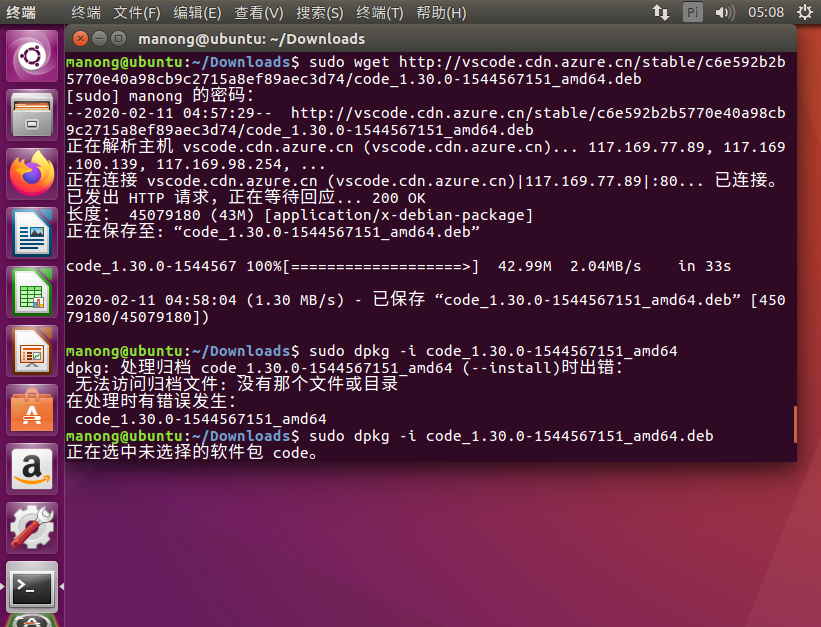
学习project第三期 学习报告

1. 任务
2. **下载Visual Studio Code的Linux版（.deb格式），并使用dpkg命令安装.deb格式的VS Code安装**
3. **使用命令下载Visual Studio Code (wget** <https://vscode.cdn.azure.cn/stable/c6e592b2b5770e40a98cb9c2715a8ef89aec3d74/code_1.30.0-1544567151_amd64.deb>**)**
4. **安装Visual Studio Code（sudo dpkg -i code\_1.30.0-1544567151\_amd64.deb）**
5. **运行（在终端任何目录下输入code）**



