

# 【C/C++ 新手大礼包 之一】

龙爸写给初学 C/C++ 的朋友们的**新手入门指南**，扫平入门的几个大障碍的**极简教程**：

1. **安装 C/C++ 编译环境**： [《C/C++ 运行环境安装配置指南——初学者极简版》](#) | [PDF版下载](#)
2. **用 VSCode 写你的第一个 Hello World**： [极简安装、配置中文语言包、C/C++ 扩展](#) | [PDF版下载](#)
3. **算法入门练习题**： [陪小学生学习 C/C++ 练习题](#)（陆续完善中）
4. **方便设置 Path 环境变量的脚本**： 在 Path 环境变量中查看、查找、添加、删除——[使用说明](#) | [PDF版下载](#)

原文发布、维护于龙爸**陪孩子学习** `Python`、`C++` 的练习项目：

- [coffeescholar/C\\_CPP-Learning](#)，作者：[爱学习的龙爸](#)
- 欢迎意见、建议和帮助，谢谢 🤗❤️❤️
- 如果对你有帮助，请支持 Star~ ✨✨✨
- 转载时请保留声明和出处，谢谢 🙏🙏🙏

## C/C++ 运行环境安装配置指南——初学者极简版

此指南主要目的是用**尽可能极简**的方式，帮助初学者**完成 C/C++ 的基础环境搭建**，**三步搞定**。  
——尤其是小学生小朋友 🤗🤗🤗

好多家长朋友很头疼，孩子在学习 `C/C++` 的第一步遇到的各种问题不知道怎么解决，例如：

- 有的是培训机构给了一个比较老旧的安装包，勉强弄了能用，却隐藏了各种问题；
- 有的是按照某个教程一步步却走不通，要么下载遇到问题，要么注册环境变量遇到问题。

通过此指南，**用尽可能简单的方式**，指导初学者一步步完成 `C/C++` 的**编译器、调试器的下载、安装、配置和更新**。

本教程仅针对 Windows 10/11 版本，不推荐 Windows 7/XP，本教程也不考虑 MacOS、Linux 等系统。

本教程发布和更新于作者陪同孩子学习 Python、C/C++ 的 Github 项目：

## 三个基本动作如下：

1. **在线安装** Scoop —— 软件包部署和升级工具，以后经常会用到；
2. **自动安装设置** gcc, gdb  
—— GNU 的 C/C++ 编译器和调试器  
—— 不需要设置环境变量等等，自动完成了
3. **测试一下**，确保安装正确。

**大功告成！** 恭喜你迈出了第一步 😊

之后，补充后续可能用到的一点点小技巧：

- **更新已经安装的软件包**
- **解决访问不了 Github.com 的问题**：下载、解压缩、运行 Fastgithub.UI  
—— 需要的话，后续补充说明。

**下载安装** VSCode **并配置** C/C++ 扩展，请自行查阅其它文档

## 1. 在线安装 Scoop

### 1.1. 运行终端

有很多种方法，这里只说最简单的：

在 Windows 10/11 中，按下 Win + x 键，在弹出菜单里选中：终端（不要选 终端管理员）或 Windows Powershell



左边是 Win11， 右边是 Win10。

## 1.2. 在线安装 Scoop

随后在打开的 终端（命令行）界面，复制粘贴下面几行命令并按回车（# 后面是注释）：

```
# 设置 Powershell 的用户策略，如果提示确认请选择 'Y'
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
# 这里用国内镜像代替了官方的安装地址
iwr -useb https://gitee.com/glsnames/scoop-installer/raw/master/bin/install.ps1 |
iex
# 安装 git
scoop install git
# 更新一下 scoop 自己的脚本和数据，执行后稍微等一等就好
scoop update
```

可能会看到类似的提示：

执行策略更改 执行策略可帮助你防止执行不信任的脚本。更改执行策略可能会产生安全风险，如 <https://go.microsoft.com/fwlink/?LinkID=135170> 中的 about\_Execution\_Policies 帮助主题所述。是否要更改执行策略? [Y] 是(Y) [A] 全是(A) [N] 否(N) [L] 全否(L) [S] 暂停(S) [?] 帮助 (默认值为"N"): **y**

输入 **y**，同意执行后续脚本。

没有什么意外的话，进度条唰唰唰的过，稍微等一等就能看到提示完成后续安装。

用 **Scoop** 来安装一些开发常用的工具非常方便，以后会经常用到。

**Scoop** 功能非常强大，远远不止这一点，建议以后有空多了解一下。

**Scoop** 是一款适用于 **Windows** 系统的 命令行软件（包）管理工具，免费、开源：[官网](#)、[Github 开源项目地址](#)。

简单来说，就是可以通过命令行工具（**PowerShell**、**CMD** 等）实现软件（包）的安装管理等需求，通过简单的一行代码实现软件的下载、安装、卸载、更新等操作。

