点击【MSYS2 发布链接】并选择下载最新版的安装程序。
- 本教程写成的时候最新版为【一键下载】。
- 下载之后采用一键安装的方式完成安装。
- 完成安装后点击“立即运行 MSYS2”会出现一个黑色的命令行窗口。



## MinGW 安装

- 在其中输入 `pacman -S mingw-w64-x86_64-gcc` 按回车进行 `gcc` 的安装。然后对于提示按 `y` 和回车。



## 环境变量配置

- 如果你使用的默认安装路径，则将 `C:\msys64\mingw64\bin` 目录添加到系统 PATH 环境变量上；如果你在安装 MSYS2 时选择了安装路径，则将 `< 安装路径 >\mingw64\bin` 目录添加到系统 PATH 环境变量上。

- 具体操作可以在搜索栏搜索“环境变量” → 点击“编辑环境变量” → 找到 PATH 变量点击“新建/添加”。如果你觉得环境变量配置有苦难，可以参考教程末尾的附录，不同的 Windows 系统版本在环境变量配置上有若干不同，如果您的系统配置和附录中内容有出入，请自行搜索 Windows 版本 + 环境变量配置。
- 按 Win + R 键，输入 cmd 并回车，在打开的命令行窗口中输入 g++ -v。
- 输出下面的内容表示安装成功。

```
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=C:/msys64/mingw64/bin/./lib/gcc/x86_64-w64-mingw32/12.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../gcc-12.1.0/configure --prefix=/mingw64 --with-local-prefix=/mingw64/local --build=x86_64-w64-mingw32
--host=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --with-native-system-header-dir=/mingw64/include --libexecdir=/mi
ngw64/lib --enable-bootstrap --enable-checking=release --with-arch=x86-64 --with-tune=generic --enable-languages=c,lto,c
++,fortran,ada,objc,obj-c++,jit --enable-shared --enable-static --enable-libatomic --enable-threads=posix --enable-graph
ite --enable-fully-dynamic-string --enable-libstdcxx-filesystem-ts --enable-libstdcxx-time --disable-libstdcxx-pch --ena
ble-lto --enable-libgomp --disable-multilib --disable-rpath --disable-win32-registry --disable-nls --disable-werror --di
sable-symvers --with-libiconv --with-system-zlib --with-gmp=/mingw64 --with-mpfr=/mingw64 --with-mpc=/mingw64 --with-isl
=/mingw64 --with-pkgversion='Rev2, Built by MSYS2 project' --with-bugurl=https://github.com/msys2/MINGW-packages/issues
--with-gnu-as --with-gnu-ld --disable-libstdcxx-debug --with-boot-ldflags=-static-libstdc++ --with-stage1-ldflags=-stati
c-libstdc++
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 12.1.0 (Rev2, Built by MSYS2 project)
```

