

Linux 常见面试问题

一、Linux 基础概念

1. 什么是 Linux？它与 Windows 的主要区别是什么？
2. Linux 的目录结构中，/bin、/sbin、/usr、/etc、/var 分别用于存放什么文件？
3. 什么是 Shell？常见的 Shell 有哪些？
4. 绝对路径和相对路径的区别是什么？如何表示当前目录和上一级目录？
5. 如何查看 Linux 系统版本和内核版本？

二、文件与目录操作

1. 如何创建一个新目录？如何递归创建多级目录？
2. 复制文件和目录的命令是什么？如何强制覆盖目标文件？
3. 移动/重命名文件的命令是什么？与复制命令有何区别？
4. 删除文件和目录的命令是什么？如何强制删除非空目录？
5. 如何查看文件内容？cat、more、less、tail 命令的区别是什么？
6. 如何查找包含特定字符串的文件？（如查找所有.log 文件中包含“error”的行）
7. 如何在指定目录下查找文件名包含“test”的文件？
8. 如何统计一个文件的行数、单词数、字符数？
9. 如何创建一个空文件？除了 touch 还有其它方法吗？
10. 如何查看文件的权限信息？rwx 分别代表什么？

三、权限管理

1. Linux 文件权限分为哪三类用户？如何通过数字表示权限（如 755、644）？
2. 如何修改文件/目录的所有者和所属组？
3. 如何递归修改目录及其子文件的权限？
4. chmod 和 chown 命令的作用分别是什么？

5. 什么是 SUID、SGID 和 Sticky Bit？它们的作用是什么？

四、进程管理

1. 如何查看系统中运行的进程？ps 和 top 命令的区别是什么？
2. 如何根据进程名或 PID 查找进程？
3. 如何终止一个进程？kill、killall、pkill 命令的区别是什么？
4. 什么是后台进程？如何将一个进程放到后台运行？如何将后台进程调回前台？
5. 如何查看进程的 CPU、内存占用情况？
6. 如何查看进程打开的文件？

五、网络操作

1. 如何查看本机 IP 地址？ifconfig 和 ip 命令的区别是什么？
2. 如何测试与目标主机的连通性？ping 命令的常用参数有哪些？
3. 如何查看端口占用情况？（如查看 8080 端口被哪个进程占用）
4. 如何查看网络连接状态？（如 ESTABLISHED、LISTEN 状态的连接）
5. 如何使用 curl 或 wget 命令测试一个 HTTP 接口的响应？
6. 如何查看 DNS 配置？如何临时修改本机 DNS？

六、日志与压缩

1. Linux 系统的日志文件通常存放在哪个目录？常用的系统日志有哪些？
2. 如何实时查看日志文件的新增内容？（如监控应用的错误日志）
3. 如何按时间、关键字筛选日志？（如筛选今天的“ERROR”日志）
4. 常用的压缩/解压缩命令有哪些？如何压缩/解压缩.tar.gz、.zip 文件？
5. 如何将一个命令的输出保存到文件中？如何同时输出到屏幕和文件？

七、用户与组

1. 如何创建、删除用户？如何修改用户密码？
2. 如何创建、删除用户组？如何将用户加入某个组？
3. 如何查看当前登录的用户？如何查看用户的登录历史？
4. sudo 命令的作用是什么？如何配置用户的 sudo 权限？

八、磁盘与存储

1. 如何查看磁盘分区和使用率？df 和 du 命令的区别是什么？
2. 如何查看某个目录的大小？如何查找磁盘中占用空间最大的文件/目录？
3. 如何挂载/卸载一个外部存储设备（如 U 盘、分区）？
4. 什么是 inode？它与文件有什么关系？

九、Shell 脚本与自动化

1. 如何编写一个简单的 Shell 脚本？脚本的首行#!/bin/bash 有什么作用？
2. 如何给 Shell 脚本添加执行权限并运行？
3. 写出一个 Shell 脚本，实现“每天凌晨 3 点备份指定目录到/backup”的功能。
4. 如何使用 crontab 设置定时任务？举例说明定时任务的格式。
5. 如何在 Shell 脚本中判断一个文件是否存在？

Linux 常见面试问题参考答案

1. 什么是 Linux？它与 Windows 的主要区别是什么？

- **Linux:** 基于 POSIX 的多用户、多任务、支持多线程和多 CPU 的开源类 Unix 操作系统，内核由 Linus Torvalds 主导开发。
- **核心区别:**
 - 开源性: Linux 开源免费，Windows 闭源收费；
 - 架构: Linux 基于 Unix 架构，Windows 有独立内核；
 - 权限管理: Linux 权限体系更严格，Windows 权限控制相对宽松；
 - 用途: Linux 多用于服务器/开发环境，Windows 多用于个人桌面。

