Linux使用指南

让一部分学生先把Linux用起来[狗头]

# 缘起

现在计算机软件的同学课程安排大一学c、c++，后续数据结构依赖c、c++，现在使用的vs6开发环境已经非常老旧，赶快让大家用上Linux编译工具链。

计划的教程内容有：

1. 配环境，linux基础操作，cmake
2. 完成c、c++作业，完成数据结构课程作业
3. 简单项目
4. 提高项目，及未来可以做那些方向

# 需要那些贡献

1. 对于有经验的同学

讨论初学者需要掌握那些知识，怎么引入，怎么安排，贡献好词好句

1. 对于初学者

指出有歧义不好理解的句子，没有表达清楚的概念

安装流程和执行流程，反馈可以丝滑运行，或者卡在哪一步、遇到那些特殊情况

学习c、c++、数据结构等课程有什么问题也可以提出来一起讨论

1. 提供c、c++、数据结构 等课程个人总结和使用Linux开发环境完成课程的经验

提供开源的项目，介绍项目思路、技术栈、源代码、文档、演示等

# 序言

You can code. They cannot. That is pretty damn cool.

-《Advice from an Old Programmer》<https://learnpythonthehardway.org/book/advice.html>

为什么计算机学生要使用linux？

1. linux是开源免费共享的，大量的服务器的操作系统是Linux、嵌入式设备也可以定制linux操作系统在有限的计算性能和能量消耗下完成更多事情。
2. Linux开放源代码，可以深入了解其构造，相应配套软件也都优先支持Linux。

# 安装linux环境

## 路线图

学习linux使用首先需要一个linux，最简单的方式是使用wsl2安装linux，也可以使用虚拟机安装linux，安装linux和Windows双系统，使用Ubuntu LiveCD不安装操作系统下体验linux，登录远程linux服务器。以下是不同方法的对比：

///// 以下内容需要重构

以下是不同方法的具体讨论：

WSL2：运行在Hyper-V虚拟化层上的Linux，功能完备

物理机双系统

物理机单系统

为什么使用wsl2？

（或许可以加一个为什么不用wsl2）

## 使用哪个linux发行版？

建议萌新选择 Ubuntu 22.04 LTS

<https://zh.wikipedia.org/zh-cn/Ubuntu>

Ubuntu是著名的Linux发行版之一，也是目前最多用户的Linux版本。

Ubuntu每6个月发布一个新版，每个版本都有代号和版本号。版本号源自发布日期，例如第一个版本，*4.10*，代表是在2004年10月发行的。下表列出了部分以前和计划中的发布：



## Linux 发行版安利：

### 0x00 wsl2自带的linux发行版 可以快速无痛体验

wsl -l -o



### 0x01 试试Debian 极其稳定快速、易于维护的 Linux：

<https://www.iplaysoft.com/debian.html>

说到 Linux 操作系统，很多人都听过 Ubuntu、CentOS、Fedora、RedHat，国产的优麒麟和 Deepin 深度等系统。除了这些，还有一个极具代表性的操作系统，它就是 Debian！

Debian (英语发音：ˈdɛbiən) 作为最老牌的 Linux 发行版之一，它在创建之初便紧密遵循 GNU 规范并坚守 Unix 和开源自由软件精神。最大优点是极其稳定快速，同时拥有强大的 APT 包管理机制和丰富的软件源，可方便安装和平滑升级软件和系统，省去很多编译安装的麻烦……

### 0x02 试试Fedora 更新快 / 软件包版本新 / 开放的 Linux 发行版系统：

<https://www.iplaysoft.com/fedora.html>

Fedora 是一款非常流行且广泛的 Linux 发行版之一！它基于 Fedora Project（由红帽资助，旨在取代 Red Had Linux 在个人领域的应用），并由全球范围的社区志愿者和开发人员构建和维护。

相比 Ubuntu、CentOS、Debian、RedHat 等，Fedora 最大的特色是追求并吸纳最新技术，一些前沿技术和最新版本软件往往能在 Fedora 上第一时间享用，软件版本包都很新，加上开发社区人气兴旺，Fedora 相比其他发行版对编程工作者或热爱尝鲜的技术爱好者有着更大的魅力……

### 0x03 试试deepin 国产linux 操作系统：

<https://www.deepin.org/zh/deepin-v23-adapted-wsl/>

在已经发布的deepin V23 beta版本中，深度操作系统正式适配了WSL！

### 0x04 发行版对比

Debian：

* Ubuntu：Canonical 公司支持，基于Debian，每年4月和10月会发布两个支持期9个月的版本，每两年的10月发布一个支持期为5年的Long Term Support版本
* Linux Mint：
* Red Hat / CentOS Stream / Fedora：红帽系列，从左到右新功能加得更快而稳定性下降
* Arch：可DIY程度高，软件包更新快（一般都是最新的内核+软件），社区活跃度高，但需要有较强动手解决问题的能力和时间。Manjaro是Arch的一个发行版
* Deepin / UOS：国内统信软件和深度科技二次开发的Linux系统，界面更接近Windows的体验

## 安装wsl2

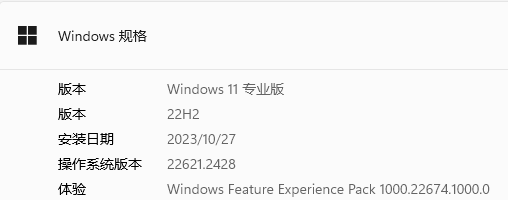
参考自：

<https://github.com/ShiqiYu/CPP/blob/main/week01/Lab01.pdf>

<https://bbs.scumaker.org/t/topic/167/3>

### 0X 00 Windows 环境确认

**安装过程在以下环境测试过：**

****

注意：

使用wsl2快速方式安装需要 Windows 10 版本 2004 及更高版本（内部版本 19041 及更高版本， 2020年9月4号正式发布）或 Windows 11（2021年10月5号正式发布）。之前安装方式参考：xxxx。（2023年底了，该不会有人用这么老的Windows吧，尤其是现在微软强制推更新）

Wslg需要Windows 11 版本22000或更高，或者加入Windows 10 预览体验计划 并升级到 Windows 10 Insider Preview build 21362+。

