

# 渲染实验环境配置

## 一、 实验环境

1. 硬件环境：主流配置，最好配备独立显卡。
2. 软件环境
  - a) 操作系统： Windows 10
  - b) Windows 集成开发环境： Microsoft Visual Studio Community 2019
  - c) 构建系统： CMake
  - d) 版本控制系统： Git
  - e) 库管理工具： vcpkg
  - f) 开发库： GLFW 库， GLAD 库， GLM 库。

## 二、 理论背景

### 1. OpenGL 简介

OpenGL (Open Graphics Library) 是指定义了一个跨编程语言、跨平台的编程接口规格的专业的图形程序接口。它用于三维图像 (二维的亦可)，是一个功能强大，调用方便的底层图形库。OpenGL 是行业领域中最广泛接纳的 2D/3D 图形 API，其自诞生至今已催生了各种计算机平台及设备上的数千优秀应用程序。

### 2. GLFW 简介

GLFW (Graphics Library Framework)，它是一个小型 C 库，允许使用 OpenGL 上下文创建和管理窗口，从而也可以使用多个监视器和视频模式。它同时提供对键盘、鼠标和操纵杆输入的访问。

### 3. GLAD 简介

由于 OpenGL 驱动版本众多，大多数函数的位置都无法在编译时确定下来，需要在运行时进行查询，使得开发过程较为繁琐。GLAD (Multi-Language GL/GLES/EGL/GLX/WGL Loader-Generator based on the official specs) 是一个开源的库，可以提供一种更为简单的方法加载所有相关的 OpenGL 函数。

### 4. GLM 简介

GLM (OpenGL Mathematics) 是专门为 OpenGL 量身定做的数学库，它是一个只有头文件的库，可以快速实现矩阵变换等各种图形学中常用的几何计算。

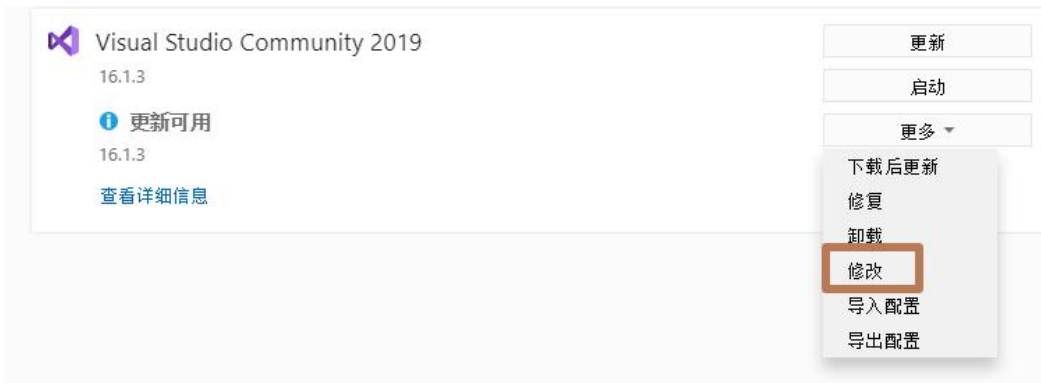
### 5. 三者的关系

OpenGL 定义标准 → 显卡厂商编写驱动 → GLAD 提供接口 → GLFW 建立窗口

## 三、 实验内容

1. 集成开发环境：Visual Studio Community 2019 的安装

- a) 需要安装“使用 C++ 的桌面开发”，“用于 Windows 的 C++ C++ CMake 工具”、“英语语言包”三项。
- b) 打开网址 <https://visualstudio.microsoft.com/zh-hans/downloads/>，选择 Visual Studio 2019 Community，点击下载 Visual Studio Installer。
- c) 安装完 Visual Studio Installer 后打开，选择安装 Visual Studio Community 2019，进入工作负载选择界面，在 Windows 一栏选择使用 C++ 的桌面开发。如果已经安装过 VS2019 的选择“修改”（不是修复）。



