

# Linux Cheatsheet

## File Management (文件管理)

列出所有文件(含隐藏)、显示大小\时间

```
ls -lha
```

树形显示目录 (显示大小 限制深度 显示隐藏文件)

```
tree -h -L -a 2
```

递归复制目录 (-i 交互, -b 备份)

```
cp -r source/ dest/
```

强制移动/重命名 (-b 备份)

```
mv -f source dest
```

强制递归删除目录

```
rm -rf dir/
```

创建软链接 (快捷方式)

```
ln -s src softlink
```

创建硬链接

```
ln src hardlink
```

添加执行权限

```
chmod +x script.sh
```

递归修改权限 (所有者全权, 其他只读)

```
chmod -R 755 dir/
```

查找当前目录下的 .py 文件

```
find . -name "*.py"
```

查找大于 100M 的文件

```
find . -type f -size +100M
```

找文件名含 log 的文件(不区分大小写)

```
find . -iname "*log*"
```

## Text Processing (文本处理)

查看文件并显示行号

```
cat -n file
```

实时追踪文件更新 (看日志神器)

```
tail -f access.log
```

查看前 10 行

```
head -n 10 file
```

按冒号分隔, 提取第1列

```
cut -d: -f1 文件名
```

排序 -> 统计频率 -> 按频率降序

```
sort | uniq -c | sort -nr
```

查找包含 ERROR 的行

```
grep "ERROR" app.log
```

实时监控并过滤日志

```
tail -f log | grep "ERROR"
```

## Disk & Archive (磁盘归档)

解压 .tar.gz 到当前目录

```
tar -zxvf file.tar.gz
```

将目录压缩为 .tar.gz

```
tar -zcvf bk.tar.gz dir/
```

查看当前文件夹总大小

```
du -sh .
```

查看当前目录下每个文件/文件夹的大小

```
du -sh *
```

按大小排序当前目录下每个文件/文件夹的大小

```
du -sh * | sort -hr
```

查看磁盘分区挂载与剩余空间 (宏观)

```
df -h
```

比较文件差异 (Git 风格)

```
diff -u old.py new.py
```

左右分栏对比, 直观对比修改

```
diff -y a.txt b.txt
```

递归比较目录差异 (仅列出不同文件)

```
diff -qr dir1/ dir2/
```

## Process (系统监控)

实时监控 (进入后按键: P=CPU排序 M=内存排序 N=PID排序 c=完整命令 k+PID=杀进程 q=退出)

```
top
```

只看 root 用户的进程

```
top -u root
```

top 的增强版, 能用鼠标操作

```
htop
```

结束进程

```
kill PID
```

## Remote (远程与传输)

远程登录 (-X 转发图形界面)

```
ssh -p 22 user@host
```

上传文件 (私有传输)

```
scp -v file user@host:/path
```

下载文件 (私有传输)

```
scp -v user@host:/path/file .
```

增量同步目录 (推荐, 支持断点续传)

```
rsync -avzP dir/ user@host:/path
```

断点续传下载文件 (公开链接)

```
wget -c URL
```

## Network Diag (网络排查)

查看 IP 地址 (替代 ifconfig)

```
ip -br a
```

查看路由表

```
ip r
```

DNS 解析查询 (+trace 看完整路径)

```
dig google.com
```

连通性测试 (发4个包)

```
ping -c 4 google.com
```

动态显示路径丢包率 (Ping+Traceroute)

```
mtr google.com
```

查看监听端口及进程 (替代 netstat)

```
ss -tunlp
```

扫描/探测端口开放情况

```
nc -zv 1.2.3.4 80
```

查看出口 IP 详情 (可以用于测试代理)

```
curl -v cip.cc
```

指定 Host 头测试虚拟主机 (-i 显示响应头) (-v 显示详细信息)

```
curl -i -H "Host: a.com" IP
```

抓取目标 IP 的包 (-w 保存 .pcap)

```
tcpdump -i eth0 dst 1.2.3.4
```

查看防火墙状态

```
ufw status
```

拒绝所有入站连接

```
sudo ufw default deny incoming
```

允许所有出站连接

```
sudo ufw default allow outgoing
```

允许 SSH 端口 (或 sudo ufw allow 22)

```
sudo ufw allow ssh
```

允许 HTTP

```
sudo ufw allow 80/tcp
```

允许 HTTPS

```
sudo ufw allow 443/tcp
```

仅允许指定 IP 访问本机 3306 端口

```
sudo ufw allow from 192.168.1.100 to any port 3306
```

启用防火墙 (先确保已允许 SSH)

```
sudo ufw enable
```

查看带编号的规则列表

```
sudo ufw status numbered
```

删除编号为 2 的规则 (根据上一步编号)

```
sudo ufw delete 2
```

图形化界面配置网络 (Wifi/IP)

```
nmtui
```

应用 YAML 网络配置, Yaml 文件通常保存在 /etc/netplan 文件夹当中

```
netplan apply
```

命令行连接 Wifi (替换 SSID 和 PASS)

```
nmcli dev wifi connect "SSID" password "PASS"
```

## Proxy & Env (代理环境)

设置终端临时代理 (http/https/all)

```
export http_proxy=http://127.0.0.1:7890
```

取消终端代理

```
unset http_proxy all_proxy
```

设置 Git 代理

```
git config --global http.proxy ...
```

Pip 安装使用代理

```
pip install ... --proxy ...
```

设置 Conda 代理 (同时设置 https 同理)

```
conda config --set proxy_servers.http http://127.0.0.1:7890
```

## Hardware (系统与硬件)

查看 OS 发行版本号

```
cat /etc/os-release
```

查看内核版本

```
uname -r
```

查看 CPU 架构/核数

```
lscpu
```

查看内存使用情况

```
free -h
```

查看 GPU 显存/进程

```
nvidia-smi
```

更强大的 GPU 实时监控工具

```
nvitop
```

查看 CUDA 编译器版本

```
nvcc -V
```

查看硬盘/分区树状结构

```
lsblk
```

查看 USB 设备列表

```
lsusb
```