### 0X 01 安装 wsl 2

<https://learn.microsoft.com/zh-cn/windows/wsl/install>

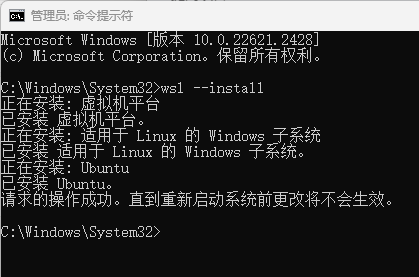
#### 使用Windows 命令提示符安装 wsl 2

1. 以管理员身份运行Windows 命令提示符



1. 输入

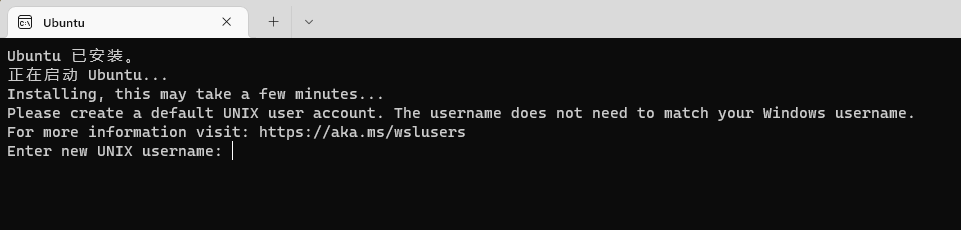
wsl --install



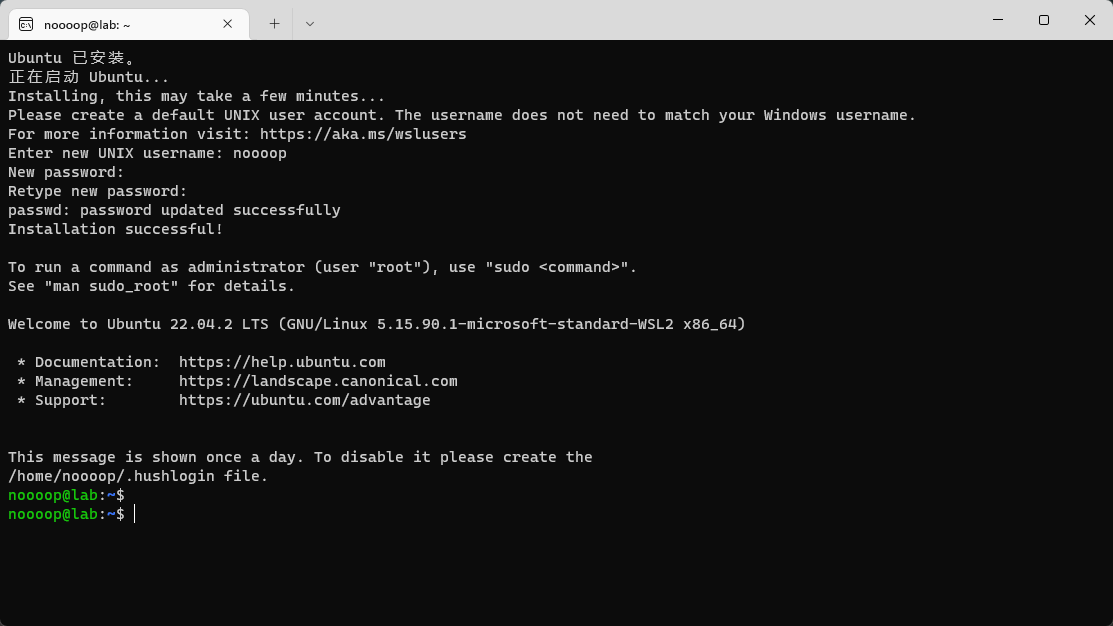
默认下装的Ubuntu-22.04，如果未能安装成功可以指定wsl --install -d Ubuntu-22.04，这里d是指distribution 发行版

1. 重启电脑继续安装
2. 输入linux 用户名和密码

用户名和密码不一定和Windows用户名密码相同，它两属于不同的体系



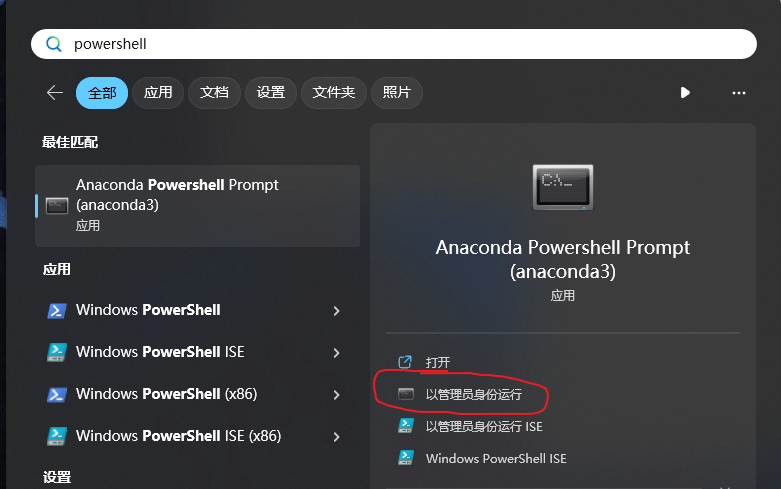
1. 是的，已经安装好了



// 以下是次要部分

#### 使用PowerShell 安装 wsl2

1. 以管理员身份运行PowerShell



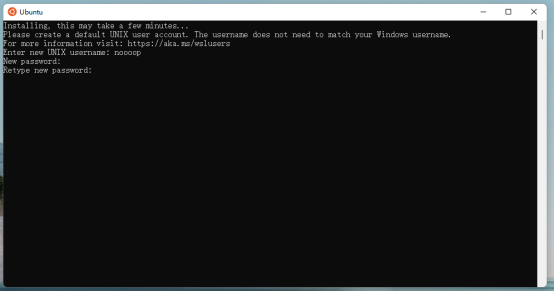
2. 输入

wsl --install

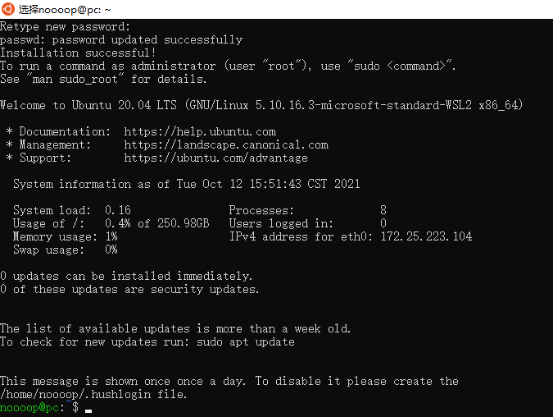


1. 重启电脑继续安装
2. 输入linux 用户名和密码

用户名和密码不一定和Windows用户名密码相同，它两属于不同的体系



1. 是的，已经安装好了



#### wsl2 安装方式历史

再次注意：

使用wsl2快速方式安装需要 Windows 10 版本 2004 及更高版本（内部版本 19041 及更高版本， 2020年9月4号正式发布）或 Windows 11（2021年10月5号正式发布）。之前安装方式参考：xxxx。（2023年底了，该不会有人用这么老的Windows吧，尤其是现在微软强制推更新）

Wslg需要Windows 11 版本22000或更高，或者加入Windows 10 预览体验计划 并升级到 Windows 10 Insider Preview build 21362+。

