

MiniOB 开发环境搭建（windows 版）

1. MiniOB 介绍

数据库系统是非常重要的系统基础组件，相对于其他课程知识，学习数据库系统难度大，要了解非常多的知识，有很多工作经验丰富的人经常使用 `mysql` 系统，但即使学习使用了很多年也很少有人精通数据库系统。实现难度更大；数据库系统涉及的知识面非常广，如果要了解或实现一个数据库系统必须对操作系统、内存、网络等知识非常精通才能构建一个比较性能好的数据库系统，这还没有涉及数据库本身领域的一些知识。现在除了单机版数据库还有很多分布式方面的多机房等相关知识，知识面很广；上手难度大，很难针对入门做数据库。

基于如此背景下，**MiniOB** 应运而生，**MiniOB** 设计的目标是让不熟悉数据库设计和实现的同学能够快速的了解与深入学习数据库内核，期望通过 **MiniOB** 相关训练之后，能够对各个数据库内核模块的功能与它们之间的关联有所了解，并能够在使用时，设计出高效的 **SQL**。面向的对象主要是在校学生，并且诸多模块做了简化，比如不考虑并发操作。

2. Docker 安装

由于 **MiniOB** 运行在 **linux** 环境中，**windows** 是没有办法直接运行的，因此我们这里为了部署 **MiniOB** 的运行、开发环境，使用了 **Docker**。**Docker** 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的 **Linux** 或 **Windows** 操作系统的机器上，也可以实现虚拟化。**Docker** 容器是一种轻量级、可移植、自包含的软件打包和部署技术。它可以将应用程序和依赖项打包在一个可移植的容器中，并提供一个一致的运行环境，无论在哪个计算机上运行都能够保证应用程序的可靠性和稳定性。**Docker** 的教程大家可以在网上自行搜索、学习。这里主要是介绍 **Docker** 的安装。

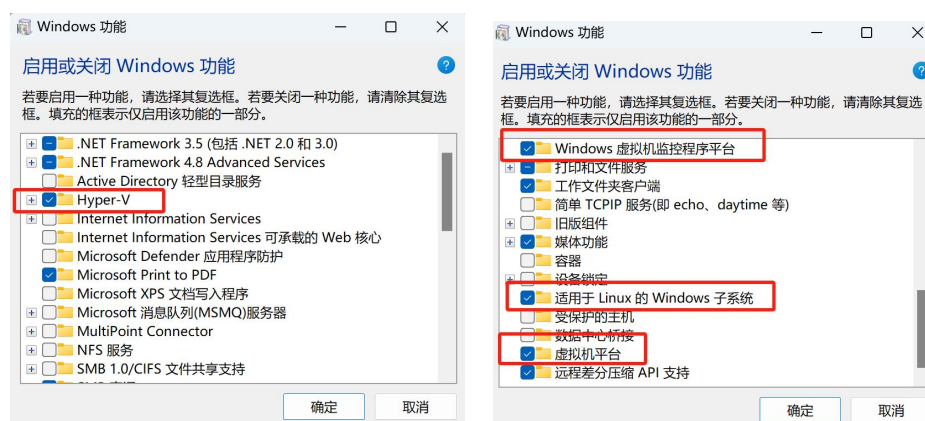
首先 **windows** 安装配置 **Docker**，要求 **windows** 系统为 **windows10** 或者 **Windows11** 版本，太低的版本不支持 **Docker**，或者安装会很麻烦。

这里介绍基于 **windows11** 安装 **Docker**。安装 **Docker** 需要开启电脑的虚拟化功能、**Windows** 的 **Hyper-V** 功能。

（1）开启电脑的虚拟化功能。需要在电脑的 **BIOS** 中进行设置，一般情况是开启状态。在 **BIOS** 设置界面中，找到“**Intel VT-x**”、“**AMD-V**”选项或者类似的 **intel virtual technology**

的选项，这是 BIOS 虚拟化的关键选项，设置为 **Enable** 就可以了。可根据自己电脑品牌在网上搜索如何开启虚拟化功能，根据教程设置。

(2) 打开 Windows 的 Hyper-V 功能。依次打开“控制面板”->“程序和功能”->“启用和关闭 windows 功能”，确保“Hyper-V”、“Windows 虚拟机监控平台”和“适用于 Linux 的 Windows 子系统”相关选项前面勾选了。



(3) WSL 配置：

使用管理员权限打开 Windows PowerShell ， 运行：

如果 windows 没有安装 WSL，可以先运行：`wsl --install`， 设置 WSL 默认版本为 2， `wsl --set-default-version 2`。 运行结果如下，

```
PS C:\Windows\system32> wsl --set-default-version 2
有关与 WSL 2 的主要区别的信息，请访问 https://aka.ms/wsl2
操作成功完成。
PS C:\Windows\system32>
```

