**CPP文档**

# 文档管理

**更改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **状态** | **日期** | **负责人** | **更改原有** |
| 1.0 | 新建 | 20241108 | 卢祥洲 | 新建文档 |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录

1 文档管理 2

2 目录 3

3 数据入库映射表 4

3.1 入库流程 4

# 参考文章

<https://blog.csdn.net/weixin_52159554/article/details/134406628>

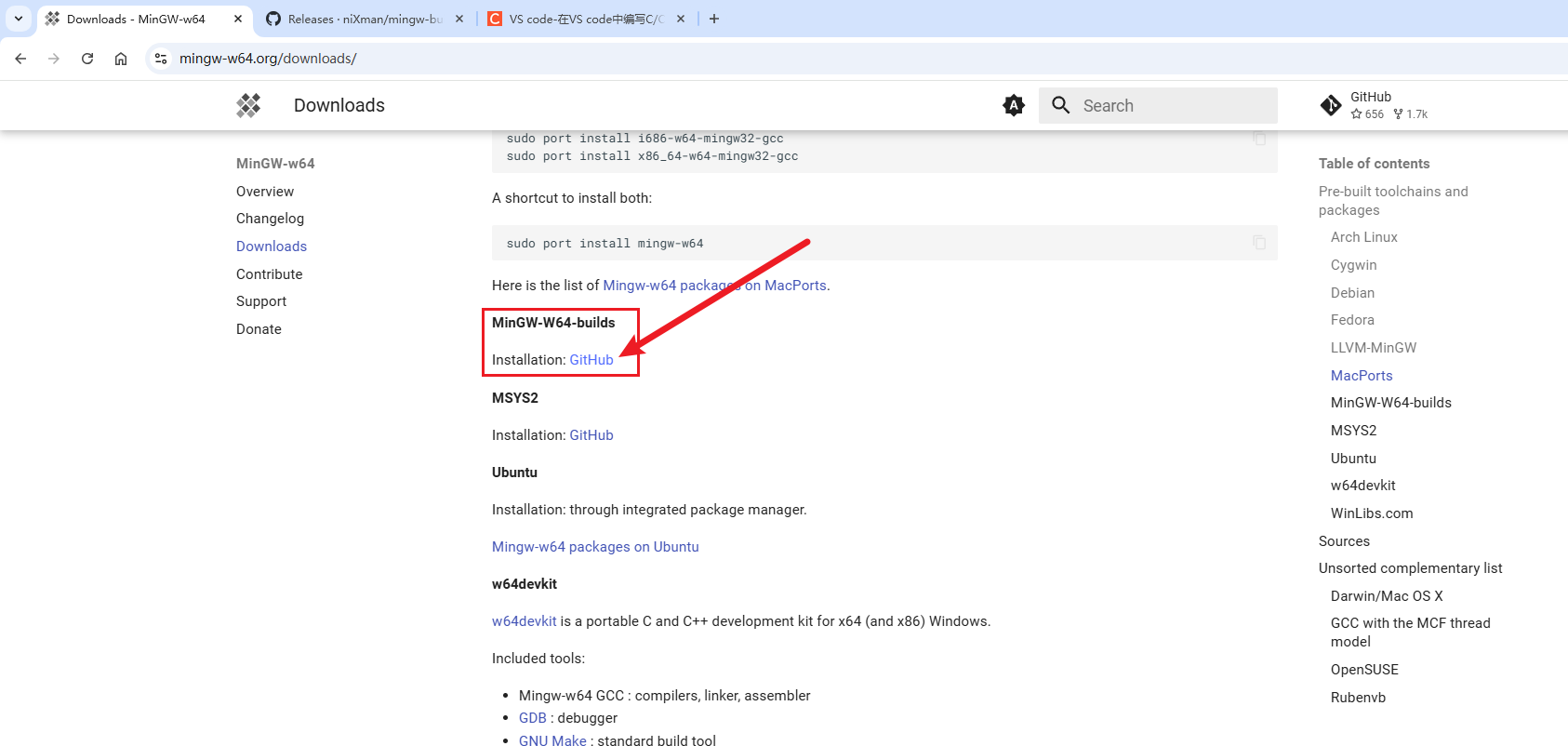
