

CS100 Recitation 0

Environment Setup

Speaker Ye Jiang (jiangye2026@shanghaitech.edu.cn)

VSPLab@SIST



上海科技大学

ShanghaiTech University

目录

1 前置知识：编辑器、编译器、IDE

2 编译器的安装

3 终端（命令行，命令提示符）

4 Visual Studio Code

前置知识：编辑器、编译器、IDE

编辑器、编译器、IDE

写代码之前，先分清三样东西。

编辑器：任何可以编辑文本的软件，例如记事本、Word、WPS 文字、手机备忘录。

但我们真正需要的是**代码编辑器**，尤其是**现代代码编辑器**：`Notepad++`、`Visual Studio Code`、`Vim`、`Sublime Text`……

编译器：用来将高级语言的代码翻译成计算机真正能执行的代码。流行的 C/C++ 编译器有 `GCC`（GNU）、`Clang`（LLVM）、`MSVC`（Microsoft）、`ICC`（Intel）等。

IDE（Integrated Development Environment，集成开发环境）：代码编辑器 + 编译器 + 调试器 + 项目管理工具 +……

立志成才 报国裕民

我们到底需要什么？

- 好的 IDE 有各种工具，例如 linter、profiler、版本管理等。
- **宇宙最强 IDE**：Microsoft Visual Studio。
- 常见的 IDE：VS、CLion、PyCharm……
- 我们实际需要的是一个 IDE——但不必是它的“打包形态”。

两条路线都行：

- 直接用 CLion 或 VS 这样的狠货；
- 用编辑器 + 编译器 + 插件自己搭一套，这也是本课程的推荐路线。

立志成才 报国裕民

编译器的安装

编译器的安装

安装方式按平台分三类：

- **Linux**：gcc/g++
- **Windows**：通过 WSL 使用 gcc/g++
- **Mac**：自带 clang/clang++

Linux（以 Ubuntu 为例）

一行命令即可安装编译所需的一整套工具（gcc、g++、make 等）：

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

在 Ubuntu24.04 默认安装的是 gcc-13，26.04 默认安装的是 gcc-15。

每个新版本的 GCC 都会增加对更新标准的支持：

- gcc-9：对 C++17 支持比较完善；
- gcc-11：改进了 C++20 的支持；
- gcc-13/14：开始逐步实现 C++23 的特性。

同时编译器优化更强，错误检查与诊断也更好。

立志成才 报国裕民

Windows（WSL）

WSL（Windows Subsystem for Linux，Windows 的 Linux 子系统）是微软官方提供的功能，本质上是一个在 Windows 上运行的**只有命令行环境的 Linux 虚拟机**。

换句话说：在 Windows 里，你可以直接打开一个 Linux 终端，运行 `apt`、`bash`、`gcc`、`python` 等工具，就像真的在 Linux 下一样。

从这学期开始，课程演示全部在 Linux 环境下进行：Windows 用户请全程使用 WSL，Mac 用户直接使用自带终端。

WSL 安装

WSL 的安装需要在**正确的网络环境**下，在管理员权限的 PowerShell 中执行：

```
wsl --install
```

会自动安装 WSL 和默认的 Ubuntu，装完**重启电脑**生效。

如果网络环境不正确，可以先去 GitHub 的 `microsoft/WSL` releases 下载最新**x86**架构的 `.msi` 手动装上 WSL 本体[1]，再执行 `wsl --install` 安装 Ubuntu；若安装卡在 0%，可加参数 `--web-download`。

装完 Ubuntu 后首次打开，会让你创建一个用户名和密码（注意密码输入时屏幕不显示字符，是正常的）。

[1]次选方案：如果实在没有正确的网络环境，可以试试这个办法。

立志成才 报国裕民

WSL 安装的版本选择

推荐使用 Ubuntu 26.04 或 Ubuntu 24.04，两种版本本课程都支持：

- `wsl --install`：Ubuntu 26.04（默认 gcc-15）；
- `wsl --install Ubuntu-24.04`：Ubuntu 24.04（默认 gcc-13）。

