Lecture 1

(注：下文中C指代ctrl，F指代fn，W指代windows，A指代alt)

> 这是一个Markdown写成的讲义，你可以将它线上转化为pdf，或者使用vscode的Markdown侧边预览进行查看。如果你想获得更美观的视觉体验还可以试试MarkText

课前准备

0.如果你使用的是苹果电脑，你可以跳过课前准备步骤。macOS和Linux之间的关系详见“UNIX史话”一节。

1.W-r，输入powershell

2.依次键入以下指令安装WSL2

```
wsl --install --no-distribution
wsl --set-default-version 2
```

3.通过Microsoft Store安装Ubuntu-24.04 LTS

4.在弹窗中输入用户名与用户密码，一般我习惯自用机直接用分发版的名称进行命名，比如用户名ubuntu，密码也是ubuntu，然后稍等片刻，在弹窗最下方出现一行以 $ 结尾的字后关闭弹窗

> 不要在出现 $ 之前关闭弹窗！如果关闭了请在Powershell中使用命令 wsl --unregister Ubuntu-24.04 并回到第三步重新安装！

5.键入以下命令将Ubuntu-24.04导出到D盘

```
wsl --export Ubuntu-24.04 D:\Ubuntu2404.tar
wsl --unregister Ubuntu-24.04
wsl --import Ubuntu-24.04 D:\Ubuntu2404 D:\Ubuntu2404.tar --version 2
```

6.这时Ubuntu中默认以root进行登录，将默认用户改为之前你所设置的用户

```
Ubuntu2404 config --default-user ubuntu
```

7.在C:\用户\<用户名>\处新建文本文档，更名为.wslconfig，使用记事本打开，输入以下内容并保存

```
[wsl2]
networkingMode=mirrored
```

这样WSL就可以使用本机的代理网络了

> 说实话，WSL的安装算是Linux各种载体中最为简便的一种了。如果你在上述步骤卡住了请及时求助，不要放弃并选择下载某个配好的Linux镜像绕路，你应当拥有使用纯净、由你定制的Linux Distro的权力！

> 你还可以尝试在VMware或者Oracle VirtualBox中安装Linux虚拟机，或者如果你备份好了文件也可以尝试安装物理机Linux。我们把它设置为了一道练习题，你可以课后试试！

Hello, Linux!

初见

当你输入

```
wsl -d Ubuntu-24.04
```

之后，你便进入了Linux。对，你只需要命令行，也就是Linux Shell就能和Linux进行互动了。
如果你想回到Powershell，只需要使用`exit`命令即可
无论什么系统，你大概率都会问出一个问题：**我的XXX文件在哪儿？**一个Windows用户的回答大概率会以磁盘的某个分区，比如“C盘”或者“D盘”开头，每当切换到一个子文件夹的时候就要使用`\`分割。比如：

```
C:\Users\Lenovo\Applications\DevCpp
```

> 磁盘分区真的有必要吗？

文件夹在英文中为document，而目录在英文中为directory，前者在Windows中更加常用，后者在Linux中更为常用。在后文中我将混用两种表达。

Linux使用一种截然不同的方式进行路径标记，在Linux中有四个特殊的目录：`/`表示根目录，`~`表示用户主目录，`..`是上一级目录，`.`是目前所在的目录。每当切换到一个子目录时使用`/`分割。

比如刚才那个文件的路径就应该表示为：

```
/mnt/c/Users/Lenovo/Applications/DevCpp
```

`/mnt`是什么？确切来讲是临时挂载目录。你可以大概理解成把c盘和d盘分别当成一个大U盘插在Linux上。

`cd`(change directory)可以用来切换目录，在命令行输入

```
cd ~
```

就可以前往用户主目录。同理，这时

```

```

```
cd ..
```

可以前往/home，

```
cd .
```

不会发生变化，因为.代表目前所在的目录。

> 当你不知道现在自己在哪儿的时候，使用pwd查看当前的目录

使用

```
clear
```

可以来清理终端上的显示信息，如果你想使用之前用过了的某条指令，试试上下方向键。

我们回来，

```
cd /mnt/d
```

那如果是我这时候 `cd ~` 了又该如何回到`/mnt/d`呢？你可以使用

```
cd -
```

来回到上一个你进入的目录。

everything is a file

> In Linux, everything is a file.