# 配置MinGW64

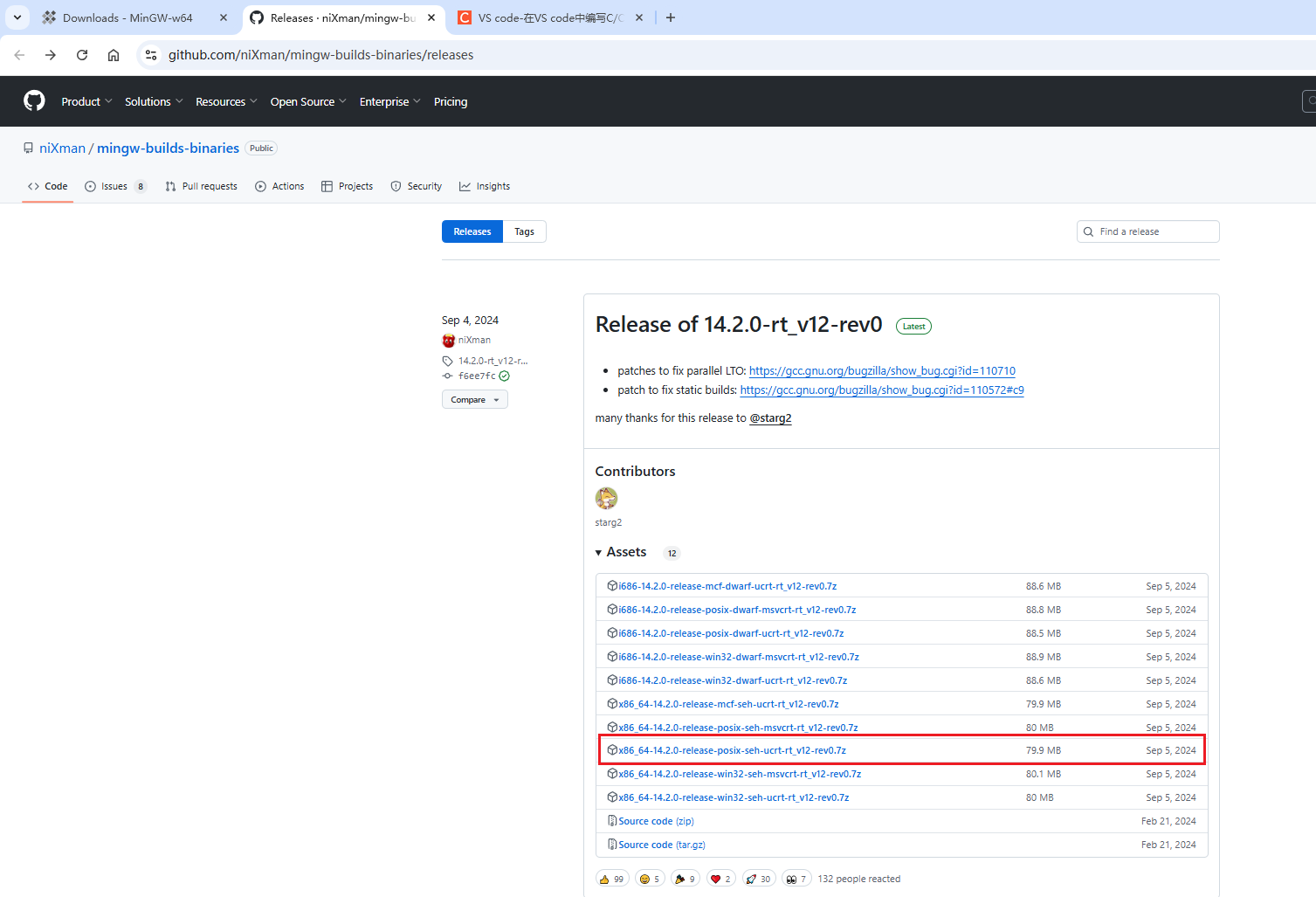
## 下载

<https://www.mingw-w64.org/downloads/>

<https://github.com/niXman/mingw-builds-binaries/releases>

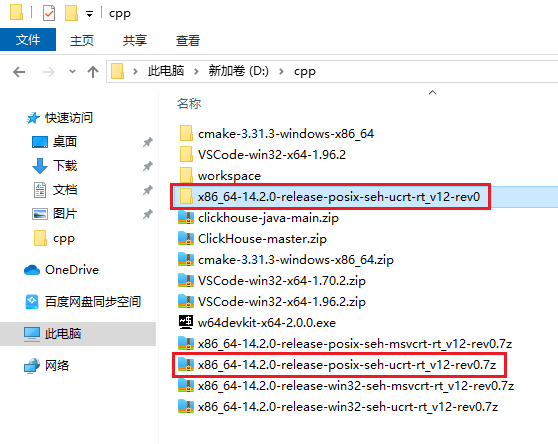
下载：[x86\_64-14.2.0-release-posix-seh-ucrt-rt\_v12-rev0.7z](https://github.com/niXman/mingw-builds-binaries/releases/download/14.2.0-rt_v12-rev0/x86_64-14.2.0-release-posix-seh-ucrt-rt_v12-rev0.7z)





