

CS100 Recitation 1

Linux Basics

Speaker Yuxuan Li (liyx22025@shanghaitech.edu.cn)

VSPLab@SIST



上海科技大学

ShanghaiTech University

目录

1 为什么使用 Linux?

2 Linux 的基本命令

3 Linux 的目录结构

4 Linux 的权限管理

立志成才 报国裕民

为什么使用 Linux?

当我们谈及 Linux 时，我们到底在说什么？

“我只想插一句话，你所说的 Linux，实际上是指 GNU/Linux。”[1]

[1] https://www.gnu.org/gnu/linux-and-gnu.html

立志成才 报国裕民

Linux 是内核，GNU 是用户空间

Linux 本身只是一个**操作系统内核**，它负责管理硬件、进程、内存与文件系统。但一个完整的操作系统，还需要一些常用的**用户空间程序**（例如 bash 和 gcc），而这些由 GNU 计划提供。

因此，严谨地说，我们平时所说的 Linux 系统，其实是**GNU/Linux**系统。

Linux 的发行版

Linux 有众多**发行版**（Distribution），常见的发行版有

- Debian 系：Debian、Ubuntu，包管理器为 `apt`。
- Centos 系：Fedora、Centos、Redhat，包管理器为 `yum` 和 `dnf`。
- Arch 系：Arch Linux，包管理器为 `pacman`。

不同发行版在包管理器、默认桌面、维护策略等有一些不同，但使用起来是差不多的。

本课程假定你使用 Ubuntu 系统（推荐 Ubuntu 24.04 或 Ubuntu 26.04）。

立志成才 报国裕民

有些事情本不应如此繁琐

Windows 是糟糕的

Windows 预装在每一台笔记本上，我们中的大多数人都习惯了，但“习惯”并不等于“合理”。
Windows 上的许多操作之所以繁琐，只是因为我们习惯了它繁琐的样子。

接下来，我们用几个具体的例子，对比同一件事在 Windows 和 Linux 上的做法

1. 软件安装
2. 系统更新
3. 路径与命令
4. 开发环境

立志成才 报国裕民

软件安装

Windows 上安装一个软件，往往要经历这样一番折腾

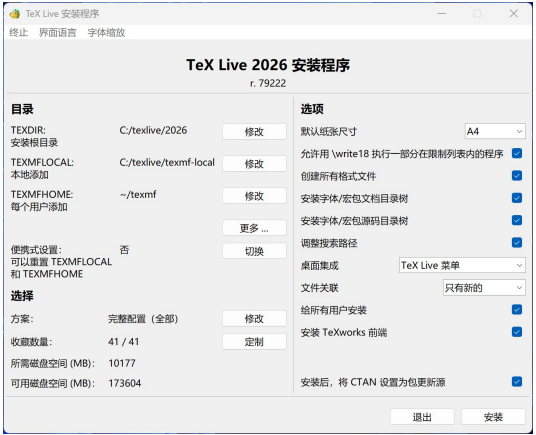
- 打开浏览器和搜索引擎，在众多结果中辨认官网，避开各类欺诈网站。
- 安装程序在下载后需要手动运行，需要一路点击“下一步”。
- 安装路径很随意，有的默认装进 `C:\Program Files`，有的需装进 `C:\`下的独立目录。
- 许多“科普”还会告诉你软件不要装 `C:\`要装 `D:\`，最终散布在各个盘符，难以管理。

立志成才 报国裕民

软件安装

Windows 上，安装 LaTeX 需要以下步骤

- 确定我要下载的东西称为“TeX Live”。
- 访问 `https://tug.org/texlive/`
- 点击“install on windows”页面上的链接。
- 点击“install-tl-windows.exe”进行下载。
- 运行安装程序，选择安装目录。
- 你可能会无法找到正确的下载对象。
- 你可能会在一大堆安装选项中感到迷失。



软件安装

Linux 上，安装 LaTeX 这样庞大的软件只需要一句话

```
sudo apt install texlive-full
```

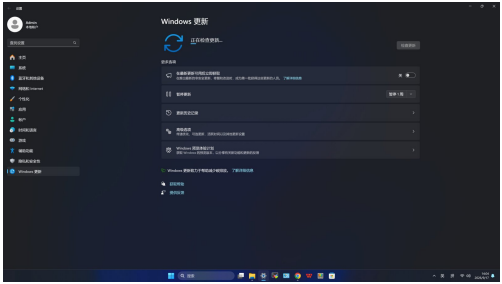
这里 apt 是 Ubuntu 的**软件包管理器**，它有三个关键特征

- 软件包集中化管理：软件集中在官方的软件仓库，不必去浏览器搜索。
- 软件安装方式统一：软件统一用 apt install 安装，自动解析依赖。
- 软件安装路径固定：在 Linux 上，软件安装的路径是有一套严格规范的。例如，执行程序会安装到 /bin，库文件会安装到 /lib，配置文件会安装到 /etc，无需也不能手动选择。

系统更新

Windows 的系统更新常常让人感到恼火

- 强制推送：更新可能在工作一半时突然发生。
- 需要重启：更新意味着重启，不可热更新。
- 不透明性：更新改动时常会破坏系统。
- 体验割裂：更新只覆盖操作系统和预装应用。



系统更新

Linux 并不区分“系统软件”与“应用软件”，它们由同一个包管理器统一管理，可以一起更新

```
sudo apt update
sudo apt upgrade
```

其中，`apt update` 用于刷新软件包索引，`apt upgrade` 用于下载并安装更新。

你可以随时执行这两条命令，更新完成后**无需重启**！

路径与命令

Windows 在路径与命令的设计上自成一派，与几乎所有其他操作系统都不同

- 路径用反斜杠\而非正斜杠 / 分隔，且从盘符 C: 或 D: 开始，没有统一的根目录。
- 路径的反斜杠\是大多数编程语言中的转义符，表示\本身要用\\转义。于是经常需要将 C:\Users\liyuxuan 这样的路径写成 C:\\Users\\liyuxuan，既别扭又容易出错。
- 系统目录组织混乱，且存在 C:\Program Files 和 C:\Program Files (x86) 这种极为糟糕的关键系统目录名称选择：路径中的空格和括号是灾难！命令行中对此需要特殊处理。
- Windows 的命令行 cmd 非常难用，命令名称和选项格式与主流操作系统格格不入：无法使用 cd、ls、mkdir、cp、rm 等常用命令，使用 / 而非 - 或 -- 表示选项（例如 dir /a）。