1. **打开Visual Studio Code，安装中文插件，Python扩展，C/C++扩展。**
2. **安装中文插件**

**按下 F1 或者 Shift + Ctrl + P 组合键**

**然后在命令行输入：Configure Display Language**

**会打开locale.json文件，将里面的 US或者en 改成 zh-CN 然后 Ctrl + S 保存**

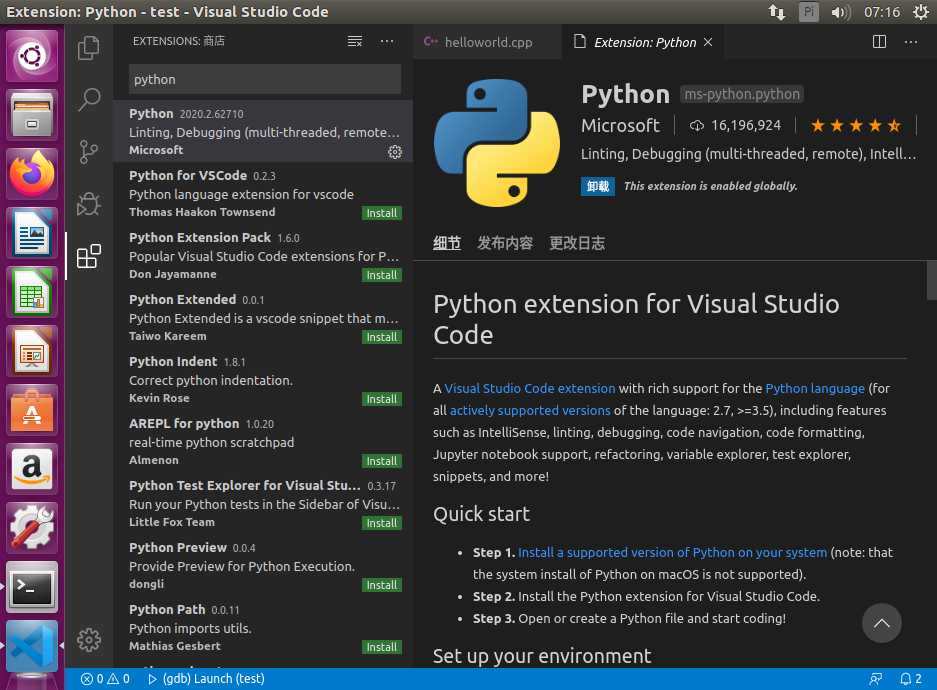
**如果不是中文版，是因为默认下载的包里没有中文语言包。**

**下载语言包:Ctrl + P，然后在命令行输入以下内容并回车：ext install MS-CEINTL.vscode-language-pack-zh-hans**

**然后再重启vscode，就已经是简体中文了。**

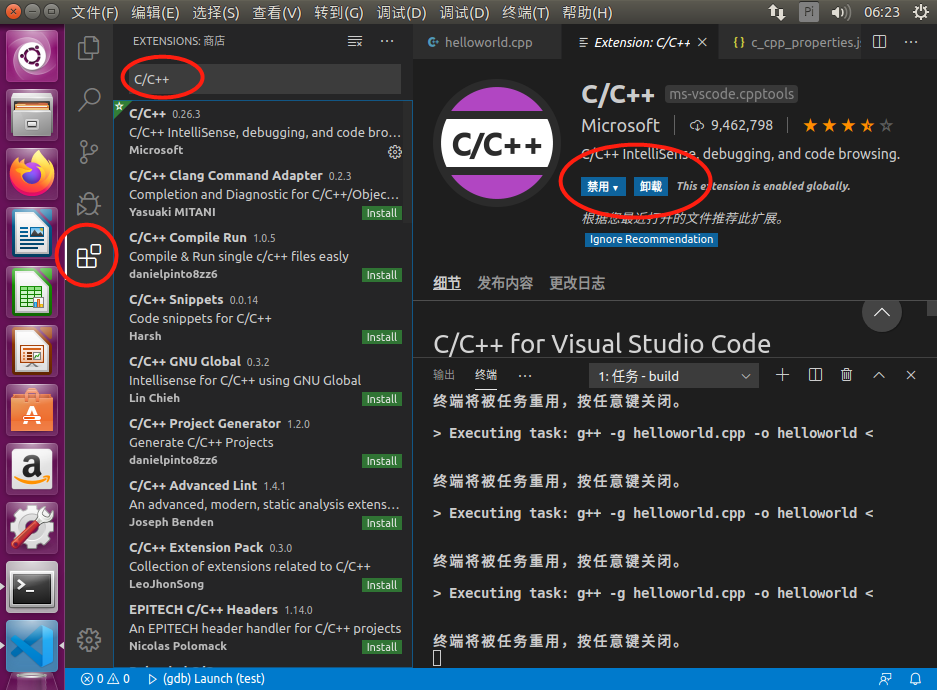
1. **Python扩展**

**直接在扩展中搜python，install即可。**



1. **C/C++扩展**

* **在扩展中安装C/C++插件**



* **生成目录和文件**

**新建文件夹【test】,并新建文件helloworld.cpp文件,文件中内容如下，**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main(int argc, char const \*argv[])**

