文件系统和权限

一、Linux文件系统（更详细介绍）

定义与原理：
Linux文件系统是操作系统管理存储设备上数据的方式，负责文件的存储、组织、检索和权限控制。它采用**树状结构**，以根目录（`/`）为起点，所有文件和目录以层级方式组织。Linux文件系统的核心是通过**文件系统驱动**与存储设备交互，结合**内核**的虚拟文件系统（VFS）层，统一管理不同类型的文件系统。

文件系统的工作机制：（了解即可）

- **Inode**：每个文件或目录都有一个唯一的索引节点（inode），存储元数据（如权限、所有者、时间戳、大小等），但不包含文件名。文件名存储在目录的inode中。
- **块存储**：文件数据存储在磁盘块中，文件系统管理块分配和访问。
- **挂载**：文件系统通过`mount`命令挂载到目录树，设备（如硬盘、U盘）与目录关联。未挂载的设备无法访问。

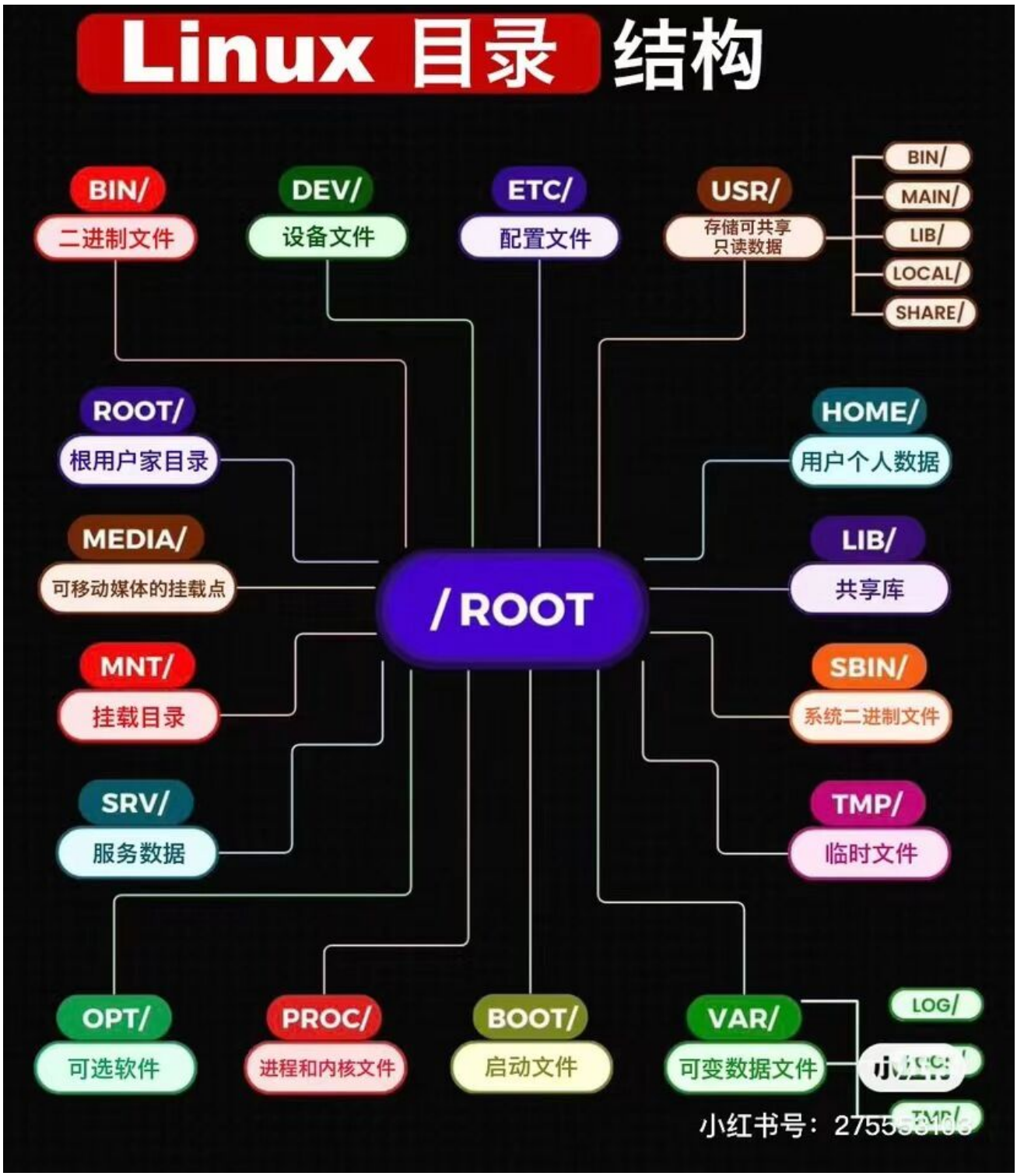
常见文件系统类型（看看特性就行，不用太死）：

- **ext4**（第四扩展文件系统）：
  - 默认文件系统，稳定且高效。
  - 支持最大2TB单文件，16TB分区。
  - 特性：日志功能（减少数据丢失）、向后兼容ext2/ext3、延迟分配（提高性能）。
  - 适用场景：通用服务器、桌面系统。
- **XFS**：
  - 高性能，适合大文件和大数据量场景。
  - 支持最大8EB分区，动态分配inode。
  - 特性：高效并行I/O、快照支持。
  - 适用场景：高性能计算、媒体存储。
- **Btrfs**（B-tree文件系统）：
  - 现代文件系统，支持快照、数据压缩、动态卷管理。
  - 特性：支持子卷、RAID、数据完整性检查。
  - 适用场景：需要快照或高级存储管理的场景，如NAS。
- **FAT32/NTFS**：
  - FAT32：跨平台兼容，适合U盘，单文件最大4GB。
  - NTFS：Windows常用，Linux可读写（通过`ntfs-3g`）。