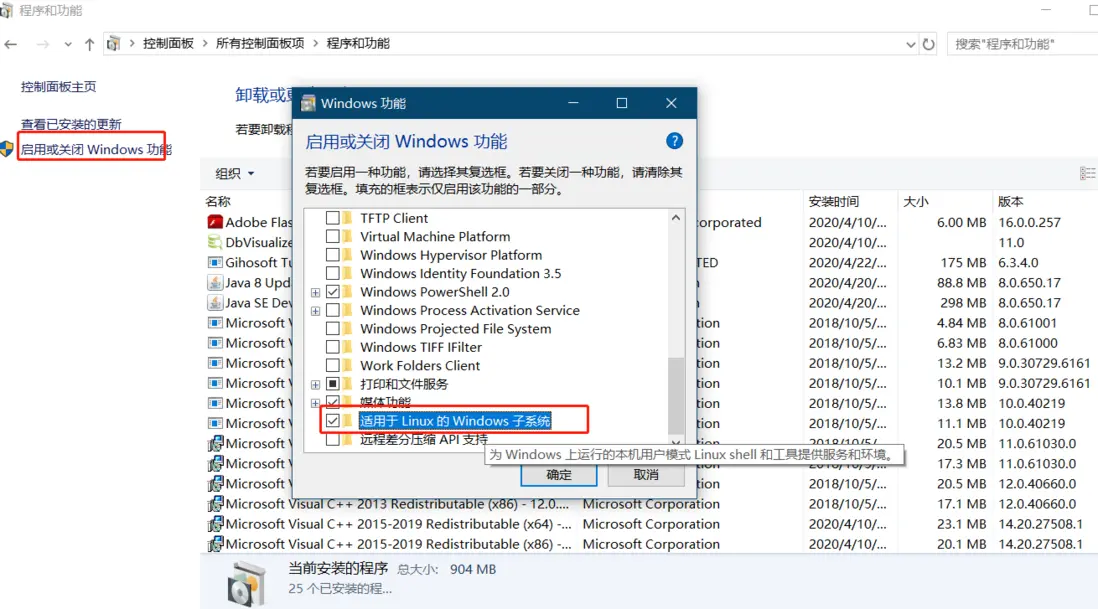
现在已经是2023年wsl2安装已经大幅简化，以下安装方式基本上是2020的安装方式，已经是历史了

1. 不需要，打开开发者模式，在应用或关闭Windows功能，勾选 适用于Linux的Windows子系统。

<https://segmentfault.com/a/1190000022987318>

<https://cloud.tencent.com/developer/article/1726747>

<https://www.cnblogs.com/yugege/p/14073908.html>



1. 不需要，更新wsl核心到wsl2

<https://github.com/ShiqiYu/CPP/blob/main/week01/Lab01.pdf>

<https://blog.51cto.com/u_15309736/5388067>

Step two: Update WSL kernel

Step three: Set WSL version as 2

“新版本快速安装方式已经默认 wsl版本为2，操作系统的wsl核心也更新为最新”这个步骤不需要了

### 0X 03 wsl2 相关配置

#### 1. 导入导出

查看所有WSL wsl -l --all -v

导出WSL wsl --export Ubuntu-20.04 d:\ubuntu20.04.tar

注销WSL wsl --unregister Ubuntu-20.04

导入WSL wsl --import Ubuntu-20.04 d:\ubuntu d:\ubuntu20.04.tar --version 2 （一定要写–version 2！）

Ubuntu修改默认登陆用户(我觉得用root挺好的[狗头]) ubuntu2004.exe config --default-user Username

#### 2. 原地导入

wsl --import-in-place Ubuntu D:\wsl\ubuntu\ext4.vhdx

原地导入权限问题 Error code: Wsl/Service/CreateInstance/MountVhd/E\_ACCESSDENIED

给Users完全控制权限

<https://blog.csdn.net/weixin_37178320/article/details/128720616>

#### 3. 不支持wslg

不能使用linux gui程序，换windows11 [狗头]

#### 4. 控制wsl内存使用量

C:\Users<UserName>.wslconfig

[wsl2]