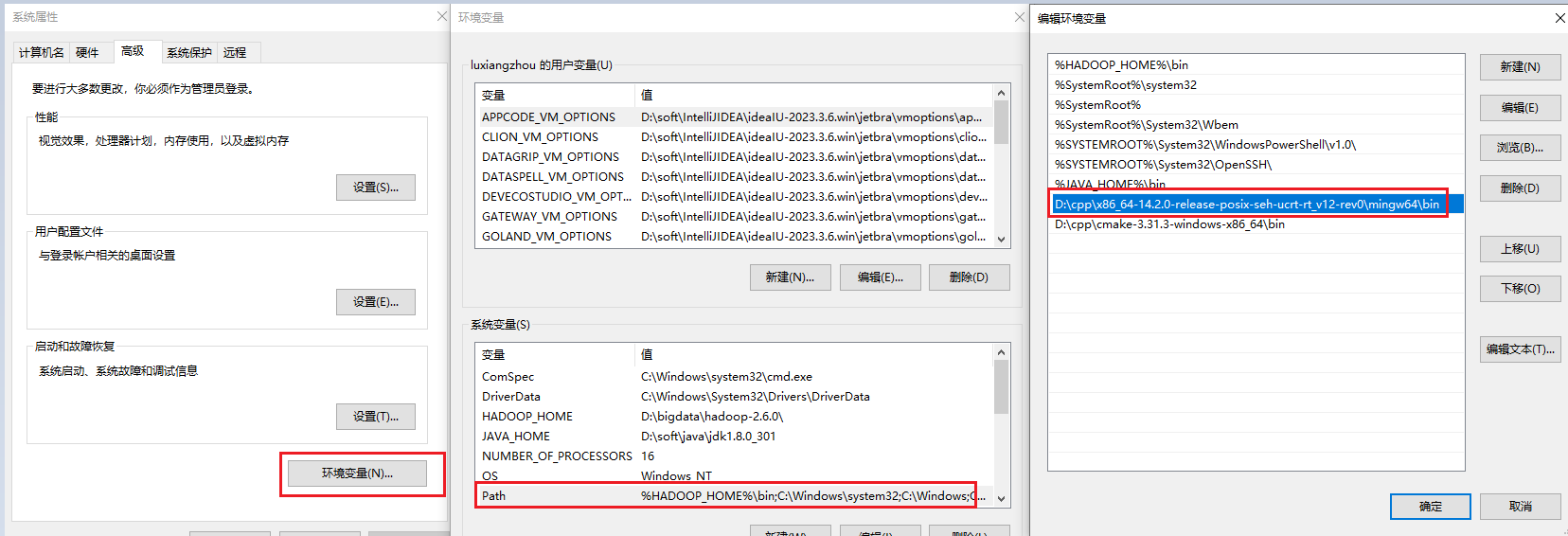
## 解压

**解压x86\_64-14.2.0-release-posix-seh-ucrt-rt\_v12-rev0.7z**



## 配置MinGW64环境变量

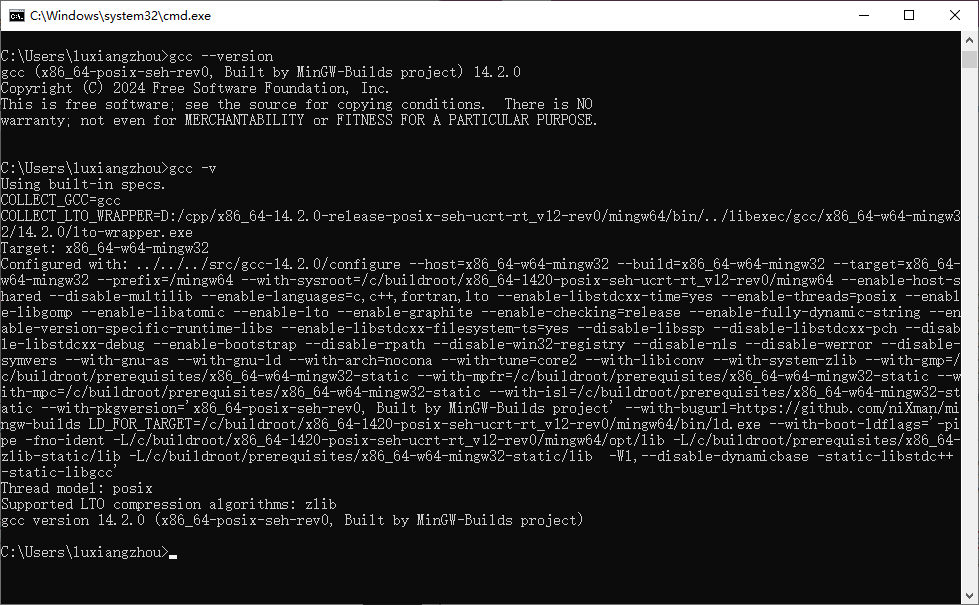
D:\cpp\x86\_64-14.2.0-release-posix-seh-ucrt-rt\_v12-rev0\mingw64\bin