2. Linux 的目录结构中，/bin、/sbin、/usr、/etc、/var 分别用于存放什么文件？

- **/bin:** 存放所有用户可执行的基本命令（如 ls、cp），系统启动时必需。

- **/sbin:** 存放仅管理员（root）可执行的系统管理命令（如 ifconfig、reboot）。
- **/usr:** 存放用户应用程序和文件，如/usr/bin（非核心命令）、/usr/lib（库文件）。
- **/etc:** 存放系统所有配置文件（如/etc/passwd、/etc/profile）。
- **/var:** 存放动态变化的文件，如日志（/var/log）、缓存（/var/cache）、邮件（/var/mail）。

3. 什么是 Shell？常见的 Shell 有哪些？

- **Shell:** Linux 内核与用户之间的交互接口，接收用户命令并翻译成内核可执行的指令。
- **常见 Shell:**
 - bash (Bourne Again Shell)：默认 Shell，兼容 Bourne Shell；
 - sh (Bourne Shell)：最早期的 Shell；
 - zsh (Z Shell)：功能更丰富，支持自动补全；
 - csh (C Shell)：语法类似 C 语言。

4. 绝对路径和相对路径的区别是什么？如何表示当前目录和上一级目录？

- **区别:**
 - 绝对路径：从根目录（/）开始的完整路径（如 /home/user/file.txt）；
 - 相对路径：从当前目录开始的路径（如 ./file.txt、../docs）。
- **当前目录:** 用 . 表示；
- **上一级目录:** 用 .. 表示。

5. 如何查看 Linux 系统版本和内核版本？

- **查看系统版本:**
 - cat /etc/os-release（通用，显示发行版信息如 Ubuntu/CentOS）；
 - lsb_release -a（需安装 lsb-core，显示详细版本）。

- **查看内核版本：**

- `uname -r`（显示内核版本号，如 5.4.0-100-generic）；
- `uname -a`（显示内核版本+系统架构+主机名等完整信息）。

6. 如何创建一个新目录？如何递归创建多级目录？

- **创建单个目录：** `mkdir` 目录名（如 `mkdir test`）；
- **递归创建多级目录：** `mkdir -p` 目录路径（如 `mkdir -p /home/user/docs/2024`，自动创建不存在的父目录）。

7. 复制文件和目录的命令是什么？如何强制覆盖目标文件？

- **复制命令：** `cp`（copy）；
 - 复制文件： `cp` 源文件 目标路径（如 `cp file.txt /tmp`）；
 - 复制目录：需加 `-r`（递归）参数， `cp -r dir1 /tmp`。
- **强制覆盖：**加 `-f` 参数， `cp -f` 源文件 目标文件（忽略覆盖提示）。

8. 移动/重命名文件的命令是什么？与复制命令有何区别？

- **命令：** `mv`（move）；
 - 重命名： `mv oldname.txt newname.txt`（同一目录）；
 - 移动： `mv file.txt /tmp`（不同目录）。
- **与 `cp` 的区别：**
 - `cp`：复制文件，源文件保留；
 - `mv`：移动/重命名文件，源文件被删除（本质是修改文件的 inode 指向）。

9. 删除文件和目录的命令是什么？如何强制删除非空目录？

- **删除命令：** `rm`（remove）；
 - 删除文件： `rm file.txt`；
 - 删除空目录： `rm -d dir1`（或 `rmdir dir1`）。
- **强制删除非空目录：** `rm -rf` 目录名（`-r` 递归删除子内容，`-f` 强制忽略提示）；
 - 示例： `rm -rf /tmp/old_dir`。

10. 如何查看文件内容？cat、more、less、tail 命令的区别是什么？

- 查看文件内容的命令：cat、more、less、tail、head 等。
- 区别：

命令	功能特点	适用场景
cat	一次性显示全部内容（适合小文件）	查看短文件、合并文件
more	分页显示（仅向下翻页，按 q 退出）	查看长文件，无回退功能
less	分页显示（支持上下翻页、搜索，按 q 退出）	查看长文件，功能最灵活
tail	默认显示文件末尾 10 行，支持实时监控	查看日志（如 tail -f log.txt 实时跟踪新增内容）

11. 如何查找包含特定字符串的文件？（如查找所有.log 文件中包含“error”的行）

- 核心命令：grep（全局搜索正则表达式并打印）；
- 示例：查找当前目录及子目录下所有.log 文件中包含“error”的行：

```
grep -r "error" --include="*.log" ./
```