file，中文译为文件。如果你问AI在Linux中文件是什么，它大概率会给你扯inode和块指针一类你听也听不懂的名词。我们不在乎那些，我们在乎的是"文件"的抽象为我们提供了一套统一的操作：查看文件，创造文件，打开文件，读文件，写文件，执行文件，关闭乃至删除文件。

查看有什么文件和目录: `ls <path>`
创建目录: `mkdir <dir_name>`
创造文件: `touch <filename>`
读文件（查看文件内容）: `cat <filename>`
执行文件: `./<filename>`
复制文件: `cp <src_path> <dst_path>`
剪切文件: `mv <src_path> <dst_path>`

删除目录: `rmdir <dir_name>`
删除文件: `rm <filename>`
这些命令还可以加上flag进行操作，比如我想把文件列成一个更长的表格，我就可以

```
ls -l
```

这里的`l`就是long的缩写。
或者如果你想删除一个目录及其中的所有文件，你可以使用

```
rm -rf <file_name>
```

其中 `r` 代表recursive，也就是递归地，而`f`是force的缩写，强制进行

> 注意，这里的执行文件前面必须加`./`，否则系统默认这个被执行的文件在`/bin`目录下

```
cd /bin
ls ls
ls mkdir
ls *dir
```

> 复制和剪切极好地体现了文件抽象的好处。你真的在乎文件的后缀名或者名称吗？

> rmdir只能用于删除空目录，如果目录中有内容则应当使用有参数的rm

> rm的操作是不可逆的，请在删除文件时仔细检查！

至于写文件，在Linux中内置了一个文本编辑器`nano`，你可以使用`nano <filename>`打开它。这个命令的默认行为是在当前路径下有这个文件的时候打开这个文件，如果没有就新建文件。好了，你已经学会nano了。
还不信？使用方向键移动光标，剩下的操作方法都在最下头写着呢，顺带一提，有些表示方法中会用`^`代表Ctrl键，M是meta键，这是一个存在于上世纪70年代键盘上的老键，现在已经不存在了。在今天的电脑上它被映射为Windows上的alt或者macOS上的command
Linux中也有一些其他的写文件命令，比如`tee`：

```
tee hello.txt << EOF
# 还记得EOF吗？它是文件结束标识符
```

管道是一个强大的工具，不过这个等会儿再说。

帮手们

你可能这时候会想：同样的操作我在Windows上用鼠标右键也能干啊，还真干不成，因为右键菜单并没有为你提供更多的选项。当你使用右键的时候，你得跟着图形界面固有的逻辑走，它给你提供什么功能你就

只能使用什么功能。命令行比你想象的强大。

tab

现在有一个叫`ThisIsAVeryLongDirectoryName`的目录，你该怎么进入这个目录？不用把整个名称都打出来，你只要`cd T`，然后再按一次`tab`键，命令就会自动为你补全。如果按一下tab没反应，那就说明匹配这个开头的名称不止一个，敲两下看看可供你补全的选项。

man和info

接下来是`man`和`info`，当你输入`man`的时候系统会提示你输入`man man`进行查看。（这里大致浏览提示并进行解释）ok，那我们应该输入`man ls`来查看ls相关的帮助文档。可以看到ls有一些可选参数。我们试试`ls -la`，以列表形式列出文件，不忽略.开头的文件。info也是同理的，你可以使用`info <name>`进行查看。

使用方向键在帮助文档中上下翻阅，使用q退出。当想要寻找特定内容时不妨试试`/<pattern_name>`进行查找，比如在`man ls`中查找递归列出文件的参数你就可以使用`/recu`进行查找，这个参数应该是-R

你可以大致看出Linux的设计思路——每个工具只做一件事，但是绝对做好，做到极致。

> cd不是一个命令，man中也没有设置cd的相关信息。因为如果cd是一条命令，那么shell将产生一棵无穷递归的进程树（以后你学了操作系统就知道这句话是什么意思了）