memory=30GB

更多高级控制功能 Advanced settings configuration in WSL | Microsoft Learn 1

## 虚拟机安装linux

## 安装双系统

## 登录远程服务器

# Linux基本操作

<https://item.jd.com/13018025.html>

<https://www.runoob.com/linux/linux-tutorial.html>

## 路线图

希望学习最少得命令和概念把linux命令行玩转起来，更多的功能以后慢慢学。

1. 文件系统及管理

文件树 大小写

home ~

pwd

ls cd

mkdir rmdir cp mv rm

文件权限 sudo chown chmod

1. wsl2 文件系统
2. 帮助文档

man

1. 查看文件

cat more less tail head echo wc

管道命令 |

重定向 > 是覆盖模式，>> 是追加模式

1. 编辑文件

vi vim 最基本的使用

1. 包管理

apt

国内源加速

list show update search install upgradable remove

1. 进程相关

top htop

tmux

ctrl-c ctrl-d

kill kill-9

// 学到这里已经能应付初学的大部分事情，后面的可以以后慢慢学

1. 链接

ln

1. 网络

wget

1. 通配符
2. 搜索文件

whatis whereis which locate find

1. 效率提高

别名 自动补全

1. 压缩解压
2. 正则表达式

## 0x01 文件系统及管理

///

这里需要一篇针对萌新的，linux文件系统的设计思路的文章

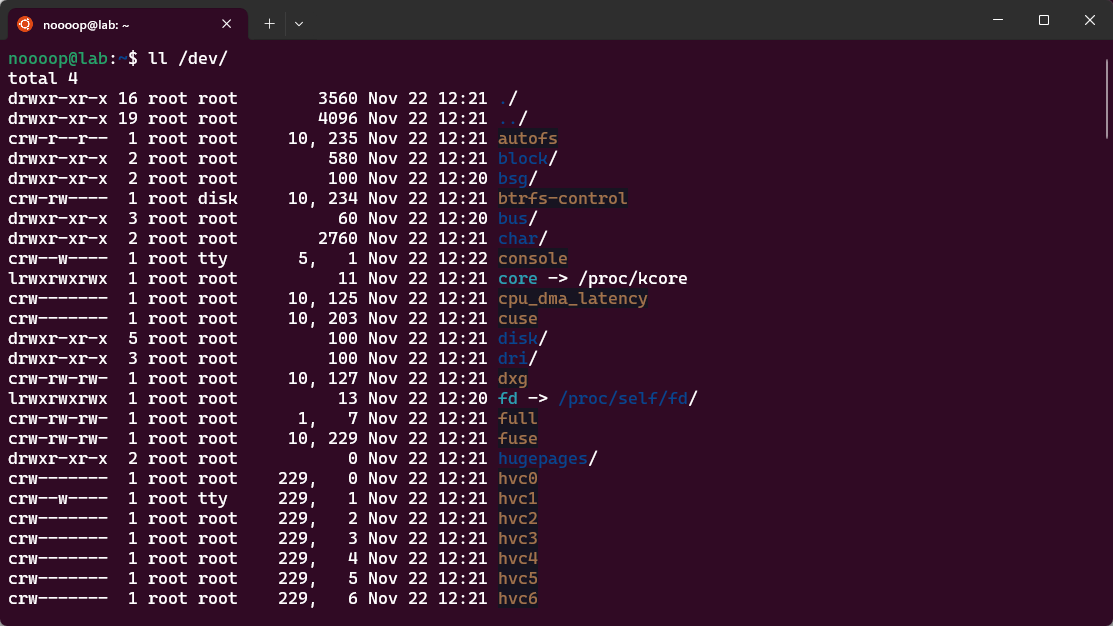
包括一切皆文件，文件操作

这不免牵扯操作系统的内容

最少涉及那些内容，又能讲的清楚

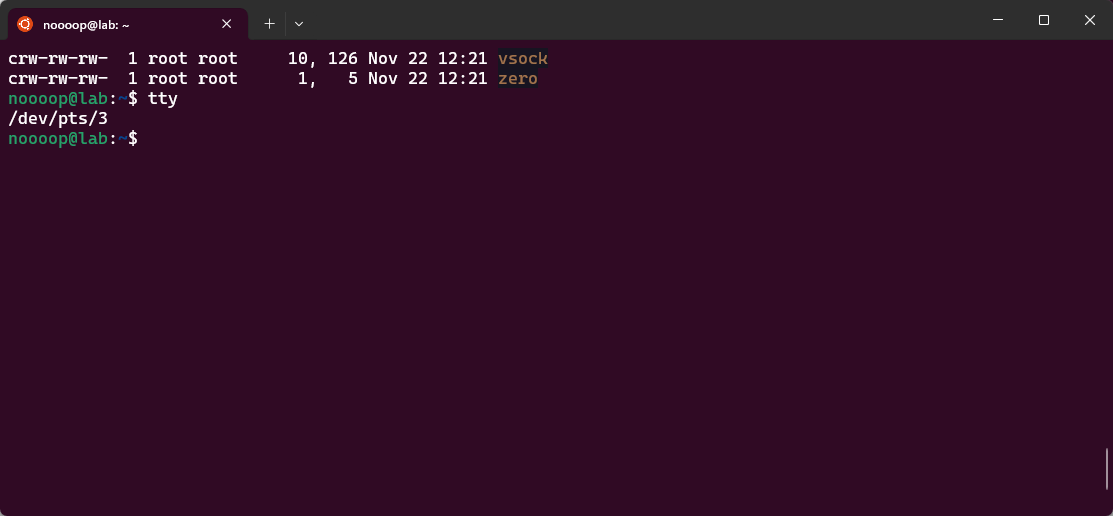
Linux 最经典的一句话是：「一切皆文件」，不仅普通的文件和目录，就连块设备、管道、socket 等，也都是统一交给文件系统管理的。