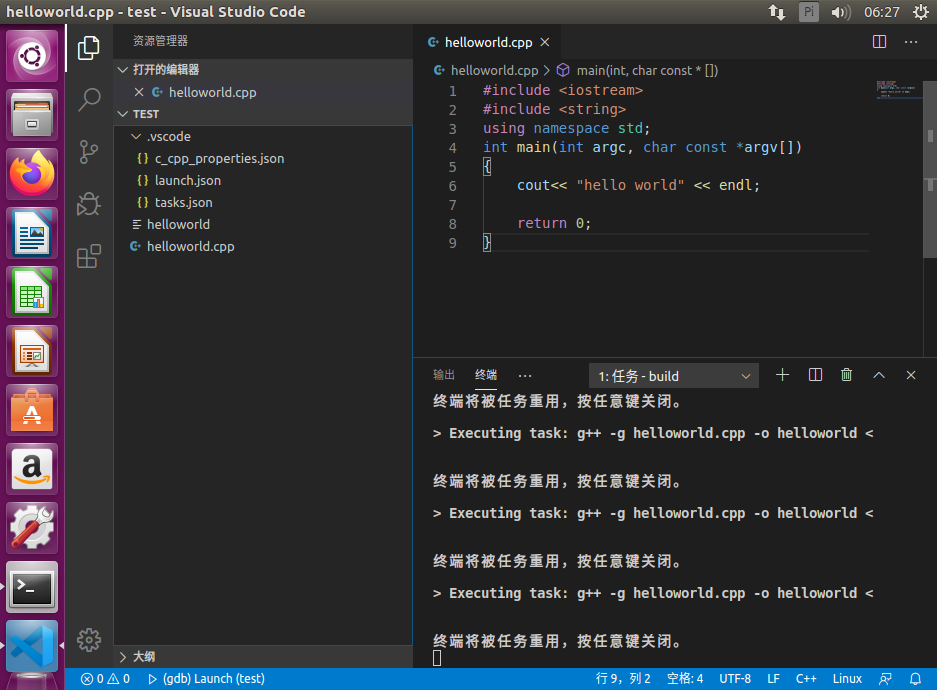
**{**

**cout<< "hello world" << endl;**

**return 0;**

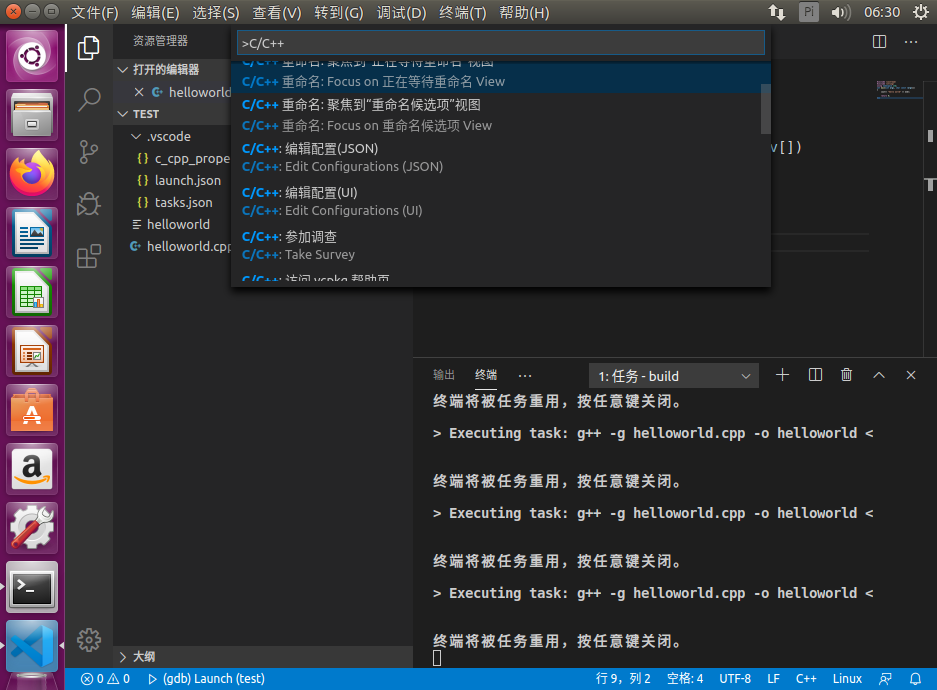
**}**

**使用vscode打开文件夹**



* **配置c++ IntelliSense**

**使用F1，打开命令选项，输入C/C++，选择C/C++:Edit configuration，生成c\_cpp\_properties.json配置文件。**



**{**

**"configurations": [**

**{**

**"name": "Linux",**

**"includePath": [**