停止命令的执行

```
sleep 1000
```

这时候你会怎么停下它，关掉这个终端再开一个？那可太不优雅了。Linux中有很多停止程序运行的方法：

- C-c: 向前台程序发送SIGINT信号
- C-z: 停止前台程序的运行并将这个前台程序转移到后台
- C-d: 发送信号杀死前台程序对应的进程

前台和后台的概念其实你不陌生，你在手机上打开了某个app，当你想要打开另一个app的时候你会上划一下退出，这时候你先打开的app就在后台运行，而你正在实时交互的app则在前台运行。

下载软件和包

你在Windows上怎么下载文件？
大概有两种情况，一种是使用下载向导，另一种是使用某个压缩包，解压，把文件放在某个路径下方。但你想，如果路径什么的由计算机决定，自己只要下载想下载的东西就行，那岂不是很好？包管理器可以满足你。
Ubuntu和Debian的包管理器是apt

从镜像源开始说起……

包管理器会从仓库查找有没有你想下载的软件或包，如果有就进行下载前准备，如果没有则返回错误信息。绝大多数Linux Distro的仓库都在美国（或者说发行这些Distro的国家或地区），为了加快下载速度，我

们需要配置镜像源。镜像源在世界各地都有，他们承担着为主仓库分流和提高所在地区Linux用户下载速度的任务。本质上镜像源是一个分布式系统，或者说是仓库的一个副本。它每隔一段时间便通过脚本与主仓库进行同步，保证镜像源中软件的时效性。
在各镜像站的官网可以直接查询到镜像源的更新方法，不过在此之前你应该先备份原先的主仓库源以防更新失败。

```
# backup
# this command is suitable for Ubuntu-24.04
sudo cp /etc/apt/sources.list.d/ubuntu.sources
/etc/apt/sources.list.d/ubuntu.sources.bak
```

然后使用`nano`编辑软件源列表

```
sudo nano /etc/apt/sources.list.d/ubuntu.sources
```

把官网的内容进行复制并保存
最后，更新包管理器的软件源

```
sudo apt update
```

等等，sudo是什么，为什么这些命令不加sudo会提示`Permission denied`？

Linux的用户策略

Linux中有用户的概念。还记得课前准备中的用户名和密码吗？用`whoami`看看输出结果，再使用`sudo whoami`看看输出结果。`sudo`，即`super user do`，以超级用户身份执行。
操作系统的一大要务就是隔离(isolation)。它假定用户的一切行为都是不可信的，都是妄图越权的，而管理员和内核是绝对可信的。文件下载本质上是一个较为“危险”的操作，你想想你折腾Windows还经常被不知不觉装了假传奇和2345。你的`ubuntu`用户本质上是一个普通用户，但是当你使用sudo后它便切换成了超级用户，可以执行一些较为“危险”的操作了。比如

```
sudo poweroff
```

可以进行关机。

使用`ls -al`命令，你可以看到类似如下的输出：

```
drwxrwxrwx 1 debian group1 4096 Jan 18 19:34 semester
```

我们从后往前一组一组看。semester显然是文件名称，Jan 18 19:34代表这个文件最近被修改的时间，group1代表的是文件所属的用户组，debian代表文件所属的用户，而drwxrwxrwx中，d代表文件是一个目录，后面三个一组，分别代表拥有者权限、同组用户权限和非同组用户权限。

```
# r is for read
# w is for write
# x is for execute
d r-x rwx ---
^  ^   ^   ^
|  |   |   +--其他用户不能读，写或执行文件
|  |   +------同组用户可以读，执行，但不可以写文件
|  +----------+
文件类型：目录 |
文件归属的用户可以读，写，执行文件
```

文件的权限可以使用chmod进行修改，比如：

```
sudo chmod 777 semester
```

三个一组，把rwx想象成二进制位，比如rwx是111，rw-是110，---是000，这样上方命令的777就是rwxrwxrwx，也就是赋予文件所有用户，同群组用户和其他用户读，写，和执行文件的权限

