Linux 常用命令速查表（带简单使用场景与示例）

文件与目录操作

- `ls`
  场景：查看当前目录下有哪些文件和文件夹。
  示例：`ls -l`（详细列表，显示权限、大小、时间等）
- `cd`
  场景：切换到其他目录工作。
  示例：`cd /var/log`（进入系统日志目录）
  `cd ..`（返回上级目录）
  `cd ~`（回到自己的家目录）
- `pwd`
  场景：不确定自己当前在哪个目录时使用。
  示例：`/home/user/projects`
- `mkdir`
  场景：创建一个新文件夹。
  示例：`mkdir backup`（创建名为 backup 的文件夹）
- `cp`
  场景：备份文件或复制到其他目录。
  示例：`cp important.txt important_backup.txt`（复制文件）
  `cp -r project/ project_backup/`（递归复制整个目录）
- `mv`
  场景：移动文件或重命名。
  示例：`mv oldname.txt newname.txt`（重命名）
  `mv file.txt /tmp/`（移动到 /tmp 目录）
- `rm`
  场景：删除不再需要的文件或目录。
  示例：`rm temp.txt`（删除文件）
  `rm -r old_project/`（删除整个目录，慎用！）
  `rm -rf /`（极端危险，永远不要运行）
- `touch`
  场景：快速创建一个空文件，或更新文件时间戳。
  示例：`touch newfile.txt`（创建空文件）
- `cat`
  场景：快速查看小文件的全部内容。
  示例：`cat /etc/os-release`（查看系统版本信息）
- `less`
  场景：查看大文件，支持上下翻页搜索。
  示例：`less big.log`（按 q 退出）
- `head` / `tail`
  场景：只看文件开头或结尾几行，常用于查看日志。

示例：`tail -n 20 access.log`（看最后20行）
`tail -f error.log`（实时监控日志变化，常用于调试）

搜索命令

- `grep`
  场景：在文件或命令输出中查找特定关键词。
  示例：`grep "error" app.log`（查找包含 error 的行）
  `ps aux | grep python`（查找运行中的 Python 进程）
- `find`
  场景：根据名字、大小、时间等条件查找文件。
  示例：`find /home -name "*.txt"`（查找家目录下所有 .txt 文件）
  `find . -mtime -7`（查找7天内修改过的文件）

系统信息

- `df -h`
  场景：磁盘快满了，想查看剩余空间。
  示例：显示所有分区使用情况（人性化单位）
- `du -sh *`
  场景：当前目录哪个文件夹占空间最大。
  示例：`du -sh /var/log/*`（查看日志目录下各文件大小）
- `free -h`
  场景：检查内存是否足够。
  示例：显示已用/可用内存
- `top`
  场景：实时查看哪个进程占用 CPU/内存最多。
  示例：运行后按 q 退出，按 1 查看每个 CPU 核心使用率

进程管理

- `ps aux`
  场景：列出当前所有进程。
  示例：`ps aux | grep nginx`（查找 nginx 进程）
- `kill`
  场景：某个程序卡死或不需要了，想结束它。
  示例：先用 `ps` 找到 PID 为 1234 的进程 → `kill 1234`（温和结束）
  若无效 → `kill -9 1234`（强制杀死）
- `Ctrl + C`
  场景：命令行运行的程序（如 ping、死循环脚本）想立刻停止。
  作用：直接在终端按下，发送 SIGINT 信号终止前台进程。
- `Ctrl + Z`
  场景：暂时暂停前台程序，想稍后继续。
  后续：`bg` 让它后台运行，或 `fg` 重新调回前台。

权限管理

- `chmod`
  场景：让脚本可以执行，或限制别人访问文件。
  示例：`chmod +x myscript.sh`（添加执行权限）
  `chmod 600 private.key`（只有所有者可读写）
- `chown`
  场景：把文件的所有者改给其他人（通常需要 sudo）。
  示例：`sudo chown www-data:www-data /var/www/html`

压缩与解压

- `tar`
  场景：备份目录或传输大文件。
  示例：`tar -czf backup.tar.gz /home/user/project`（压缩）
  `tar -xzf backup.tar.gz`（解压到当前目录）
- `zip` / `unzip`
  场景：和 Windows 用户交换文件时常用。
  示例：`zip -r archive.zip folder/`（压缩文件夹）
  `unzip archive.zip`（解压）