## macOS 系统

- 安装命令行工具：

```
xcode-select --install
```

- 也可以安装 Homebrew 软件包管理器之后安装 gcc 软件包。

## Linux 系统（Debian、Ubuntu 等）

- 打开终端输入：

```
sudo apt-get install build-essential
```

## Linux 系统（CentOS、Fedora 等）

- 打开终端输入：

```
sudo yum install g++
```

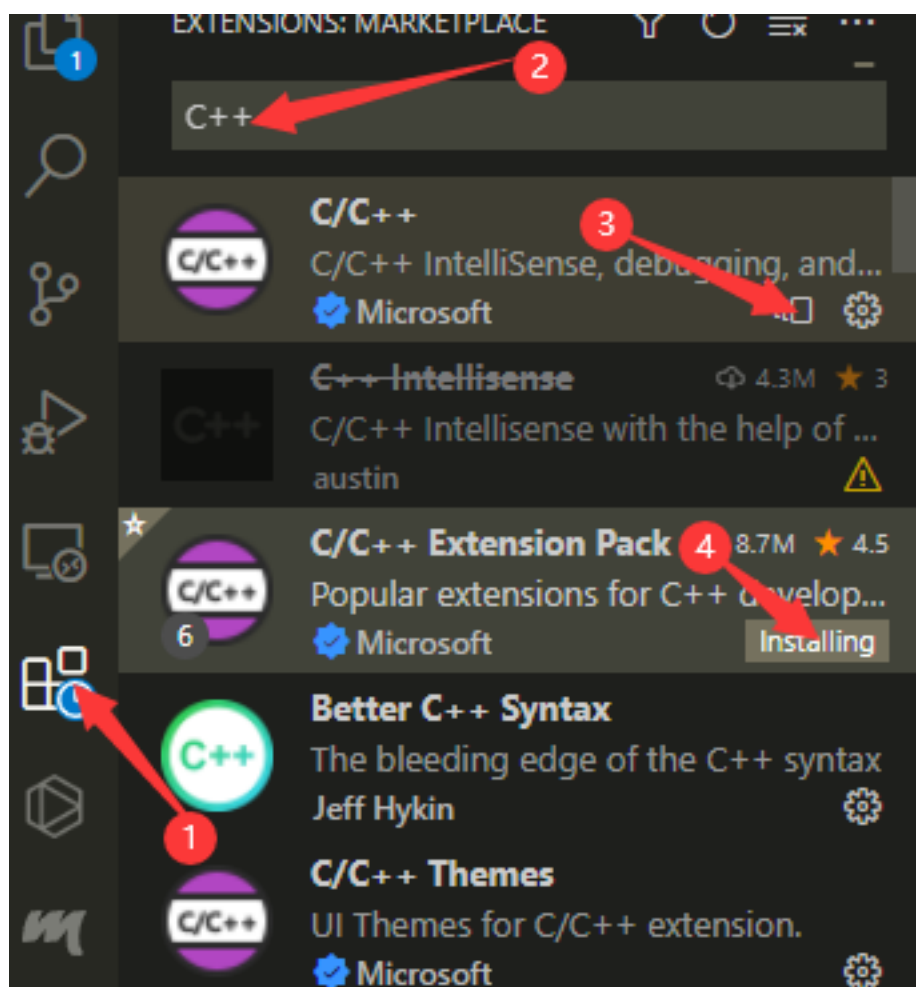
## 编辑器推荐

编辑器是程序员用来写代码的工具，虽然真正的高手可以使用记事本写代码，但是一个好的编辑器可以大大提升程序开发的效率，帮助程序员减轻很多负担。本教程推荐使用免费、开源、轻量级但是扩展性优秀的 Visual Studio Code 作为编辑器，此编辑器在跨平台方面也做得非常优秀。

### Visual Studio Code 下载及安装

- 打开【VS Code 官网】下载操作系统对应版本进行安装。

### VS Code 环境配置



如图所示，

- 首先打开侧边栏插件栏目
- 搜索 “C++”。

- 安装 C/C++ 插件和 C/C++ Extension Pack 插件。

## (WSL 2 专用) 通过 VS Code 访问 WSL 2 项目

- 在 VS Code 插件项目中搜索 “Remote Development” 并进行安装。
- 打开 Ubuntu 20.04 LTS 终端，输入命令 `code .` ( `.` 表示当前目录 )。
- 第一次输入可能会进行一些下载，耐心等待即可。
- 接下来会弹出 VS Code 窗口，现在就可以通过 VS Code 访问 Linux 系统的家目录了 ( 此时 VS Code 顶栏的 “终端” 打开的也是 Linux 命令行， “终端” 的作用下一节介绍 )。
- 在弹出的该 VS Code 窗口下，也可以按照上一小节的操作安装 C/C++ 插件。注意，在该窗口下安装的插件，与正常状态进入 VS Code 安装的插件并不互通。

## 编写程序与运行

- 在当前目录新建一个文件命名为 `hello.cpp`。
- 在 `hello.cpp` 中输入如下代码：

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, world!" << endl;
}
```

- 打开 VS Code 顶栏的 “Terminal/终端”。
- 在终端中输入：