修改文件权限显然也是一种危险操作，需要使用超级用户权限

```
2025.1.25: chmod在WSL中可能并不起作用。如果不起作用，请使用'sudo nano
/etc/wsl.conf'然后输入
[automount]
options = "metadata"
打开另一个powershell，输入
wsl --shutdown
wsl -d Ubuntu-24.04
再尝试一下修改权限
```

当然，你还可以使用sudo -i直接切换到超级用户进行任意操作。你的终端的提示字符从$（或者在macOS上，%）变成了#，这时你可以使用exit退出超级用户登录
请注意权限问题，超级用户进行操作的一些文件普通用户可能无法进行相同权限的操作！
实质上，用户策略还有一个更为历史的视角。20世纪70年代，计算机还没有降低成本向家庭用户普及，在高校或科研所中的计算机通常被不同组别的用户进行使用，而受制于当时的中央处理器性能，计算某个参数可能要花几小时甚至几天。因此，UNIX系统自诞生之初就担负着隔离和持久使用两大要务。隔离的思想催生出了UNIX的用户策略，持久化的思想催生了分时多任务。（关于分时多任务可以参考*The Linux Time-Sharing System*一文）

下载

```
sudo apt install <pack_name>
```

可以用于下载包/软件，同理，`remove`用于卸载指定的包。
你应该定期清理包管理器的缓存并进行包的更新，防止环境兼容问题

```
# 从源拉取最新的包信息
sudo apt update

# 下载所有可行的升级
sudo apt upgrade

# 卸载所有可能用不到的依赖
sudo apt autoremove

# 清理下载产生的缓存
sudo apt clean
```

这多爽！统一的命令进行优雅的升级操作和干净的卸载操作，不用担心任务管理器中突然多出来一堆`Visual Studio C++ 20XX`了

当然，还可以直接从某些官网拉取文件进行下载，这里要使用`curl`或者`wget`，比如下载*The C Programming Language*：

```
wget <URL>
```

正确的方法

到现在为止，你已经基本掌握了Linux的操作方法。你在使用过程中理所当然地会遇到一些你没见过的命令和你没见过的报错。这时候你会怎么做？慌了，还是着急私信某个群友？

多查，多看，自己先试试

Read The F**king Manual 和 Search The F**king Web是Linux创始人Linus应对不假思索直接邮件骚扰他提问的人的标准回复语。你应当明白，大佬们很忙，你打扰他们大概率不会得到满意的答复，你在耽误他们投入专业工作或者在工作之余进行休息的时间，你还在破坏国内本就不太好的计算机学术生态。

那怎么办？你应该掌握科学的解决问题的方法。遇到问题先尝试自己折腾，用科学的方法自己折腾不明白再用科学的方法提问

什么是Manual？Manual是官方手册，官方文档，它本质上代表着解铃还须系铃人的追根溯源精神，没有人能比一个程序的创造者更懂这个程序了。比如当你看到Python中的某个新的原生功能或者遇到了新的问题，你应该首先查这个，而不是这个。同理，当你发现你的JavaScript中有什么你不理解的新特性或者报错，你应该首先查这个而不是这个。不是说后者不能用，而是你在后者中浪费时间的期望远高于前者。

什么是Web？web本质上是寻找可靠的搜索引擎和可靠的论坛资源。当你遇到某个你不理解的新概念或者你没见过的新工具时，你应该用Google而不是百度，当你想问某个常见报错的含义时你应该查Stack Overflow而不是CSDN。要明白Stack Overflow已经纯粹到一定程度了，它甚至没有采用分布式架构，在提问的时候没有插入图片的功能，但正因为它的纯粹，所有在它上头的提问都有头有尾，指向性极强。互联网很强大，只是你的打开方式不对。

可是我的问题依旧没法解决，怎么办？那就提问吧，说出你的问题的详细表象，附上截图或者源码、log而不是拍屏，你尝试了什么修复方法？比如：

```
我用WSL2的git push失败了，我在.wslconfig中将网络设置成了镜像，也尝试了开启或者关闭代理，都不行，最后都提示connection timed out
```

