

算法竞赛的编程环境配置

同济大学CPCLab 叶茂尧

工欲善其事，必先利其器。——《论语·卫灵公》

由于大部分算法竞赛的比赛及评测环境均为 Linux gcc，为了避免由于编程环境差异导致的问题，也为了自己训练的方便，建议大家参考本文档配置一个自己觉得比较舒服的开发环境。

基本原则

- 不要用 Visual Studio

VS 和 gcc 的差异还是较为明显的，这一点上过高程的同学应该深有体会。使用 VS，可能会遇到包括但不限于网上的题解复制到本地编译不过、本地程序提交评测发现编译错误、同样程序表现不同等问题。

- 一定要掌握基本的命令行操作

ICPC 现场比赛环境通常是 Linux，可以使用的开发工具可能只有 CodeBlocks 或者不带插件的 Visual Studio Code 等（而且我在比赛中遇到过 CodeBlocks 崩溃的事情），不会命令行的话可能会懵；另外，熟练使用命令行也算程序员最最基本的素养了。

补充阅读：编辑器、编译器与IDE

为了省事，通常不要求初学者区分这三个概念，因此你可能听到把 Visual Studio 或者 Dev C++ 称为“编译器”的叫法。但严谨地说，这是三个不同的东西。

- 编辑器

编辑器 (editor) 是用于编辑文本文件的工具。大家熟悉的记事本、Microsoft Word 等等都属于编辑器。由于编程这件事的专业性，也有很多专用于编程的编辑器，例如 Vim、Emacs、Visual Studio Code、Sublime Text、~~Notepad++~~ 等等。编辑器可能提供语法高亮、自动补全、代码格式化、语法检查等功能。

按：Notepad++ 作者在程序官网和更新日志中公然发表分裂中国的言论，强烈不推荐使用。

- 编译器

编译器 (compiler) 是负责将源代码翻译成可执行文件的程序（编译器实际的定义比这宽泛，但是作为算法竞赛选手暂时可以不必关心）。

常见的 C/C++ 编译器有 MSVC（也称 cl，即 VS 所使用的编译器）、gcc/g++（Dev C++ 内置的编译器就是它）、clang、icc 等。

按：严格意义上，这些都是**编译器驱动程序**，包含了预处理器、编译器、链接器等多个部分。有兴趣的同学可以自行阅读[《深入理解计算机系统》第七章](#)以了解相关知识，这对学习大型项目管理以及系统编程非常有帮助。

- IDE

IDE 是 Integrated Development Environment（集成开发环境）的缩写。使用者可以在 IDE 中管理项目、编写代码、编译运行并调试。Visual Studio、Dev C++ 等软件都属于 IDE。

最省事的轻量IDE：Dev C++

一个非常非常有年头的 IDE 了，由于作为中学信息竞赛的半官方指定工具（大部分省提供的环境里都会预置），被竞赛选手广泛使用。本已被开发者放弃，但 Embarcadero 公司于 2020 年接手并提供了较新的版本。

无论是官方旧版（5.11）还是我为高程做的定制版（5.11 + gcc 9.2.0），都不再推荐使用。高程不使用官方最新版的原因是其对 GB 系列中文编码的支持问题，而算法竞赛选手没有使用这种编码的需求（要选编码也一定是 UTF-8）。

优点：

- 安装方便，在 Windows 中下载安装包再双击即可，不必额外下载编译器
- 支持单文件编译运行，创建一个新文件就能开始写代码，不用管理项目，非常方便
- 心智负担小，不需要掌握其他概念和知识就能使用