**"${workspaceFolder}/\*\*"**

**],**

**"defines": [],**

**"compilerPath": "/usr/bin/gcc",**

**"cStandard": "c11",**

**"cppStandard": "c++17",**

**"intelliSenseMode": "clang-x64"**

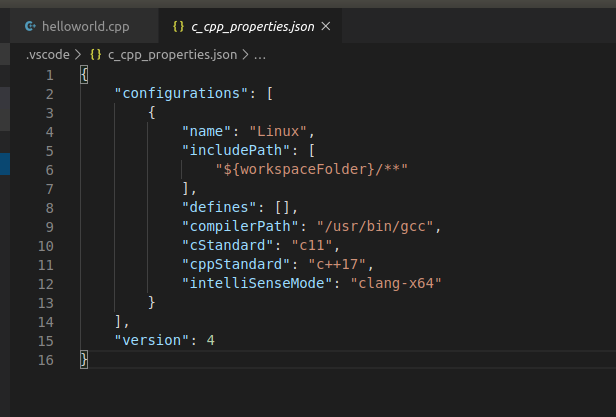
**}**

**],**

**"version": 4**

**}**

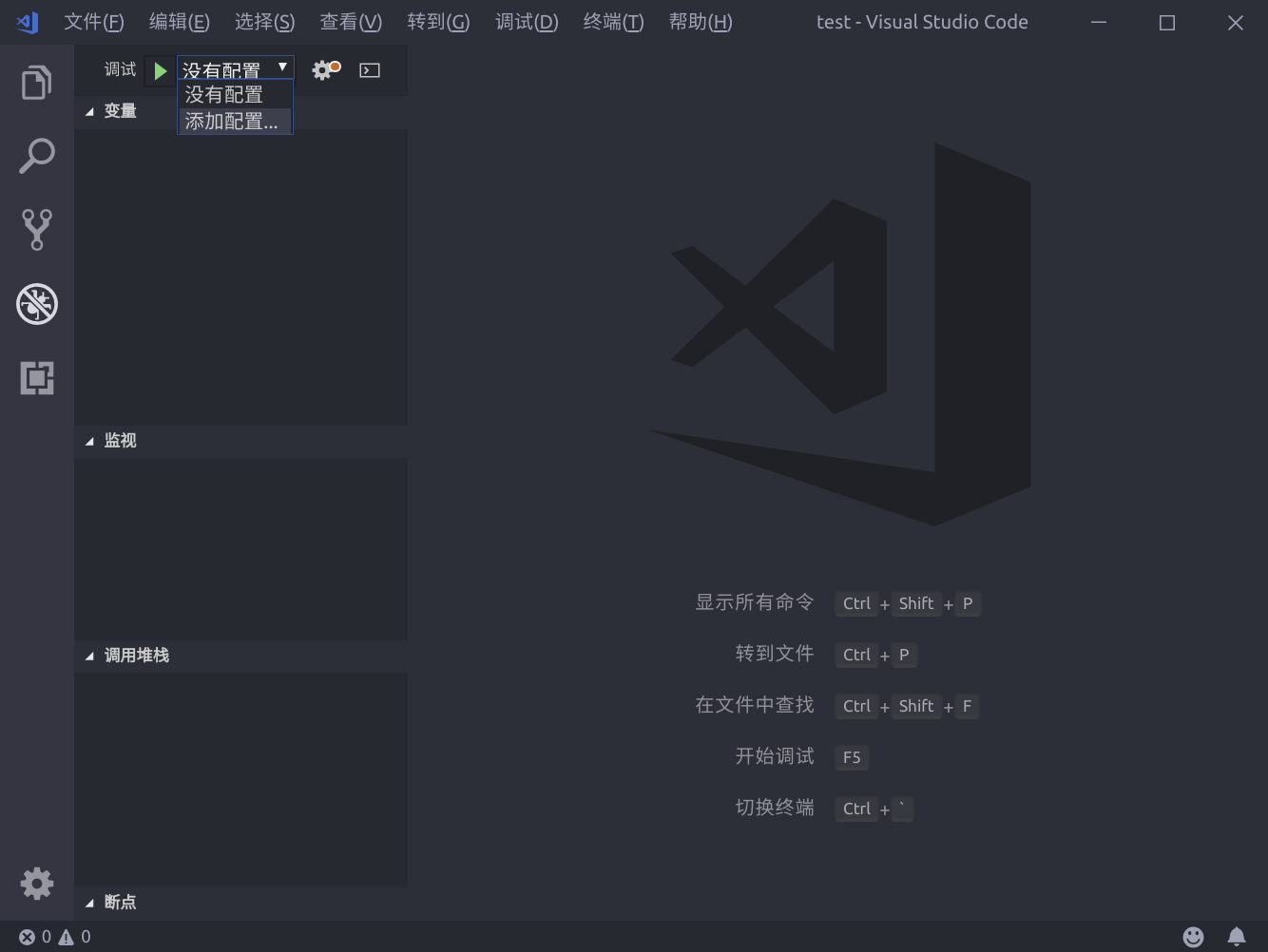
**其中最主要为"includePath"的引用和库的路径，根据引用内容进行配置。**



