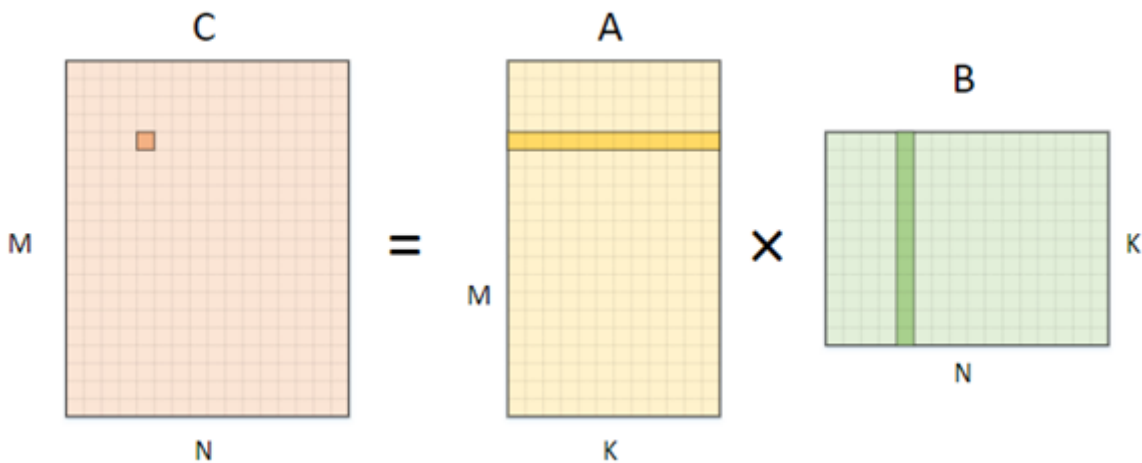


实验2：Linux环境下C语言编程

1. 实验目的

- (1) 熟悉Linux下可执行文件的生成过程
- (2) 掌握gcc的使用
- (3) 掌握Linux下C语言开发

2. 实验内容



(1) 安装开源openblas库，编译执行test_cblas_dgemm.c。熟悉cblas_dgemm的接口参数，test_cblas_dgemm.c用的列主序的方式，请改为行主序编译运行。

(2) time_dgemm.c中新增naive_dgemm实现，并在代码中新增与cblas_dgemm计算结果做对比，验证naive_dgemm实现正确。

3. Visual Studio Code (VS Code) 安装配置

Visual Studio Code (以下简称 VS Code) 是一个由微软开发，同时支持 Windows、Linux 和 macOS 等操作系统且开放源代码的代码编辑器，用 TypeScript 编写，支持多种语言的开发。