缺点：

- 类型提示、代码补全等功能比现代编辑器差很多，而且经常不可用
- 写代码的时候没有实时检查
- 对单步调试功能的支持非常糟糕
- 需要另外学习命令行操作

推荐对象

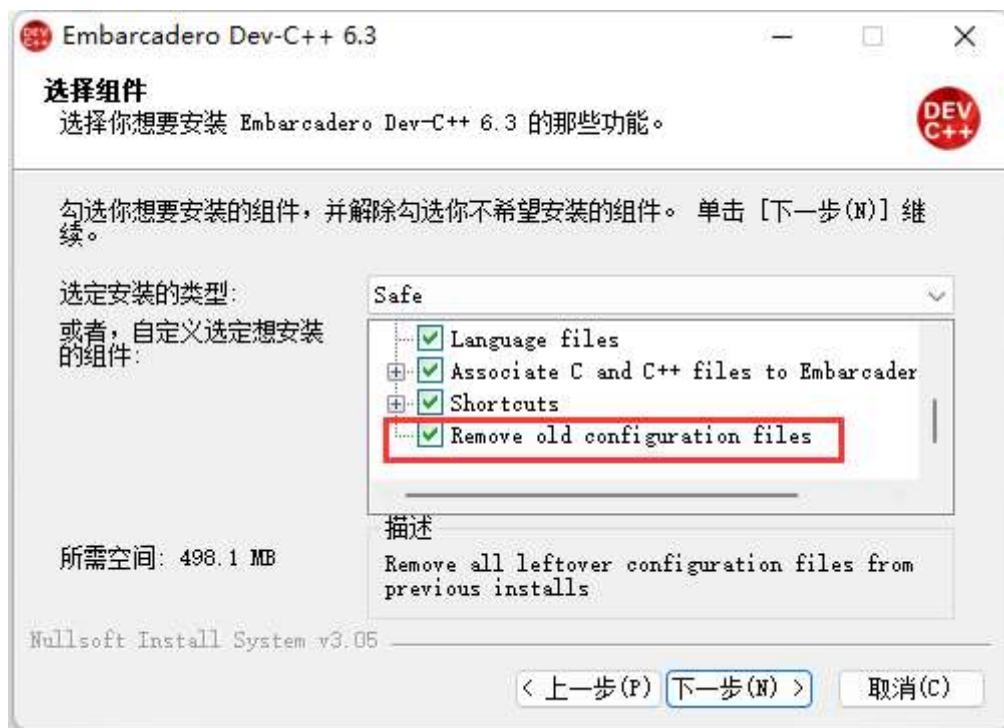
- 不想折腾的同学
- 计算机用得还不太熟，没能力折腾的同学（但不能总是这样，得学会折腾的）

简要安装指南

于[github](#)下载安装文件：（也可在群文件中下载，或自行于网络搜索）

▼ Assets 8		
 Embarcadero_Dev-Cpp_6.3_No_Compiler_Portable.7z	6.16 MB	31 Jan 2021
 Embarcadero_Dev-Cpp_6.3_No_Compiler_Setup.exe	8.09 MB	31 Jan 2021
 Embarcadero_Dev-Cpp_6.3_No_Compiler_Setup.zip	7.85 MB	31 Jan 2021
 Embarcadero_Dev-Cpp_6.3_TDM-GCC_9.2_Portable.7z	72.1 MB	31 Jan 2021
 Embarcadero_Dev-Cpp_6.3_TDM-GCC_9.2_Setup.exe	70.7 MB	31 Jan 2021
 Embarcadero_Dev-Cpp_6.3_TDM-GCC_9.2_Setup.zip	70.4 MB	31 Jan 2021
 Source code (zip)		31 Jan 2021
 Source code (tar.gz)		31 Jan 2021

按默认配置安装即可。若以前安装过其他版本的 Dev C++，安装时建议勾选“Remove old configuration file”以规避可能的问题。



建议配置

打开工具-编译选项，对于 TDM-GCC 9.2.0 64-bit Release 与 TDM-GCC 9.2.0 64-bit Debug 均进行如下设定：

在代码生成/优化-代码警告中，勾选“显示最多警告信息(-Wall)”和“显示较多警告信息（-Wextra）”两项。

■ 吐槽一下这个翻译，-Wextra 明明是“额外警告信息”。。。



这两项会使编译器进行严格的检查，对可能不规范或产生错误的地方进行警告，有助于查错。

另外，对于 TDM-GCC 9.2.0 64-bit Release 配置，建议额外在代码生成/优化页面中，将“优化级别”设置为 High。



这会让编译器进行很多优化工作，相当程度地提升代码运行速度，尤其在较多使用 STL 的情况下有巨大的作用。

注意事项与使用技巧

- 想要使用单步调试，将编译器切换至 Debug，否则会出错
- 可以使用快捷键 Ctrl+Shift+A 快速格式化代码。在菜单栏的"Astyle"中可以设置格式化选项。
 - 括号风格推荐 Stroustrup 或者 Allman，缩进风格强烈建议 Spaces，Tab 宽度 2 或 4 均可，由使用习惯决定（不推荐 8）。“最大行数”是翻译错误，实际指“每行最多字符数”，建议勾选该选项，保持 80 的默认配置。
- 如果代码有改动，使用 F11 编译运行；如果代码没有改动，只需要重新运行一遍，F10 会更快。