## 查看gcc版本

gcc --version

gcc -v



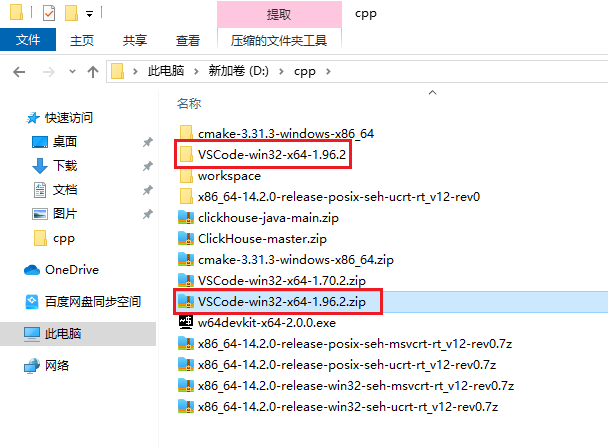
# 安装vscode

## 下载

<https://code.visualstudio.com/>

## 解压

**解压VSCode-win32-x64-1.96.2.zip，并打开vscode**



## 安装Chinese、C/C++插件



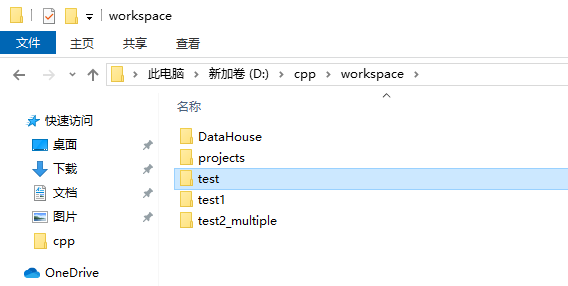


## 重启

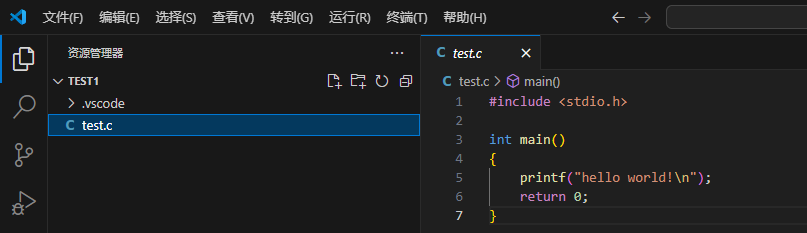
重启VSCode，这一步很重要，让前面的设置生效，要不然后面会出现问题的。

# 开发C/C++代码

## 创建项目文件夹



## 编写test.c



## 设置c\_cpp\_properties.json（非必须）

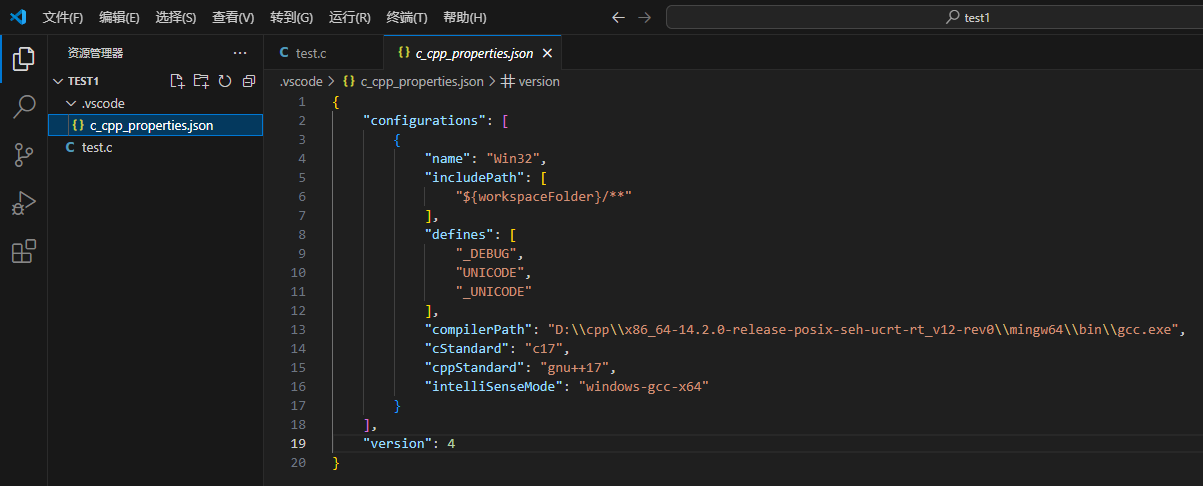
**设置C/C++编译的选项：c\_cpp\_properties.json**

 鼠标点在.c的源文件内部，按ctrl+shift+p快捷键，在弹出的界面中选择：【C/C++：编辑配置(UI)】，然后进入设置页面。





其他的选项就不再修改。当修改完以上内容后，就在你打开的文件夹下生成了一个.vscode的文件夹，这个文件夹很重要，里边放的都是配置文件，以后是可以复用的。同时你看到在.vscode文件夹下生一个新的文件c\_cpp\_properties.json,这个文件中是关于C/C++的配置属性。