  - 适用场景：移动存储设备。

- **tmpfs**：
  - 基于内存的文件系统，数据存储在RAM中，重启丢失。
  - 适用场景：临时文件存储，如`/tmp`。
- **NFS**（网络文件系统）：
  - 用于远程文件共享，允许多台Linux主机访问同一文件系统。
  - 适用场景：分布式系统、文件共享。

文件系统目录结构（详细说明）：

- `/`：根目录，所有文件系统的起点。
- `/etc`：系统配置文件，如：
  - `/etc/fstab`：定义挂载点。
  - `/etc/passwd`：用户信息。
  - `/etc/hosts`：主机名解析。
- `/home`：用户主目录，每个用户有子目录（如`/home/user1`），存储个人文件。
- `/var`：动态数据目录，如：
  - `/var/log`：日志文件（如`syslog`、`messages`）。
  - `/var/www`：Web服务器文件。
- `/tmp`：临时文件目录，通常具有Sticky Bit权限。
- `/usr`：用户程序和数据：
  - `/usr/bin`：用户可执行程序。
  - `/usr/lib`：库文件。
  - `/usr/local`：本地安装的软件。
- `/dev`：设备文件，如：
  - `/dev/sda`：第一块硬盘。
  - `/dev/null`：空设备，丢弃输出。
- `/proc`：虚拟文件系统，存储运行时信息，如：
  - `/proc/cpuinfo`：CPU信息。
  - `/proc/meminfo`：内存信息。
- `/boot`：存储内核和启动相关文件，如`vmlinuz`（内核镜像）。



二、Linux文件和目录权限

权限机制原理：

- Linux是多用户系统，权限控制基于**用户（User）、组（Group）、其他（Others）**，通过**读（r）、写（w）、执行（x）**三种权限管理访问。
- 权限存储在文件或目录的**inode**中，通过`ls -l`查看。