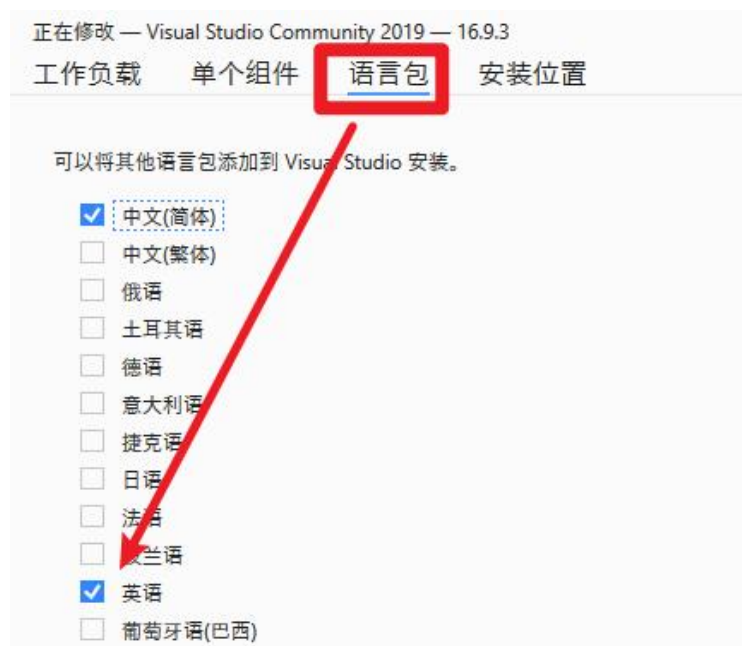
- d) 安装选择界面中，勾选“C++桌面开发”。



- e) 然后转到“单个组件”界面，在“编译器、生成工具和运行时”一栏选择用于“Windows 的 C++ CMake 工具”。



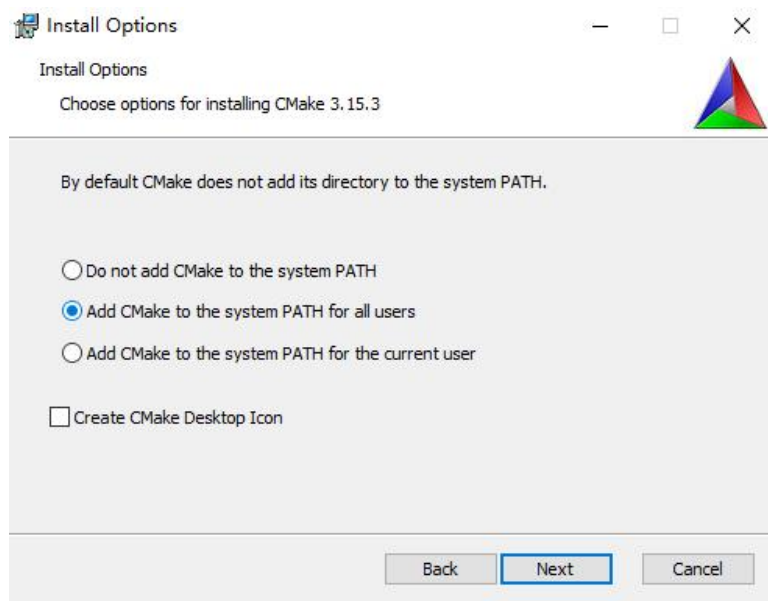
f) 然后转到“语言包”勾选“英语”。



g) 如果 C 盘空间不足，可以在“安装位置”栏目中修改自己的安装路径。然后点击程序界面右下角的安装按钮进行安装，VS 安装期间可以先进行后续的环境配置。

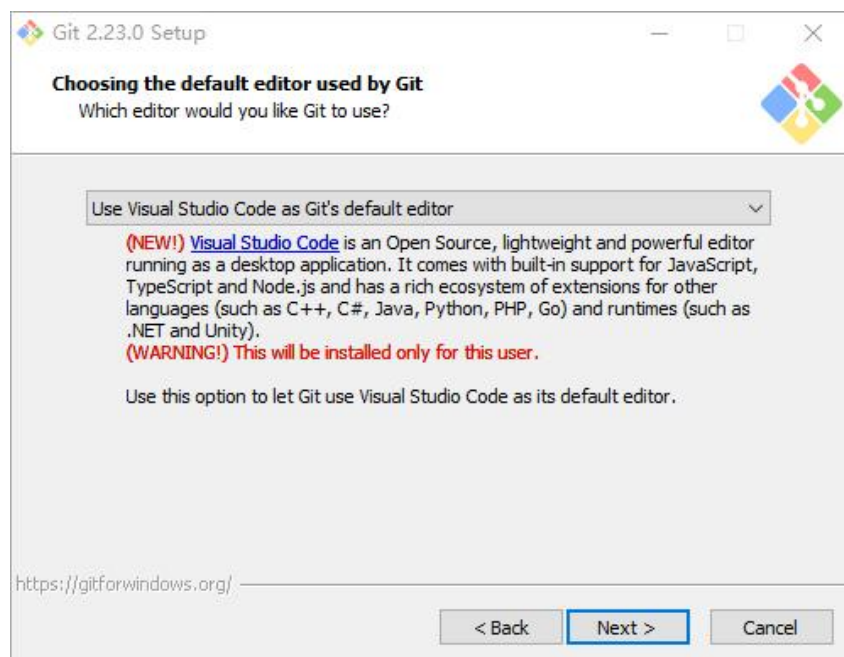
## 2. CMake 的安装

- 打开网址 <https://cmake.org/download/>，下载对应平台的 CMake 安装包（或使用提供的 VS2019 软件包中的安装包）。
- 打开安装包，按流程安装 CMake，其中有修改系统变量的选项选择第二项。



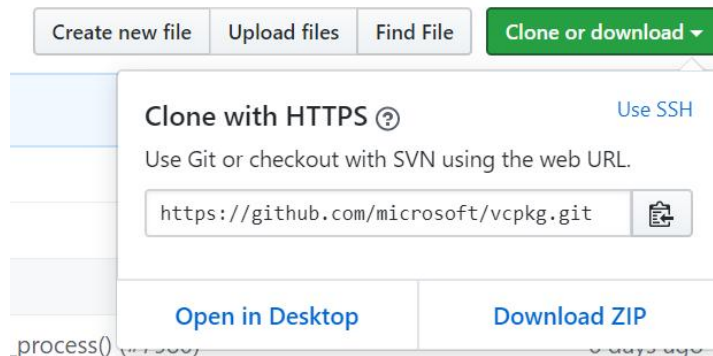
### 3. Git 的安装

- a) 打开网址 <https://git-scm.com/>，下载对应平台的 Git 安装包（或使用我们提供的 VS2019 软件包中安装包）。
- b) 打开安装包，按流程安装 Git，其中有选择编辑器的选项，有安装 VSCode 的建议选择 VSCode 作为默认的编辑工具。