立志成才 报国裕民

POSIX：类 Unix 系统的共同规范

Windows 才是那个异类

Windows 的这些特性并非来自深思熟虑的工程设计，而是一次次历史妥协的结果。
Windows 虽然是全球最流行的桌面操作系统，但它才是那个异类！

Windows 之外的几乎所有操作系统（包括 Linux 的各个发行版和 MacOS）都遵循了一套统一的接口标准——**POSIX**（Portable Operating System Interface），它规定了命令、路径、进程、文件等应当如何工作。POSIX 是所有类 Unix 系统共同遵循的规范，具体来说，它意味着

- 存在统一的 / 根目录，使用正斜杠 / 作为路径分隔符（例如 `/home/liyuxuan`）。
- 存在 `/bin`、`/lib`、`/etc` 等标准目录，这被称为**FHS**（Filesystem Hierarchy Standard）。
- 存在 `cd`、`ls`、`mkdir`、`cp`、`rm` 等标准命令，- 表示短选项（`-l`），-- 表示长选项（`--help`）。

Windows 是主流操作系统中唯一与 Unix 没有继承关系、也不遵循 POSIX 的系统！

立志成才 报国裕民

开发环境

Windows 是编程学习的最大阻碍

Windows 糟糕的命令行体验使得许多教程无法使用：我该在哪里敲这个命令?

Windows 上编程，往往要先请出一个庞大的**IDE**（Integrated Development Environment）

- 开发 C/C++ 通常要安装体积巨大的 Visual Studio，其使用专有的 MSVC 编译器。
- 开发 Python 通常要安装 Anaconda，其核心的环境管理功能被臃肿的 GUI 界面层层包裹。
- 把“编译、运行”包装在层层按钮之后，容易让人困在表象，忽略背后真正发生了什么。
- 当真的需要敲命令时，又要面对糟糕的 `cmd`，甚至不知道命令该在哪里输入。

开发环境

Linux 上，C 编译器和 Python 解释器都是很常规的命令行工具，很容易使用

```
gcc main.c -o main
python main.py
```

至于你在哪里编写 `main.c` 和 `main.py`？这是编辑器的工作，你可以使用最流行的 VSCode 或历史悠久的 Vim，但这不重要：用什么编写代码和用什么运行代码，是毫无关系的两件事。

每个工具只做好一件事，再通过命令行组合起来，这正是 Unix 哲学。

立志成才 报国裕民

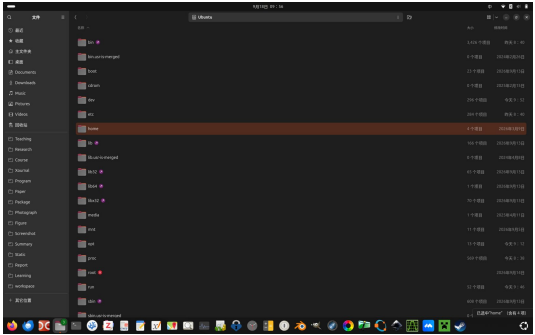
Linux 不只是命令行

Linux 是一个成熟的桌面操作系统：它有完善的图形界面，日常使用与任何系统无异！

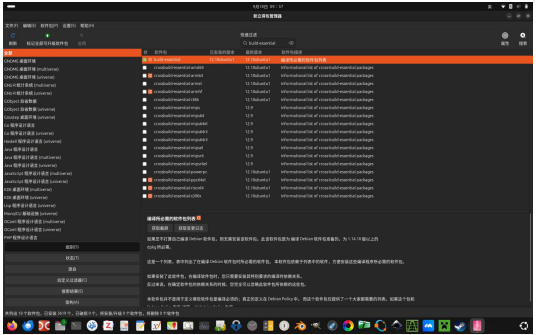
- 你可以用鼠标点击连接蓝牙和无线网络（我并不知道通过命令行怎么做）。
- 在 `cd` 和 `ls` 之外，Ubuntu 也有图形化的文件浏览器 Nautilus（文件）。
- 在 `apt` 之外，Ubuntu 也有图形化的软件商店 Synaptic（新立得）。
- 高效的命令行是 Linux 强大的一面，但不代表一切事情都需要通过命令行完成。

立志成才 报国裕民

Linux 不只是命令行



Nautilus



Synaptic

Linux 也有丰富的软件生态

Linux 常常被质疑的另外一点是软件生态：我需要使用的软件会不会没有 Linux 支持？

- 有些商业软件实际上有非常全面的 Linux 支持，商业软件和开源系统并不矛盾。
- 有些商业软件实际上有功能相当甚至更好的开源软件替代！
- 当然，我们需要承认，确实有一部分软件只提供了 Windows 支持。