- -r：递归搜索子目录；
- --include="*.log"：仅匹配.log 后缀的文件。

12. 如何在指定目录下查找文件名包含“test”的文件？

- 核心命令：find（按文件名/属性等查找文件）；
- 示例：在/home 目录下查找文件名包含“test”的文件：

```
find /home -name "*test*"
```

- -name "*test*"：模糊匹配（*表示任意字符）；
- 若忽略大小写，用-iname "*test*"。

13. 如何统计一个文件的行数、单词数、字符数？

- 核心命令：wc（word count）；

- 常用参数：
 - -l: 统计行数 (line)；
 - -w: 统计单词数 (word)；
 - -c: 统计字符数 (byte)；
- 示例：统计 file.txt 的行数、单词数、字符数：

```
wc -lwc file.txt # 依次输出：行数 单词数 字符数 文件名
```

14. 如何创建一个空文件？除了 touch 还有别的方法吗？

- 方法 1 (touch)：touch 文件名（如 touch empty.txt，若文件已存在则更新修改时间）；
- 其他方法：
 - 重定向空输出：> empty.txt（覆盖已有文件，慎用）；
 - 新建文件并退出（不写入内容）：vim empty.txt（打开后按:wq 保存退出）；
 - 管道空输出：cat /dev/null > empty.txt（/dev/null 为“空设备”）。

15. 如何查看文件的权限信息？rwx 分别代表什么？

- 查看权限：ls -l 文件名（如 ls -l file.txt），输出格式示例：

```
-rwxr-xr-- 1 user group 1234 May 20 10:00 file.txt
```

- 第一列（-rwxr-xr--）即为权限信息。
- rwx 含义：
 - r (read)：读权限（数字表示为 4）；
 - w (write)：写权限（数字表示为 2）；
 - x (execute)：执行权限（数字表示为 1）；
 - -: 无对应权限（数字表示为 0）。

16. Linux 文件权限分为哪三类用户？如何通过数字表示权限？

- 三类用户（权限列的 9 位字符分为 3 组，每组 3 位）：
 1. 所有者 (Owner)：文件的创建者（权限列第 1-3 位，如 rwx）；

- 2. 所属组 (Group)：所有者所在的用户组（权限列第 4-6 位，如 r-x）；
- 3. 其他用户 (Others)：除所有者和所属组外的所有用户（权限列第 7-9 位，如 r--）。
- **数字表示权限：**每组权限的 rwx 对应数字之和（r=4、w=2、x=1）；
 - 示例：rwxr-xr-- → 所有者 7（4+2+1）、所属组 5（4+0+1）、其他 4（4+0+0），即 754。

17. 如何修改文件/目录的所有者和所属组？

- **修改所有者：**chown 新所有者 文件名（需 root 权限）；
 - 示例：将 file.txt 的所有者改为 user1：chown user1 file.txt。
- **修改所属组：**
 - 方法 1：chgrp 新所属组 文件名（如 chgrp group1 file.txt）；
 - 方法 2：chown :新所属组 文件名（如 chown :group1 file.txt）。
- **同时修改所有者和所属组：**chown 新所有者:新所属组 文件名（如 chown user1:group1 file.txt）。

18. 如何递归修改目录及其子文件的权限？

- 核心命令：chmod（修改权限），加-R（递归）参数；
- 示例 1（符号权限）：将/home/docs 目录及子文件的所有者权限设为 rwx，所属组和其他用户设为 r-x：

```
chmod -R u=rwx,g=rx,o=rx /home/docs
```

- 示例 2（数字权限）：将/home/docs 目录及子文件权限设为 755：

```
chmod -R 755 /home/docs
```

19. chmod 和 chown 命令的作用分别是什么？

- **chmod：**全称“change mode”，用于修改文件/目录的**权限**（如 rwx）；
 - 示例：chmod 755 file.txt。
- **chown：**全称“change owner”，用于修改文件/目录的**所有者和所属组**；
 - 示例：chown user1:group1 file.txt。

20. 什么是 SUID、SGID 和 Sticky Bit？它们的作用是什么？

- 三者均为“特殊权限”，在权限列的第 1 位（常规权限前）表示：
 - **SUID** (Set User ID)：权限位表示为 s（所有者执行位），用户执行文件时临时获得文件所有者的权限；
 - 示例：/bin/passwd（普通用户执行时临时获得 root 权限，以修改/etc/passwd）。
 - **SGID** (Set Group ID)：权限位表示为 s（所属组执行位），用户执行文件时临时获得文件所属组的权限；若用于目录，新创建的文件/目录会继承该目录的所属组。
 - **Sticky Bit** (粘滞位)：权限位表示为 t（其他用户执行位），仅对目录有效；目录内的文件仅所有者、root 可删除，其他用户即使有写权限也无法删除；
 - 示例：/tmp 目录（默认权限 1777，1 表示粘滞位）。