## 设置tasks.json（必须）

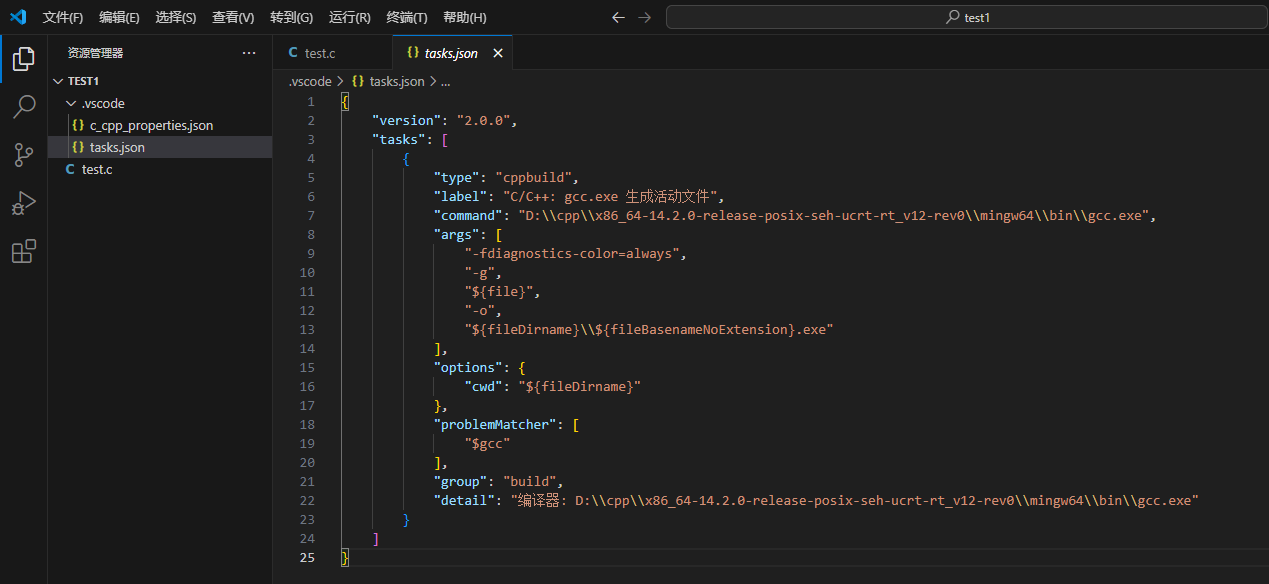
**创建执行任务：tasks.json**

1、打开test.c，打开“终端--->配置任务”，才能在配置文件框中找到【C/C++:gcc.exe 生成活动文件】；

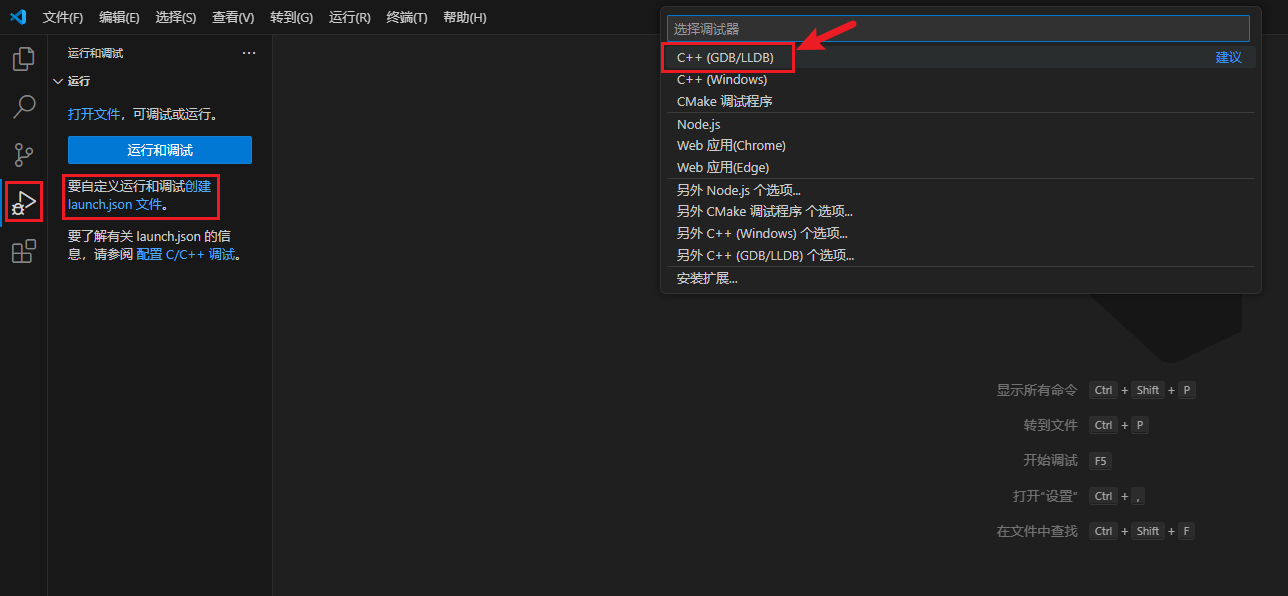
2、在.vscode文件夹中生成tasks.json文件；