VS Code是一款应用非常广的IDE，强烈建议大家使用VS Code配合远端的Linux环境。

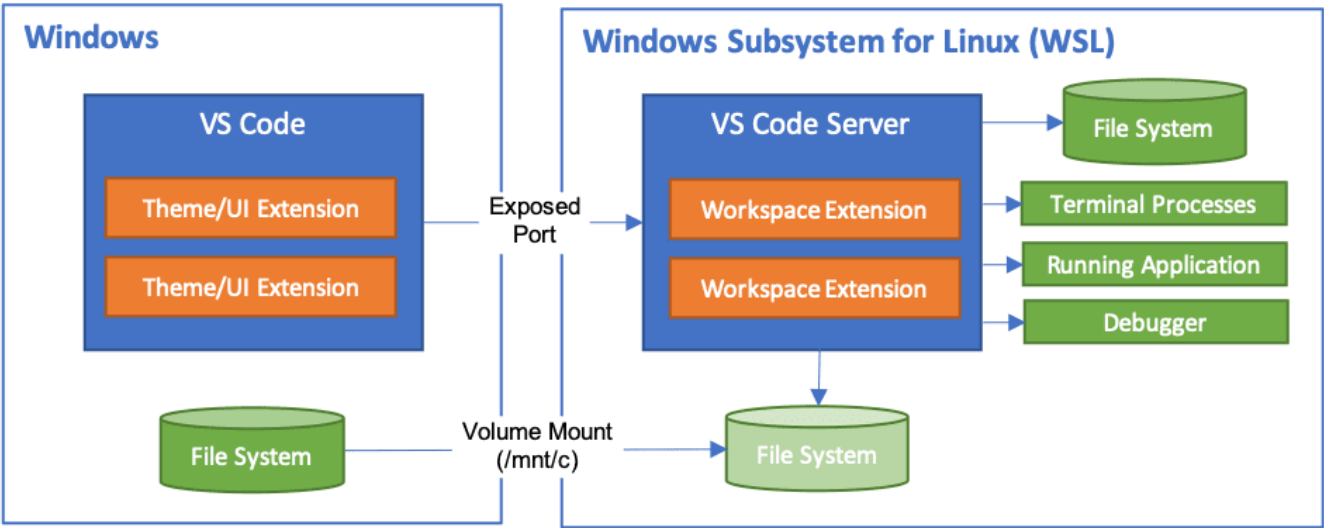
3.1 安装

从 <https://code.visualstudio.com/download> 下载对应的安装包，下载完成后双击运行。默认的安装路径是 `C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code`。

3.2 WSL 远程开发

VS Code的远程开发功能非常好用，可以非常方便的在windows端启动VS Code，连接到远端Linux服务器，提供在Linux本地开发的体验。

WSL远程开发，即文件、编译环境、运行环境都是在WSL环境中，在windows中启动VS Code，通过ssh连接到WSL环境，编辑远端WSL的文件。文件更新后再在终端执行对应的编译、运行命令。



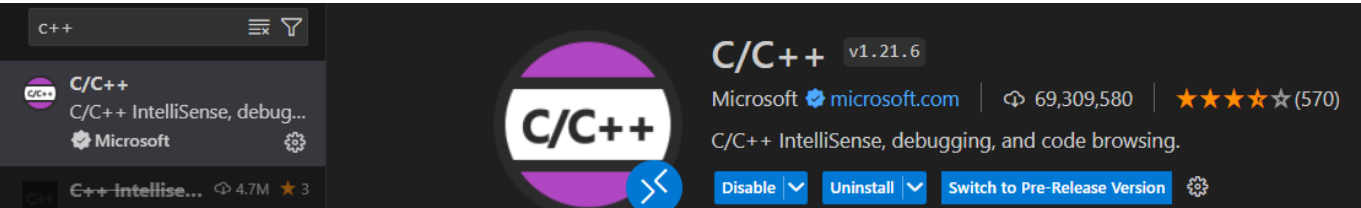
图片来源

因为WSL在实验1中已经安装好，只需要参照官方教程做以下配置：

1. Install the WSL extension，左边工具栏有个Extensions图表（快捷键是Ctrl+Shift+X），然后输入WSL即可搜索到对应的插件，点击安装。
2. 安装插件后点击左晓娇绿色的图表，从弹出来的选项中选择 Connect to WSL，稍等片刻即可连接到WSL环境。
3. 默认打开的是HOME目录，在菜单栏选择File->Open Folder（快捷键Ctrl+O），再选择自己需要打开的目录，若没有则可以在终端下用命令行新建目录。

更详细的说明请参考：[Remote development in WSL](#)

为了更好地支持语言高亮、自动补全等功能，还需安装如下的C/C++插件，根据名字搜索点击安装即可。



4. 实验报告要求

（1）PDF格式，要有课程名、学期、姓名、学号四个基本信息，其他无要求，注意排版

(2) 列出实验环境：OS版本，gcc版本，CPU（型号、频率、物理核数），内存大小。

```
1. uname -a查看操作系统内核版本，lsb_release -a查看发行版本
2. lscpu 查看CPU信息
3. free查看内存
```

(3) test_cblas_dgemm.c 修改为行主序后结果有什么不同？

(4) time_dgemm.c 分别测试M=N=K：256，1024，4096，8192时，以下面表格的形式记录两者的duration和 gflops的值。从数据中可以发现什么规律，可以尝试自己分析下。数据有波动是正常的，可以运行多次取平均。

	256	1024	4096	8192
cblas_dgemm duration				
naive_dgemm duration				
cblas_dgemm gflops				
naive_dgemm gflops				

(5) 碰到的问题及解决办法

(6) 将代码和报告提交到远端仓库