网络命令

- `ping`
  场景：检查网络是否通畅。
  示例：`ping baidu.com`（Ctrl+C 停止）
- `wget`
  场景：直接在终端下载文件。
  示例：`wget https://example.com/file.iso`
- `curl`
  场景：测试 API、下载文件或提交数据。
  示例：`curl -O https://example.com/file.txt`

包管理（以 Ubuntu 为例）

- `apt`
  场景：安装软件、更新系统。
  示例：`sudo apt update`（更新软件源）
  `sudo apt install nginx`（安装 nginx）
  `sudo apt upgrade`（升级所有软件包）

Vim 基本操作（图片中列出的常用命令）

场景：在服务器上编辑配置文件（如 nginx.conf、.bashrc）时常用。

- 进入编辑模式：按 `i`
- 保存并退出：按 Esc → `:wq` → Enter

- 强制退出不保存：按 Esc → `:q!` → Enter
- 常用移动：`h` 左、`j` 下、`k` 上、`l` 右
- 删除当前行：`dd`
- 撤销：`u`
- 搜索：`/` + 关键词 → Enter，`n` 下一个

其他实用技巧

- `sudo !!`
  场景：运行命令忘记加 sudo，想用上一条命令补上。
  效果：用管理员权限重新执行上一条命令。
- `history | grep cd`
  场景：想找之前执行过的某个命令。
- `man <命令>`
  场景：不记得命令具体用法时查看手册。
  示例：`man ls`

> **提醒**：`Ctrl + C` 是最常用的中断方式，能快速终止正在运行的前台命令（如无限循环的测试程序）。关闭终端窗口本身不会杀死进程，需小心后台进程残留。

Linux 危险操作安全提示

总结

- 在执行任何会修改、删除、格式化或影响系统运行的命令前，**三思并核对路径**。
- **优先备份**：重要数据先备份（快照、外部存储或版本控制）。
- **最小权限原则**：尽量避免以 root 或 sudo 身份执行操作；在必要时短期提升权限。
- 使用 `--dry-run` / 先 `ls` / 用 `echo` 测试变量展开结果。把危险命令分解并先在非生产环境验证。

常见危险命令与注意事项

下面按类别列出常见危险操作，说明危险原因并给出更安全的替代方法。

1) 删除类（`rm` 相关）

- 危险示例：
  - `rm -rf /path/to/dir`、`rm -rf *`、`rm -rf $DIR`（当变量为空时会变成 `rm -rf` 当前目录）
  - `rm -f /` 或带 `--no-preserve-root`（会尝试删除根文件系统）
- 危险原因：不可恢复、容易误删重要文件或系统文件。
- 安全做法：
  - 禁止使用该命令！！！
  - 禁止使用该命令！！！
  - 禁止使用该命令！！！
  - 禁止使用该命令！！！
  - 禁止使用该命令！！！

2) 覆盖与重定向（`>`、`>>`、`tee`、`echo`）

- 危险示例：
  - `echo "..." > /etc/passwd`、`> important.log`、`somecommand > /dev/sda`
  - `command > $FILE` 当 `$FILE` 为空或错误时会覆盖当前目录下某个文件
- 危险原因：重定向会直接覆盖目标文件，造成配置丢失或设备被覆盖。

- 安全做法：
  - 使用 >> 追加而非覆盖（如果意图是追加）。
  - 先 `tee -a file`（追加）或 `tee file` 与 sudo 结合时要谨慎。
  - 在重定向前 `ls -l` 或 `test -n "$FILE"` 检查变量。
  - 对重要配置文件使用版本控制（git）或保存备份副本：`cp file file.bak`。

3) 格式化与分区（mkfs、fdisk、parted）

- 危险示例：`mkfs.ext4 /dev/sdb1`、`mkfs -t xfs /dev/sda`、误选磁盘进行分区操作。
- 危险原因：会擦除整个分区/磁盘数据，通常不可恢复。
- 安全做法：
  - 在操作前用 `lsblk`、`blkid`、`fdisk -l` 等命令**反复确认设备名**与挂载点。
  - 不在生产机器上直接操作真实磁盘，先在虚拟机或测试盘上演练。
  - 如果可能，先卸载（umount）再操作并确保目标不是系统盘。

4) 清零、低级写入（dd、shred）

- 危险示例：`dd if=/dev/zero of=/dev/sda bs=1M`、`dd if=/dev/zero of=/dev/sdb conv=fsync`、`shred -n 1 /dev/sda`。
- 危险原因：会将设备填满零值或随机数据，立即破坏文件系统及分区结构。
- 安全做法：
  - 仅在明确知道目标且已备份时使用，先 `lsblk` 确认。
  - 在脚本中避免直接使用设备名变量，若必须使用，先打印并让人工确认。

