1. 建立项目

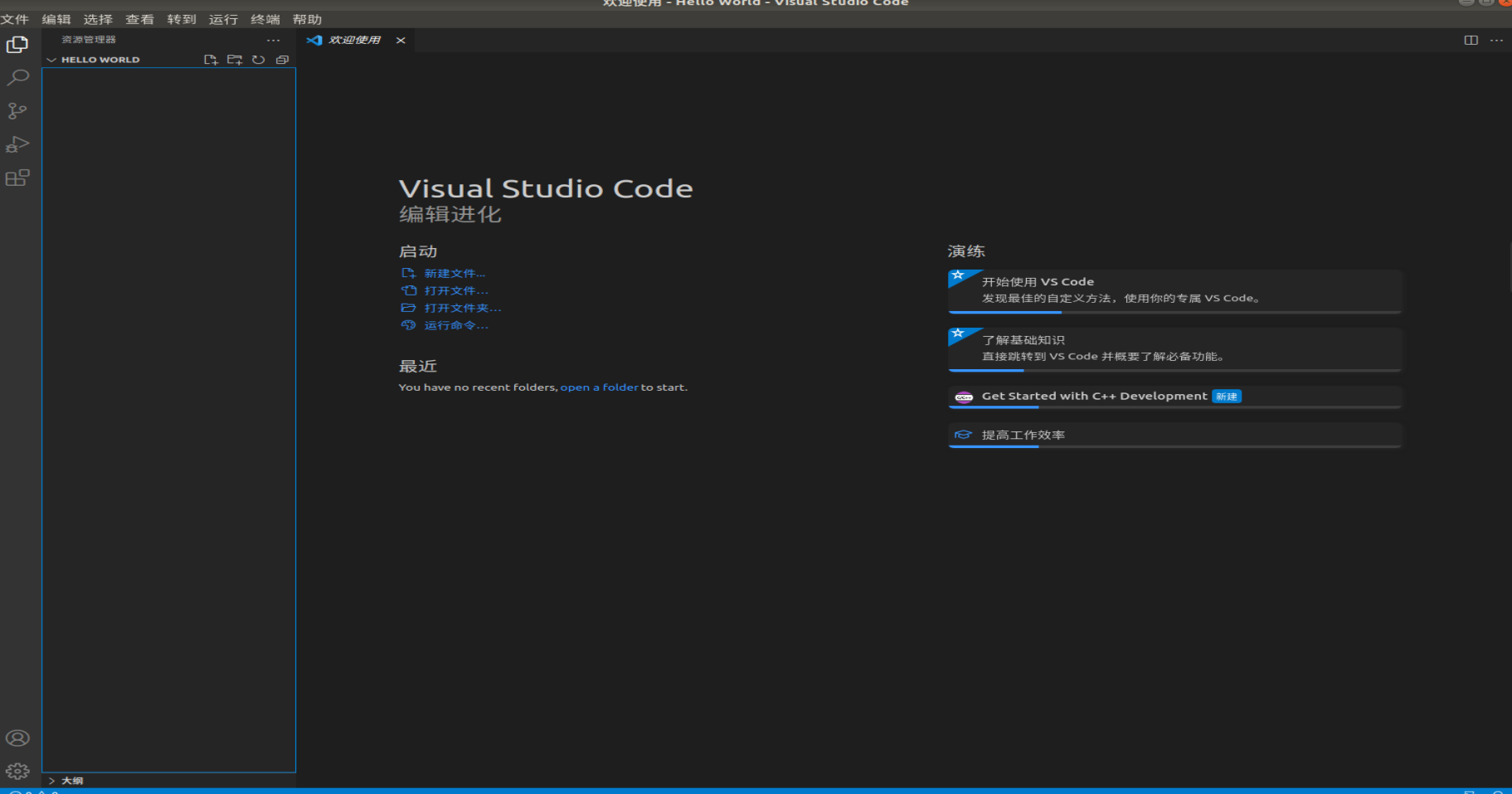
VSCode以文件夹来区分不同的项目，我们先建立一个文件夹：



打开VSCode，在左上角的文件下拉列表中选择打开文件夹：



选择我们刚刚建立的文件夹，就可以在左侧资源管理器中查看该文件夹了：

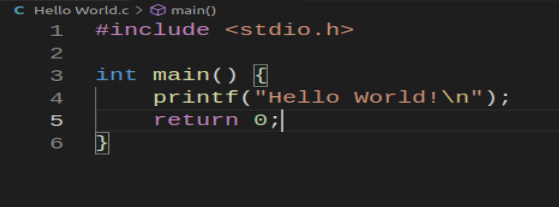


1. 编写程序

在资源管理器顶部分别有几个按键，分别是新建文件、新建文件夹、刷新资源管理器和折叠文件夹，我们新建一个文件，命名为Hello World.c：



在右侧的文本编辑器中，我们就可以编写代码了：

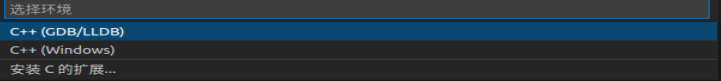


这样，我们一个简单的程序就编写完成了。

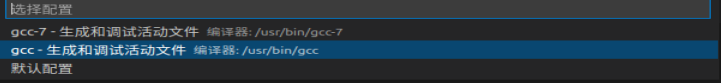
1. 运行程序

编写完程序后，我们按下F5按键，即可运行程序。需要注意的是，在第一次运行程序时要配置程序编译的环境，我们会在下一节详细讲解。

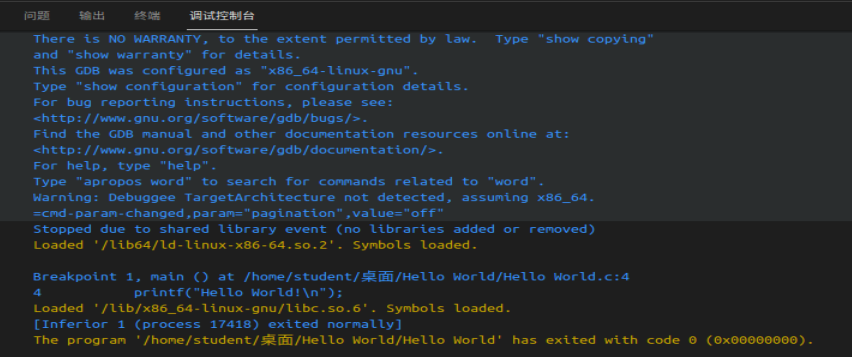
按下F5会提示选择调试环境，我们选择第一个：



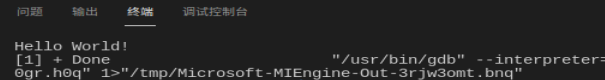
接着是选择编译环境，我们选择gcc即可：



选择完成后，程序开始运行，下方会弹出窗口，其中调试控制台会输出程序运行的基本信息：



终端窗口会打印程序的输出：