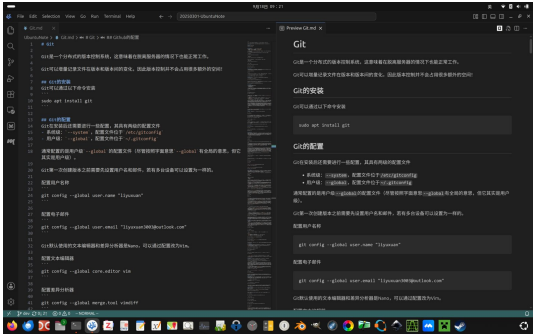
立志成才 报国裕民

文档与表格

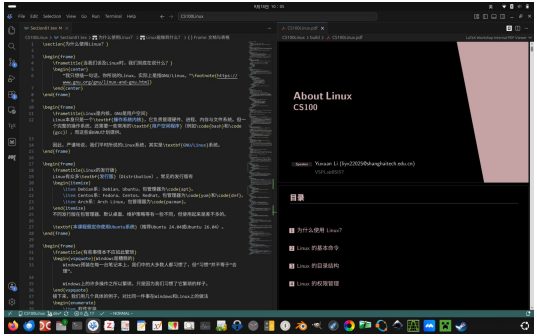
- 图形化编辑文档本身，其实就是一件糟糕的事。
- Word：Markdown 或 LaTeX 写作是更好的选择，内容与格式分离，纯文本便于版本管理。
- PowerPoint：Markdown 有 Marp，LaTeX 有 Beamer，幻灯片本质只是一种特殊的文档。
- Excel：表格数据可以用 `.csv` 存储，表格分析可以用 `pandas` 等 Python 库处理。
- 当然，如果仍然需要传统 Office 软件，开源的有 LibreOffice 可用，商业的有 WPS 可用。
- PDF 阅读器通常是系统预装的，Ubuntu 会搭配 Evince（文档查看器）。

立志成才 报国裕民

文档与表格



Markdown



LaTeX

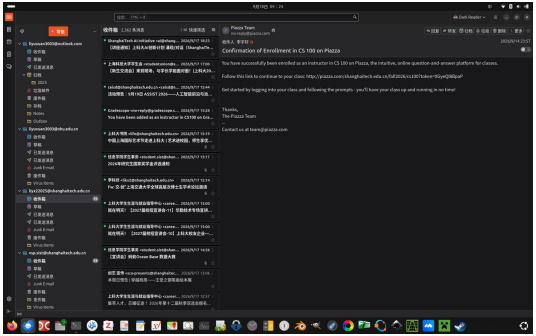
立志成才 报国裕民

- 浏览器可以使用开源的 Firefox，当然 Chrome 也是同样可用的。
- 邮件客户端可以使用开源的 Thunderbird，它的体验远好于现在的 Outlook (New)。

网络与通信



Firefox

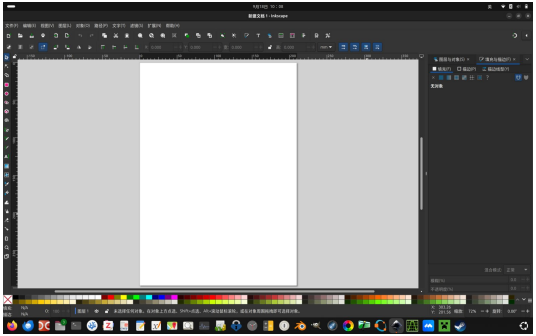


Thunderbird

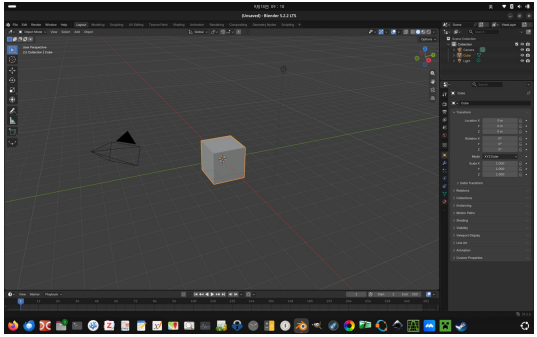
图形与影像

- 位图处理：可以使用开源的 GIMP，替代 Adobe Photoshop。
- 矢量绘图：可以使用开源的 Inkscape，替代 Adobe Illustrator。
- 照片处理：可以使用开源的 RawTherapee 和 Digikam，替代 Adobe Lightroom。
- 三维创作：可以使用开源的 Blender，它是该领域的主力生产工具！

图形与影像



Inkscape



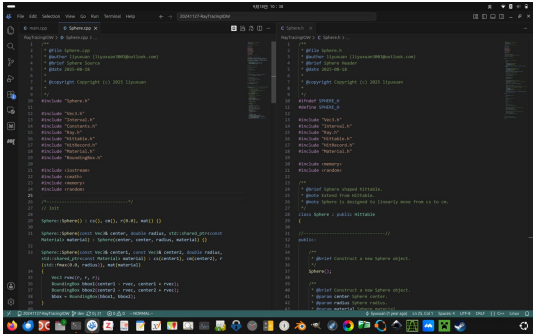
Blender

开发工具

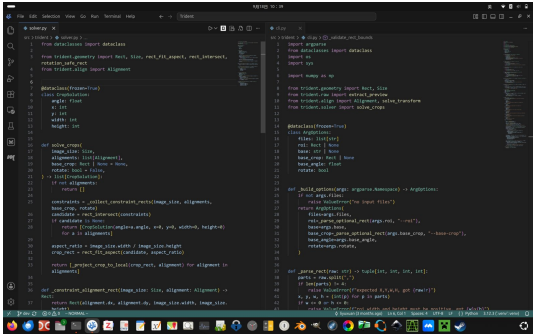
- C/C++ 开发：直接使用 `gcc`、`g++`、`make` 等命令。
- Python 开发：直接使用 `python` 命令，配合 `uv`、`conda` 管理虚拟环境。
- ARM 嵌入式开发不必依赖 Keil，可以使用 `gcc-arm-none-eabi` 的开源工具链替代。
- 代码编辑器：VSCode 同样可用，VSCodium 则是更为纯净的开源构建版本。