Wait a Minute! Show me！

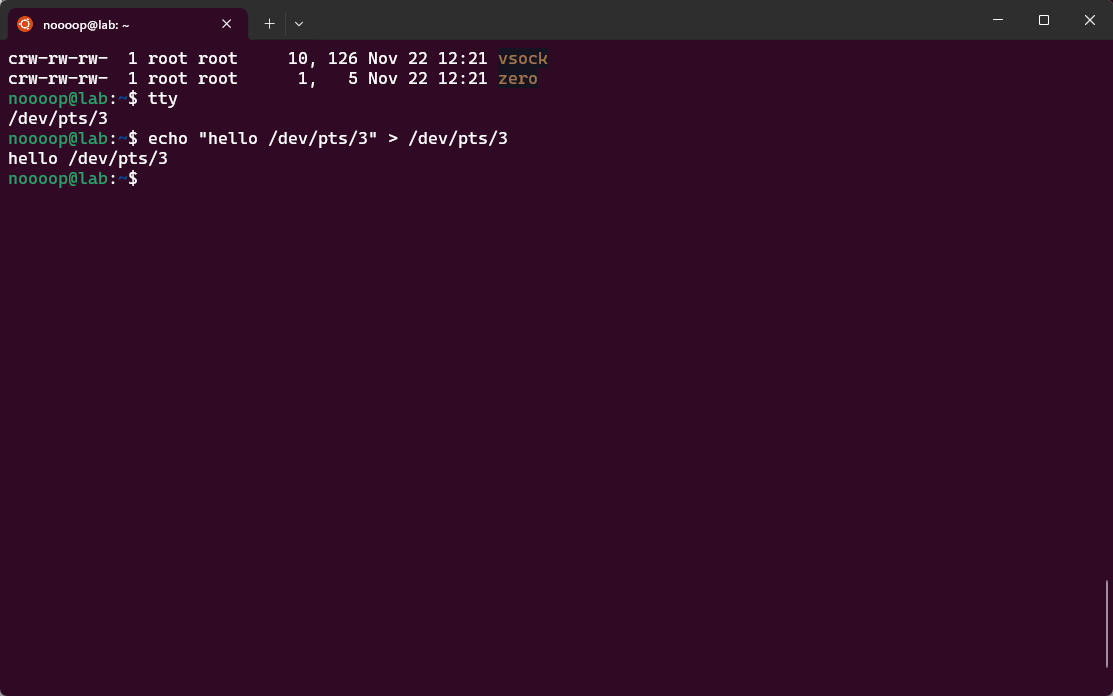


所有的硬件包括虚拟硬件都在 /etc 这个目录

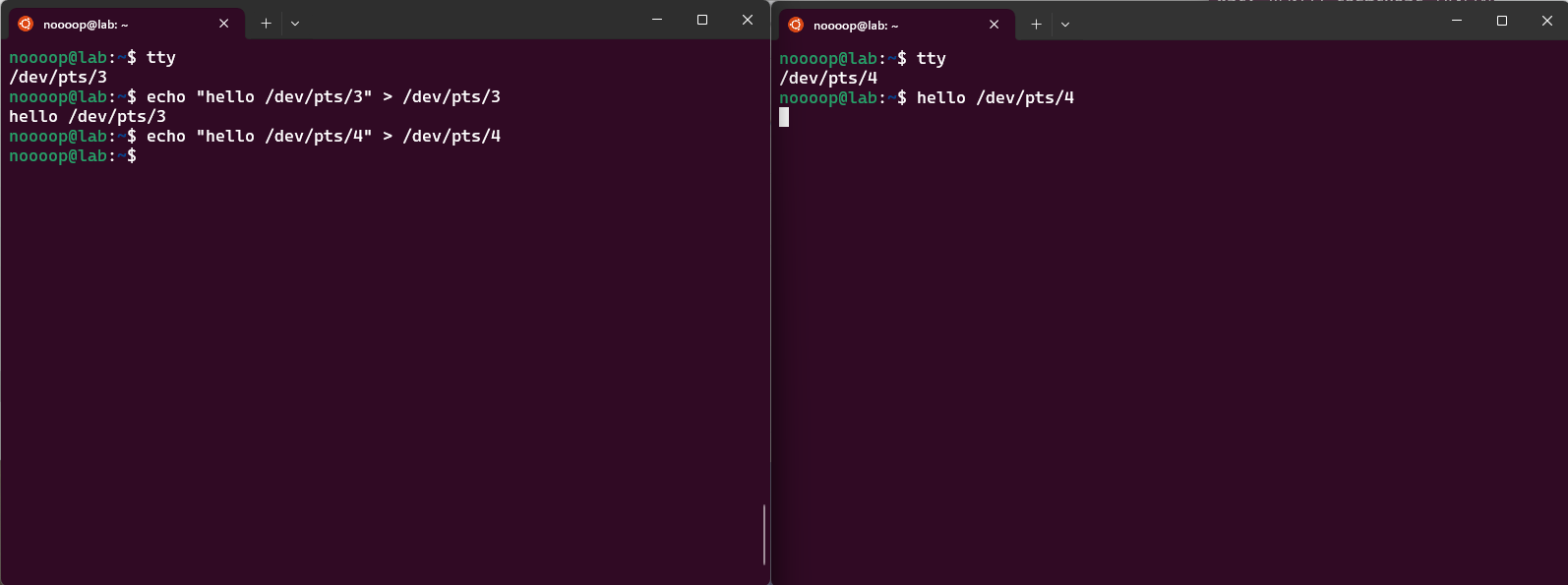
比如正在交互的这个终端



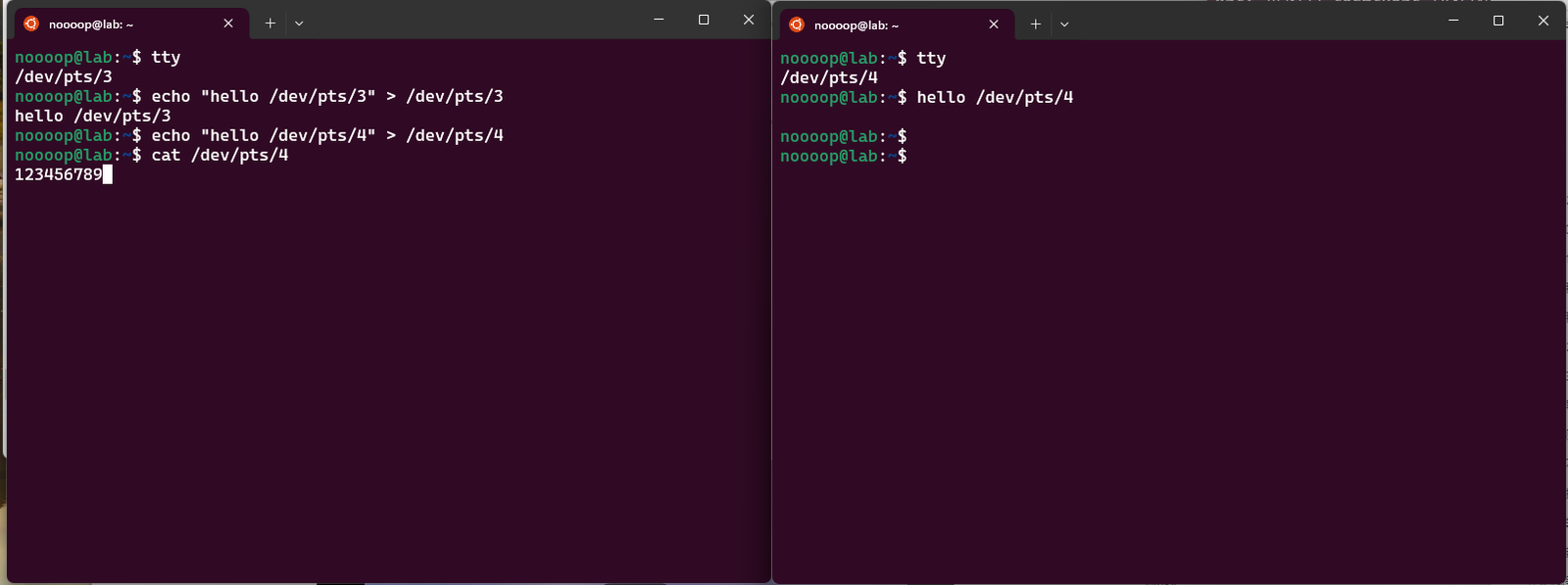
使用tty命令，这个终端设备挂在/dev/pts/3，我们跟它打个招呼



我们还可以跟另外一个终端打招呼



甚至可以把另一个终端输入的内容拿回来



（不要有大胆的想法，因为用户权限，你没办法拿到其他用户的输入）

好，我们继续

Linux 最经典的一句话是：「一切皆文件」。电脑里面所有资源，包括但不限于硬盘里的文件，都以文件的方式挂载在文件系统树上，有约定俗称的一套目录结构。学会操作文件就学会操作计算机上的所有资源。

<https://www.runoob.com/linux/linux-file-content-manage.html>

<https://www.runoob.com/linux/linux-file-attr-permission.html>

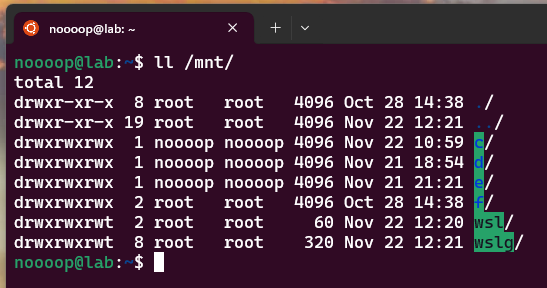
<https://www.runoob.com/linux/linux-user-manage.html>