装好后用 `cat /etc/os-release` 确认版本，后面换源时选择对应的脚本。

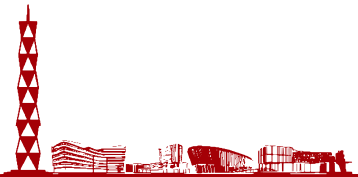
立志成才 报国裕民

双系统：更好的选择

WSL 只有命令行环境，如果你希望体验完整的 Linux 桌面，可以考虑**双系统**：

- 在磁盘上划出一块空间，在保留 Windows 的情况下安装完整的 Linux 系统；
- 开机时可以选择启动哪个系统；
- 推荐 Ubuntu 26.04 或 24.04，也可以使用其他发行版。

我们真诚鼓励所有学生尝试双系统方案，体验在一个完整的 Linux 桌面系统上工作。



WSL 环境配置：换源

首次进入 WSL 后，先把软件源换到国内镜像，否则 apt 下载很慢。

先执行 `cat /etc/os-release` 确认 Ubuntu 版本（看 `VERSION_CODENAME`），然后按版本选择换源脚本：

- Ubuntu24.04（noble）：`magic_code_2404.md`
- Ubuntu26.04（resolute）：`magic_code_2604.md`

版本必须匹配

24.04 的机器用 26.04 的源（或反过来）会导致仓库里所有包都比系统旧，`apt install` 会报依赖错误。

WSL 环境配置：安装编译工具

上海科技大学
ShanghaiTech University

WSL 与 Linux 配置编译环境

按顺序执行，`build-essential` 会一次装齐 `gcc`、`g++`、`make` 等工具：

```
sudo apt update
sudo apt upgrade
sudo apt install build-essential
```


立志成才 报国裕民

Mac 环境配置

Mac 不用额外配置，系统自带 clang 编译器。用下面的命令检查版本即可：

```
clang --version
```

终端（命令行，命令提示符）

终端是什么

终端有很多种，叫法也很多：

- Terminal
- Shell
- Command Line
- Command Prompt
- ……

本质上都是同一件东西：一个用文字命令和计算机对话的窗口。

打开终端

- **Linux（Ubuntu）**：按 `Ctrl+Alt+T`，或在应用列表中搜索 Terminal；
- **Mac**：`Cmd`+空格 搜索“终端”（Terminal）；
- **Windows（WSL）**：开始菜单搜索 Ubuntu，或打开 PowerShell 后输入 `wsl` 进入。

立志成才 报国裕民

在终端使用指令

bash 常见命令（不一定全都掌握，但 cd 必须会）：

- `ls`：列出当前工作目录（Current Working Directory，CWD）下的文件；
  - 默认情况下打开终端，工作目录是用户家目录 `~`（Linux 为 `/home/用户名`，Mac 为 `/Users/用户名`）；
- `cd somePath`：将工作目录切换到 `somePath`，可以是相对或绝对路径；
  - `cd ..`：切换到上一级目录；
  - `cd 'My Folder'`：切换到 `My Folder`，加引号是因为有空格；
- `mkdir CS100`：在当前工作目录下创建名为 `CS100` 的文件夹；
- `rmdir CS100`：删除这个文件夹（要求文件夹为空）；
- `rm file`：删除名为 `file` 的文件；
- `rm -r CS100`：删除名为 `CS100` 的文件夹及其所有内容。

验证编译器安装：PATH

输入 gcc，shell 会在 PATH 环境变量所包含的路径中寻找名为 gcc 的可执行文件，它通常位于 /usr/bin。

如果编译器的安装正确，应当看到如下输出：

```
gcc: fatal error: no input files
compilation terminated.
```

这不是报错，而是编译器在告诉你“给我点活干”。

查看编译器版本

给 gcc 加上 --version 参数即可查看版本：

```
gcc --version
```

输出形如：

```
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
```

顺带一提：gcc 和 clang 是 C 编译器，g++ 和 clang++ 是 C++ 编译器。

立志成才 报国裕民

编译运行一个 Hello World 程序

假设你的 CS100 课程的文件夹位于 `~/courses/CS100`。

在 `~/courses/CS100/tmp` 创建 `hello.c`，用 VSCode 打开，输入如下代码：

```c
#include <stdio.h>
int main() {
    puts("Hello world");
    return 0;
}
```