立志成才 报国裕民

开发工具



C++

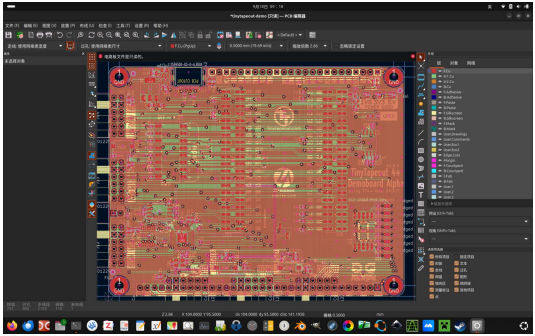


Python

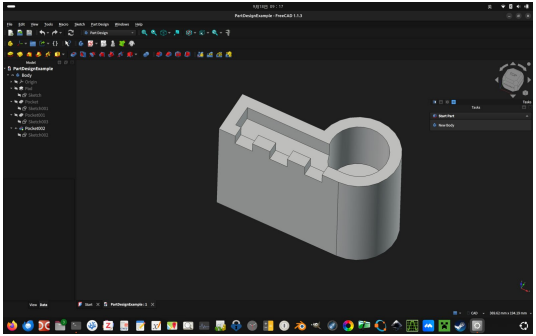
工程软件

- 电路设计：使用开源的 KiCad（立创 EDA 也是基于 KiCad!），替代 Altium 等商业应用。
- 机械设计：使用开源的 FreeCad，替代 Inventor 和 SolidWorks 等商业应用。
- FPGA：Vivado 提供 Linux 版本，仿真可以使用开源的 Verilator 与 GTKWave。
- 符号计算：Mathematica 提供 Linux 版本。
- 数值计算：Matlab 提供 Linux 版本，也有命令兼容的开源替代 Octave 可用。

工程软件



KiCad



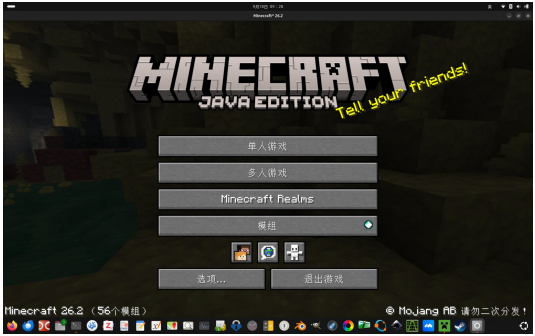
FreeCAD

游戏

- 游戏常常被认为是 Windows 平台的专属，但其实 Linux 上也可以畅玩大部分游戏！
- Minecraft 原生支持 Linux 系统，可以直接玩（仅 Java 版，不支持 Bedrock 版）。
- Steam 提供了兼容层 Proton，**绝大多数 Steam 游戏都可以直接运行**！
- Steam OS 自身就是一个基于 Arch Linux 修改的系统。

立志成才 报国裕民

游戏



Minecraft



Age of Empires II

关于显卡

曾几何时，在 Linux 上使用 NVIDIA 显卡是一件痛苦的事情，甚至因此诞生了 Linus Torvalds 非常著名的“FUCK YOU NVIDIA”的演讲名场面。

但那已经是十多年前的事了。如今 NVIDIA 的显卡在 Linux 上都有良好的官方驱动支持，绝大多数情况下，对于游戏和神经网络训练，显卡开箱即用。



开始使用 Linux

本课程提供了两种使用 Linux 系统的方式

- **WSL**（Windows Subsystem for Linux）：适用于 Windows 的 Linux 子系统，是微软官方提供的功能，本质上可以认为是一个在 Windows 上运行但**只有命令行环境的 Linux 虚拟机**。该方案适用于想继续使用 Windows 作为主力系统，只是想获得一个 Linux 开发环境的学生。
- **双系统**：你可以在磁盘上划出一块空间，在保留当前 Windows 系统的情况下，安装完整的 Linux 系统。开机时可以选择启动哪个系统。该方案适合希望体验 Linux 桌面环境的学生。
- 推荐使用 Ubuntu 26.04 或 Ubuntu 24.04，但你也可以使用其他 Linux 发行版。
- 此外，对于 MacOS 用户，你可以直接在 MacOS 上进行开发，无需 Linux 虚拟机！

我们真诚鼓励所有学生尝试双系统方案[2]，体验在一个完整的 Linux 桌面系统上工作。

[2]HEAD TA 承诺为在安装双系统的过程中遇到技术问题的学生提供技术支持。

立志成才 报国裕民

WSL：最简单的开始方式

WSL 的安装需要在**正确的网络环境**下，在 PowerShell 上执行

```
wsl --install -d Ubuntu-26.04
```

WSL 在安装后（可能需要重启一次）可以通过以下命令启动

```
wsl
```

WSL：最简单的开始方式

- WSL 只有命令行环境，但这不意味着你需要在命令行上编写代码。推荐在 Windows 上安装 VSCode 作为代码编辑器，VSCode 可以通过“连接到 WSL”访问 WSL 上的文件。
- VSCode 下载链接：`https://code.visualstudio.com/Download`

双系统：更好的选择

- 下载 Ubuntu 24.04 或 26.04 的系统镜像（ISO）。
- 使用 BalenaEtcher 或 Rufus 将镜像写入 U 盘，制作启动盘。
- 若 Windows 启用了 BitLocker，**请务必先关闭，否则可能导致 Windows 损坏**。
- 在 Windows 的磁盘管理中压缩卷，留出未分配空间用于安装 Ubuntu。
- `https://ubuntu.com/download/desktop?version=24.04`
- `https://ubuntu.com/download/desktop?version=26.04`
- `https://etcher.balena.io/`
- `https://rufus.ie/en/`

双系统：更好的选择

- 将启动盘插入电脑，重启，按 F2、F10 或 DEL 进入 BIOS，选择从 U 盘启动。
- 在图形界面中完成安装，注意选择中文语言和第三方驱动。
- 在分区配置中选择手动分区，在未分配空间上创建分区并挂载至 /。
- 文件系统推荐 Btrfs 而非默认的 Ext4。
- **创建分区会格式化对应区域，请务必不要格式化 Windows 正在使用的分区！**
- 在 Linux 上我们更推荐使用 VSCode 的开源构建版本 VSCodium。
- `https://vscodium.com/`

C 语言的最小工作示例

首先，安装 C/C++ 开发环境（build-essential 打包了 gcc、g++、make 等工具）

```
sudo apt install build-essential
```

随后，创建一个 main.c 的文件，写入以下内容

```c
#include <stdio.h>

int main(void) {
    printf("Hello World!\n");
    return 0;
}
```