21. 如何查看系统中运行的进程？ps 和 top 命令的区别是什么？

- 查看进程的命令：ps、top、htop（需安装）等。
- ps 与 top 的区别：

命令	功能特点	适用场景
ps	静态快照（仅显示执行命令时的进程状态）	查看特定时刻的进程信息（如 ps aux 查看所有进程）
top	动态实时更新（默认每 3 秒刷新，按 q 退出）	监控进程的 CPU、内存占用变化，排查资源占用问题

22. 如何根据进程名或 PID 查找进程？

- 根据进程名查找：
 - ps aux | grep 进程名（如 ps aux | grep nginx）；
 - pgrep 进程名（直接输出匹配进程的 PID，如 pgrep nginx）。
- 根据 PID 查找：
 - ps -p PID（如 ps -p 1234，查看 PID 为 1234 的进程）；

- top -p PID（实时监控指定 PID 的进程）。

23. 如何终止一个进程？kill、killall、pkill 命令的区别是什么？

- **终止进程的核心原理：**向进程发送信号（默认发送 15 号信号 SIGTERM，请求进程正常退出；若无效，用 9 号信号 SIGKILL 强制终止）。
- **命令区别：**

命令	功能特点	示例
kill	需指定 PID，仅终止单个进程	kill 1234（发送 SIGTERM）、 kill -9 1234（强制终止）
killall	需指定 进程名 ，终止所有匹配名称的进程	killall nginx（终止所有 nginx 进程）
pkill	需指定 进程名/条件 ，终止所有匹配的进程（类似 pgrep + kill）	pkill nginx（终止所有 nginx 进程）

24. 什么是后台进程？如何将一个进程放到后台运行？如何将后台进程调回前台？

- **后台进程：**不占用终端输入输出的进程，终端可继续执行其他命令（进程状态标记为&）。
- **将进程放到后台：**
 - 1. 启动时直接后台运行：命令后加&，如 nginx &;
 - 2. 已前台运行的进程：按 Ctrl + Z 暂停进程，再用 bg（background）命令恢复到后台（如 bg %1，%1 表示第 1 个后台进程）。
- **将后台进程调回前台：**fg %进程编号（如 fg %1，调回第 1 个后台进程）；若只有一个后台进程，直接输入 fg。

25. 如何查看进程的 CPU、内存占用情况？

- **实时查看：**
 - top：默认按 CPU 占用排序，显示所有进程的 CPU、内存占比（%Cpu、%Mem 列）；
 - htop：top 的增强版，界面更友好（需安装）。
- **静态查看：**

- ps aux: %cpu 列显示 CPU 占用率, %mem 列显示内存占用率;
- ps -p PID -o %cpu,%mem: 查看指定 PID 的 CPU 和内存占用 (如 ps -p 1234 -o %cpu,%mem)。

26. 如何查看进程打开的文件?

- 核心命令: lsof (list open files), 需 root 权限;
- 示例 1: 查看 PID 为 1234 的进程打开的所有文件:

```
lsof -p 1234
```

- 示例 2: 查看 nginx 进程打开的文件:

```
lsof -c nginx # -c 按进程名前缀匹配
```

27. 如何查看本机 IP 地址? ifconfig 和 ip 命令的区别是什么?

- 查看 IP 地址的命令:
 - 传统命令: ifconfig (需安装 net-tools 包, 部分系统默认不自带);
 - 现代命令: ip addr (或 ip a, 属于 iproute2 工具集, 默认自带)。
- ifconfig 与 ip 的区别:

命令	所属工具集	功能支持	现状
ifconfig	net-tools	仅支持 IPv4, 功能较旧	逐步被淘汰, 需手动安装
ip	iproute2	支持 IPv4/IPv6、路由、隧道等, 功能全面	主流推荐, 系统默认自带

28. 如何测试与目标主机的连通性? ping 命令的常用参数有哪些?