使用编辑器与命令行

可能有同学会有疑问：IDE 可以一个软件解决开发中的几乎所有问题，集成度更高，难道不应该比先拿编辑器写好代码、再去命令行里编译运行方便得多吗？为什么那么多程序员要选择使用编辑器呢？

不使用 IDE 的原因有很多。软件开发是一项复杂的工作，这就意味着合格的 IDE 需要集成非常多功能与组件，从而变得臃肿而庞大。Visual Studio 动辄几十 GB 的体积足以使人望而却步。那么，对于希望寻找轻量级解决方案的人来说，编辑器无疑是更好的。

更重要的是，编辑器有更好的可定制性。我们可以很轻松地更换使用的编译器、构建系统，可以通过插件来定制各种想要的功能——代码的检查、格式化、版本管理，清理空格、彩虹括号、快捷注释，甚至是取名助手、划词翻译、刷知乎.....通过恰当的配置，可以在 Visual Studio Code 这样的编辑器中完成几乎所有工作。

如果这还不足以说服你，还有一个无法反驳的理由：Linux 下是没有 VS 的。当然，Clion 也是出色的 IDE，但它并不免费。而出色的编辑器与插件完全可以提供相当的体验。

另外，命令行对于开发者的重要性无需赘言。对于有志于计算机相关方向的同学，不熟练掌握命令行恐怕是说不过去的。

安装 GCC

无论你想要使用什么样的编辑器，首先都要安装编译器。对于使用 Linux 或者 MacOS 的同学，这是一件简单的事情：直接通过包管理器安装即可。例如，Ubuntu 系统可以使用：

```
sudo apt install g++
```

但在 Windows 系统下，你可能会遇到一些困扰。诚然，Windows 也有类似 Scoop 和 Chocolatey 一类优秀的第三方包管理器，但由于众所周知的、不可言说的原因，国内的网络环境不支持我们流畅地使用它们。

如果在网上搜索“Windows 安装g++”，你大概率会看到 MinGW、MinGW-w64、CygWin、msys2 等概念；由于同样的网络原因，你下载的 MinGW-w64 的安装程序也不一定能正常工作。对于初学者，我的建议是：暂时忘掉它们。

若你已经安装了 Dev C++，那你可以借用其自带的编译器。找到它的安装目录，将其下 TDM-GCC-64\bin 目录加入到 PATH 环境变量中即可。（如果你不知道什么是环境变量，请参见[补充阅读](#)）。

否则，你可以下载 TDM GCC。Windows 下为何有这么多五花八门的 GCC 很难用一两句话说清楚，值得庆幸的是，在编写算法题的体验上，它们并没有什么差异。前往官网 (<https://jmeubank.github.io/tdm-gcc/download/>) 下载 tdm64-gcc-10.3.0-2.exe，双击安装即可。你也不必再操心环境变量的事情。

补充阅读：PATH

环境变量是在操作系统中设定的一些参数，通常用于全局配置。应用程序可以读取到环境变量的值，并据此决定自己的行为。例如，很多应用程序若检测到用户设定了环境变量 HTTP_PROXY 及 HTTPS_PROXY，就会选择通过代理访问网络。

PATH 是最常用的环境变量之一。考虑这样一个问题：当你在命令行敲下指令、想要执行一个程序的时候（如果没有过这样的体验，可以试试打开一个命令行，敲下 notepad 或者 explorer，看看会发生什么），系统怎么知道你要执行的程序在哪里呢？答案就在 PATH 这个环境变量中。