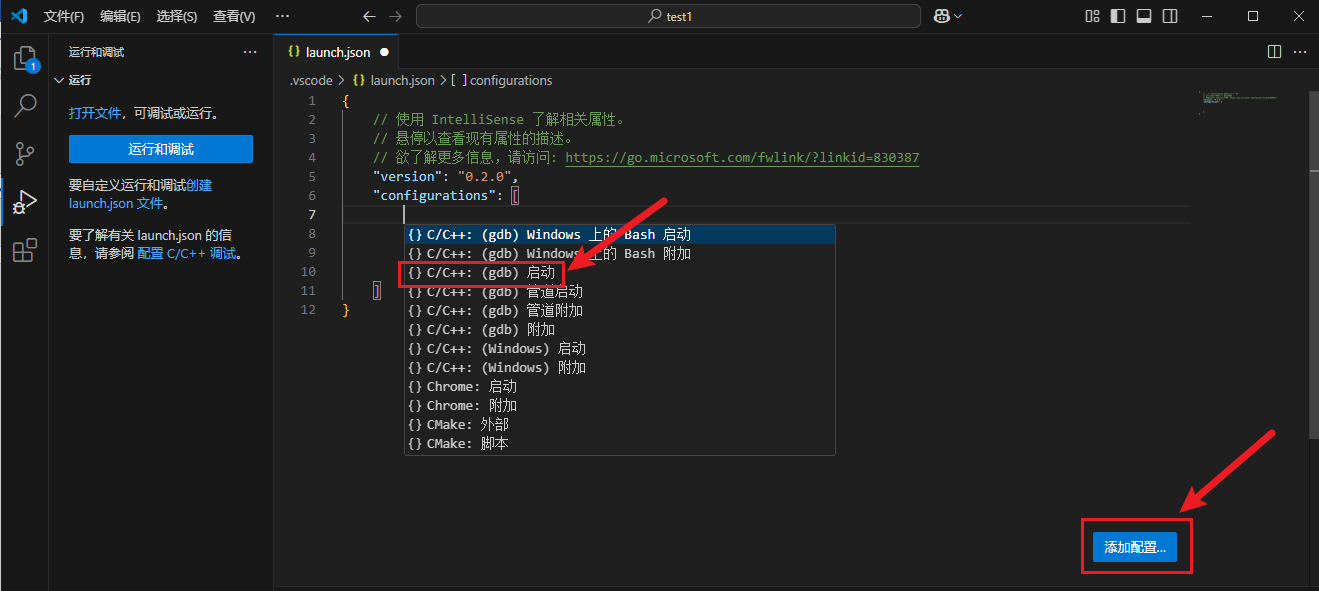


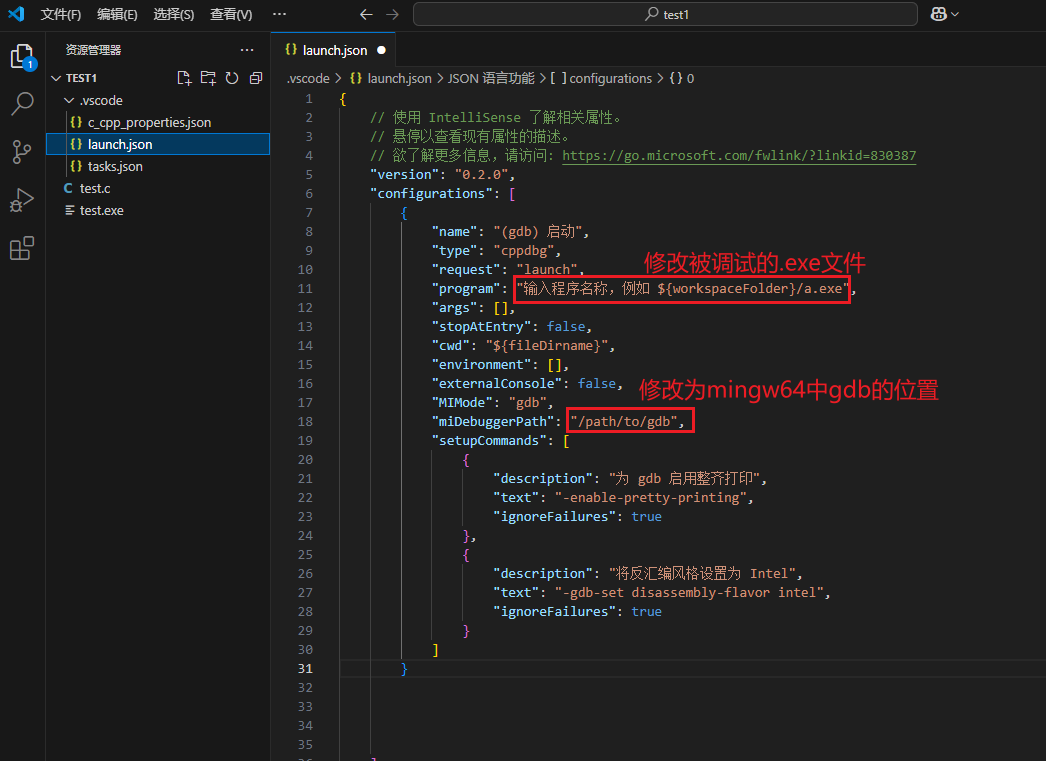




## 设置launch.json（非必须）







修改"program"为："**${workspaceFolder}\\${workspaceRootFolderName}.exe**"，这个内容和tasks.json中的可支持文件的路径匹配。