- 测试连通性: ping 命令 (基于 ICMP 协议, 发送 echo 请求包);
 - 示例: 测试与 www.baidu.com 的连通性: ping www.baidu.com。
- 常用参数:
 - -c 次数: 指定发送包的次数 (如 ping -c 3 www.baidu.com, 发送 3 次后停止);

- `-s` 字节数：指定数据包大小（如 `ping -s 1024 www.baidu.com`，发送 1024 字节的包）；
- `-i` 间隔：指定发送包的时间间隔（单位：秒，如 `ping -i 2 www.baidu.com`，每 2 秒发一次）。

29. 如何查看端口占用情况？（如查看 8080 端口被哪个进程占用）

- 核心命令：`netstat`（旧）或 `ss`（新），配合 `grep`；
- 示例 1（`ss` 命令，推荐）：查看 8080 端口的占用情况：

```
ss -tulnp | grep 8080
```

- `-t`：TCP 端口；`-u`：UDP 端口；`-l`：监听中端口；`-n`：显示 IP/端口（不解析域名）；`-p`：显示占用进程。

- 示例 2（`netstat` 命令，需安装 `net-tools`）：

```
netstat -tulnp | grep 8080
```

30. 如何查看网络连接状态？（如 ESTABLISHED、LISTEN 状态的连接）

- 核心命令：`ss` 或 `netstat`，过滤连接状态；
- 示例 1（查看所有 TCP 连接状态）：

```
ss -t state all
```

- 示例 2（查看 ESTABLISHED（已建立）和 LISTEN（监听中）状态的 TCP 连接）：

```
ss -t state ESTABLISHED,LISTEN
```

- 示例 3（`netstat` 命令）：

```
netstat -tna | grep -E "ESTABLISHED|LISTEN"
```

- `-a`：显示所有连接（包括监听和非监听）；`-t`：TCP 连接；`-n`：不解析域名。

31. 如何使用 curl 或 wget 命令测试一个 HTTP 接口的响应？

- **curl 命令**（发送 HTTP 请求，显示响应内容）：
 - 发送 GET 请求：`curl https://api.example.com/test`（默认 GET）；

- 显示详细请求/响应头: `curl -v https://api.example.com/test`;
- 发送 POST 请求 (带参数): `curl -X POST -d "name=test" https://api.example.com/login`。
- **wget 命令** (下载 HTTP 响应内容到文件, 默认不显示响应体):
 - 下载响应到文件: `wget https://api.example.com/test -O response.txt` (-O 指定输出文件);
 - 显示详细日志: `wget -d https://api.example.com/test` (-d 开启调试模式)。

32. 如何查看 DNS 配置? 如何临时修改本机 DNS?

- **查看 DNS 配置:**
 - 方法 1: 查看 `/etc/resolv.conf` 文件 (DNS 服务器列表, `nameserver` 开头的行);
 - 示例: `cat /etc/resolv.conf` → `nameserver 8.8.8.8`。
 - 方法 2: `nmcli dev show | grep DNS` (NetworkManager 管理的 DNS, 适用于图形化系统)。
- **临时修改 DNS:** 直接编辑 `/etc/resolv.conf`, 添加 `nameserver` DNS 服务器 IP;
 - 示例: `echo "nameserver 114.114.114.114" >> /etc/resolv.conf`;
 - 注意: 重启网络服务后可能失效, 永久修改需配置网络管理工具 (如 `/etc/network/interfaces` 或 `nmcli`)。

33. Linux 系统的日志文件通常存放在哪个目录? 常用的系统日志有哪些?

- **日志文件存放目录:** `/var/log` (所有系统和应用日志默认存放于此)。
- **常用系统日志:**
 - `/var/log/messages`: 核心系统日志 (如内核、进程启动/停止信息, CentOS/RHEL 系统);
 - `/var/log/syslog`: 核心系统日志 (Ubuntu/Debian 系统, 类似 `messages`);
 - `/var/log/auth.log` (Ubuntu) // `/var/log/secure` (CentOS): 认证日志 (如 SSH 登录、sudo 操作);

- /var/log/dmesg：内核启动日志（硬件检测、驱动信息）；
- /var/log/boot.log：系统启动过程日志。

34. 如何实时查看日志文件的新增内容？（如监控应用的错误日志）

- 核心命令：tail -f（-f 表示“follow”，实时跟踪文件新增内容）；
- 示例：实时监控/var/log/app/error.log 的新增日志：

```
tail -f /var/log/app/error.log
```