你可以通过 echo %PATH% 这个指令查看其值（若使用 Linux，则为 echo \$PATH）。同样的方法也适用于其他环境变量。

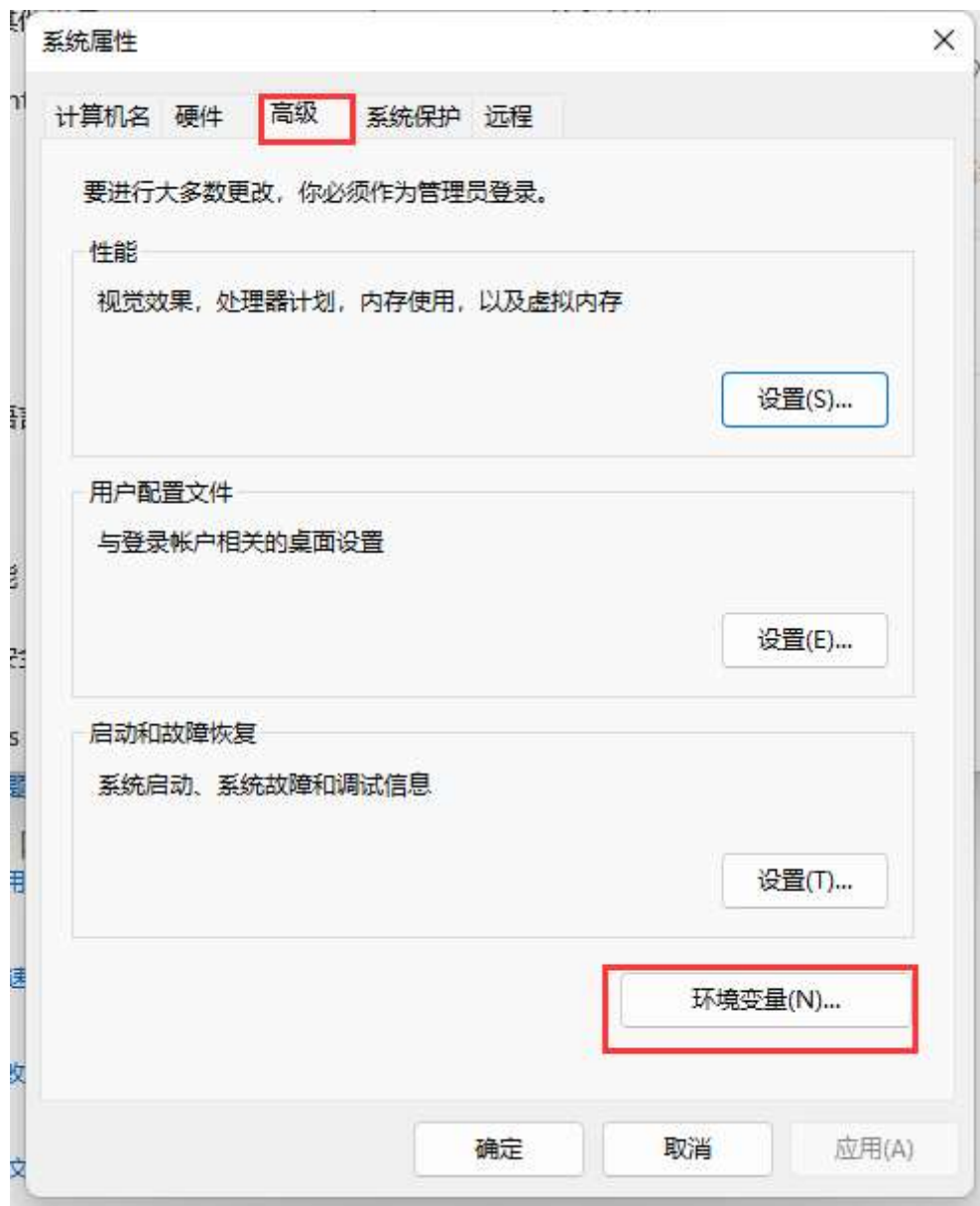
```
C:\Users\my\ye>echo %PATH%
C:\Program Files (x86)\Razer\ChromaBroadcast\bin;C:\Program Files\Razer\ChromaBroadcast\bin;C:\Program Files (x86)\VMware\VMware Workstation\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR;C:\Program Files\dotnet\;C:\Program Files (x86)\NetSarang\Xftp 7\;C:\Program Files (x86)\NetSarang\Xshell 7\;C:\Windows\system32\config\systemprofile\AppData\Local\Microsoft\WindowsApps;C:\Program Files\Microsoft SQL Server\150\Tools\Binn\;C:\Program Files (x86)\dotnet\;C:\TDM-GCC-64\bin;C:\Users\my\ye\scoop\apps\gcc\current\bin;C:\Users\my\ye\go\bin;C:\Users\my\ye\scoop\apps\nodejs-lts\current\bin;C:\Users\my\ye\scoop\apps\nodejs-lts\current;C:\Users\my\ye\scoop\apps\vscode\current\bin;C:\Users\my\ye\scoop\apps\openjdk\current\bin;C:\Users\my\ye\scoop\apps\oh-my-posh\current\bin;C:\Users\my\ye\scoop\apps\miniconda3\current\scripts;C:\Users\my\ye\scoop\apps\miniconda3\current\Library\bin;C:\Users\my\ye\scoop\apps\llvm\current\bin;C:\Users\my\ye\scoop\apps\dotnet-sdk\current;C:\Users\my\ye\scoop\apps\cuda\current\bin;C:\Users\my\ye\scoop\apps\gsudo\current;C:\Users\my\ye\scoop\apps\rustup\current\cargo\bin;C:\Users\my\ye\scoop\shims;C:\Users\my\ye\AppData\Local\Microsoft\WindowsApps;C:\Users\my\ye\dotnet\tools;C:\Users\my\ye\dotnet\tools;D:\Dev\zlib123d\ll\x64\dll_x64;C:\Users\my\ye\dotnet\tools;D:\Ktago\TensorRT-8.2.3.0\lib;C:\Program Files (x86)\Tencent\QQGameTest\Hall.37793\
```