其灵感来源于 **macOS** 的 **Homebrew** [开源项目地址](#)，**Mac** 用户就比较熟悉了。

**Scoop** 不仅可用于安装软件包，还可以更新、批量更新，甚至导出安装列表，重装系统后一次性批量安装。支持很多常用软件，包括 **VSCode**、钉钉、微信等，比如**最强视频音频编解码工具**：**ffmpeg**。

尤其**对于开发人员而言**，更方便安装指定版本、安装多个版本、运行指定版本，在很多情况下非常有用。

### 1.3. 切换为国内镜像（可跳过）

**Scoop** 正常情况下是连接到 **Github.com** 来获取相应的更新数据等等，

因为国内访问 **Github.com** 不稳定，切换为国内镜像会比较方便一些。

**如果你之前用官方的方式安装过** **Scoop**，可以用下面命令切换为国内镜像：

```
# 配置仓库为国内镜像，感谢 Scoop 团队，感谢国内镜像提供者
scoop config SCOOP_REPO https://gitee.com/glsnames/scoop-installer
# 配置主桶为国内镜像
cd $env:SCOOP\buckets\Main
git remote set-url origin https://gitee.com/scoop-bucket/main.git
```

感谢 Scoop 团队，感谢国内镜像提供者 🙌 ❤️ ❤️

后面会推荐访问 `Github.com` 的工具 `FastGithub`，虽然访问速度不快但至少能访问。

## 2. 自动安装设置 gcc, gdb

随后，用 `Scoop` 帮助我们自动下载安装 `GNU` 的 `C/C++` 编译器和调试器：`gcc`，`gdb`——`g++` 已经包含在 `gcc` 里。

在 终端（命令行）界面 粘贴或输入下面命令：

```
# 先装一个下载工具（可以跳过）
scoop install aria2
# 安装 gcc gdb
scoop install gcc gdb
```

这个过程中，`Scoop` 会自动安装所需的 `7zip`，不需要设置环境变量等等，因为 `Scoop` 已经自动完成了。

## 3. 测试一下，确保安装正确

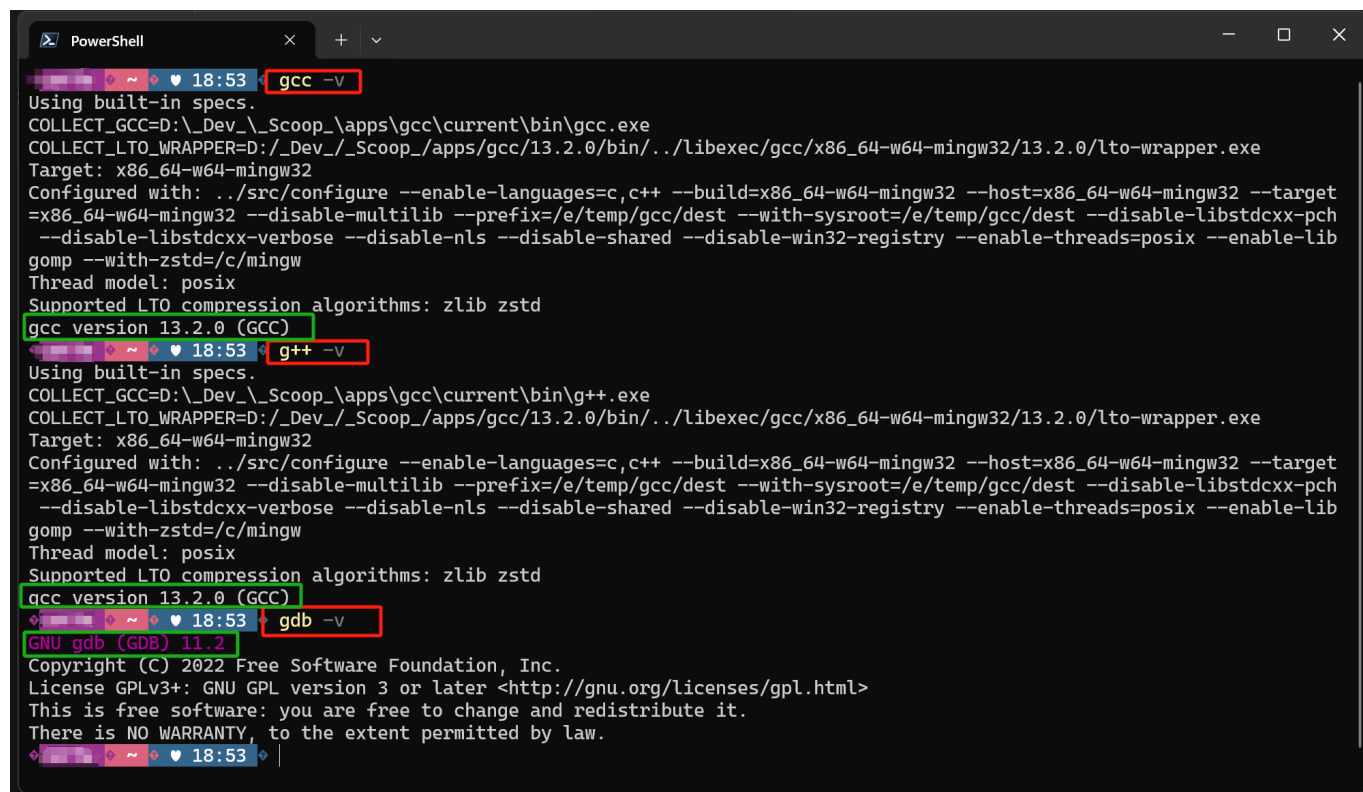
分别运行测试一下，在 终端（命令行）界面 中分别执行下面的命令：

```
gcc -v
g++ -v
gdb -v
```