C 语言的最小工作示例

通过以下命令将 main.c 编译为执行程序 MyProgram（Linux 的执行程序没有后缀）

```
gcc -o MyProgram main.c
```

通过以下命令运行 MyProgram

```
./MyProgram
```


立志成才 报国裕民

Linux 的基本命令

软件包管理器

在 Linux 系统中，几乎所有发行版都采用了某种软件包管理器，例如，Ubuntu 的软件包管理器就是 apt，你可以使用 apt 命令完成大部分软件的搜索、下载、安装、更新、卸载！

立志成才 报国裕民

命令 apt

通过 apt 可以进行更新，更新会分为两步

```
sudo apt update
sudo apt upgrade
```

其中，update 的作用是刷新软件包索引，以了解有哪些软件可以更新，upgrade 的作用是真正更新软件包，sudo 的含义是以 root 身份（Linux 下的超级用户）运行。在 Linux 系统中，系统软件和应用软件是一体的。在安装完系统后，应当立即运行一次上述两条命令进行系统更新。

使用 sudo 需输入密码。在 Linux 下输入密码是没有回显的，盲输按回车即可。

立志成才 报国裕民

命令 apt

通过 apt 可以安装软件，例如以下命令会安装 vim，这是一个命令行上的编辑器

```
1 sudo apt install vim
```

你不需要指定安装路径！这在 Linux 系统下有一套严格的规范。

立志成才 报国裕民

命令 pwd

pwd 用于显示当前路径（Print Working Directory）

```
1 pwd
```

你可能会得到类似于 /home/liyuxuan 的结果。事实上，当前路径总是会显示在你的命令提示符上，只不过当前用户的家目录会用 ~ 省略表示。终端启动时，通常默认位于家目录。

立志成才 报国裕民

命令 ls

ls 用于显示当前路径下的文件（List Directory Contents）

```
ls
```

搭配 -l 显示详细信息，搭配 -a 显示隐藏文件，也可以查看指定路径的文件

```
ls -l
ls -a
ls /home
```

Linux 命令中，参数以空格分隔，选项以 - 或 -- 开头（短选项/长选项）。

命令 cd

cd 用于移动当前路径（Change Directory）

```
cd /home
```

回到家目录，你可以尝试不同的路径表示方法

```
cd /home/liyuxuan
cd liyuxuan
```

Linux 中，若路径以根目录 / 开头，那就是绝对路径，否则是相对路径。

命令 cd

有两个特殊的目录：用 . 代表当前目录，用 .. 代表上一级目录

```
cd .
cd ..
```

Linux 中，所有以 . 开头的文件都是隐藏文件，用 `ls -a` 查看。

立志成才 报国裕民

命令 touch

touch 用于刷新时间戳，但如果目标文件不存在，它会创建一个空文件

```
touch file.txt
```

你可以用 ls -l 查看文件的时间戳（即最后修改时间）。

命令 mkdir

`mkdir` 用于创建目录（Make Directory）

```
1 mkdir dir
```

你可以在 `ls -l` 中观察第一列，若以 `d` 开头则为目录，若以 `-` 开头则为普通文件。

立志成才 报国裕民

命令 echo

echo 用于打印文字，请注意任何包含空格的参数需要用双引号包裹

```
1 echo "Hello World"
```

通过 > 和 >> 可以将命令行上的输出重定向到文件，> 是覆盖，>> 是追加

```
1 echo "Hello World" > file.txt
2 echo "Hi!" >> file.txt
```


立志成才 报国裕民

命令 cat

cat 用于查看文件内容（Concatenate）

```
1 cat file.txt
```

该命令实际上是 Concatenate 的缩写，但不妨理解为一只猫挠了一下文件（^_^）。

立志成才 报国裕民

命令 vim

`vim` 是一个命令行上的代码编辑器（你可以用它来编辑任何文本文件！）

```
vim main.c
```

Vim 有一套非常高效的快捷键体系，但它的使用方式常常会让新手感到困惑，牢记以下几点可以让你至少把它当成一个普通编辑器来使用。Vim 启动默认处于普通模式，键盘上几乎每个按键都是快捷键！你会发现你无法编辑文件，正确的做法是按下 `I`，这会让你进入插入模式，这就和普通编辑器一致了。完成后，你需要按 `ESC` 回到普通模式，输入 `:wq` 进行保存和退出[3]。

Vim 常用于编辑需要 `root` 权限的配置文件：这种情况下你无法使用 VSCode 或 VSCodium！

[3]关于 Vim 一个经久不衰的笑话是：如何退出 Vim?

命令 mv

mv 用于重命名或移动文件（Move File）

```
mv file.txt file.md
mv file.txt dir
mv file.txt dir/file.md
```

- 若目标是一个已经存在的目录，文件会被移动到该目录下。
- 若目标不存在或不是目录，文件会被重命名并移动至目标指定的路径。

mv 不区分操作对象是文件还是目录，也可以用来重命名或移动目录。

命令 cp

cp 用于复制文件（Copy File）

```
cp main.c main.cpp
cp main.c dir
cp main.c dir/main.cpp
```

搭配 -r 选项可以递归复制目录

```
cp -r dir source
```

命令 rm

rm 用于删除文件（Remove File）

```
rm main.c
```

搭配 -r 选项可以删除目录

```
rm -r dir
```

特别需要强调的是，Linux 的删除是没有回收站的，三思而后行！

按下回车前请三思

命令行非常高效且强大，但错误的使用会造成格外严重的后果！

- 我正在操作的文件是我想要操作的吗？
- 我是否位于正确的目录？这是在本地还是在服务器？
- 我是否真的理解我在做什么？尤其是当这条命令是由 AI 提供的。

胆大心细，勇敢地探索和学习，谨慎地实施和执行，**对自己的行为负责**。

著名的危险命令

以下是一条非常著名的危险命令，**绝对不要在任何情况下尝试！**

```
sudo rm -rf /
```

其中，`sudo` 用于提权，`rm` 用于删除，`-r` 允许递归删除，`-f` 允许强制删除，这会跳过部分带保护的文件的删除确认，最后，删除对象是 `/` 即根目录，因此这条命令会高效且悄无声息地抹去这台 Linux 系统上的一切数据，甚至包括操作系统自身（WSL 上执行还会抹除 Windows）！