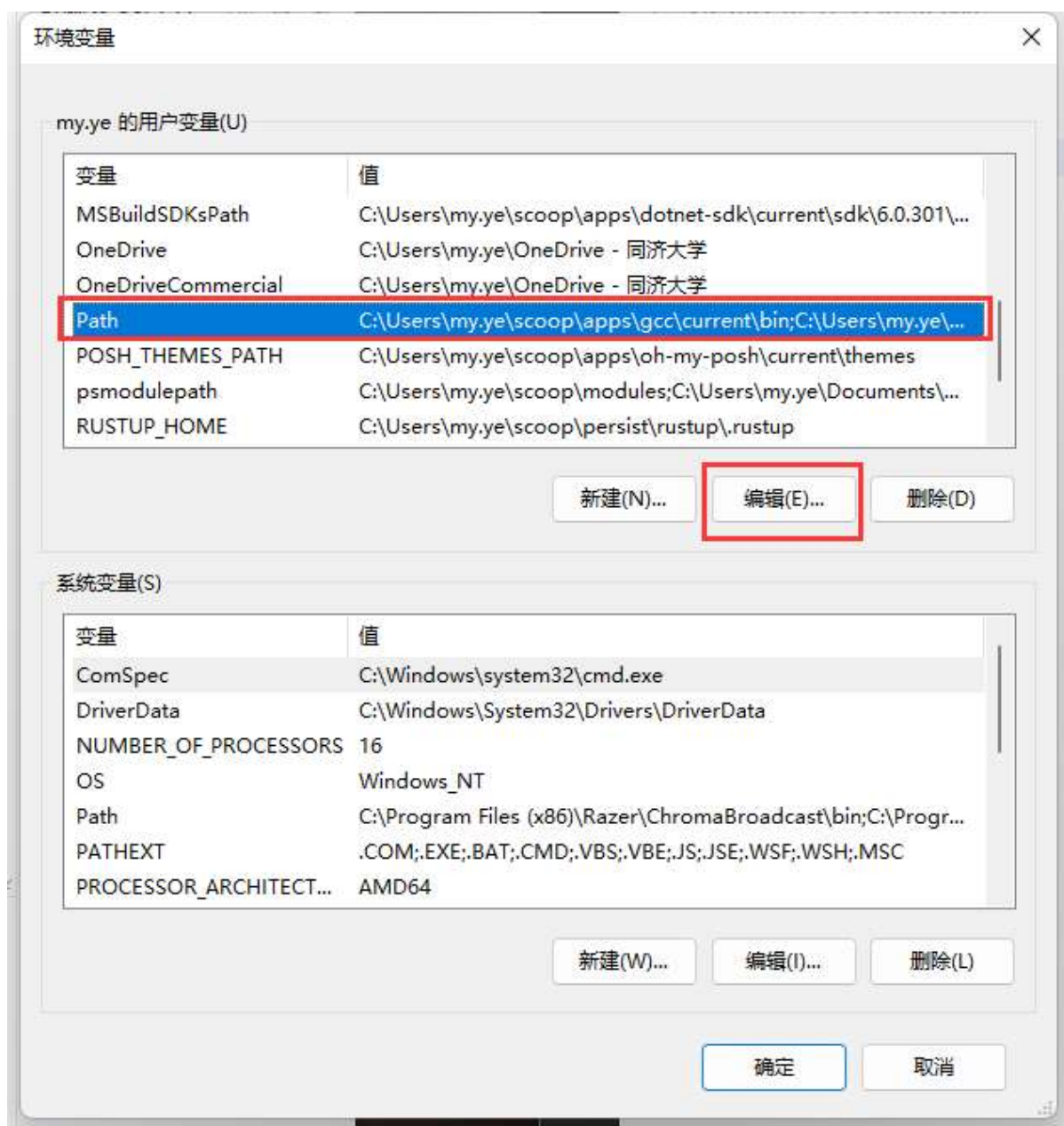

可以看到，PATH 的值是若干用分号分割的路径。当在命令行输入程序名时，系统会依次在这些目录下查找是否存在相应的可执行文件。

因此，如果我们想要在命令行中使用 g++ 进行编译，就需要将 g++.exe 这个可执行文件所在的目录添加在 PATH 中。这一操作可以在图形化界面中完成：

在开始菜单中搜索“环境变量”，或者通过设置-系统-相关链接-高级系统设置，均可打开“系统属性”对话框。在其中“高级”选项卡下，可以找到设置环境变量的入口。







在命令行下使用编译器

在阅读本节之前，请确保已经正确安装了 GCC。可以通过 `g++ --version` 命令来验证这一点。

```
C:\Users\my.ye>g++ --version
g++ (tdm64-1) 10.3.0
Copyright (c) 2020 Free Software Foundation, Inc.
本程序是自由软件；请参看源代码的版权声明。本软件没有任何担保；
包括没有适销性和某一专用目的下的适用性担保。
```

出现版本信息，就表示接下来可以正常使用了。

现在，使用编辑器编写一份代码，假设其保存在 `test.cpp` 中。可以使用如下指令将其编译：

```
g++ test.cpp -o test
```

其中，`-o` 意味指定“output”，即编译得到的可执行文件名称。你也可以把“test”改成你喜欢的任意名字。

如果编译正确完成，则不会有任何输出；否则，命令行中会显示错误提示，如下图所示：

```
C:\Users\my.ye\dev\demo>g++ test.cpp -o test
test.cpp: In function 'int main()':
test.cpp:7:13: error: expected ';' before '}' token
   7 |         return 0
     |         ^
   8 |     }
     |     ~~~~~
```

修改程序，直至完成编译，当前目录下会出现名为“test”的可执行文件。

```
C:\Users\my.ye\dev\demo>g++ test.cpp -o test

C:\Users\my.ye\dev\demo>dir
驱动器 C 中的卷是 Windows
卷的序列号是 60A8-A85C

C:\Users\my.ye\dev\demo 的目录

2022/07/17  12:14    <DIR>          .