如果安装失败，则会提示找不到对应的命令，如：

gcc: The term 'gcc' is not recognized as a name of a cmdlet, function, script file, or executable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.

安装正确的情况下，在每个命令输出中能看到提示和版本号，如图中绿色部分：



```
PowerShell
18:53 gcc -v
Using built-in specs.
COLLECT_GCC=D:\_Dev\_Scoop\_apps\gcc\current\bin\gcc.exe
COLLECT_LTO_WRAPPER=D:/_Dev/_Scoop_/apps/gcc/13.2.0/bin/./libexec/gcc/x86_64-w64-mingw32/13.2.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../src/configure --enable-languages=c,c++ --build=x86_64-w64-mingw32 --host=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --disable-multilib --prefix=/e/temp/gcc/dest --with-sysroot=/e/temp/gcc/dest --disable-libstdcxx-pch --disable-libstdcxx-verbose --disable-nls --disable-shared --disable-win32-registry --enable-threads=posix --enable-libgomp --with-zstd=/c/mingw
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 13.2.0 (GCC)
18:53 g++ -v
Using built-in specs.
COLLECT_GCC=D:\_Dev\_Scoop\_apps\gcc\current\bin\g++.exe
COLLECT_LTO_WRAPPER=D:/_Dev/_Scoop_/apps/gcc/13.2.0/bin/./libexec/gcc/x86_64-w64-mingw32/13.2.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../src/configure --enable-languages=c,c++ --build=x86_64-w64-mingw32 --host=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --disable-multilib --prefix=/e/temp/gcc/dest --with-sysroot=/e/temp/gcc/dest --disable-libstdcxx-pch --disable-libstdcxx-verbose --disable-nls --disable-shared --disable-win32-registry --enable-threads=posix --enable-libgomp --with-zstd=/c/mingw
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 13.2.0 (GCC)
18:53 gdb -v
GNU gdb (GDB) 11.2
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

## 4. 后续可能用到的

### 4.1. 更新已经安装的软件包

同样，以后可以在 终端（命令行）界面 中执行下面的命令，更新所有安装了的工具包：