能力越大，责任越大，请谨慎对待你手中掌握的权限。

命令 ln

`ln` 用于创建链接（Link File），选项 `-s` 代表创建的是软链接

```
ln -s dir link
```

软链接的工作方式类似于快捷方式：当访问 `link` 时，其实是在访问 `dir` 目录下的内容，但和快捷方式有所不同的是，你的路径不会发生跳转，你仍然会位于 `/home/liyuxuan/link` 下。

软链接在 `ls -l` 下会显示为 `l` 类型，代表这是一个链接。

立志成才 报国裕民

软链接与硬链接

链接分为两种，软链接（由 `ln -s` 创建）和硬链接（由 `ln` 创建）

- 软链接类似于 C++ 语言中的指针（本质上是存储地址的变量）。
- 硬链接类似于 C++ 语言中的引用（本质上是一个别名）。

在文件系统层面，文件名和文件本体（inode）通常是一一对应的。硬链接不创建新的文件，而是令一个文件 inode 能关联到多个文件名上。换言之，如果将硬链接理解为文件名和 inode 间的链接关系，那么所有文件其实都是硬链接：有些 inode 只关联到一个文件名，有些 inode 会关联到多个文件名。硬链接命令的作用是在 inode 和文件名之间创建额外的链接。当删除文件时，本质上是在减少 inode 的链接计数，当 inode 的链接计数归零时，空间才会真正被释放。

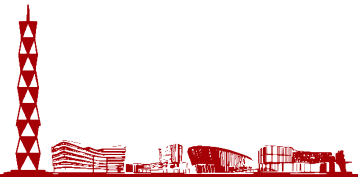
立志成才 报国裕民

软链接与硬链接

软链接则是完全不同的东西，它是一个独立的文件（有自己的 inode），只不过这个特殊的文件存储的内容是一个路径，使得你在访问它的时候实际访问到的是别的地方，由此实现了链接。

- 软链接可以链接目录或文件，硬链接通常只被允许链接文件。
- 软链接可以跨文件系统链接，硬链接则不可以（inode 对每个文件系统独立）。
- 软链接只是一个普通文件，删除软链接无影响，删除源则所有软链接失效。

最佳实践：除非你很清楚自己要做什么，**总是使用软链接**！



获取帮助

几乎所有的命令都可以通过 `--help` 选项查看帮助

```
ls --help
```

除此之外，使用 `man` 可以查看某个命令文档

```
man ls
```

命令补全

你不必完整输入命令！下面展示了输入命令时如何通过按 Tab 来自动补全

```
mkdi<Tab>
mkdir /h<Tab>
mkdir /home/li<Tab>
mkdir /home/liyuxuan/vsp
```

简而言之，按 Tab 可以帮助你补全命令和已经存在的路径。但请理解补全不是魔法，它并不能预知你的想法，例如这里的 vsp 肯定无法补全出来，因为这是你将要创建的目录的名称。

命令历史

你不必重复输入已经输入过的命令！按向上键和向下键可以翻阅输入过的历史命令。

除此之外，使用 `history` 可以查看所有输入过的命令

```
history
```

如果你要搜索包含特定内容的历史命令，可以用管道 `|` 连接 `grep`

```
history | grep mkdir
```

管道 `|` 用于将一个命令的输出重定向到另一个命令的输入，`grep` 用于文本匹配。

立志成才 报国裕民

命令 tar

在 Linux 中，经常会看到 .tar.gz 后缀的压缩包

- .tar 代表归档，即将多个文件打包为单个文件，由 tar 命令完成。
- .gz 代表压缩，即将单个文件进行压缩，由 gzip 命令完成。

但事实上，通过恰当的选项，可以用 tar 命令一步完成 .tar.gz 的压缩和解压缩。

立志成才 报国裕民

命令 tar

tar 用于压缩的命令是

```
tar -czvf package.tar.gz dir
```

tar 用于解压的命令是

```
tar -xzvf package.tar.gz
```

其中，选项 -c 代表 Create，选项 -x 代表 Extract，选项 -v 代表 Verbose，这会列出所有压缩和解压的文件（需要静默输出时可以省略），选项 -z 代表使用 Gzip 格式的压缩算法，选项 -f 代表 File，即指定操作的归档文件。这些选项中 -f 必须放在最后，因为其后要跟随一个参数。

此外，在 Linux 的命令中，短选项允许连写是一个普遍支持的特性。

立志成才 报国裕民

Linux 的目录结构

Linux 的目录结构

Linux 系统的目录结构遵循文件系统层次标准（Filesystem Hierarchy Standard，FHS）。这实际上是几乎所有类 Unix 系统遵循的规范：你甚至可以在 MacOS 上找到相似的目录层次结构！

立志成才 报国裕民

表: Linux 的根目录结构

目录	用途	目录	用途
/bin	用户执行程序	/media	自动挂载点
/lib	动态与静态库	/mnt	手动挂载点
/sbin	系统执行程序	/boot	启动引导
/home	普通用户家目录	/dev	设备文件
/root	超级用户家目录	/sys	设备属性
/etc	配置文件	/proc	进程信息
/var	可变文件	/run	进程数据
/usr	系统资源	/tmp	临时数据
/opt	可选软件	/srv	服务数据

目录 /bin /lib /sbin

/bin 目录存放了所有用户可以执行的命令

```
ls /bin | grep ls
ls /bin | grep mkdir
```

/bin 目录也会包含大部分通过 apt 安装的程序。对于 Linux 系统，它不会区分所谓“系统自带的基本命令”和“用户安装的应用程序”，因为它们两者本质上都是可以执行的二进制程序。

例如，安装 build-essential 包，这包含了 C/C++ 开发的基本工具

```
sudo apt install build-essential
```