<https://www.runoob.com/linux/linux-system-contents.html>

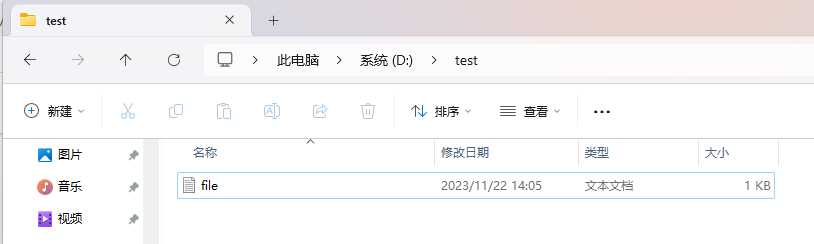
## 0x02 wsl2 文件系统

1. 在linux下访问Windows 硬盘分区内容

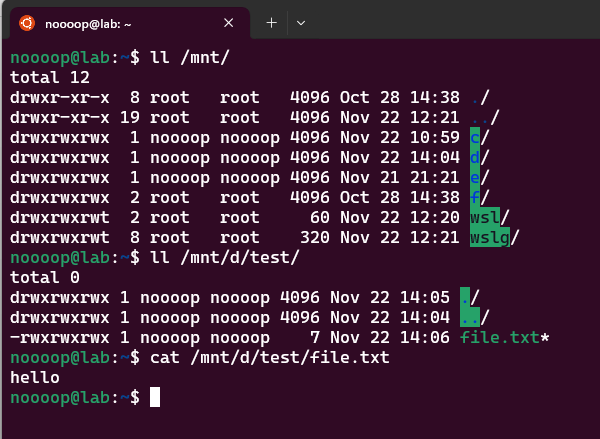
Windows 硬盘分区内容 挂载在/mnt 目录下



比如

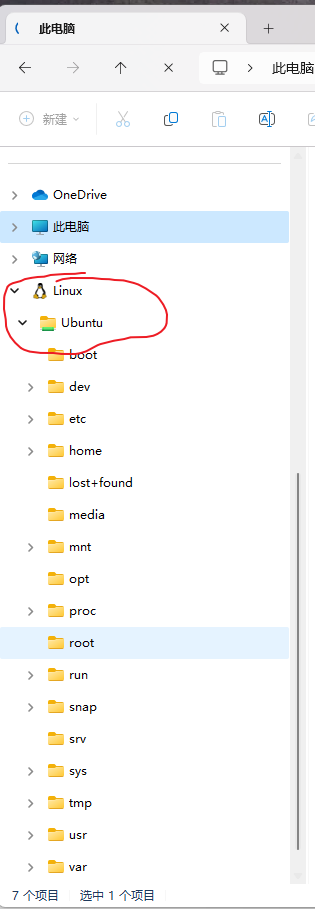


可以在linux下 /mnt/d/test/file.txt 访问



1. Windows下访问linux目录

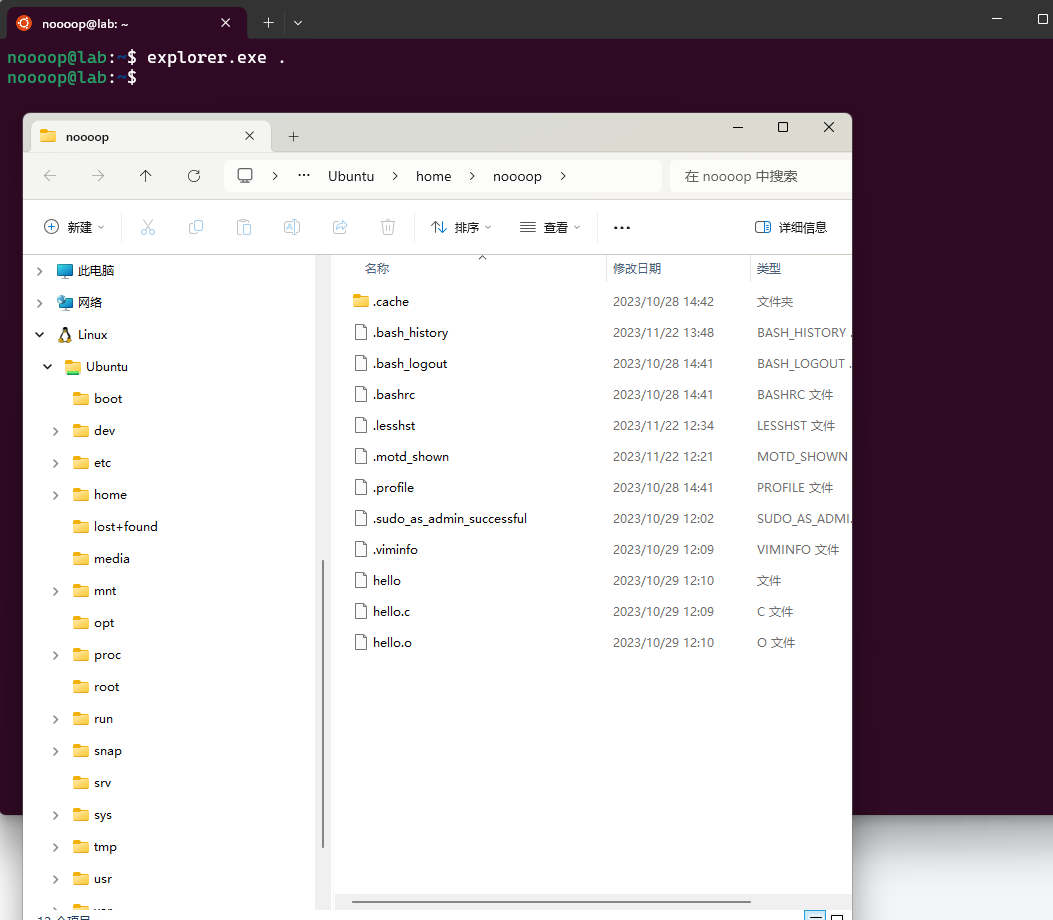
Windows 文件资源管理器 可以直接打开



执行

$ explorer.exe .

Windows 文件资源管理器将打开并显示该分发的目录路径



## 0x03 其他基本操作

这是个慢慢积累的过程，需要的时候现查去菜鸟教程看看

<https://www.runoob.com/linux/linux-tutorial.html>

// 这里需要一个b站教程

# c语言程序设计

<https://www.runoob.com/cprogramming/c-tutorial.html>

## 路线图

c语言程序设计这门课至少包含四层内容：

1. c语言语法本身和c语言标准库。有点像游戏拓荒环境，熟悉基本操作逻辑和初始装备，可以出新手村上路了。
2. 程序是如何在电脑中运行的。因为是第一门专业课，老师要将大量计算机知识穿插进来，不少内容后续要单独讲一门大课，比如《离散数学》《数字逻辑》《组成原理》《系统结构》《微机系统与接口技术》《汇编语言》《操作系统》《编译原理》。大部分概念都没办法在短时间讲透，肯定听的云里雾里。慢慢学会懂的。
3. 数据结构和算法。在基本语法和如何运行之上，计算机有一套系统解决问题的方法，称之为算法，存储数据的方法称为数据结构。特别简单的算法题可以通过动脑筋想出来，大部分问题的解决方法需要系统地学习后续课程《数据结构与算法分析》。

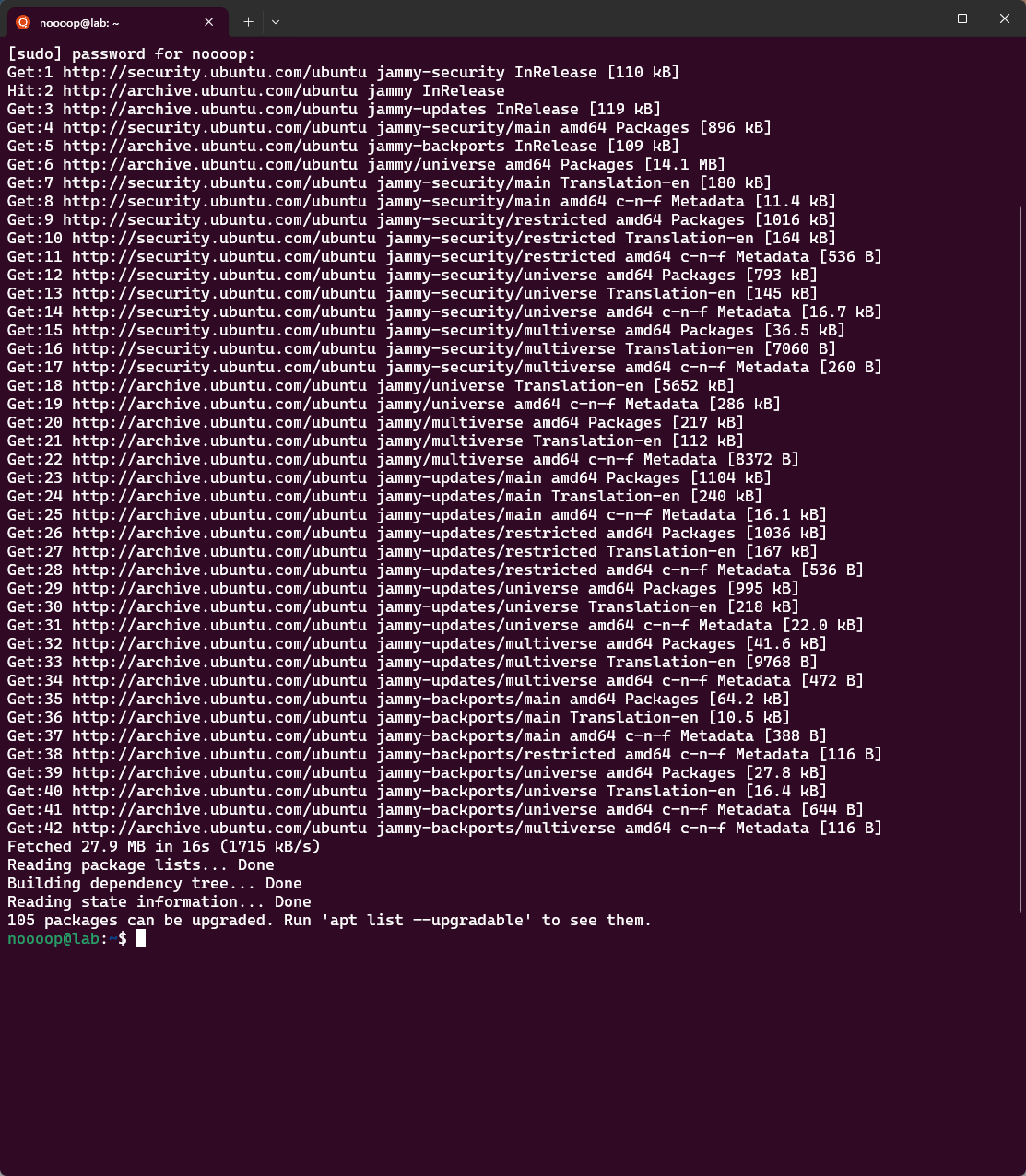
很多老师会引入一些算法题当做c语言课后题、上机题，很多学生做不出来感觉受挫，因为很多题目的解法看上去只涉及c语言本身，实际上有一套完整的分析问题解决问题的方法。甚至很多问题来自于《离散数学》，《数值计算》，《计算几何》，奥数式的学会几个特例问题的特别解法是没啥意义的，需要系统学习。暂时做不出来没有关系。