- **文件与目录权限的区别**：
  - 文件：
    - `r`：可读取文件内容（如`cat file.txt`）。
    - `w`：可修改文件内容。
    - `x`：可执行文件（如脚本或二进制文件）。
  - 目录：
    - `r`：可列出目录内容（如`ls dir`）。
    - `w`：可在目录中创建、删除、重命名文件。
    - `x`：可进入目录（如`cd dir`）。

权限表示（扩展说明）：

- **字符表示**：
  - 示例：`-rwxr-xr-x`：
    - 第一个字符：文件类型（`-`普通文件，`d`目录，`l`符号链接，`b`块设备，`c`字符设备）。
    - 后九个字符：三组权限（所有者、组、其他），每组3位（rwx）。
    - 示例解析：`rwxr-xr-x`表示所有者有读写执行权限，组和其他用户有读和执行权限。
- **八进制表示**：
  - `r=4`，`w=2`，`x=1`，无权限=0。
  - 每组权限的数字是r、w、x的和：
    - `rwx` = 4+2+1 = 7
    - `rw-` = 4+2 = 6
    - `r-x` = 4+1 = 5
  - 示例：`rwxr-xr-x` = `755`，`rw-r--r--` = `644`。
- **查看权限**：

```
ls -l file.txt # 查看文件权限
ls -ld dir # 查看目录权限
```

特殊权限（看看就行）：

- **SUID（Set User ID，4xxx）**：
  - 文件执行时以文件所有者的权限运行。
  - 示例：`/usr/bin/passwd`（修改密码需要root权限）显示为`rwsr-xr-x`。

    - 设置：`chmod u+s file` 或 `chmod 4755 file`。
- **SGID（Set Group ID，2xxx）**：
    - 文件：执行时以文件所属组的权限运行。
    - 目录：新建文件继承目录的所属组。
    - 示例：`drwxrwsr-x`（SGID目录）。
    - 设置：`chmod g+s dir` 或 `chmod 2755 dir`。
- **Sticky Bit（1xxx）**：
    - 目录中只有文件所有者或root才能删除文件，防止其他用户误删。
    - 示例：`/tmp` 显示为 `drwxrwxrwt`。
    - 设置：`chmod +t dir` 或 `chmod 1777 dir`。

权限与用户/组关系：

- **用户和组**：
    - 每个文件有**所有者（owner）**和**所属组（group）**，通过 `ls -l` 查看（如 `user:group`）。
    - 用户通过 `/etc/passwd` 定义，组通过 `/etc/group` 定义。
- **默认权限**：（看看就行）
    - 新建文件/目录的权限由 `umask` 决定。
    - 默认权限：文件 `666`（rw-rw-rw-），目录 `777`（rwxrwxrwx）。
    - `umask` 减去默认权限：如 `umask 022`，新建文件权限为 `666-022=644`，目录为 `777-022=755`。
    - 查看/设置umask：

```
umask # 查看当前umask
umask 022 # 设置umask
```

ACL（访问控制列表）：（看看就行）

- 标准权限只支持用户、组、其他三类，ACL提供更细粒度的权限控制。
- 查看ACL：`getfacl file.txt`
- 设置ACL：

```
setfacl -m u:user2:rw file.txt  # 给user2添加读写权限
setfacl -m g:group1:r dir  # 给group1添加读权限
```

三、常用操作文件或目录的命令（更详细说明）

以下是操作文件和目录的常用命令，包含更详细的用法和场景。

1. 文件操作

- **创建文件**：（重点了解）
  - `touch`：创建空文件或更新时间戳。

```
touch file.txt  # 创建空文件
touch -t 202501011200 file.txt  # 设置时间戳为2025年1月1日12:00
```

  - `echo`：创建并写入内容。

```
echo "Hello" > file.txt  # 覆盖写入
echo "World" >> file.txt  # 追加写入
```