1. 配置编译环境

在程序运行完后，我们查看资源管理器，会发现文件夹中多了几个文件：



我们由下而上，Hello World.c即我们编写的程序文件，称为源程序;Hello World文件是gcc工具编译生成的可执行文件，我们编写的程序都要编译成可执行文件才能正常运行；.vscode是一个文件夹（Linux中，以.开头的文件为隐藏文件夹），里面有我们的程序编译配置文件，点开后里面有两个文件：

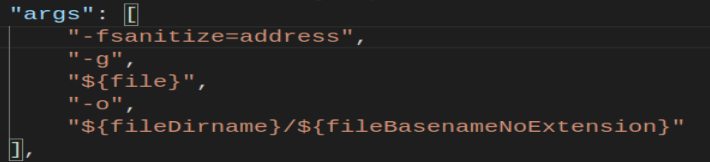


在上一节中，我们并没有编写这两个文件，而是由应用自动生成的，但有时候我们需要自己修改这两个文件，来满足我们的一些特殊要求。一般来说，task.json负责程序编译，launch.json负责程序执行。

下面是一个简单的task.json文件，一般我们只需要修改“args”部分，这里面填写的都是gcc编译的一些选项，这里面-g选项可以让我们的执行程序可调试，${file}指源程序，-o选项为输出选项，后面紧跟着输出文件地址。



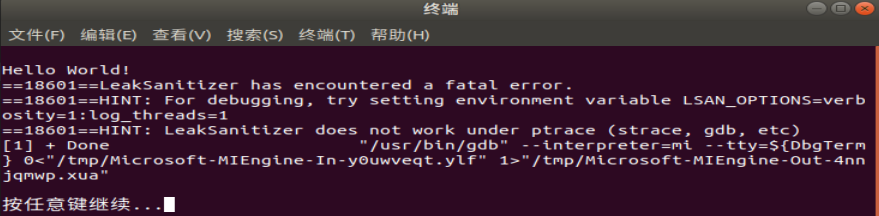
在今后的学习中，我们会在其中添加“-fsanitize=address”选项：



launch.json文件一般不需要修改，这里只介绍一个选项：



当该选项为false时，程序输出会在VSCode的终端中打印，即上一节展示的方式；如果为true，则会在Linux的终端中打印：



修改完成后，再次按下F5即可应用修改。