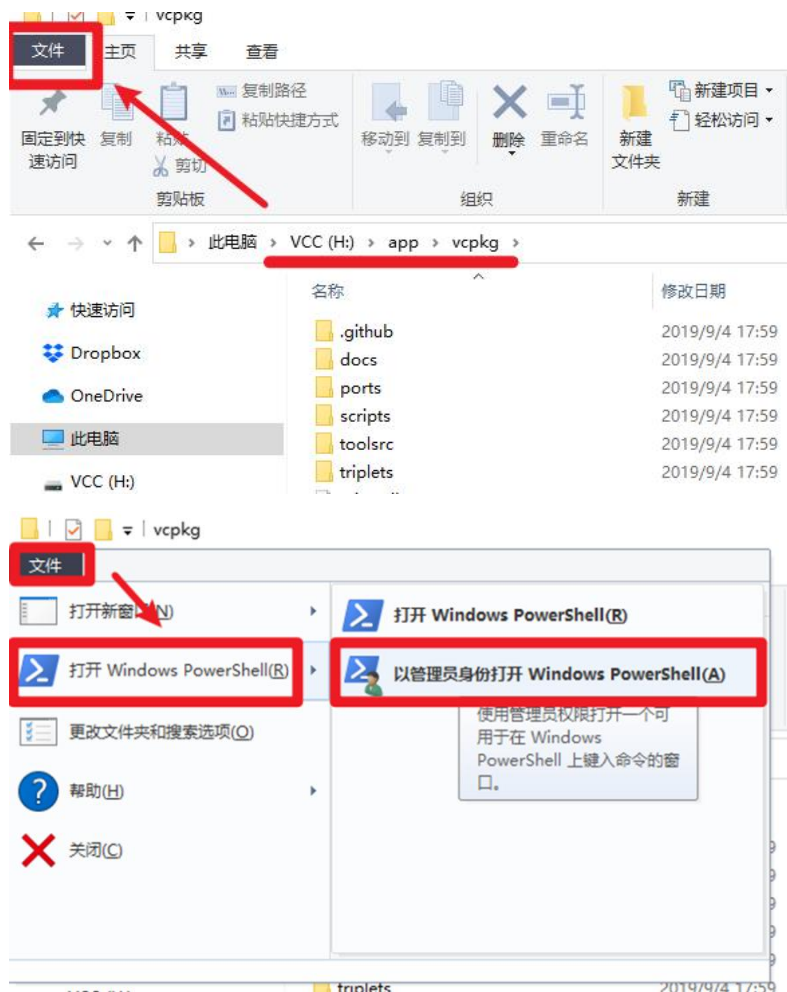


### 4. vcpkg 的安装【安装目录不要带中文，建议 C 盘或 D 盘根目录】

- a) 打开网址 <https://github.com/microsoft/vcpkg/>，使用 Git 克隆仓库到安装目录，或直接下载当前版本的源代码解压到安装目录。（建议使用提供的 VS2019 软件包中的 vcpkg 压缩包）



- b) 进入到**你自己的 vcpkg 文件**，点击左上角文件，使用管理员身份打开 Powershell。



- c) 运行目录下的 bootstrap 引导脚本，执行 `.\bootstrap-vcpkg.bat`，构建 vcpkg。