* **Launch**

**在debug界面中选择添加配置，然后选择才c++(gdb/lgdb)选项，生成launch.json 顾名思义此文件主要服务于调试时的加载控制**

**Linux下VS Code的C++工程配置**



**{**

**"version": "0.2.0",**

**"configurations": [**

**{**

**"name": "(gdb) Launch",**

**"type": "cppdbg",**

**"request": "launch",**

**"program": "${workspaceFolder}/helloworld",**

**"args": [],**

**"stopAtEntry": false,**

**"cwd": "${workspaceFolder}",**

**"environment": [],**

**"externalConsole": true,**

**"MIMode": "gdb",**

**"preLaunchTask": "build",**

**"setupCommands": [**

**{**

**"description": "Enable pretty-printing for gdb",**

**"text": "-enable-pretty-printing",**

**"ignoreFailures": true**

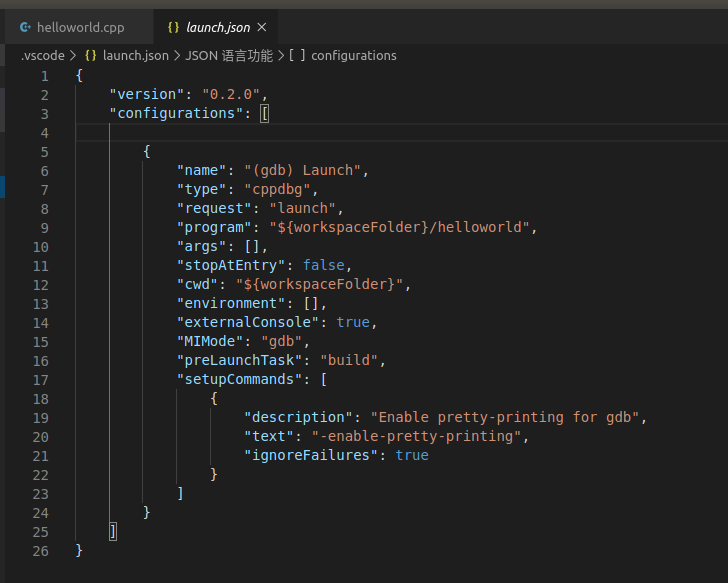
**}**

**]**

**}**

**]**

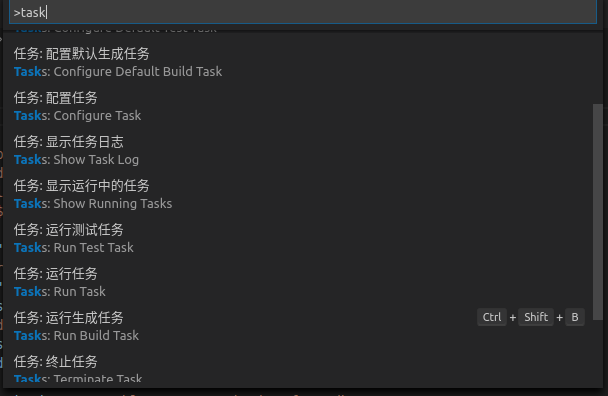
**}**



**需要注意的参数为"program"，此为需要调试的目标文件，应当设置为编译输出的文件位置；其次需要添加"preLaunchTask"，此项的名字应与下面所建的tasks.json中的任务名称一致。**

* **tasks.json**