修改"miDebuggerPath"为："**D:\\cpp\\x86\_64-14.2.0-release-posix-seh-ucrt-rt\_v12-rev0\\mingw64\\bin\\gdb.exe**

"，这个路径根据自己实际放MinGW-w64的文件路径调整。

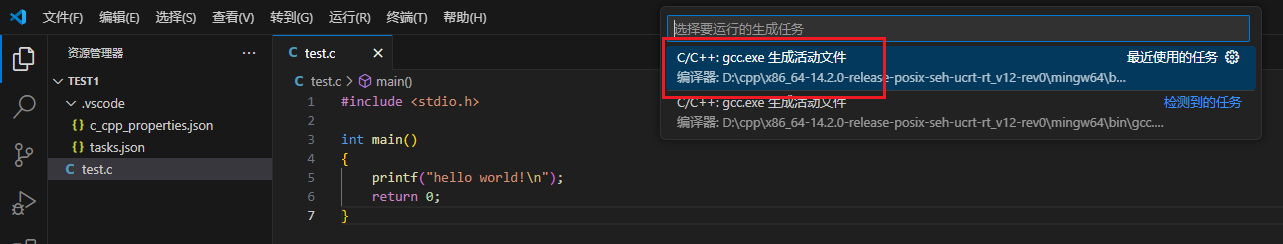
## 编译和运行一个c代码

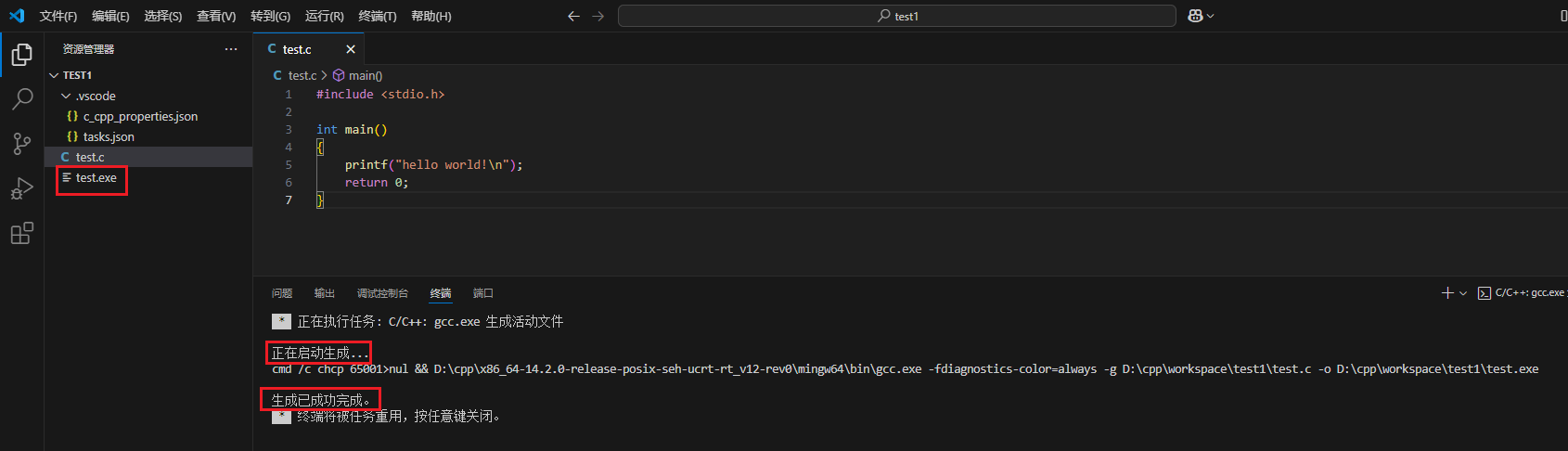
### 编译

1、终端--->运行生成任务--->C/C++:gcc.exe 生成活动文件

2、生成可执行文件 “.exe ”文件；



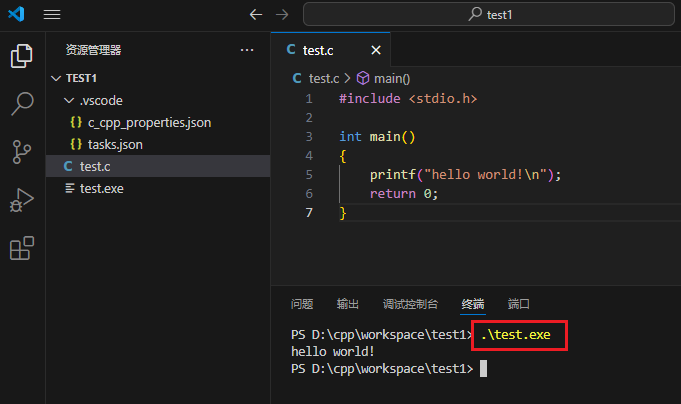




### 运行

按键盘的ctrl+`快捷键，打开VSCode自带的终端，就可以运行代码了。

这时输入：.\可执行程序的名字，即可以执行代码。



## 编译和运行多个c代码

修改tasks.json文件：

编译多个源代码文件：

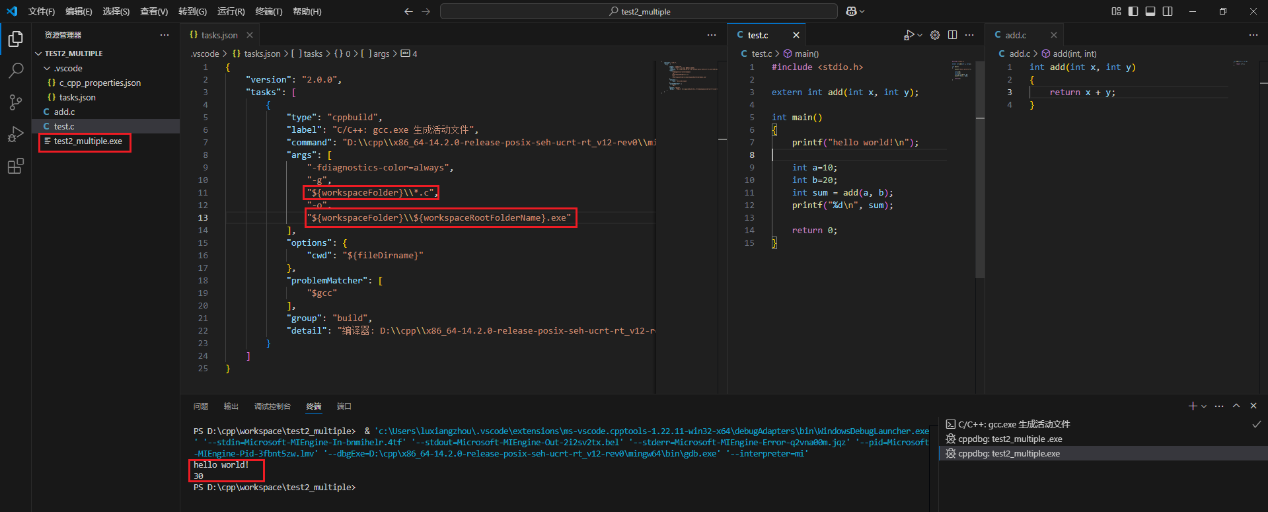
通过把"${file}"修改为"${workspaceFolder}\\\*.c"，可以构建当前工作区中的所有C文件。

通过把"${file}"修改为"**${workspaceFolder}\\\*.cpp**"，可以构建当前工作区中的所有C++文件。