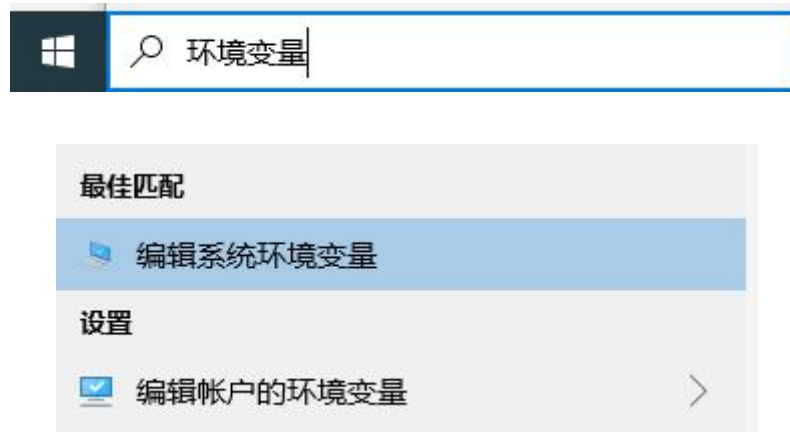
```
PS H:\app\vcpkg> .\bootstrap-vcpkg.bat
Building vcpkg.exe ...
```

- d) 构建完成后，而且 VS2019 也安装完毕后。执行 `.\vcpkg integrate install` 命令，将 vcpkg 聚合到 visual studio，这样用 vcpkg 安装的库就可以直接在 visual studio 中使用了

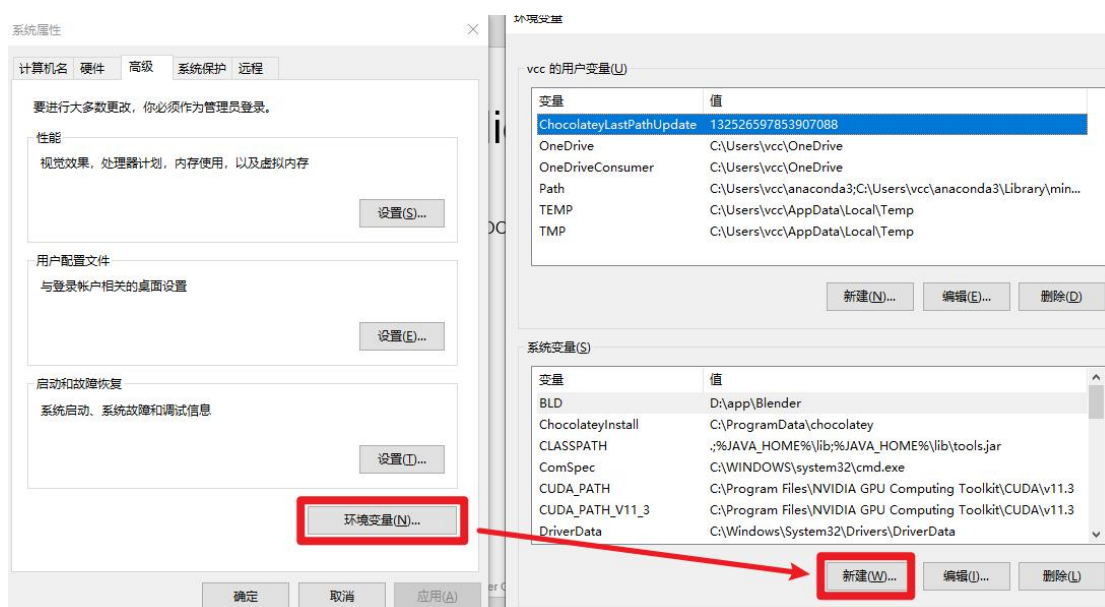
```
Building vcpkg.exe... done.
PS H:\app\vcpkg> .\vcpkg.exe integrate install
```