2022/07/17  12:12    <DIR>          ..
2022/07/17  12:14                   113 test.cpp
2022/07/17  12:14             2,881,931 test.exe
                2 个文件      2,882,044 字节
                2 个目录 460,844,015,616 可用字节
```

可以通过在命令行中键入 test 来执行：（若你使用 Linux，则需要显式指定路径，即使用 ./test）

```
C:\Users\my.ye\dev\demo>test
Hello world!
```

当然，我们也可以把编译与运行融合在一行：

```
g++ test.cpp -o test && test
```

编译选项可以以任意顺序附加在命令之后。通常，我们使用 -Wall -Wextra -O2 三个编译选项，以开启严格的代码警告，同时启用优化。

如果你有在命令行中使用 gdb 进行单步调试的需求，则不要使用 -O2，改为使用 -g，以产生调试信息。

gdb 是一个非常强大的单步调试工具，强烈建议至少掌握其基本使用。可以通过网络寻找教程。

选择你的编辑器

现在，你已经有能力在命令行中编译运行程序了。接下来要做的事情就是挑选一个合适的编辑器以改善开发体验。这里有很多不错的选项：非常极客范的 Vim/Emacs，或者是 Sublime Text。当然，最受欢迎的编辑器无疑是 Visual Studio Code 了。

Vim 学习曲线十分陡峭，但熟练掌握后效率极高。另外，日后工作中难免会遇到没有图形界面可用的紧急状况，掌握 Vim 的基本操作可以拿来应急。因此，强烈推荐适当学习一下。Vim 有很多交互式的教程，用于快速入门，不难在互联网上找到。

如果你掌握了 Vim，可以在 VS Code 中开启 Vim 模式。

现代化的选择：VS Code + clangd + clang-format

VS Code 的强大与全面很难用一两句话说清。它开放的插件生态为其赋予了无限的可能。

初学者在配置 VS Code 时，可能会因为一些网络资料的引导而花大量的时间配置 json 文件，来试图做到类似 IDE 中“一键编译运行”的效果。这个做法不能说不不对，但存在几个问题：

- 配置非常麻烦，且对每个文件夹/工作区都需要相应的配置文件
- 大部分教程配置的实际上是单步调试任务，运行速度会比较慢
- 依赖于官方的 C/C++ 插件，而其开发体验比不上第三方插件 clangd