修改编译输出的文件名：

通过把"${fileDirname}\\${fileBasenameNoExtension}.exe"，修改为:"${workspaceFolder}\\myProgram.exe",可以把编译输出的文件名硬编码为myProgram.exe，而不是与C文件同名。

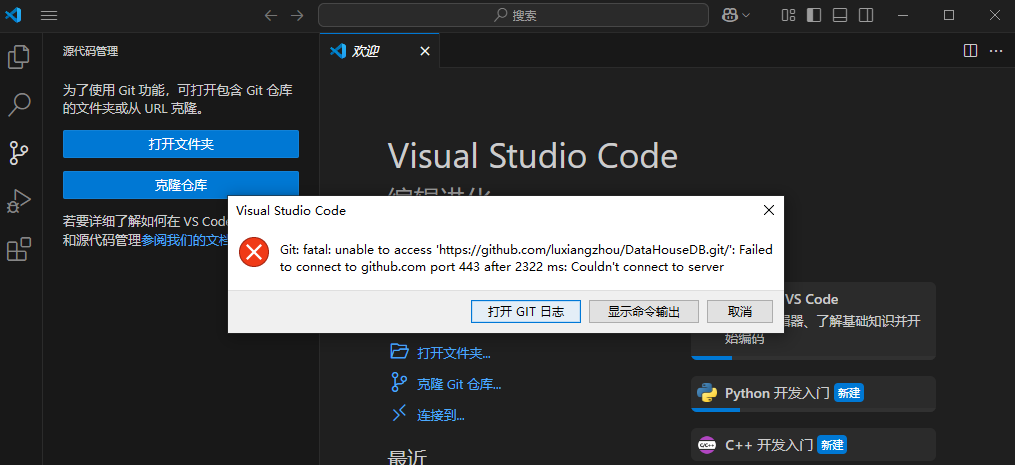
通过把"${fileDirname}\\${fileBasenameNoExtension}.exe"，修改为:"**${workspaceFolder}\\${workspaceRootFolderName}.exe**",可以把编译输出的文件名为：打开的工作区文件夹名字.exe。



# git

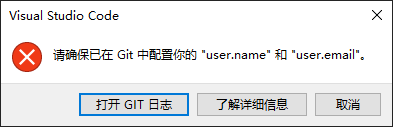
## VSCode连接GitHub报错

修改host文件，把140.82.114.4 github.com添加到C:\Windows\System32\drivers\etc里的hosts文件的最后一行，然后保存。



## VSCode Git提交失败

请确保已在 Git 中配置你的"user,name"和"user,email"。



git config --global user.name "luxiangzhou"

git config --global user.email "472773804@qq.com"