通过 `wsl --update` 命令，WSL 将进行升级并检查相关缺少内容。

```
PS C:\Windows\system32> wsl --update
正在安装: 适用于 Linux 的 Windows 子系统
已安装 适用于 Linux 的 Windows 子系统。
PS C:\Windows\system32>
```

手动方式：（`wsl --install` 没有安装成功可以采用手动方式安装），【参考：<https://blog.csdn.net/yutingwu816/article/details/128190030?spm=1001.2014.3001.5506>】、【https://blog.csdn.net/maomao_op/article/details/135932907】

使用管理员权限打开 Windows PowerShell ， 运行：

- `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
- `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`
- 安装 `wsl_update_x64`。
- 设置 `wsl` 默认版本为 2。（命令参考前面命令）

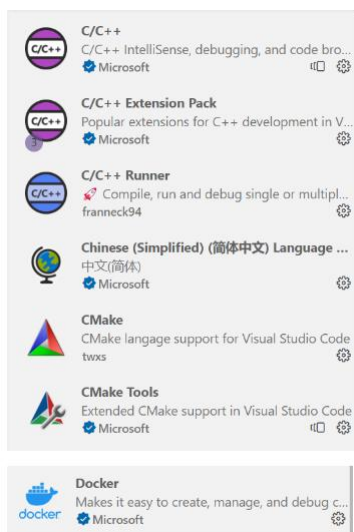
（4）Docker 安装。在 Docker 官网下载 Docker Desktop

双击 Docker Desktop Installer 开始安装，默认勾选内容，直接点 ok 开始安装。

安装后，能够成功运行 Docker，表示安装成功。

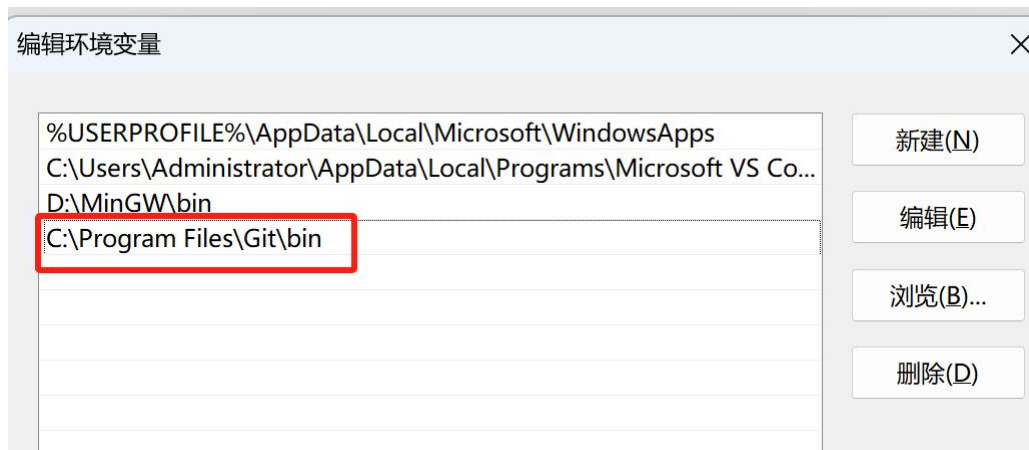
（5）VS code 安装

安装 MinGW，然后 Vs code 安装后，添加下来扩展组件。



（6）git 安装

下载 git，安装，全部默认选项即可。添加环境变量：



(7) MiniOB 的部署

使用 `vscode` 打开一个文件夹。然后 `ctrl+shift+~` (快速新建一个终端)。



确保是你想要的路径。

运行: `git clone https://github.com/oceanbase/miniob.git`

将 `miniob` 拉到本地。



然后将获取的文件与 **docker** 容器映射连接

```
docker run -d --name forttest --privileged -v $PWD/miniob:/root/miniob oceanbase/miniob
```

```
PS E:\forttest> docker run -d --name forttest --privileged -v $PWD/miniob:/root/miniob oceanbase/miniob
Unable to find image 'oceanbase/miniob:latest' locally
latest: Pulling from oceanbase/miniob
b237fe92c417: Pull complete
3e217228f4be: Pull complete
e12074e2167c: Pull complete
da8a63f342bf: Pull complete
7ef8b0caaaea: Pull complete
3d7ff68e7a43: Pull complete
7c47215316d3: Pull complete
86623ea74627: Pull complete
```

到此，基本把 **miniob** 部署好了，然后就是需要让 **miniob** 运行起来和基于 **miniob** 进行编码、编译和运行了。