你现在就可以在 /bin 下找到 gcc 和 make 等工具了

```
ls /bin | grep gcc
ls /bin | grep make
```

/lib 目录存放了所有动态库和静态库

- 库的本质是预先编译好的通用二进制片段，它可以和应用程序进行链接。
- 动态链接库：运行时链接，后缀为 .so，例如 C 库 libc.so
- 静态链接库：编译时链接，后缀为 .a，例如 C 库 libc.a

你可以在 /lib/x86_64-linux-gnu 下找到 libc.so 和 libc.a

```
1 ls /lib/x86_64-linux-gnu | grep libc.so
2 ls /lib/x86_64-linux-gnu | grep libc.a
```

通常，动态链接是较为常用的。

当你在 C 程序中写下 `#include <stdio.h>` 以及 `printf` 并编译时，函数 `printf` 的二进制并不会被链接到你的程序中，真正发生的事情是，当你的程序运行到 `printf` 函数时，系统会找到 `libc.so` 并运行对应的二进制，这就是所谓“运行时链接”的含义。动态链接机制使系统中海量的二进制程序不必重复包含 C 库的片段，而是可以共享一个 C 库，从而节约大量空间！

然而，静态链接在某些情况下也有其意义。

当你的 C 程序最终是在某个裸机的单片机上运行时，根本没有操作系统进行动态链接，此时就需要在编译阶段静态链接用到的函数。空间上也很划算，只需要带上 C 库中真正用到的部分！

目录 /bin /lib /sbin

/sbin 目录存放了所有 root 才能执行的命令

```
ls /sbin | grep poweroff
ls /sbin | grep reboot
```

这是因为，在多人使用的服务器上，普通用户显然不能随意关机和重启！

立志成才 报国裕民

目录 /home /root

/home 和 root 都是用户的家目录

- /home 是普通用户的家目录，每个用户在 /home 下有同名的家目录。
- 用户 liyuxuan 的家目录位于 /home/liyuxuan。
- 用户的家目录始终可以用 ~ 表示（取决于当前登录的用户）。
- /root 是超级用户的家目录，这是 root 作为超级用户在架构上享有的额外特权：其家目录并非位于通常的 /home/root，而是相应地高挂了一级，直接位于根目录 /root 下。

立志成才 报国裕民

/etc 下存放配置文件，比较典型的例子是 hosts 文件

```
sudo vim /etc/hosts
```

你可以在这里设置本地的 DNS 解析，例如

```
185.199.108.133 raw.githubusercontent.com
```

/var 下存放可变文件，比较典型的例子是 apt 的日志

```
cat /var/log/apt/history.log
cat /var/log/apt/term.log
```

你可以在两个文件中分别看到 apt 最近的更新记录和当时的命令行输出内容。

/usr 的全称是 Unix System Resource，请注意其和 User 无关！

如果你足够细心，仔细观察了 ls -l / 的输出，你会注意到 /bin /lib /sbin 实际上分别是指向 /usr/bin /usr/lib /usr/sbin 的软链接！这是因为早期的实践中，前者存储系统所需的命令和库，后者存储用户安装的命令和库，但是，随着时代发展，现在主流的 Linux 发行版都选择将这两个目录用软链接的方式合并在一起，不再区分到底是系统所需还是用户安装。

目录 /opt

/opt 用于安装可选的第三方大型软件，例如

- Wolfram Mathematica
- Xilinx Vivado
- 腾讯会议

这些软件不遵守标准的目录结构，因此统一在 /opt 下为每个软件分配独立的目录。

立志成才 报国裕民

/media 和 /mnt 分别是自动挂载点和手动挂载点，例如当插入 U 盘后

```
ls /media/liyuxuan
```

假如 U 盘的名称是 udisk，那么你就会在该位置看到出现了一个名为 udisk 的目录，该过程被称为挂载（Mount），这一点和 Windows 中插入 U 盘会出现新的盘符的行为是不太一样的。

通常移动存储设备都可以被自动挂载至 /media，极少数情况需要手动挂载至 /mnt。

目录 /boot

/boot 目录包含了操作系统正常启动所必须的文件。事实上，当在安装系统进行分区时，你可能会注意到除了被挂载于 / 的主分区，还会自动产生一个被挂载于 /boot/efi 的启动分区。

立志成才 报国裕民

/dev 用于存放设备文件，例如

```
ls -l /dev | grep nvme
df -h
```

第一条命令的输出可能会看到 nvme0n1p1 和 nvme0n1p2，这对应了在 NVMe 上连接的 SSD 的第一个分区（p1）和第二个分区（p2）。第二条命令用于查看磁盘占用，观察该命令的输出可以发现，挂载点 / 对应的是 /dev/nvme0n1p1，挂载点 /boot/efi 对应的是 /dev/nvme0n1p2。

在 Linux 系统中，**一切皆文件**。磁盘设备本身也被视为一个文件，这并不是一个真实存在的文件，而是通过文件读写的方式包装了对磁盘设备的数据写入和读取。而挂载的作用，就是将磁盘设备的直接读写操作，转换为我们在文件系统层面所看到的对文件的创建、修改、删除。

设备文件在 `ls -l` 下会显示为 `b` 或 `c` 类型，代表这是一个块设备或字符设备。

/sys 用于配置设备属性，例如

```
echo 0 | sudo tee /sys/class/leds/asus::kbd_backlight/brightness
echo 1 | sudo tee /sys/class/leds/asus::kbd_backlight/brightness
```

第一条命令将关闭键盘灯，第二条命令将开启键盘灯。

这里 tee 最终发挥的作用就是将 0 和 1 写入 brightness 文件（重定向 > 在这里会有权限问题）。/sys 下的文件是设备属性的反映，通过对 /sys 下文件的读写可以查看和控制设备属性。

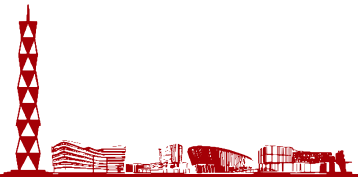
/proc 是进程的映射，例如

```
ls /proc
ps -A
```

第一条命令会看到大量数字命名的目录，例如 248 3533 55991。第二条命令用于查看全部进程。通过对照，我们可以发现，进程号和 /proc 下的数字命名的目录是一一对应的！