编译

打开终端并切换到 `~/courses/CS100/tmp`，输入如下命令：

```
1 gcc hello.c -o hello
```

将会得到可执行文件 `hello`。该文件名由紧跟在 `-o` 之后的字符串指定。

立志成才 报国裕民

运行

在终端里输入 `./hello` 即可运行 `hello`。

`./` 表示当前工作目录，因为 Linux/Mac 默认不会在当前目录下搜索可执行文件。

输出：

```
Hello world
```

向编译器传递参数


```
1 gcc hello.c -o hello
```

跟在 gcc 之后的各项是传递给 gcc 的**参数**。

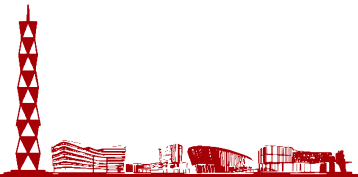
- hello.c 是它需要编译的文件名，如果不在当前工作目录下，需要指明路径，例如 ../../CS101/pa1/a.c；
- -o hello 指明生成的可执行文件的名字，也可以带有路径；Linux/Mac 下可执行文件通常无后缀名；
- 还可以有其它参数，例如 -std=c17 指明采用 2017 年发布的语言标准 C17，-lm 链接数学库，--save-temps 保留编译过程中产生的临时文件；
- 以上参数的顺序无所谓。

Visual Studio Code

什么是 VSCode

VSCode 是一个现代的**代码编辑器**。

- 它不能帮你**编译**程序——“为什么我的代码在 VSCode 上跑出来是这样……?”这类问题的答案往往不在编辑器;
- 但它有大量好用的插件，可以帮你一键调用编译器、方便地调试代码、版本管理、在远程/容器里开发，甚至上 QQ、逛知乎、听网易云、炒股……



下载、安装 VSCode

- 去官网（`code.visualstudio.com`）下载安装包；
- **Linux**：下载 `.deb` 文件，双击安装或 `sudo apt install ./code_*.deb`；
- **Mac**：下载 `.zip` 解压后拖入 Applications；
- **Windows（WSL 用户）**：VSCode 装在 Windows 上，然后安装**WSL**扩展，通过左下角绿色图标连接 WSL。

立志成才 报国裕民

配置 VSCode

VSCode 有 $N+1$ 套配置：

- 每个工作区/文件夹可以有自己的配置，以 `json` 文件的形式存放在 `.vscode` 目录下（. 开头的文件/文件夹默认隐藏，文件管理器里按 `Ctrl+h` 或 `ls -a` 可以显示）；
- 另外有一套全局配置，也是 `json` 文件的形式，存在于某个特定的位置；
- 某一项设置如果在当前工作区/文件夹的配置中出现了，就采用这里指定的值，否则采用全局配置中的值。

立志成才 报国裕民

安装插件（扩展，extension）

以下四个插件已经足够：

- **C/C++**：提供 C/C++ 代码高亮、静态分析等功能；
- **Chinese (Simplified)**：中文语言包，界面汉化；
- **vscode-icons**：更好的文件图标显示（安装后记得 enable）；
- **GlassIt-VSC**：按 `Ctrl+Alt+z` 变透明，按 `Ctrl+Alt+c` 变回来。

不要装太多

不建议安装很多 C/C++ 相关的插件（例如 Clang Command Adapter、C/C++ Runner 等），否则配置起来很困难。个人不建议安装 VSCode 推荐的 C/C++ Extension Pack。

立志成才 报国裕民

全局配置

按 `Ctrl+,`，或者左下角的齿轮 →Settings：

- Editor: Format On Save 设为 true；
- Editor: Format On Type 设为 true；
- Editor: Accept Suggestion On Enter 个人建议设为 off，避免和换行冲突（按 `Tab` 仍然可以接受补全）。