```
g++ hello.cpp -o hello
```

- 如果没有报错你可看到你的文件夹中出现了一个新的可执行文件。在 Windows 系统下为 `hello.exe`，在其他系统下为 `hello`。
- 在终端输入：

```
./hello
```

- 即可看到成功输出 “Hello, world!” 的字样。

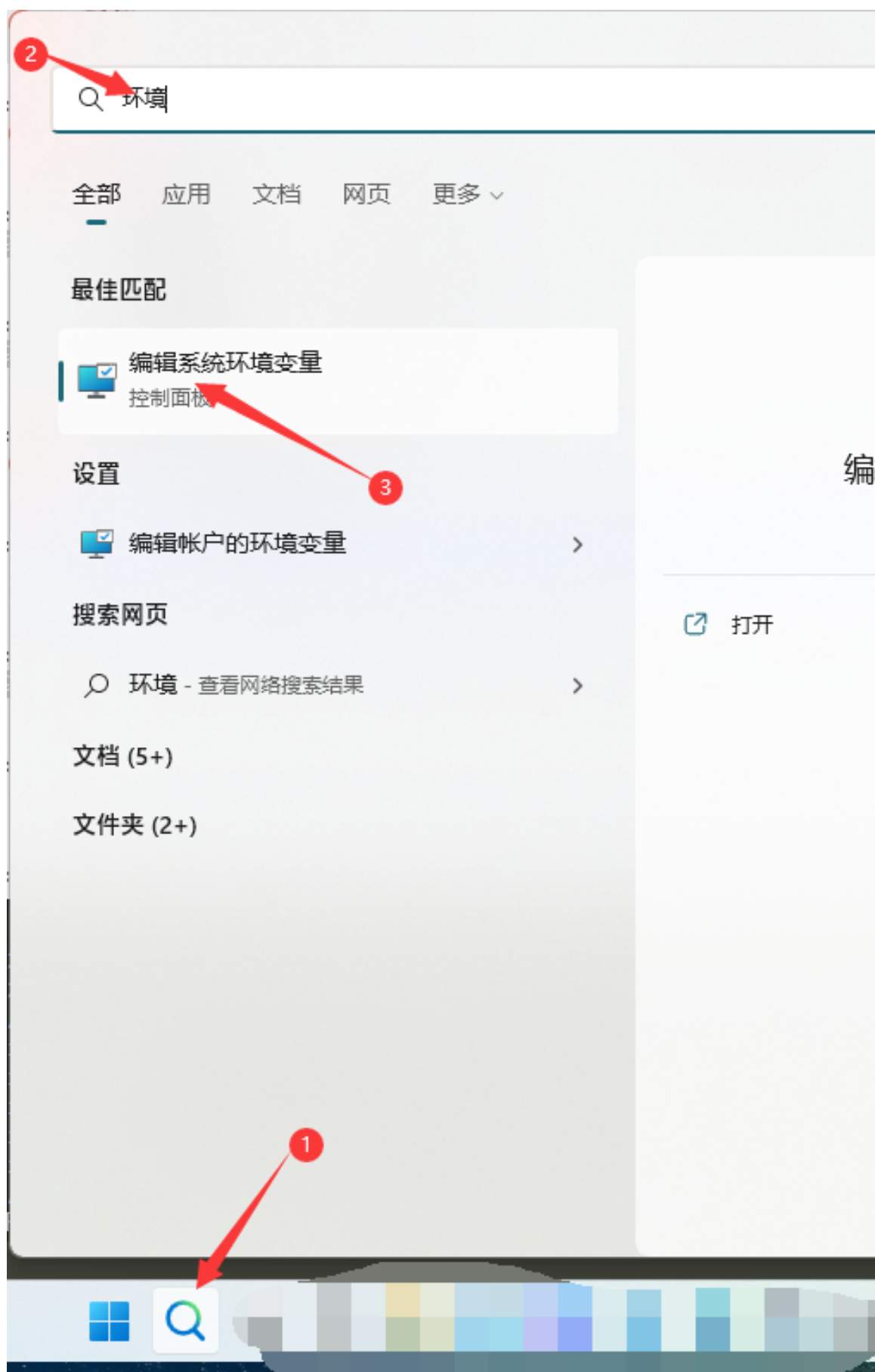
**祝贺你！你已经成功配置了 C++ 的开发环境！**



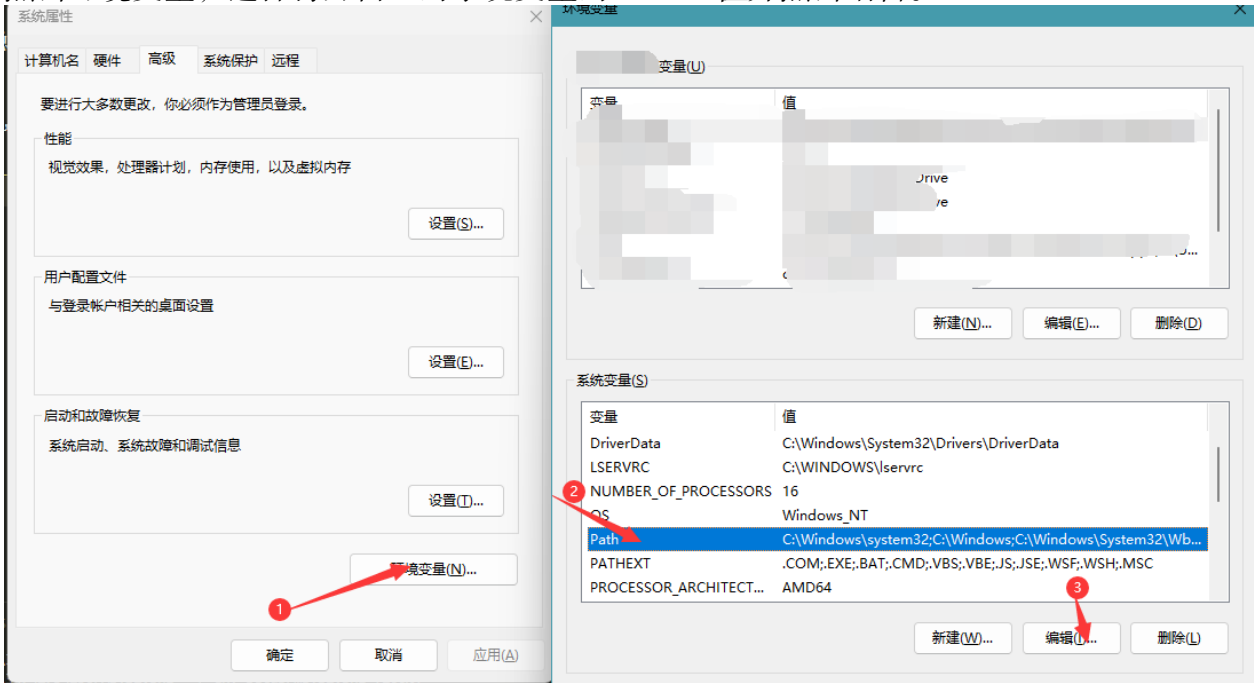
## 附录：环境变量配置说明

由于教程编写者只有 Windows 11 的环境，故而以 Windows 11 为演示，其他环境大同小异。略有不同但总体操作相同，如果有问题请自行使用搜索引擎搜索对应系统版本的环境变量编辑方式。