- **查看文件内容**：
  - `cat`：显示全部内容。（重点了解）

```
cat file.txt
cat -n file.txt  # 显示行号
```

  - `less`：分页查看，支持上下翻页（按`q`退出）。（看看得了）

```
less file.txt
```

  - `head` / `tail`：查看文件开头或结尾。（看看taii）

```
head -n 5 file.txt  # 前5行
tail -n 5 file.txt  # 后5行
tail -f /var/log/syslog  # 实时监控日志
```

- **编辑文件**：
  - `nano`：简单编辑器，适合新手。（不常用）

```
nano file.txt # 编辑，Ctrl+O保存，Ctrl+X退出
```

- `vim`：功能强大，适合高级用户。（乘用）

```
vim file.txt # 进入vim，按i编辑，:wq保存退出（会这几个就够用）
```

- **复制文件**：

```
cp source.txt dest.txt # 复制文件
cp -r dir1 dir2 # 递归复制目录
```

- **移动/重命名文件**：

```
mv file.txt /path/to/dir/ # 移动文件
mv oldname.txt newname.txt # 重命名
```

- **删除文件**：

```
rm file.txt # 删除文件
rm -r dir # 删除目录及其内容
rm -f file.txt # 强制删除（无提示）
```

2. 目录操作

- **创建目录**：

```
mkdir mydir # 创建目录
mkdir -p /path/to/nested/dir # 递归创建
```

- **查看目录内容**：

```
ls -l # 详细列表（含权限）
ls -a # 显示隐藏文件（如`.bashrc`）
```

- **切换目录**：

```
cd /path/to/dir # 进入目录
cd .. # 返回上一级
cd - # 返回上一次目录
cd ~ # 返回用户主目录
```

- **删除目录**：

```
rmdir mydir # 删除空目录
rm -r mydir # 删除非空目录
rm -rf mydir # 强制删除（谨慎使用）
```

3. 文件查找与搜索（重点了解find命令）

- **按名称查找（`find`）**：

```
find / -name "file.txt" # 在根目录查找
find . -type f -name "*.txt" # 查找当前目录下所有txt文件
find /home -mtime -7 # 查找7天内修改的文件
```

- **按内容查找（`grep`）**：

```
grep "error" /var/log/syslog # 搜索含“error”的行
grep -r "keyword" /path # 递归搜索目录
grep -i "keyword" file.txt # 忽略大小写
grep -n "keyword" file.txt # 显示行号
```

4. 文件链接

- **符号链接**（软链接）：

```
ln -s /path/to/original linkname # 创建软链接
```

- **硬链接**：

```
ln /path/to/original linkname # 创建硬链接
```

- **区别**：

- 软链接：指向路径，类似快捷方式，原始文件删除后失效。
- 硬链接：指向同一inode，删除原始文件后仍有效。

四、修改权限的命令

以下是修改权限的命令，包含更多选项和实际场景。

1. `chmod`（更改权限）

- **功能**：修改文件或目录的读、写、执行权限。
- **语法**：

```
chmod [选项] 权限 文件/目录
```

- **八进制模式**（常用）：

```
chmod 755 script.sh # 所有者：rwx，组和其他：r-x
chmod 644 data.txt # 所有者：rw-，组和其他：r--
chmod 700 secret.txt # 仅所有者有权限
```

- **符号模式**（更灵活）：
  - 格式：`[ugoa][+-=][rwx]`
  - 示例：

```
chmod u+rwx file.txt # 给所有者添加读写执行权限
chmod g-w file.txt # 移除组的写权限
chmod o=r file.txt # 设置其他用户只读
chmod a+x script.sh # 所有人添加执行权限
```

- **递归修改**：

```
chmod -R 755 /path/to/dir # 递归修改目录及其内容
```

- **设置特殊权限**：
  - SUID：`chmod 4755 file`（显示为`rwsr-xr-x`）。

- SGID：`chmod 2755 dir`（显示为 `rwxr-sr-x`）。
- Sticky Bit：`chmod 1777 dir`（显示为 `rwxrwxrwt`）。