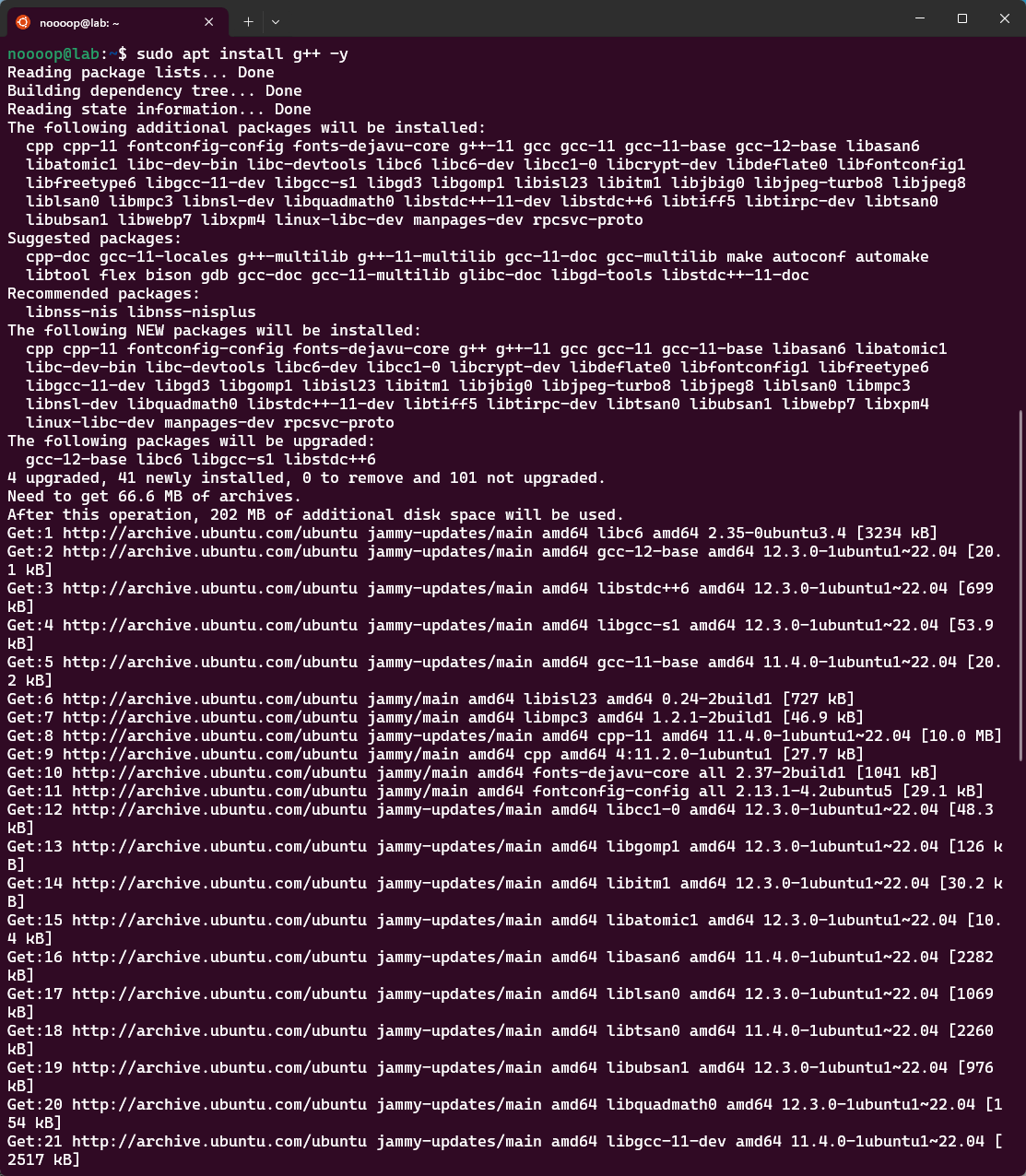
1. 复杂项目开发。一方面，更复杂的项目需要引入更多的库函数，比如操作系统相关的库函数；网络相关的库函数；如果要做嵌入式开发，需要引入嵌入式板自带的库函数把集成的功能调用起来。另一方面，复杂项目管理本身都是就是一门后续需要学的课程 《现代软件工程》。

### 0x01 安装Linux 编译工具链

sudo apt update

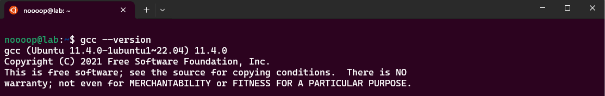
sudo apt install g++ -y





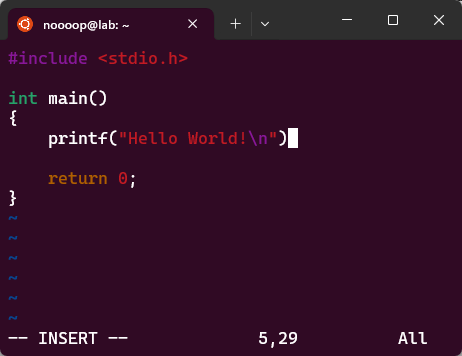
### 0x02 验证安装成功

gcc --version



### 0x03 写第一个c程序

vim hello.c 编辑



编译

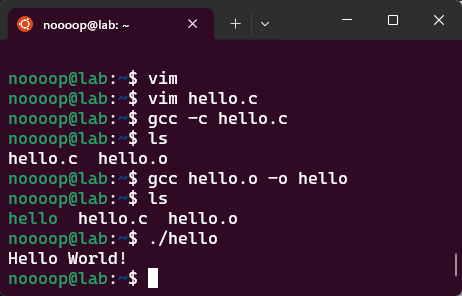
gcc -c hello.c

链接

gcc hello.o -o hello

运行

./hello



运行程序为什么要再文件名之前加./:

<https://blog.csdn.net/u011724770/article/details/121929920>

背景介绍：

计算机发明后的相当长时间，只运行一个程序，把存储程序的介质插入计算机，按下开关启动，读取程序，执行程序，如果有输出，使用打印机（物理的打印）在纸上，停机。

随着计算机能力越来越强，计算机可以同时执行多个程序，输出设备变成显示器，但远古的很多术语比如“打印”传承了下来。为了让多个程序同时执行，不发生冲突，计算机操作系统管理和调度程序。暂时可以简单认为，每个运行程序运行在独占的“虚拟机”里，有单个独占的cpu核心，从0开始连续无限的内存，程序之间不可见，运行期间也不会互相影响。在程序看来好像跟原始的单台机器运行一个程序一样。

操作“虚拟化”是怎么实现的，程序之间不可见又如何通信，等等，都会在《操作系统》这门课里面学习。

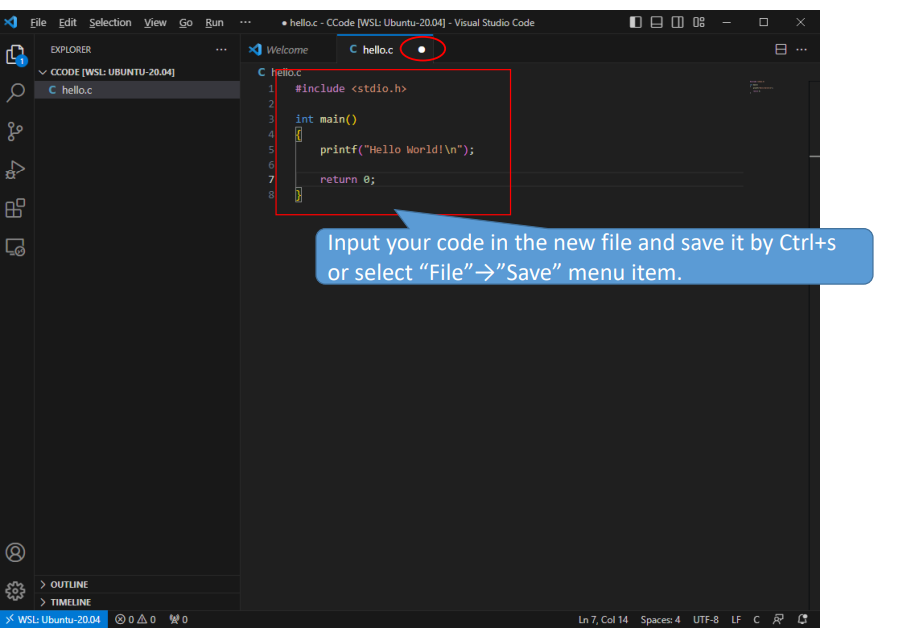
### 0x04 配置和使用vscode

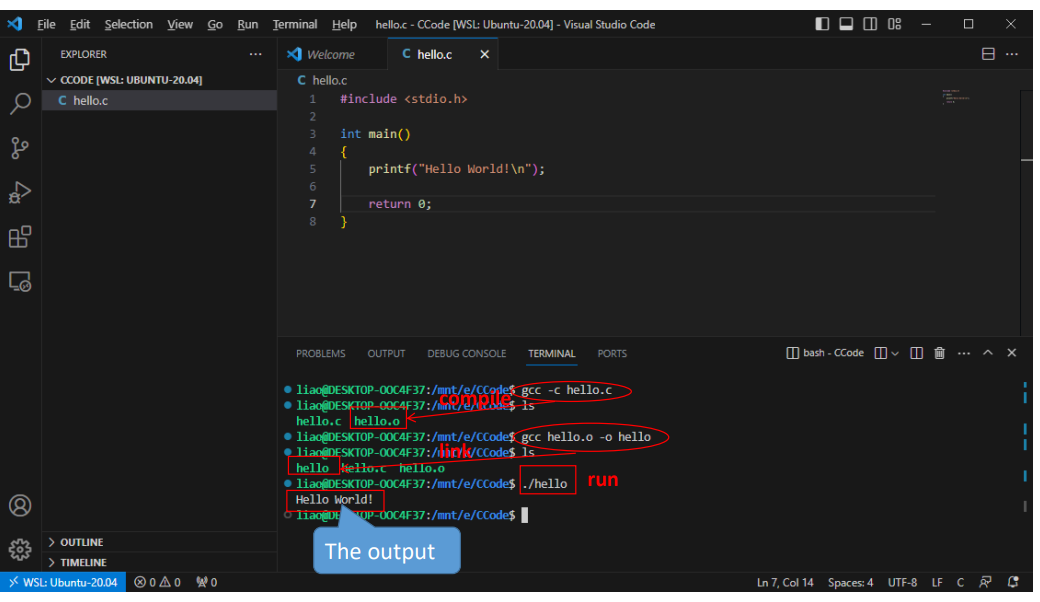
安装 VSCode编辑器

安装C/C++ plugin

安装 WSL plugin

//需要更多文字描述





### 0x05 C语言参考资料

网络教程

// 菜鸟教程不知道好不好，现放着占位

<https://www.runoob.com/cprogramming/c-tutorial.html>

《C程序设计语言，The C Programming Language》

《C Primer Plus》

《C和指针》

《C陷阱与缺陷》

《C专家编程》

《程序员的自我修养 -- 链接、装载与库》

最好不要看的书

谭浩强的《C语言程序设计》

### 0x06 OJ

在线评测系统（英语：Online Judge，缩写OJ）是一种在算法竞赛竞赛中用来测试参赛程序的在线系统，也可以用于平时练习。

算法竞赛通常采取黑盒测试，事先准备好一些测试数据，然后用它们来测试选手的程序。

在在线评测系统中，用户需要提交源代码至服务器，服务器会编译用户的源代码，然后执行源代码生成的可执行文件（或用解释方式执行，或直接执行脚本文件），得到其输出的结果，并与正确结果比较。

<https://zh.wikipedia.org/zh-hans/%E5%9C%A8%E7%BA%BF%E8%AF%84%E6%B5%8B%E7%B3%BB%E7%BB%9F>

// 离线演示一下oj的原理

// 着重讲清楚输入输出重定向

刷一些简单题，练习一下基本的c语言知识。

// 这里需要一个简单题list 网页

算法题等学习 《数据结构与算法分析》课程再刷不迟

注：Linux的使用并不是单纯算法，而是包含了输入输出重定向等，因此在OJ刷题中不建议使用核心代码模式（即给定函数头，只需要实现内部功能的模式）

# c++语言程序设计

## 路线图

c++相比c语言增加了什么？

# 简单项目

C语言搓一个HTTP服务器

# 提高项目

## 嵌入式开发

Arduino

STM32

树莓派

路由器