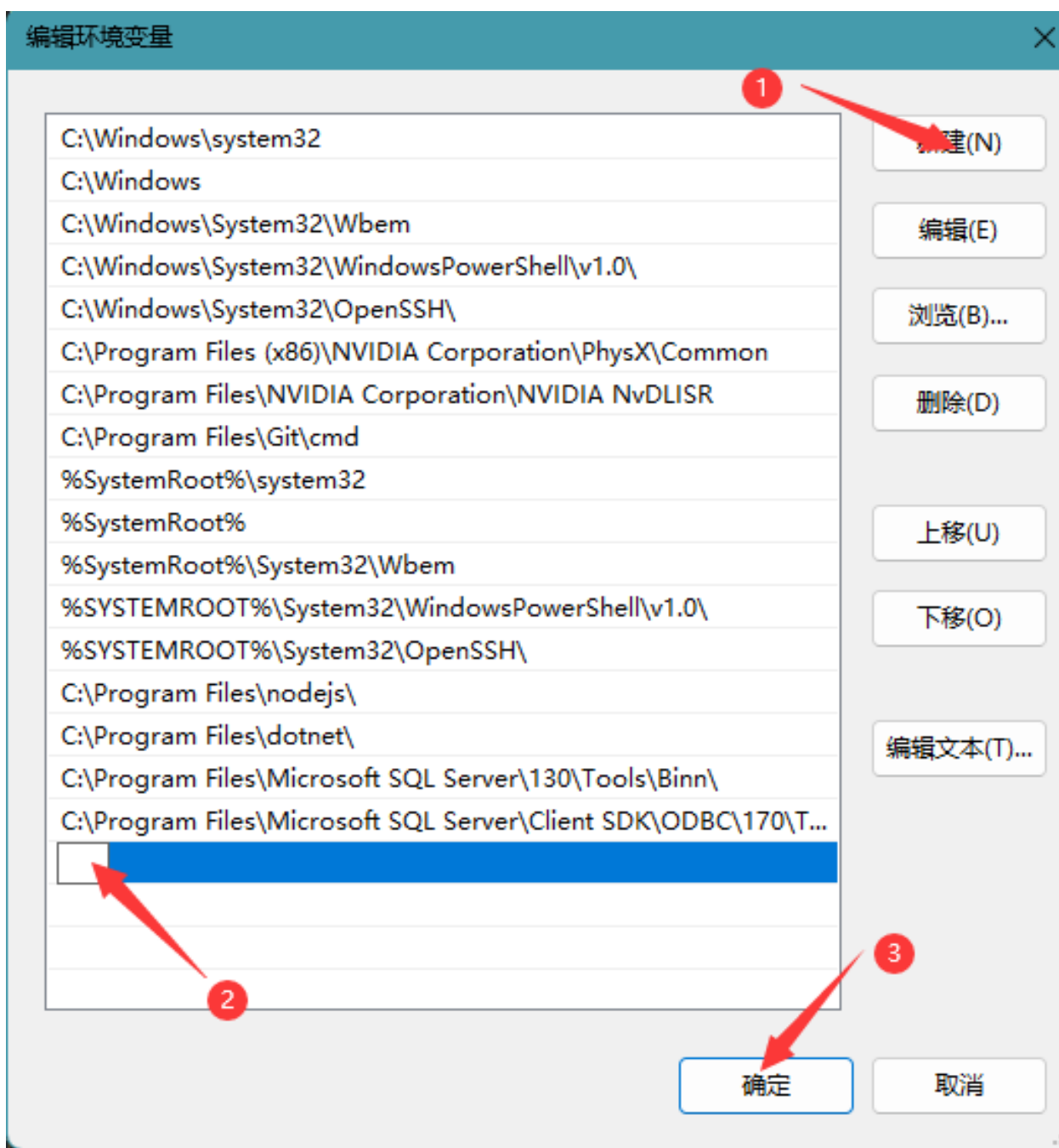
- 打开系统搜索栏搜索“环境变量”并选择编辑系统环境变量。



- 点击环境变量，选择打开窗口的系统变量的 PATH 值并点击编辑。



- 点击新建，然后输入要插入的环境变量值。点击确定



- 依次将打开的窗口点击确定关闭。