更推荐的做法是：仅将 VS Code 作为编辑器使用，使用命令行编译运行程序；如果有调试的需求，命令行下使用 gdb 也可以满足。

这并没有看起来那么麻烦：首先，VS Code 有内嵌的终端，可以通过快捷键 `Ctrl + ~`（键盘左上角的那个飘号）唤起与隐藏；其次，可以安装 Code Runner 插件，用快捷键代替手动键入命令。

如果你发现 `Ctrl + ~` 快捷键失效了，请检查是不是和其他应用的快捷键发生了冲突。默认设置下，QQ拼音会占用此快捷键：



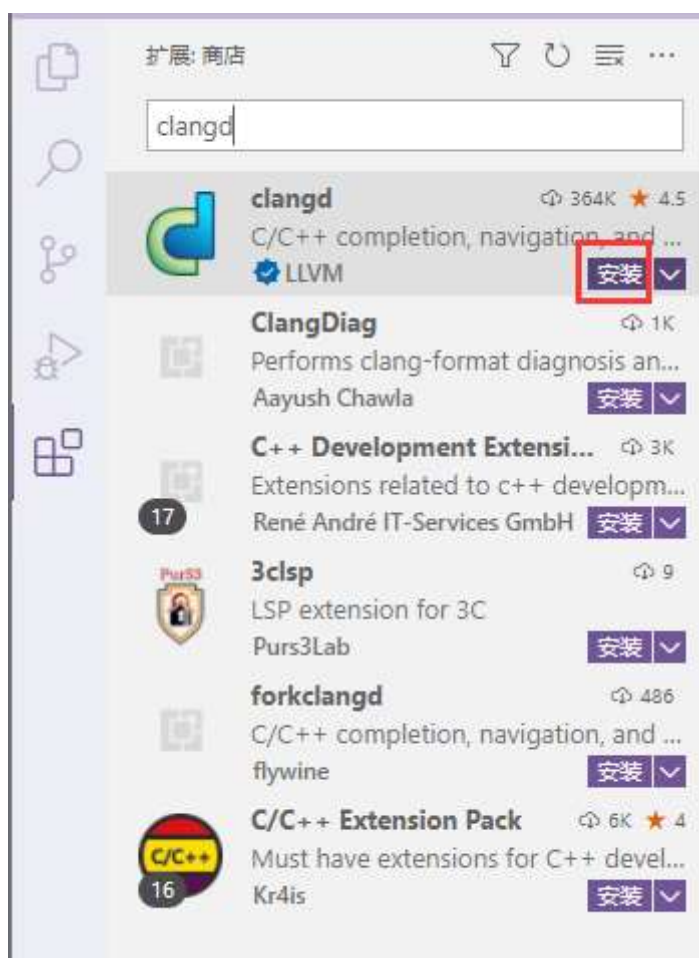
VS Code 的安装、配置与使用。

首先，[下载](#) 并安装 VS Code。它的初始界面是英文，如果你需要中文界面，按下 F1（请记住这个操作，VS Code 的大部分功能都可以通过按下它再搜索关键词以使用），搜索“Configure Display Language”，在其中选择“中文（简体）”，等待相应插件自动完成下载安装，并按提示重新加载 VS Code 即可。