**在命令窗口中输入task，选择task： configure task选项生成tasks.json文件**



**{**

**"version": "2.0.0",**

**"tasks": [**

**{**

**"label": "build",**

**"type": "shell",**

**"command": "g++",**

**"args":[**

**"-g","helloworld.cpp","-o","helloworld"**

**],**

**"group": {**

**"kind": "build",**

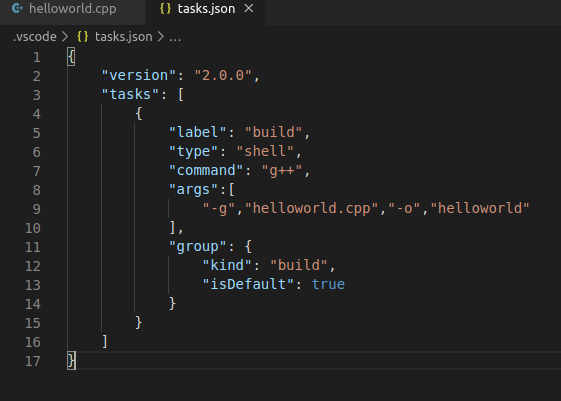
**"isDefault": true**

**}**

**}**

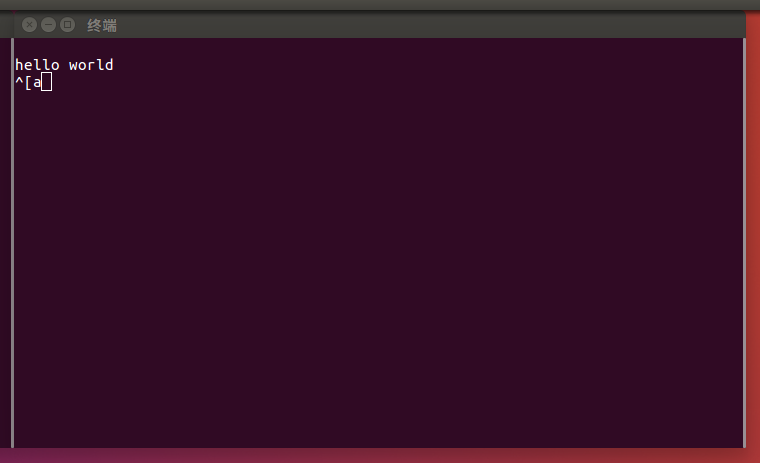
**]**

**}**

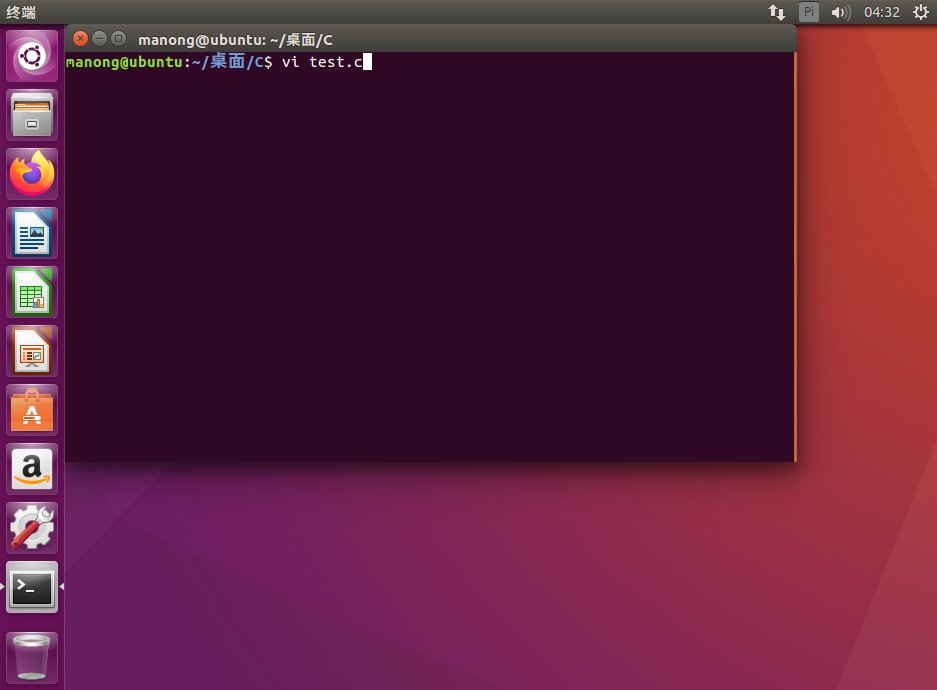


**注意launch.json中的"preLaunchTask"调用与“label”相同的task。**

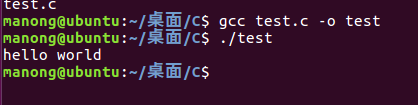
* **按下F5就可以开始调试了**



1. **使用Visual Studio Code完成C语言版的Hello,World，并查阅资料完成launch.json和tasks.json的创建和编写（使用快捷键Ctrl+Shift+P）以编译及运行该Hello,World。在这期间，你可能需要去查阅编译器gcc的基本用法。VSCode的官网也会有编译示例，可作为学习参考资料。**
2. **在指定文件夹下打开终端，通过vi test.c这点指令进入vim编辑器**



1. **用vim中的指令编写C语言的Hello World并保存**
2. **利用gcc指令（gcc test.c -o test）编译该文件**
3. **用指令/test运行**



**二、困难和结论**

1. **这期project要记住的命令不多，看着教程来就行了**
2. **不过gcc真是一个强大的编译器**