那你大概率会得到设置全局的`git config`的有效回复。

同理的，电脑协会的官方群中如果你想买新电脑求群友推荐，你就应当说明预算多少，日常用啥软件，平时打啥游戏这种最基本的约束条件而不是甩一句"我想换电脑"然后让群友盲猜。正确的提问方法会使你在今后的学习生涯中能够得到足够的训练，获得足够的知识。如果你不知道如何正确提问，详见"拓展阅读"一栏的《提问的智慧》和《别想弱智一样提问》两篇文章

总之，科学的解决问题的方法和科学的提出问题的方法是你学计算机最应该掌握的东西。

为什么你应该试试Linux？

你应当使用正确的工具解决问题。
在Linux中，每个工具只解决一个垂类的问题，但都保证解决到极致。
在Linux中，工具通过管道进行联系，而不是像Windows一样许多时候依赖于手工复制粘贴。比如我想新建一个文本文档并输入一句话：

```
echo "This is an English Sentence" > hello.txt
```

然后再向这个文件中追加另一句话：

```
echo "This is another sentence" >> hello.txt
```

管道使得你可以近乎随意地重定向命令的输出，目标可以是一个文件，也可以是另一个命令行工具。前面的`tee`是一个例子，接下来还有一个：

```
# ifconfig用于查看本机的网络信息
ifconfig > net.txt
```

或者重定向到另一个命令行工具中，再进行排序。我们查看本机网络信息，并晒选以ether开头的信息，然后以字典序进行排序

```
ifconfig | grep ether* | sort
# grep用来过滤文本，sort在没有flag的时候用来对文本进行字典序升序排序
```

不够，还不够……

```
ifconfig | grep ether* | sort | tee ethernet.txt | wc -l
wc -m < ethernet.txt
# wc -l统计输入文本的行数，wc -m统计输入文本的字符数
```

你真的感觉到你在使用"命令行工具"吗？如果有的话，你感觉到文件的输出和命令行工具的输出的区别了吗？这就是对UNIX哲学中"Everything is a file"的更深层次的认识。不管你是狭义的"文件"，目录，命令行工具乃至I/O外设，它们共享着近乎统一的数据结构和统一的交互方式。

想想看，同样的任务交给Windows你会怎么做？而在这个过程中，你又要按多少次C-c和C-v呢？

再比方说，你有一个新的项目，这个项目的文件夹有如下的层次关系：

```
+
|-bin
|-util
|   |
|   +-lib
+-cache
+-exe
|  |
|  +-gui
|  |  |
|  |  +-log
|  |
|  +-cli
|     +-trace
```

我在Linux中只要

```
mkdir -p bin util/lib cache exe/gui/log exe/cli/trace
```

这一句话，那Windows呢？你要进出文件夹多少次，然后按多少次新建文件夹呢？

Linux拥有严格的用户策略，虽说还是时常爆出漏洞，但至少它是清晰的。
Linux对分时多任务的支持比Windows强，DOS作为一个1996年发行的计算机系统依然仅支持单次单个作业
Linux也对各种编程环境更加友好，许多的官方文档中的Windows安装方法下只会冷不丁给你来一句`Please use Windows Subsystem for Linux or MSYS2`

当然，Linux并不是万能的，许多任务Windows比Linux好得多
比如生态强大的图形界面，Linux的桌面套件时至今日都没有Windows的桌面套件那么成熟
再比如历史包容性。许多发行于Windows XP乃至Windows98的软件都可以在Windows11上原生运行，但是Linux在历史包容上较为缺乏

为什么我说这么多关于Linux和Windows孰优孰劣的争论呢，是因为我很喜欢带节奏吗？不。所有这些都是只有你浸淫在双系统中才能体悟到的。当你遇到一个新的工具的时候你应该自行通过搜索来了解它的特点甚至试用一下，而不是望而却步。当你问出我该不该试试nano的时候就已经表现出你太习惯于在自己的舒适圈呆着了。试试用呗，不习惯再回来！