使用 Ctrl + , 或左下的齿轮按钮可以打开设置面板。颜色主题、字体字号的设置，可以自行探索。

VS Code 大部分强大的功能均由种种扩展提供。使用快捷键 Ctrl + Shift + X 可以打开扩展页面。我们搜索“clang”：

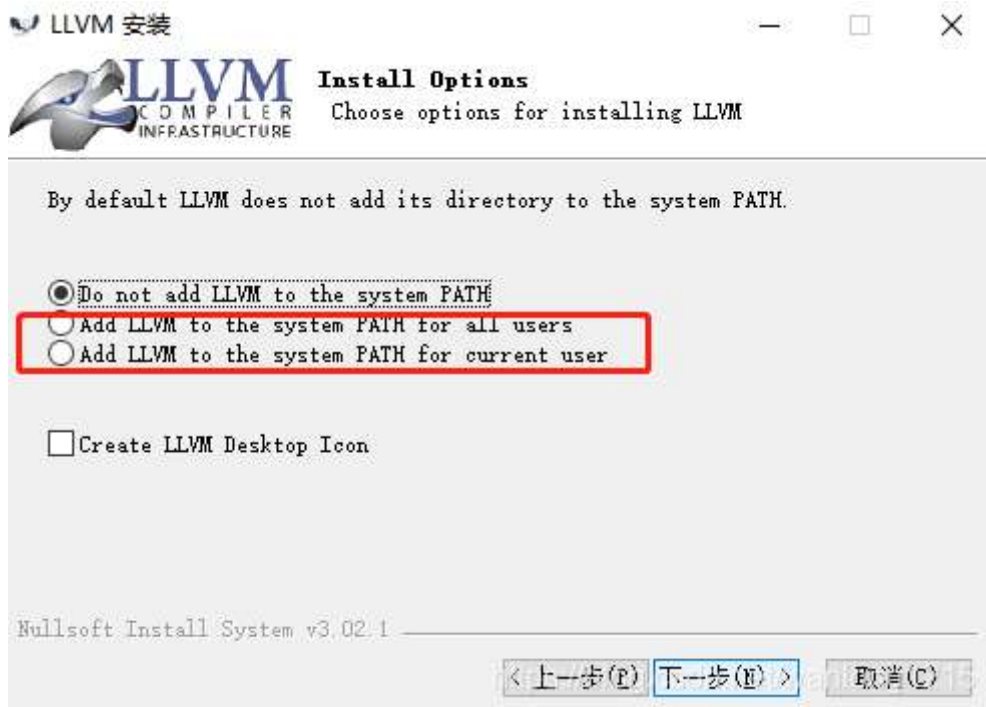


点击“安装”即可。

扩展商店中的“clangd”仅仅只是在 VS Code 与 clangd 语言服务之间建立的一个桥梁，想要使用它，需要安装 clangd 的本体。VS Code 中的 clangd 扩展可以自动下载安装相应语言服务（或者按 F1 后使用“clangd: Download Language Server”），但不幸的是，由于众所周知的网络原因，这个下载存在相当大的失败的可能性。另外，我们通常会使用另外一个扩展 clang-format 来进行代码格式化，所以在这里推荐大家直接下载安装 LLVM 全家桶：

进入官方下载页面，(<https://github.com/llvm/llvm-project/releases/tag/llvmorg-14.0.6>)，下载并安装 LLVM-14.0.6-win64.exe（当然，这也可能因为同样的网络问题而失败，因此我会在群文件里也发一份）。

安装时注意选择将其加入环境变量 PATH（否则还需要手动操作一遍），如下图所示：



在命令行中输入 `clangd --version`，若出现版本信息，则意味着安装成功。

```
C:\Users\my.ye>clangd --version
clangd version 14.0.5
Features: windows
Platform: x86_64-pc-windows-msvc
```

另外一个值得推荐的扩展是 `clang-format`。尽管 `clangd` 本身具备格式化代码的功能，但 `clang-format` 可以允许基于预设的风格去定制代码格式。确保 LLVM 已经被添加到了环境变量 PATH 中，然后在扩展商店安装 `clang-format` 即可。具体的风格定制，可以自行在互联网查找资料和示例。

不出意外的话，就可以享受到流畅的开发体验了。

VS Code 使用注意事项与技巧

- VS Code 的使用逻辑是基于文件夹或工作区的。除非只是简单浏览一下文件，否则都应该先打开一个文件夹，然后再进行工作。
- 可以安装 Code Runner 这一扩展免去每次打开终端、输入命令的麻烦。
- VS Code 有很多实用的快捷键和小功能，例如 `Ctrl+/将当前行转为注释，Shift+Alt+F 格式化代码，Shift+Alt+DownArrow 将当前行复制一份插入下方，F2 重命名选中的变量或者函数，Alt+LeftArrow 快速回到上一次浏览的位置等等，可以点击“帮助-编辑器操场”来交互式地学习`

我想用 Linux

上述所有的配置均是基于 Windows 的。如果想要一个正宗的 Linux 开发环境，成本最低的方案是使用 Windows 10 提供的 Linux 子系统功能（WSL）。其配置和使用不难在网上找到清晰的教程。在完成安装后，可以通过 VS Code 的 Remote-WSL 扩展连接到 Linux 子系统，并获得和在 Windows 下几乎完全相同的使用体验。

■ 高程选手请注意：使用此功能需要在 cmd 中关闭“使用旧版控制台”选项。

Linux 下配置开发环境比 Windows 简单很多。例如，在 Ubuntu 中，只需要：

```
sudo apt install g++ clangd-12 clang-format-12
```

就可以解决所有问题。建议有兴趣的同学自行尝试。