/run 用于存放进程的数据，/tmp 用于存放任何临时文件，系统重启后自动清空。

/srv 用于存放本机对外提供的服务的数据（适用于服务器）。



Linux 的权限管理

超级用户

Linux 系统的超级用户称为 root，通常来说，修改家目录外的文件都需要 root 权限，普通用户可以通过 sudo 临时提权以 root 身份执行命令（sudo 的含义是 Superuser Do）

```
sudo apt install build-essential
sudo vim /etc/hosts
```

当使用 sudo 时，你会被要求输入自己的密码（接下来一段时间不用再次输入）。

立志成才 报国裕民

超级用户

Linux 系统中，使用 su 命令（Superuser）可以切换至 root 用户，此时你会被要求输入 root 用户的密码。然而，大部分发行版出于安全原因都禁止直接以 root 用户登录，并且未设置 root 密码，这种情况下，输入任何密码都会被认为是错误的！但有趣的是，你可以使用下面的命令

```
1 sudo su
```

换言之，临时以超级用户身份宣布将自己提升至超级用户，只需要输入自己的密码！

立志成才 报国裕民

超级用户

当然，这并非意味着任何用户都可以轻松登录 root 用户！

因为其实并不是所有用户都有使用 sudo 的权限。通常来说，在一台多人使用的服务器中，只有极少数作为管理员的用户才能使用 sudo 命令。给予 sudo 的权限等同于提供 root 用户！

立志成才 报国裕民

超级用户

在 root 下工作是危险的！引入 sudo 机制的意义在于，让那些可以使用 root 用户的人平常仍然在自己的低权限用户下工作，仅在必要时，显式以 sudo 获得临时提权，避免误操作。

立志成才 报国裕民

权限管理机制

Linux 系统有一套完善的权限管理机制，它可以保证多个用户同时使用，但各个用户又无法触碰他人的数据。即便在只有一个用户的个人电脑上，理解权限管理的概念也是很有价值的。

立志成才 报国裕民

权限管理机制

尝试以下命令

```
mkdir dir
touch file.txt
ln -s dir link
ls -l
```

你应当会看到以下输出

```
drwxrwxr-x 1 liyuxuan liyuxuan    0 Mar  5 10:21 dir
-rw-rw-r-- 1 liyuxuan liyuxuan    0 Mar  5 10:21 file.txt
lrwxrwxrwx 1 liyuxuan liyuxuan    3 Mar  5 10:21 link -> dir
```


立志成才 报国裕民

文件所有权

Linux 中每个文件都有两个所有权的概念：**所有者**（User）、**所属组**（Group）

- 所有者位于 `ls -l` 输出的第三列，该例中所有者为 `liyuxuan`
- 所属组位于 `ls -l` 输出的第四列，该例中所属组为 `liyuxuan`

所属组的概念服务于一个文件需要被多个用户共享的情况。不过，每个用户都会对应一个同名且仅包含其自身的主组，例如，这里的组 `liyuxuan` 就只包含了 `liyuxuan` 这一个用户。

默认情况下，创建文件的用户和其主组会自动成为文件的所有者和所属组。

立志成才 报国裕民

文件权限

Linux 中的权限有三类：**可读**（r）、**可写**（w）、**可执行**（x）

- r：最高位（r=4），读取文件，列出目录中的文件，“可读”。
- w：中间位（w=2），修改文件，增删目录中的文件，“可写”。
- x：最低位（x=1），执行文件，进入目录，“可执行”。

例如，rwx 代表可读可写可执行（记为 7），r-- 代表只读（记为 4）。

在 Linux 中，执行文件是通过权限而非后缀区分的！例如脚本 test.py 只有在添加执行权限之后才能以 ./test.py 的方式运行，否则就只能显式调用解释器 python test.py 运行。

文件权限

Linux 中每个文件对于三种用户角色（所有者、所属组、其他人）都会定义 rwx 权限

- rw-rw-r--，所有者 rw-，所属组 rw-，其他人 r--，记为 664。
- rwxrwxr-x，所有者 rwx，所属组 rwx，其他人 r-x，记为 775。
- rwxrwxrwx，所有者 rwx，所属组 rwx，其他人 rwx，记为 777。

实际上 664、775、777 分别是文件、目录、软链接的默认权限。

默认情况下创建的文件，对所有者和所属组是可读可写，对其他人可读。目录一般应当是可以进入的，所以均增加了可执行。软链接的权限没有意义，实际取决于被链接文件的权限。

立志成才 报国裕民

文件权限

既然默认对其他人是可读的，那每个用户的隐私如何保证？答案在家目录的权限

```
1 ls -l /home
```

注意到家目录 `liyuxuan` 的权限是 `750` 而非默认的 `775`，这保护了下面的文件！

立志成才 报国裕民

工作目录的创建尝试

尽管用户应当在自己的家目录 ~ 即 /home/liyuxuan 下工作，但实践中 ~ 下往往会产生许多杂乱的用户配置文件和用户安装软件。如果有 root 权限，也可以选择额外创建一个和家目录平级的 /home/workspace 作为自己专用的工作目录（/home 下创建目录需要 root 权限）

```
1 sudo mkdir workspace
```

接下来要做两件事：将目录的所有权转给自己，将目录的权限设置为 750。

chown 用于修改所有权（Change Ownership），冒号前后分别指定所有者和所属组

```
1 sudo chown liyuxuan:liyuxuan workspace
```

由于在执行命令前，该目录还属于 root，故需加上 sudo 进行提权。

命令 chmod

chmod 用于修改权限（Change Mode），可以直接用数字标识

```
chmod 750 workspace
```

chmod 的用法很灵活，例如为 test.py 添加执行权限

```
chmod +x test.py
```

谨慎处理权限问题！遇到权限不足时随意设置 sudo chmod 777 是不负责任的行为。

立志成才 报国裕民

Thank You