

我能够使用一些命令和编写 shell 脚本熟练的操作系统，对系统中的文件进行管理，比如修改配置文件，查找日志等。管理系统中的用户和文件的权限。查看系统运行状态,查看进程、服务等。

## 文件管理

我掌握基本的文件操作。使用 touch 创建一个文件。rm 删除文件或目录等。对日志文件**查看**。内容较少可以使用 cat 输出全部内容。内容过多可以使用 more, less 分屏查看内容。head 可以看文件前几行。tail 也能动态的查看文件新增的内容用于日志追踪。

在项目中会进行**查找**.log 文件的操作，可以使用 grep 命令。ls /log | grep "\*.log"。查找文件可以用 find 命令，find /log -name "\*.log"。根据文件名查找-name，还有其它参数：-type, -user, -size 根据需求选择。

我们在项目中也会有**修改文本内容**的操作。例如替换配置文件中的域名。可以使用 sed 命令，sed "s/aa.com/bb.com/g config.ini。能使用正则表达式，删除匹配的行,也能添加，使用 tee 命令 echo "port=3306" | tee -a config.ini。我还掌握 **vim 的使用**，在命令模式中使用 dd 删除光标所在行，yy 和 p 可以复制一行和粘贴，数字加 G 可以跳转到多少行。进入插入模式就可以任意修改内容。ESC 回到命令模式:wq 就保存退出。

**对文本内容截取**，在我们项目中也是常用的操作。可以使用 cut。以一个字符或多个字符进行分割行，获取需要的某列。比如：cat "access.log" | cut -d : -f 1。awk 也能完成这样的操作，它还能做一些运算。比如统计目录大小，先 cd 进入到目录中，使用 ll | awk '{sum+=\$5;print sum}'

对文件的**解压缩**也是常用操作，比如数据备份和还原等，可使用 zip 或 tar 命令。例如 tar -zcvf backup.tar.gz file1 file2 dir1，对文件和目录进行压缩打包。解压文件操作 tar -zxvf filename;

## 用户管理

我还掌握用户和组的管理，用户创建和删除。组的创建，删除。设置用户和组的密码用 passwd 和 gpasswd。还有管理组中的成员，例如将 jerry 加入到 test 组：gpasswd -a jerry test，删除用 -d 参数。-M 重新设置组员。查看组成员：getent group test。查看用户所属的组使用 id 命令：id jerry;

## 权限管理

我们在项目还会遇到权限问题，使用 chmod 为文件的拥有者，组成员和其他用户设置读写执行的权限。例如为 shell 脚本添加可执行权限：chmod ugo+x clearLog.sh。还可以使用 -R 参数对所有子目录和文件修改。还有更改文件的拥有者：chown; 更改文件所属组 chgrp。

## 权限&用户管理

我们在项目中还会遇到权限问题，例如 jerry 用户无法访问某目录，我们可以将用户和文件设置为同一组，对于用户可以使用 gpasswd 命令：gpasswd -a jerry tom; 对于文件：使用 chgrp 命令。还可以是 chmod 修改文件的权限，chmod 777/744 目录/文件

## 进程管理

我还能够管理系统中的进程。使用 ps -ef 能查看当前系统的进程。使用 -aux 参数查看所有用户执行的进程。还可以使用 top 命令实时的系统状态，cpu 使用率，内存占用比等。根据查看到进程的 PID，可以使用 kill 命令结束进程。

## 服务管理

我们在项目中会管理服务程序，httpd，mysql 等，在安装之后我们需要启动它，使用 systemctl 查看服务的状态，开启关闭。设置服务自启或关闭自启。例如将 httpd 服务设置自启动：systemctl enable httpd。

## 网络管理

我还掌握一些关于网络操作。使用 ifconfig 或者 ip addr 查看本机的 ip 地址。使用 ping ip 或者域名。查看是否与目标主机网络联通。使用 netstat 命令查看端口的使用情况。比如：那个进程使用了 80 端口：netstat -tunlp | grep ":80"。也可以使用简单的命令 lsof -i :80 查看。还有防火墙的开启关闭等。还有使用 firewall-cmd 命令加--add-port=80/tcp，对 80 端口放行，这是临时的，需要永久的加上--permanent 参数。然后重载防火墙--reload。

## 磁盘管理

对于磁盘的管理我也有一些掌握。使用 du -ah 查看当前路径下每个文件的大小。df -h 查看整个系统磁盘使用的情况。还有当外部的硬盘连接时，使用 mount 挂载到某目录下，进行访问。卸载硬盘使用 umount 命令。还有使用 fdisk 对磁盘进行分区。

## 软件安装

对于软件的安装，我掌握多种方式：使用 yum 安装 httpd：yum -y install httpd。或者下载二进制的安装包，使用 rpm -ivh 加文件名安装。还有就是下载软件的源码，例如使用 wget 或 curl 下载 lrzsz 源码压缩包，解压后可以修改安装路径，使用 make 命令进行编译，编译完成后使用 make install 命令安装软件。

## Shell 脚本编写

在 shell 脚本中，可以直接使用一些**特殊的变量**：\$\*：接收运行脚本时传入的所有参数；\$#：传入脚本参数的个数；\$?：记录上一条命令是否成功执行；\$\$：当前脚本的执行的进程 id；还有位置变量；shell 脚本最大支持 9 个参数，分别用 \$1-\$9 接收。

在脚本中掌握**流程控制**语句的使用：if 条件判断，case 条件选择，循环 for，while，until。其中最重要的就是 if 条件判断。使用 **test 命令做各种测试**；比如判断一个目录或文件是否存在，可以使用 test -e 路径；或者用中括号的形式表达：[ -e 路径 ]；判断是否是文件或者目录，分别用 -f 和 -d 参数。还有数字测试：比如判断 age 变量是否大于 18：[ \$age -gt 18 ]；-lt 判断是否小于，-eq 和 -ne 判断是否等于或者不等于。多条件的可以使用 -a 或 -o 连接，表示逻辑与和逻辑或；也可以使用 && 和 || 更直观；

在脚本中也会进行**重定向**的操作；输出重定向符 >；追加重定向符 >>；错误重定向符 2>，比如 find / -name "\*.log" > info.log 2> error.log，分别把正常输出和错误输出保存到 info.log 和 error.log 中；

## 定时任务

我们在项目中会经常使用到定时任务 crontab，临时任务 at。查看定时任务 crontab -l，-e 参数进行任务编辑。例如每天晚上 3 点执行备份脚本：0 3 \* \* \* /shell/backup.sh。还有临时任务的使用：at 命令，比如在今天 9:30 要执行数据处理脚本，at -f 脚本绝对路径 9:30。使用 -l 参数可以查看到已添加的临时任务，-r 参数加上任务 id 可以删除定时任务。

以上就是我对 linux 掌握的情况，可能存在遗漏的部分。还能进行其它部分补充。