- 进阶：跟踪多个日志文件：tail -f /var/log/app/error.log /var/log/app/access.log。

35. 如何按时间、关键字筛选日志？（如筛选今天的“ERROR”日志）

- 按关键字筛选：grep 命令；
 - 示例：筛选 error.log 中包含“ERROR”的行：grep "ERROR" /var/log/app/error.log。
- 按时间筛选（需日志包含时间格式，如 2024-05-20 14:30:00）：
 - 方法 1：grep "2024-05-20" /var/log/app/error.log | grep "ERROR"（先筛选今天的日志，再筛选“ERROR”）；
 - 方法 2：sed 命令按时间范围筛选（如筛选 2024-05-20 14:00-15:00 的日志）：

```
sed -n '/2024-05-20 14:/,/2024-05-20 15:/p' /var/log/app/error.log
```

36. 常用的压缩/解压缩命令有哪些？如何压缩/解压缩.tar.gz、.zip 文件？

- 常用压缩格式及命令：

压缩格式	压缩命令	解压缩命令
.tar.gz（tar 压缩+gzip）	tar -zcvf 压缩包.tar.gz 源文件/目录	tar -zxvf 压缩包.tar.gz -C 目标目录
.zip	zip 压缩包.zip 源文件/目录（需安装 zip）	unzip 压缩包.zip -d 目标目录（需安装 unzip）

压缩格式	压缩命令	解压缩命令
.gz（仅文件）	gzip 源文件（压缩后删除源文件）	gunzip 压缩包.gz

- **参数说明（tar 命令）：**
 - -z：用 gzip 压缩/解压缩；
 - -c：创建压缩包；
 - -x：解压缩；
 - -v：显示详细过程；
 - -f：指定压缩包文件名；
 - -C：指定解压缩的目标目录。

37. 如何将一个命令的输出保存到文件中？如何同时输出到屏幕和文件？

- **将输出保存到文件：**使用重定向符号；
 - >：覆盖写入文件（如 `ls -l > file_list.txt`，若文件已存在则清空）；
 - >>：追加写入文件（如 `echo "new line" >> file_list.txt`，在文件末尾添加内容）。
- **同时输出到屏幕和文件：**使用 tee 命令；
 - 示例：`ls -l | tee file_list.txt`（屏幕显示 ls 结果，同时写入 file_list.txt）；
 - 追加模式：`ls -l | tee -a file_list.txt`（-a 表示追加）。

38. 如何创建、删除用户？如何修改用户密码？

- **创建用户：**useradd（或 adduser，部分系统 adduser 是 useradd 的封装）；
 - 示例：创建用户 user1 并指定家目录：`useradd -d /home/user1 -m user1`（-m 自动创建家目录）。
- **删除用户：**userdel；
 - 仅删除用户：`userdel user1`；

- 同时删除家目录：`userdel -r user1`（`-r` 递归删除家目录和邮件目录）。
- 修改用户密码：`passwd`；
 - 修改当前用户密码：直接输入 `passwd`，按提示输入新密码；
 - 管理员修改其他用户密码：`passwd user1`（无需输入旧密码）。

39. 如何创建、删除用户组？如何将用户加入某个组？

- 创建用户组：`groupadd`；
 - 示例：创建用户组 `group1`：`groupadd group1`。
- 删除用户组：`groupdel`；
 - 示例：删除用户组 `group1`：`groupdel group1`（需确保组内无用户，否则删除失败）。
- 将用户加入用户组：
 - 方法 1（临时加入，当前会话有效）：`newgrp group1`（切换当前用户的有效组为 `group1`）；
 - 方法 2（永久加入，作为附加组）：`usermod -aG group1 user1`（`-a` 追加，`-G` 指定附加组列表）；
 - 验证：`id user1`（查看用户的所属组信息）。

40. 如何查看当前登录的用户？如何查看用户的登录历史？

- 查看当前登录的用户：
 - `who`：显示当前登录的用户名、终端、登录时间、IP 地址；
 - `w`：显示当前登录用户及正在执行的命令（比 `who` 更详细）；
 - `users`：仅显示当前登录的用户名（去重）。
- 查看用户的登录历史：
 - `last`：显示所有用户的登录历史（从 `/var/log/wtmp` 读取，包括登录时间、终端、IP、退出时间）；
 - `lastlog`：显示每个用户的最后一次登录信息（包括未登录过的用户）；
 - 查看失败登录记录：`lastb`（从 `/var/log/btmp` 读取，需 `root` 权限）。

41. sudo 命令的作用是什么？如何配置用户的 sudo 权限？

- **sudo 命令的作用：**允许普通用户在知道自己密码的情况下，临时以 root（或其他指定用户）身份执行特定命令，避免直接使用 root 登录带来的风险。
- **配置用户的 sudo 权限：**
 1. 编辑 sudo 配置文件（需 root 权限）：visudo（推荐，会自动语法检查）或 vim /etc/sudoers；
 2. 配置规则格式：用户名 主机名=（可切换的用户） 可执行的命令；
 - 示例 1：允许 user1 在所有主机上以 root 身份执行任何命令：
user1 ALL=(ALL) ALL
 - 示例 2：允许 user1 无需输入密码执行 ls、cp 命令：
user1 ALL=(ALL) NOPASSWD: /bin/ls, /bin/cp
 3. 保存退出后，user1 即可使用 sudo 命令（如 sudo ls /root）。

