

Linux 采用单一根目录 / 的树状结构，不像 Windows 有 C:、D: 盘。所有文件和目录都从 / 开始向下展开。

/ 根目录

- Linux 文件系统的起点
- 所有目录都是它的子目录

/home 普通用户的主目录

- 每个用户一个子目录，例如：/home/username
- 你的代码、数据、conda 环境、实验结果通常都放在这里
- ~ 表示当前用户的 home 目录

/usr 存放系统级软件和库

- /usr/bin: 系统命令
- /usr/local: 手动安装的软件

`cd ./home` 切换工作目录

`ls` 显示目录列表

`pwd` 以绝对路径的方式显示用户当前工作目录（用于查找数据集的绝对路径）

`cat file` 显示文件的内容

`gedit file` 编辑文件

`mkdir data` 创建 data 文件夹

`rm -rf data` 移除 data 文件夹

`cp demo.py ./model` 复制文件 demo.py 到 model 文件夹

`mv demo.py ./model` 移动文件 demo.py 到 model 文件夹

`chmod 777 data` 修改 data 文件夹读写权限

`chown` 修改 data 文件夹所有者

`ps aux` 列出所有进程的详细信息

`grep` 筛选命令

`watch 'ps aux | grep python'` 监测 python 代码进程

`kill ***` 以 ps aux 展示的进程号杀掉执行中的进程

`df -h` 查看磁盘空间

`du -h` 查看文件夹大小

`tmux new -s <session-name>` 新建的 tmux 主窗口，确保程序本地电脑关机仍能运行

`tmux ls` 查看 tmux 主窗口

`tmux attach -t <session-name>` 恢复 tmux 主窗口

`scp <username>@<ip>:src_dir dst_dir` 从远程服务器往本地服务器 copy 文件

`scp -r <username>@<ip>:src_dir dst_dir` 递归拷贝

`scp src_dir <username>@<ip>:dst_dir` 从本地服务器往远程服务器 copy 文件

`scp -r src_dir <username>@<ip>:dst_dir` 递归拷贝

```
for file in "*.*"; do mv "$file" "${file/.xxx/_123.xxx}"; done    文件批量重命名  
zip -r dst_zip_file src_dir  压缩  
unzip zip_file  解压  
tar -cvf dst.tar src_dir/  压缩  
tar -xvf tar_file  解压  
  
conda create -n env_name python=3.10  #创建基于 python3.10 的 conda 环境，并且安装依赖包。  
conda activate env_name  切换到新环境  
conda deactivate  退出当前环境
```

which python 查找当前使用的命令的绝对路径

which pip

sudo apt install curl 安装

sudo apt purge curl 卸载

~/python -m pip install --upgrade torch torchvision 更新 PyTorch

~/python -m pip install package_name 将模块安装到当前 python 解释器环境下

~/python -m pip uninstall package_name 卸载当前 python 解释器环境下的某个安装包

安装某个模块时，添加清华源镜像：

~/python -m pip install -i https://pypi.tuna.tsinghua.edu.cn/simple tensorflow-gpu==1.4.1

export 设置环境变量

gedit ~/.bashrc

export PATH=/path/to/your/dir:\$PATH 加到 PATH 开头；

export PATH=\$PATH:/path/to/your/dir 加到 PATH 末尾；

source ~/.bashrc 更新环境变量