## 5. 系统环境变量设置

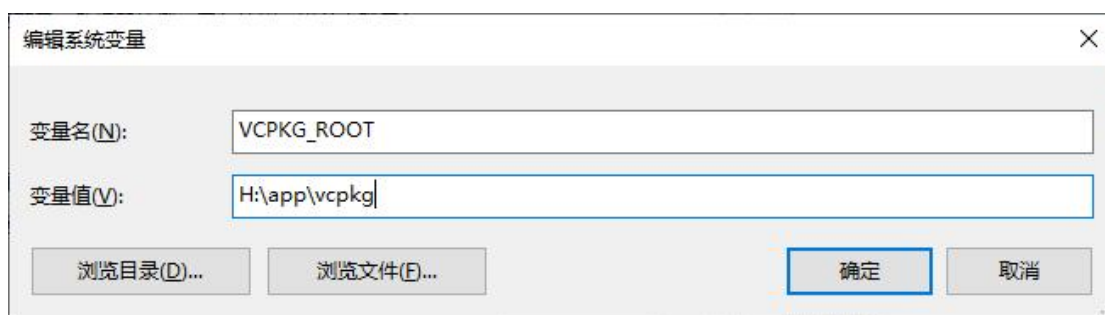
- a) 在 win10 搜索栏搜索环境变量，点击编辑系统环境变量。



- b) 在环境变量界面中选择新建。

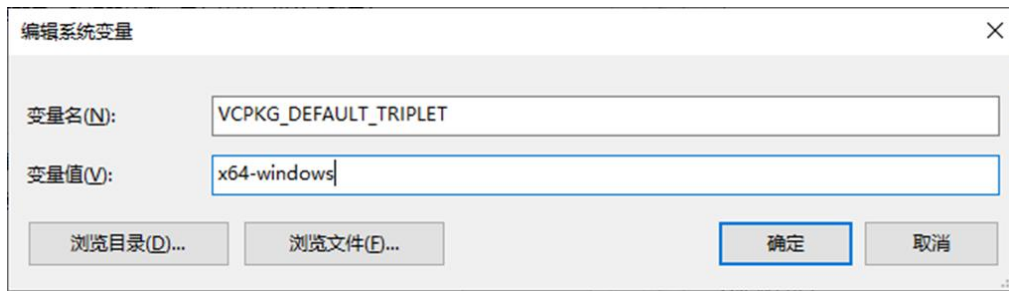


- c) 新建 VCPKG\_ROOT, 位置为**你自己的 vcpkg 文件夹路径**, 用于 CMake 找到 vcpkg。

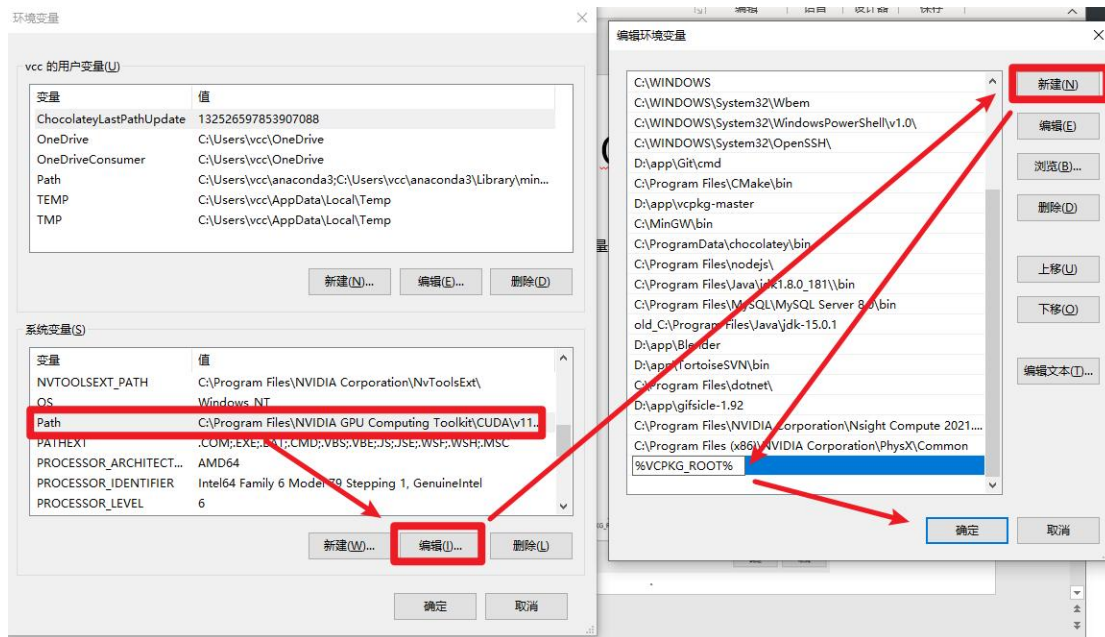


- d) 新建 VCPKG\_DEFAULT\_TRIPLET, 值为 x64-windows, 设置 vcpkg 默认库构架

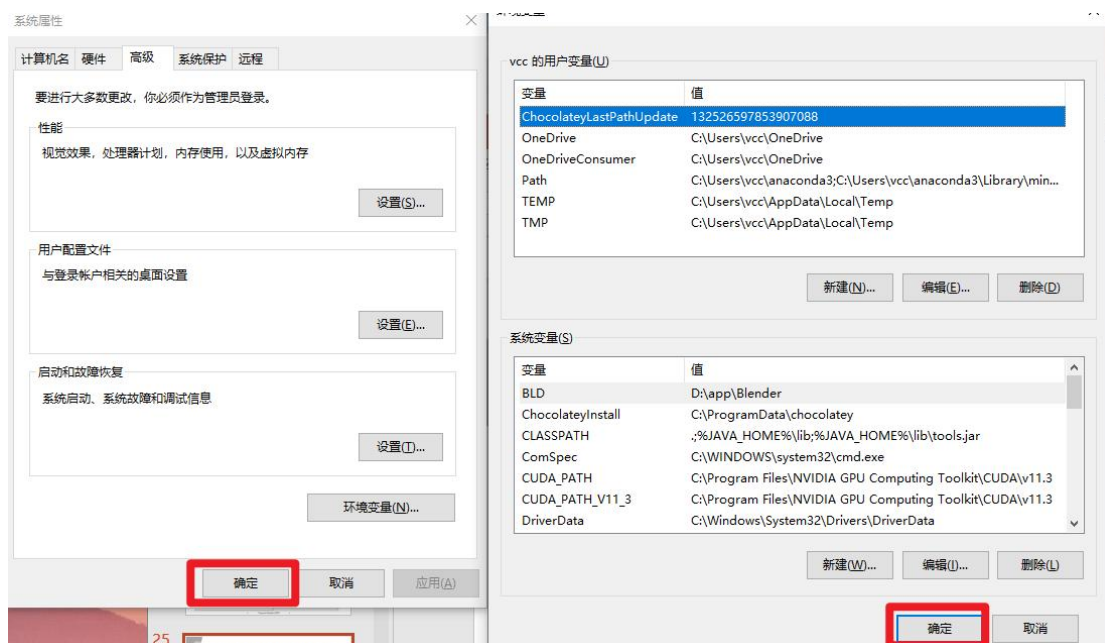




- e) 双击 Path，在最后一行新建路径：%VCPKG\_ROOT%，方便命令行在任意位置使用 vcpkg



- f) 添加变量之后记得点击“确定”按钮，才能添加成功