42. 如何查看磁盘分区和使用率？df 和 du 命令的区别是什么？

- **查看磁盘分区和使用率：**df（disk free）命令；
 - 示例：df -h（-h 表示“human-readable”，用 GB/MB 显示容量，更易读）；
 - 输出包含：文件系统（分区）、总容量、已用容量、可用容量、使用率、挂载点。
- **df 与 du 的区别：**

命令	功能特点	统计对象	适用场景
df	查看文件系统（分区）的整体容量和使用率	基于文件系统的 inode 信息，统计快	检查分区是否满了
du	查看文件/目录占用的磁盘空间	遍历文件内容，统计每个文件的实际大小	查找大文件/目录

43. 如何查看某个目录的大小？如何查找磁盘中占用空间最大的文件/目录？

- **查看某个目录的大小：**du -sh 目录名；

- 示例：查看/home 目录的大小：du -sh /home;
- 参数说明：-s (summarize, 仅显示总大小)、-h (人类可读格式)。
- 查找磁盘中占用空间最大的文件/目录：

- 方法 1 (查找/目录下占用最大的前 10 个目录)：

```
du -ah / --max-depth=1 | sort -rh | head -n 10
```

- -a: 包含文件; --max-depth=1: 仅统计一级目录; sort -rh: 按大小倒序排序; head -n 10: 显示前 10 条。

- 方法 2 (查找最大的 10 个文件)：

```
find / -type f -exec du -sh {} \; | sort -rh | head -n 10
```

- -type f: 仅查找文件; -exec du -sh {} \;;: 对每个文件执行 du 命令。

44. 如何挂载/卸载一个外部存储设备 (如 U 盘、分区) ?

- 前提：需知道设备的挂载点 (如/dev/sdb1, 可通过 lsblk 或 fdisk -l 查看)。
- 挂载步骤：
 1. 创建挂载点目录 (如/mnt/usb) : mkdir -p /mnt/usb;
 2. 执行挂载命令 (以 ext4 文件系统为例)：

```
mount /dev/sdb1 /mnt/usb
```
 3. 验证：df -h 查看是否挂载成功。
- 卸载步骤：
 1. 确保设备未被占用 (若占用, 用 fuser -m /mnt/usb 查找占用进程并终止)；
 2. 执行卸载命令：umount /mnt/usb (或 umount /dev/sdb1)；
 3. 验证：df -h 查看设备是否已卸载。
- 注意：mount 为临时挂载, 重启后失效; 永久挂载需编辑/etc/fstab 文件。

45. 什么是 inode？它与文件有什么关系？

- **inode**: 全称“index node”，是 Linux 文件系统中用于存储文件**元数据**（描述文件属性的数据）的结构，每个文件对应一个唯一的 inode。
- **inode 存储的元数据**: 文件类型（普通文件/目录/链接）、权限（rwx）、所有者/所属组、文件大小、修改时间（mtime）、访问时间（atime）、变更时间（ctime）、指向文件数据块的指针等。
- **与文件的关系**:
 - 文件的“文件名”仅用于用户识别，实际存储在目录的 inode 中；
 - 系统访问文件时，先通过文件名找到对应的 inode，再通过 inode 中的指针找到文件的实际数据块；
 - 一个 inode 可对应多个文件名（即“硬链接”），多个文件名指向同一个 inode 和数据块。

46. 如何编写一个简单的 Shell 脚本？脚本的首行 `#!/bin/bash` 有什么作用？

- **编写简单的 Shell 脚本**（示例：输出“Hello World”并显示当前日期）：
 1. 创建脚本文件：vim hello.sh;
 2. 写入内容：

```
#!/bin/bash
# 这是一个简单的 Shell 脚本（注释以#开头）
echo "Hello World!" # 输出字符串
echo "Current Date: $(date)" # 执行 date 命令并输出结果
```
- **首行 `#!/bin/bash` 的作用**: 称为“shebang”（#!），指定执行该脚本的解释器路径；
 - 告诉系统：用/bin/bash 这个 Shell 解释器来执行脚本内容；
 - 若不指定，系统会使用默认 Shell（通常也是 bash），但指定后可确保脚本在不同环境下的一致性。