5) 终止进程（kill、killall、pkill）

- 危险示例：`kill -9 -1`、`killall -9 processname`、`pkill -u root`。
- 危险原因：可能会终止系统关键进程（包括 init/systemd），导致系统崩溃或重启；kill -9 -1 会杀掉几乎所有可杀进程。
- 安全做法：
  - 先用 `ps aux | grep` 或 `pgrep -a` 确认 pid 与命令行参数。
  - 优先发送温和信号 `kill -15 PID`；只有在无响应时才用 `-9`。

- 避免对 root 用户全部进程执行 `pkill/killall`。
- 在共享服务器上，建议先联系相关用户或管理员。

6) 权限与所有权操作（`chmod`、`chown`、`setfacl`）

- 危险示例：`chmod -R 777 /`、`chown -R user:group /`、错误地改变 `/etc` 下文件属主。
- 危险原因：会暴露敏感文件、破坏安全策略或导致程序无法正常运行。
- 安全做法：
  - 只对特定文件或目录设置最小必要权限；先 `chmod` 单个文件测试后再批量更改。
  - 使用 `--reference` 或 `find -exec` 精确控制。
  - 对系统目录慎重，优先使用 ACL 或 sudoers 管理访问控制。

7) 系统控制命令（`reboot`、`shutdown`、`init`、`systemctl`）

- 危险示例：在公共服务器上随意执行 `reboot`、`shutdown -h now`、或者停用关键服务 `systemctl stop sshd`。
- 危险原因：会中断所有用户的会话与任务，影响他人工作。
- 安全做法：
  - 在执行前广播通知、安排维护窗口并征求同意。
  - 使用 `wall` 或电子邮件通知同一时间的使用者。
  - 如果只是重启单个服务，确认依赖并优先做 graceful 重启：`systemctl restart service`。

8) 非受信任脚本/二进制（`curl|sh`、`wget|sh`）

- 危险示例：`curl http://example.com/install.sh | sh`
- 危险原因：直接执行远程代码可能包含恶意逻辑，导致后门、权限提升或数据泄露。
- 安全做法：
  - 先下载脚本并审查内容：`curl -O URL && less install.sh`，再执行。
  - 只从官方可信源下载、并验证签名或哈希值。

9) 计划任务/自动化脚本（crontab）

- 危险示例：在 crontab 中写入会删除或覆盖文件的脚本而没有日志或锁机制，导致竞争条件或重复执行。
- 危险原因：无监控的自动化任务会在无人察觉的情况下造成数据损失。
- 安全做法：
    - crontab 的脚本中加入日志、错误处理与锁（`flock`），并把输出邮件到管理员。
    - 在部署前本地测试脚本。

10) 环境变量与不安全变量展开

- 危险示例：`rm -rf $MYDIR/*` 当 `$MYDIR` 未定义或为空时风险极大。
- 危险原因：变量未设置或含有空格/特殊字符会改变命令行为。
- 安全做法：
    - 在脚本中使用检查：`[ -n "$MYDIR" ] || { echo "MYDIR empty"; exit 1; }`。
    - 使用双引号保护变量：`"$MYDIR"`。

操作前检查清单（执行危险操作前必须做）

1. 是否有最近的备份或快照？
2. 目标路径或设备名是否已反复确认（使用 `lsblk` / `mount` / `df -h` / `readlink -f`）？
3. 是否在非生产或测试环境先行验证？
4. 是否以非 root 用户测试过相同行为？
5. 是否在命令中加入 `--dry-run` 或先 `echo` 命令结果？
6. 是否通知了受影响的其他用户并获得同意？
7. 是否准备了回滚或恢复方案（备份位置、快照、恢复步骤）？

误操作恢复建议（在发生误删/破坏后）

- 立刻停止对受影响磁盘的写入操作（挂载为只读或卸载），避免覆盖被删除数据块。

- 优先从备份或快照恢复。

- 当无备份时，可尝试数据恢复工具（示例）：`extundelete`（ext 文件系统）、`testdisk`、`photorec`。注意：这些工具不是万无一失，成功率依赖于操作后写入情况。

- 如果疑似磁盘分区表被破坏，可尝试 `gdisk`、`testdisk` 恢复分区表。