工具就摆在那里，为什么不去试试呢？

可我还是好菜啊

Just happy hacking, 像业余者一样自信（见拓展阅读）

UNIX史话

20世纪60年代中叶，MIT，Bell Labs和通用电气合作开发了Multics操作系统，然而这个操作系统由于过度臃肿惨遭失败
20世纪70年代，Ken Thompson和Dennis Ritchie基于Multics进行简化，设计出了UNIX
1973年，UNIX v4发行，编写语言从汇编语言变为了C。没错，C本质上是汇编语言的副产物。不久后，UNIX v6发行，风靡计算机市场
1975年，ARPA详述了UNIX分时系统作为阿帕网（因特网的前身）服务器操作系统的优势。
1978年BSD出版发行，脱离UNIX独立发展并进行维护。
20世纪80年代中叶，Plan9从UNIX中脱胎，打着原生UTF支持和扩展的文件抽象设计理念问世，然而由于它并没有解决UNIX的固有问题且与UNIX兼容性极差惨遭失败。
20世纪90年代初，UNIX出现了内斗，自由软件和开源软件从此分道扬镳。
1991年，Linux初版内核发行。
2000年，基于BSD的Darwin初版发行，成为了当今macOS的前身，2001年3月，macOS X发行（注：macOS X与前九代macOS有着从内核层面上的本质区别，前九代macOS是另一个闭源的操作系统）（没错，macOS是开源的！）
总之，以UNIX为起点，我们熟知的计算机世界的一层地基被铺好了。

公开课群号：967768744
电脑协会2群：760543989

练习题

注：练习题顺序与难度无关，仅供娱乐！

Linux

(1)思考一下，磁盘分区真的有必要吗，为什么Windows时至今日都在保留磁盘分区这个设定？
(2)分别在 `man 1` 和 `man 3` 中查看`printf`相关的内容
(3)分别使用`tree`和`pstree`列出当前文件和当前进程
(4)了解通配符相关的内容，尝试列出你的某个目录中以r开头以c结尾的文件
(5)使用`lscpu`查看本机CPU和高速缓存相关的信息

(6)使用neofetch，并将命令的输出结果截图发在课程群中装哔
(7)尝试在VMware或者Oracle VirtualBox中安装Ubuntu或者Debian。如果你备份好了数据也可以尝试一下物理机安装
(8)find可以用于查找文件，在man find中查看一下基本的使用方法（呃……这也太长了！要不试试"实用链接"一栏的tldr？）
(?)DANGER ZONE: 在explainshell（见实用链接一栏）中查找以下命令含义：

```
sudo rm -rf /*
sudo mv /* /dev
:(){ :|:& };:
```

（注：在实机上运行这些命令后果自负！！！）

互联网

(1)许多教程会告诉你C-c用于强行退出程序。这种说法有什么不严谨之处？（提示：使用反证法）
(2)白嫖2024年诺贝尔物理学奖得主之一Jeoffrey Hinton的论文（你总不想给Nature交一个月几十美金的会费吧）
(3)在Google上查找正则表达式该如何使用
(4)使用ping查看主机是否能够与某个URL对应的服务器进行连接
(5)查一查Markdown该怎么写
(6)你知道D语言吗？它曾是第一个挑战C++地位的编程语言。大致了解一下它（你不用学，现在几乎没人用D了）
(7)DuckDuckGo是近两年异军突起的一个搜索引擎，试着用一用它
(8)Wolfram Alpha可以用来查询数学和自然科学相关的信息。请在Wolfram Alpha中解微分方程：y'''y - yx = y'' + y'x
(?)地狱难度：Overleaf是一个在线LaTeX集成开发环境，请在**百度**中寻找Overleaf官网

拓展阅读

Windows Subsystem for Linux
UNIX的官方介绍
终端是什么？
鸟哥的Linux私房菜 中文Linux教材中为数不多的正经货，不过它所基于的Linux Distro，也就是CentOS已经于2024年6月30日起无限期停止维护。如果你要看的话，选读普适于整个Linux界的知识
命令行的艺术
提问的智慧 和 别像弱智一样提问
如何对抗技术焦虑？像业余者那样自信！

实用链接

Windows Subsystem for Linux官方文档
man7.org，在线的帮助文档，你可以使用C-f进行匹配查找
explainshell，可以用于直接查找命令的含义
tldr，man和info太长了不想看？看看这个，直接复制粘贴就能使用的命令实例