工作区配置

1. 为这门课程创建文件夹，例如 `~/courses/CS100`；
2. `File→Open Folder...`（快捷键 `Ctrl+k`，`Ctrl+o`）打开 `CS100` 文件夹。以后每一次都要先打开 VSCode，再打开文件夹，再编辑文件；而不是直接双击文件用 VSCode 打开，否则工作区的配置无法起效；
3. 创建 `~/courses/CS100/.vscode` 文件夹，创建文件 `.vscode/c_cpp_properties.json`，内容见课堂上传的 `vscode-config-files.md`——注意阅读“注意事项”。

代码格式化：先看一段反面教材

先看一段**能编译**但没人想读的代码：

```
1 int main(){int x=1,y=2;printf("%d\n",x+y);
2     return 0;}
```

缩进、空格、换行、大括号位置，统称**代码风格**（code style）。风格不影响程序对错，但严重影响可读性——包括两天后的你自己、批改作业的 TA，以及跟你协作的队友。

手动保持风格：写时费心、改时易漏、协作时易吵。所以：**交给工具自动做**。

立志成才 报国裕民

clang-format

clang-format 是 LLVM 项目出品的 C/C++ 自动格式化工具。

- **VSCode 内无需安装**：C/C++ 插件内置了它，也不需要其它格式化插件；
- **Linux 终端使用需安装**：`sudo apt install clang-format`，之后可以直接在终端里跑 `clang-format -i hello.c`；
- 风格由配置文件 `.clang-format` 决定，格式化时从当前文件所在目录**逐级向上查找**；
- 因此把它放在 `~/courses/CS100/`下（与 `.vscode` 同级），课程内所有代码统一生效；
- 我们在 4.4 已开启 `Format On Save`：`Ctrl+s` **保存的瞬间，代码自动变整齐**。

立志成才 报国裕民

.clang-format

在 ~/courses/CS100/创建 .clang-format（内容见 vscode-config-files.md）：

```
Language:         Cpp
BasedOnStyle:     LLVM
IndentWidth:      4
AccessModifierOffset: -4
ColumnLimit:      100
```

- Language: Cpp 对 C 和 C++ 都生效（没有 C 这个选项）；
- IndentWidth: 4：缩进 4 格；ColumnLimit: 100：超长行自动折行；
- 想手动格式化：右键 →Format Document（快捷键 Alt+Shift+f）；
- 试一试：故意把 hello.c 弄乱，然后 Ctrl+s。

Makefile：自动构建

到目前为止，编译靠手敲：`gcc hello.c -o hello`。很快你会发现：命令越来越长、文件越来越多，还要靠向上键（↑）翻历史命令。

`Makefile`：把编译规则写进一个文件，以后每次只敲一个词：`make`。

最小的 Makefile 只有两行（与源码同目录创建文件 `Makefile`，注意大小写）：

```
hello: hello.c
    gcc hello.c -o hello
```

Makefile：规则的含义

一条规则 =**目标**：**依赖**+**命令**。意思：想生成 `hello`，需要 `hello.c`，办法是执行下面那行命令。

- **命令行首必须是一个 Tab**，不是空格（否则报 `missing separator`，最经典的坑）；
- `make` 比较时间戳：`hello` 不存在或比 `hello.c` 旧才重新编译，否则提示 `'hello' is up to date`；
- 试一试：连跑两次 `make`；`touch hello.c` 后再 `make`。

Makefile：加上变量

把编译器和选项集中到一处：

```
CC      = gcc
CFLAGS = -Wall -Wextra -Wpedantic -std=c17

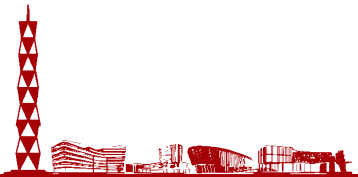
hello: hello.c
    $(CC) $(CFLAGS) hello.c -o hello

clean:
    rm -f hello
```

- `$(CC)`、`$(CFLAGS)` 取变量的值；换编译器、改选项只动一处；
- `make` 构建，`make clean` 清理产物；
- 多文件、增量编译、模式规则等内容，课程后面用到时再讲。

配置总结

- 目前为止，以上配置已经足够，不要胡乱粘贴别人的配置文件；
- 遇到任何问题请首选官网（code.visualstudio.com）和 StackOverflow，或者在 Piazza 提问，或单独找 TA。



Thank You