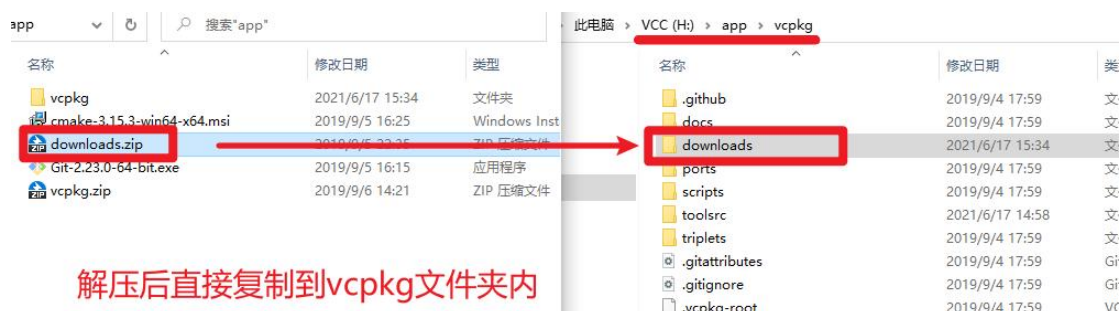


- g) 当前面的操作都弄好后，要验证路径是否添加成功，随便一个文件夹内开一个终端输入：vcpkg，执行后没用跳出错误就说明 vcpkg 环境配置成功。

```
D:\> vcpkg
Commands:
  vcpkg search [pat]           Search for package
  vcpkg install <pkg>...       Install a package
  vcpkg remove <pkg>...        Uninstall a package
  vcpkg remove --outdated      Uninstall all out-
```

## 6. OpenGL 库安装: GLFW, GLAD, GLM

- a) 如果觉得自己的网络下载 github 仓库很快可以直接进行下一步, 否则可以先将实验提供的 downloads 压缩包解压到 vcpkg 文件夹中。



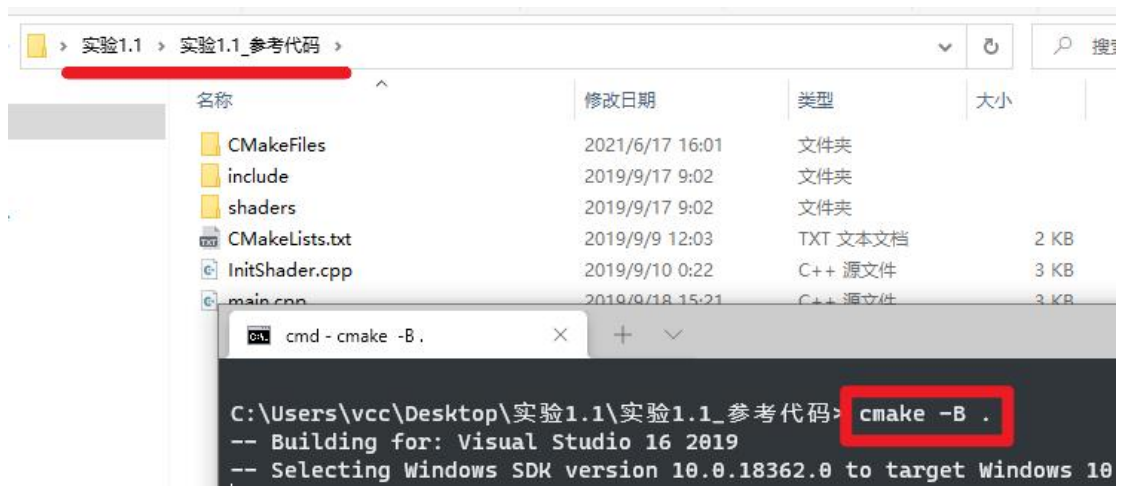
- b) 随便一个路径下打开命令行, 在命令行中输入命令 `vcpkg install glfw3 glad glm`
- c) 等待 glfw、glad 和 glm 完成编译即可

```
PS D:\> vcpkg install glad glfw3 glm
Computing installation plan...
The following packages are already installed:
  glad[core,loader]:x64-windows -> 0.1.34#1
  glfw3[core]:x64-windows -> 3.3.4
  glm[core]:x64-windows -> 0.9.9.8#1
Package glad:x64-windows is already installed
Package glfw3:x64-windows is already installed
Package glm:x64-windows is already installed
```