47. 如何给 Shell 脚本添加执行权限并运行？

- **添加执行权限**: 使用 chmod 命令，给脚本所有者添加执行权限（+x）；
 - 示例：chmod u+x hello.sh（仅所有者可执行）；
 - 或 chmod +x hello.sh（所有用户可执行）。

- 运行脚本:

1. 绝对路径运行: `/home/user/hello.sh` (需知道脚本的完整路径);
2. 相对路径运行: 若当前目录在脚本所在目录, `./hello.sh` (. 表示当前目录, 必须加, 否则系统会在 PATH 中查找);
3. 通过解释器运行 (无需执行权限): `bash hello.sh` (直接调用 bash 解释器执行脚本)。

48. 写出一个 Shell 脚本, 实现“每天凌晨 3 点备份指定目录到/backup”的功能。

- 脚本名称: `backup_dir.sh` (需替换 `/path/to/source_dir` 为实际要备份的目录);

```
#!/bin/bash
# 每天凌晨 3 点备份指定目录到/backup

# 1. 定义变量 (源目录、备份目录、备份文件名)
SOURCE_DIR="/path/to/source_dir" # 要备份的源目录
BACKUP_DIR="/backup"             # 备份目标目录
BACKUP_FILENAME="backup_$(date +%Y%m%d).tar.gz" # 备份文件名 (含日期,
如 backup_20240520.tar.gz)

# 2. 检查备份目录是否存在, 不存在则创建
if [ ! -d "$BACKUP_DIR" ]; then
    mkdir -p "$BACKUP_DIR"
fi

# 3. 执行备份 (tar 压缩源目录到备份目录)
tar -zcvf "$BACKUP_DIR/$BACKUP_FILENAME" "$SOURCE_DIR"

# 4. 备份完成后输出日志 (可选)
echo "Backup completed at $(date). File saved to: $BACKUP_DIR/$BACKUP_FILENAME" >> "$BACKUP_DIR/backup_log.txt"
```

- 后续步骤:

1. 给脚本添加执行权限: `chmod +x backup_dir.sh`;
2. 用 `crontab` 设置定时任务 (每天凌晨 3 点执行): `crontab -e`, 添加一行:

```
0 3 * * * /path/to/backup_dir.sh
```

49. 如何使用 crontab 设置定时任务？举例说明定时任务的格式。

- **crontab 命令：**用于管理用户的定时任务（每个用户有独立的 crontab 文件）；
 - 编辑定时任务：crontab -e（首次使用会提示选择编辑器，如 vim）；
 - 查看定时任务：crontab -l；
 - 删除定时任务：crontab -r。

- **定时任务格式**（共 6 个字段，空格分隔）：

*	*	*	*	*	要执行的命令/脚本路径
分	时	日	月	周	命令

- 字段说明：

- 分（0-59）：每分钟用*，每 5 分钟用*/5；
- 时（0-23）：凌晨 3 点用 3；
- 日（1-31）：每月 1 日用 1；
- 月（1-12）：每年 5 月用 5；
- 周（0-7，0 和 7 均表示周日）：每周日用 0。

- **示例：**

1. 每天凌晨 3 点执行/home/user/backup.sh：0 3 * * *
/home/user/backup.sh；
2. 每周一到周五的 18 点 30 分执行/home/user/clean.sh：30 18 * * 1-5
/home/user/clean.sh；
3. 每月 1 日和 15 日的凌晨 2 点执行/home/user/check.sh：0 2 1,15 * *
/home/user/check.sh。

50. 如何在 Shell 脚本中判断一个文件是否存在？

- **核心：**使用 Shell 的条件判断语句 if [条件]; then ... fi，配合文件测试运算符-e（exists，检查文件是否存在）。
- **示例脚本**（判断/home/user/file.txt 是否存在）：

```
#!/bin/bash
```

```
FILE="/home/user/file.txt"
```

```
# 判断文件是否存在
```

```
if [ -e "$FILE" ]; then
    echo "File $FILE exists."
    # 进一步判断文件类型 (可选)
    if [ -f "$FILE" ]; then # -f: 是否为普通文件
        echo "$FILE is a regular file."
    elif [ -d "$FILE" ]; then # -d: 是否为目录
        echo "$FILE is a directory."
    fi
else
    echo "File $FILE does not exist."
fi
```

- 常用文件测试运算符:

- -e: 文件是否存在;
- -f: 是否为普通文件;
- -d: 是否为目录;
- -r: 是否有读权限;
- -w: 是否有写权限;
- -x: 是否有执行权限。