2. `chown`（更改所有者）

- **功能**：更改文件或目录的所有者。
- **语法**：

```
chown [选项] 用户[:组] 文件/目录
```

- **示例**：

```
chown user1 file.txt  # 更改所有者为user1
chown user1:group1 file.txt  # 更改所有者和组
chown -R user1:group1 /path/to/dir  # 递归更改
chown :group1 file.txt  # 仅更改组
```

- **选项**：
  - `-R`：递归更改。
  - `--reference=ref_file`：复制另一文件的权限：

```
chown --reference=ref.txt file.txt
```

3. `chgrp`（更改所属组）

- **功能**：更改文件或目录的所属组。
- **语法**：

```
chgrp [选项] 组 文件/目录
```

- **示例**：

```
chgrp group1 file.txt  # 更改所属组
chgrp -R group1 /path/to/dir  # 递归更改
```

- **选项**：

- `-R`：递归更改。
- `--reference=ref_file`：复制另一文件的组。

4. 其他权限相关命令（了解）

- **umask**：设置新建文件/目录的默认权限。

```
umask # 查看当前umask（如022）
umask 027 # 设置umask，新文件权限为640，目录为750
```

- **chattr**：设置文件属性（如不可修改）。

```
chattr +i file.txt # 设置不可修改（即使root也无法更改）
chattr -i file.txt # 移除不可修改属性
lsattr file.txt # 查看属性
```

- **setfacl/getfacl**（ACL管理）：

```
setfacl -m u:user2:rw file.txt # 给user2添加读写权限
getfacl file.txt # 查看ACL
setfacl -x u:user2 file.txt # 移除user2的ACL
```

五、实际应用场景（暂时了解就够）

1. **设置Web服务器目录**：
   - 场景：配置Nginx的Web目录，需确保www-data用户有权限。

```
mkdir /var/www/html
chown -R www-data:www-data /var/www/html
chmod -R 755 /var/www/html # 所有者读写执行，其他用户读执行
```

2. **创建共享目录**：
   - 场景：团队共享目录，新文件自动继承组权限。

```
mkdir /shared
chgrp -R team /shared
chmod -R 2775 /shared  # SGID确保新文件继承team组
```

3. **保护敏感文件**：
   - 场景：防止重要日志被修改。

```
touch /var/log/secure.log
chown root:root /var/log/secure.log
chmod 600 /var/log/secure.log  # 仅root可读写
chattr +i /var/log/secure.log  # 防止修改
```

4. **清理临时文件**：
   - 场景：删除7天未修改的临时文件。

```
find /tmp -type f -mtime +7 -exec rm -f {} \;
```

5. **设置可执行脚本**：
   - 场景：创建并运行bash脚本。

```
echo '#!/bin/bash' > myscript.sh
echo 'echo Hello World' >> myscript.sh
chmod +x myscript.sh  # 添加执行权限
./myscript.sh  # 运行脚本
```

六、补充说明与高级操作（看看就行）

1. **权限调试技巧**：
   - **检查权限错误**：

```
strace -o trace.log command  # 跟踪系统调用，定位权限问题
tail -f /var/log/syslog  # 查看系统日志
```