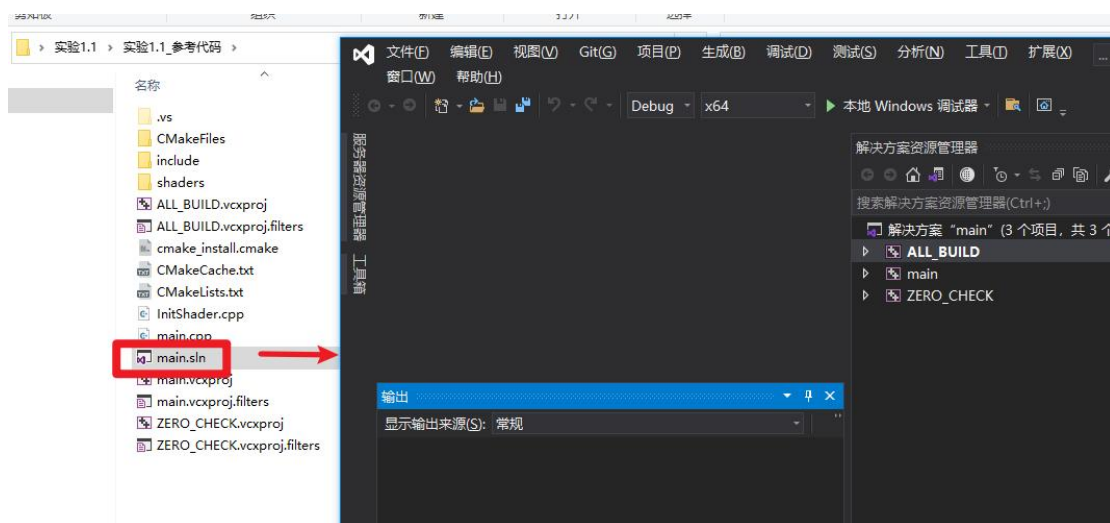
## 7. 构建并运行实验 1.1 【注意事项: 除了前面 vcpkg 的安装之外, 之后的命令执行并不需要管理员权限】

- a) 解压实验 1.1 代码并进入到实验 1.1\_参考代码文件夹, 在当前文件夹下打开命令行。然后执行 `cmake -B .`, 注意命令行最后有一个点。

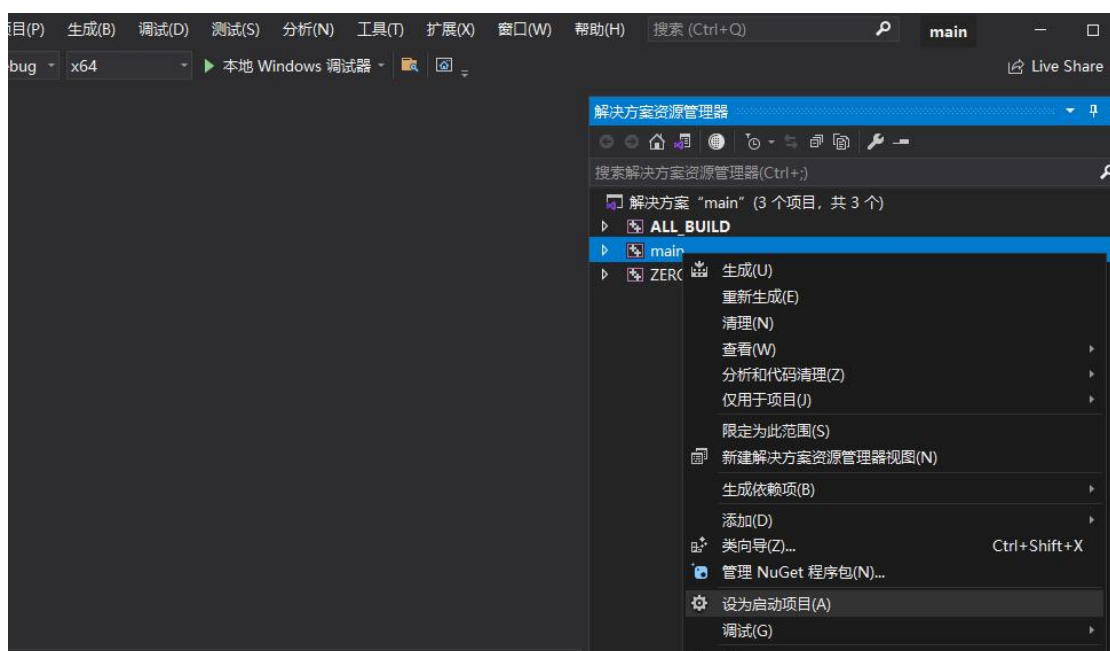




b) Cmake 命令执行后会在当前项目文件夹内生成 main.sln 文件，点击打开。



c) 在出现的 VS 界面中，可以看到“解决方案管理器”里面右键点击“main”项目，将其设置为启动项。



- d) 在完成实验后，点击菜单栏下方的“本地 Windows 调试器”编译运行程序，就可以顺利运行。