```
# 第一次会自动安装 git
scoop update *
```

也可以仅更新特定的软件包：

```
# 可选，查看安装了哪些工具包
scoop list
# 更新特定的软件包，支持多个
scoop update git gcc gdb
```

## 4.2（可选）下载、解压缩、运行 Fastgithub.UI

后续学习中，经常需要访问 `Github.com`，然而国内访问 `Github.com` 经常不稳定。

如果这时候总是提示网络问题，可以用开源、免费的 `FastGithub` 工具。

因为项目原作者已经删库，下面是众多备份之一：托管在国外 `Github.com`

源代码项目地址: [GitHub - WangGithubUser/FastGitHub: GitHub加速神器，解决GitHub打不开、用户头像无法加载、Releases无法上传下载、git-clone、git-pull、git-push失败等问题](#)

最新 2.15 版的压缩包国外下载: [fastgithub\\_win-x64.zip](#)

当然，可能现在还无法稳定访问 `Github.com`，下面是备份之二：托管在国内 `Gitee.com`

源代码项目地址: [\[uid/FastGithub\]\(FastGithub: FastGithub\)](#)

2.14 版国内直接下载地址: [fastgithub\\_win-x64.zip](#)

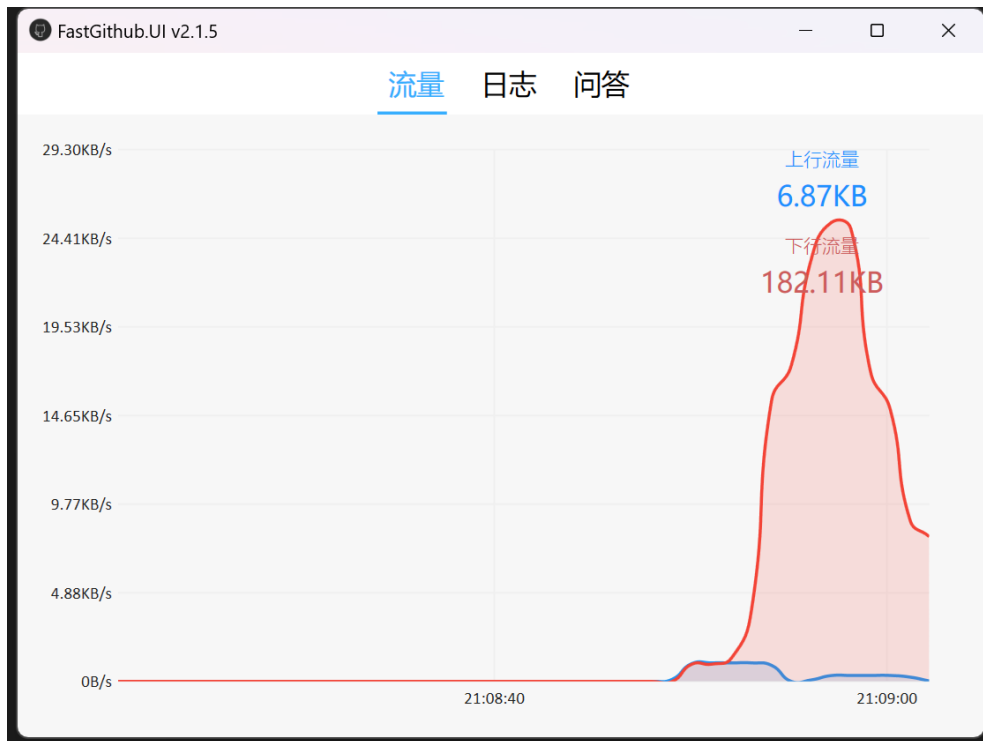
使用方式：

解压缩到本地文件夹，双击运行其中的 `FastGithub.UI.exe`

然后访问一下 `Github.com` 上的页面，例如：

爱学习的龙爸 <https://github.com/coffeescholar>

看到下面的实时流量统计图，说明已经起作用了：



很多文章中也会推荐其它类似 FastGithub 的工具，如 Steam++ 等，相比之下 FastGithub 属于最简单、功能单一的一个。——推测因为与 Steam++ 等利益相关（有广告），FastGithub 原作者删除了原来在 Github 上的项目，

但幸好有喜欢这个工具的其它朋友备份并继续维护着 FastGithub，所以上面分享的并不是原作者的项目地址，而是备份项目地址。并且在原项目基础上略做了改进和修改了 bug。

感谢原作者，也感谢这些热心的备份者和维护者（还有很多类似备份），谢谢你们 🙌 ❤️ ❤️

注意：FastGithub 工具可以配置为系统服务，在 Windows 启动时自动运行，

这里暂不展开，有兴趣可自行查阅其项目说明，如运行原理、源代码等等。

请注意作者的 **免责声明**：

- fastgithub 不具备“翻墙”功能,也没有相关的计划
- fastgithub 不支持Windows7等已被发行方停止支持的操作系统，并且也不会主动提供支持
- fastgithub 不能为您的游戏加速
- fastgithub 没有主动在github之外的任何渠道发布



# 安装 VSCode，配置中文语言包、C/C++ 扩展

---

有了 C/C++ 编译器、调试器，接下来用什么样的代码编辑软件？

有很多选择，推荐免费、开源的 VSCode —— Visual Studio Code。

龙爸继续肝，熬夜用若干头发换了一份极简教程：

[用 VSCode 写你的第一个 Hello World：极简安装、配置中文语言包、C/C++ 扩展](#)