   - **测试权限**：

```
su - user2  # 切换到user2测试访问
cat file.txt  # 检查是否可读
```

2. **性能优化**：
   - 使用`noatime`挂载选项减少访问时间更新：

     ```
     mount -o remount,noatime /dev/sda1 /data
     ```

   - 调整文件系统缓存：

     ```
     sysctl vm.swappiness=10  # 减少swap使用
     ```

3. **安全注意事项**：
   - 避免`777`权限，限制不必要的写权限。
   - 定期检查敏感文件权限：

     ```
     ls -l /etc/passwd /etc/shadow  # 应为644和600
     ```

   - 使用SELinux或AppArmor增强权限控制。
4. **备份与恢复**：
   - 备份文件：

     ```
     tar -czf backup.tar.gz /path/to/dir  # 压缩备份
     ```

   - 恢复：

     ```
     tar -xzf backup.tar.gz -C /path  # 解压恢复
     ```

uname: 显示系统信息:内核版本、机器类型等。

uname -r: 显示正在运行的Linux内核的发行版本。

uptime: 显示当前时间、系统运行时间、用户和负载平均值。

hostname: 显示系统主机名。

hostname -i: 显示当前主机的ip地址。

last reboot: 查看系统最近一次的重启时间以及重启之前的系统登录信息。

date: 显示当前日期和时间。

timedatectl: 设置系统时间与日期。

cal: 快速查看当前/指定年份日历。

w: 显示目前登入系统的用户信息。

whoami: 显示当前用户的用户名。

finger username: 显示名为"username"的用户的信息。

用户管理

id: 显示用户的UID、GID、组。

last: 显示最后登录用户的列表。

who: 显示当前登录的用户。

groupadd admin: 创建一个名为admin的新用户组。

adduser Sam: 创建一个名为Sam的新用户。

userdel Sam: 删除名为Sam的用户。

usermod: 修改现有用户的属性。

硬件

dmesg: 内核会将开机过程信息存储在环形缓冲区中。

cat /proc/cpuinfo: 显示CPU的详细信息。

cat /proc/meminfo: 显示详细的系统内存使用信息。

lshw: 列出系统的详细硬件配置。

lsblk: 列出所有可用的块设备的信息。

free -m: 显示系统内存使用情况。

lspci -tv: 以树状格式详细显示PCI设备信息。

lsusb -tv: 以树状格式详细显示USB设备信息。

dmidecode: 显示系统BIOS中的硬件信息。

hdparm -i /dev/sda: 显示磁盘/dev/sda的信息。

badblocks -s /dev/sda:
用于检查设备 /dev/sda 是否存在坏块,并显示检测进度。

安装源(编译)

./configure: 检查系统兼容性并生成用于软件安装的makefile。

make: 按照Makefile中的指令编译代码。

make install: 将编译后的代码安装到指定的系统位置

ls -al: 以长格式列出所有文件的详细信息。

pwd: 显示当前工作目录的路径。

mkdir dir1: 创建一个名为 dir1 的新目录。

rm file1: 删除名为file1的文件。

rm -f file2: 强制删除名为file2的文件。

rm -r dir1: 递归删除目录 dir1 及其内容。

rm -rf dir1: 强制删除目录 dir1 及其内容。

cp file1 file2: 复制 file1,创建或覆盖 file2。

cp -r dir1 dir2: 将dir1复制到dir2,包括子目录。

mv file1 file2: 将 file1 重命名或移动到 file2。

ln -s /oath/to/file_name link_name: 创建名为 link_name 到 file_name 的符号链接。@笨熊费键盘

touch file1: 创建一个名为 file1 的空文件。

cat > file1: 创建/覆盖 file1,等待标准输入。

more file1: 逐页显示file1的内容。

head file1: 显示file1的前十行。

tail file1: 显示file1的最后十行。

gpg -c file1: 使用对称密码对file1进行加密,需要提供密码短语

gpg file2.gpg: 解密file2.gpg,提示输入密码。

wc: 统计文件中的字数、行数和字符数。

xargs: 使用管道或文件提供的参数执行命令。

登录

ssh user@hostname: 向指定主机名发起SSH连接。

ssh -p port_number user@hostname:
使用特定端口发起 SSH 连接。

Connect to the host via telnet default port 23:
通过SSH默认端口22安全连接到系统。

telnet host: 通过telnet默认端口23连接到主机。

安装包

rpm -i pkg_name.rpm:
使用RPM包管理器安装"pkg_name.rpm"包。

rpm -e pkg_name: 卸载指定的RPM包。

dnf install pkg_name: 使用DNF安装指定的包。

pacman -S: 使用Pacman安装指定的包。

目录遍历

cd ..: 切换到父目录。

cd: 将当前目录更改为用户的主目录。

cd /mnt: 将当前目录更改为 "/mnt"。

流程相关

ps：显示当前进程的快照。

ps aux | grep telnet：显示运行中的 telnet 进程的详细信息。

pmap：显示进程的内存映射。

top：显示运行任务的动态实时视图。

kill 1234：终止具有 PID 1234 的进程。

killall proc：终止所有名为 'proc' 的进程。

pkill process-name：终止具有指定名称的进程。

bg：在后台恢复暂停的作业。

fg：将暂停的作业带到前台。

fg n：将作业编号为 'n' 的作业移到前台运行。

lsof：列出所有打开的文件和进程。

renice 19 PID：更改具有给定 PID 的进程的优先级。

pgrep firefox：显示 firefox 进程的进程 ID。

pstree：显示运行中的进程树。

磁盘使用情况

df -h：显示所有已挂载文件系统可读的磁盘空间使用情况。

df -i：显示所有已挂载文件系统的inode使用情况。

fdisk -l：列出所有驱动器上的分区及其信息。

du -sh/dir1：
显示 /dir1 目录的总磁盘使用大小的摘要,以可读方式呈现。

findmnt：显示所有已挂载文件系统及其属性的列表。

mount device-path mount-point：
将设备挂载到指定的文件系统挂载点上。

搜索

grep pattern file：
在文件中搜索给定的模式。

grep- r pattern dir1：
在“dir1”目录及其子目录中递归搜索指定的“pattern”。

locate file：
使用预建的数据库查找名为 "file" 的文件。

find /home-name index：
递归搜索“/home”目录中名为“index”的文件。

find /home-size +10000k：
在/home目录中查找大小超过10000k的文件。

网络

ip addr show：显示所有网络接口及其信息。

ip address add 192.168.0.1/24 dev eth0：
将IP地址192.168.0.1分配给eth0接口。

ifconfig：显示网络接口及其配置。

ping host：发送ICMP数据包,测量与"host"之间的往返时间。

whois domain：检索并显示域名的注册信息。

dig domain：查询DNS,提供域名的DNS信息。

dig -x host：将IP地址解析为主机名,显示DNS信息。

host gexample.com：对域名进行IP查找。

wget file_path：从指定路径下载文件。

netstat：显示各种与网络相关的信息和统计数据。

压缩/存档

tar -cf backup.tar/home/ubuntu：
创建一个名为backup.tar的tar归档文件,其中包含 /home/ubuntu
目录的内容。

tar -xf backup.tar：从 backup.tar 归档文件中提取文件。

tar -zcvf backup.tar.gz/home/ubuntu：
创建一个名为backup.tar.gz的压缩tar归档文件,其中包含/home/ubuntu目录的内容。笨熊费键盘

gzip file1：将文件 file1 压缩为 file1.gz,并删除原始文件。

日志文件传输

scp file.txt remoteuser@remote_host:/remote/directory:
将file.txt复制到远程主机的指定目录。

rsync -a /home/ubuntu/backup/:
将源目录的内容同步到目标目录,保留属性。

rsync -a /var/www/web/user@remote_host:/backup/web_backup/：
同步本地目录到远程,保留属性。

文件权限

chmod 644 /data/:
设置文件/data/test.c的权限为所有者读/写,组和其他人只读。

chmod 755 /dir1:
将目录
/dir1的权限设置为对所有者可读/可写/可执行,对组和其他用户可读/可执行

chown bob:devops filename:
将文件'filename'的所有者更改为'bob'并将所属组更改为 'devops'。

chown ownername：更改目录的所有者和所属组。