

作为开发者，这些常用命令不得不会，掌握这些命令，工作上会事半功倍，提供工作效率。

一. 文件和目录

cd命令，用于切换当前目录，它的参数是要切换到的目录的路径，可以是绝对路径，也可以是相对路径。

```
cd /home 进入 '/ home' 目录
cd .. 返回上一级目录
cd ../.. 返回上两级目录
cd          进入个人的主目录
cd ~user1 进入个人的主目录
cd - 返回上次所在的目录
```

pwd命令，显示工作路径

```
[root@mailvip ~]# pwd
/root
```

ls命令，查看文件与目录的命令，list之意

```
ls 查看目录中的文件
ls -l 显示文件和目录的详细资料
ls -a 列出全部文件，包含隐藏文件
ls -R 连同子目录的内容一起列出（递归列出），等于该目录下的所有文件都会显示出来
ls [0-9] 显示包含数字的文件名和目录名
```

cp命令，用于复制文件，copy之意，它还可以把多个文件一次性地复制到一个目录下

```
-a : 将文件的特性一起复制
-p : 连同文件的属性一起复制，而非使用默认方式，与-a相似，常用于备份
-i : 若目标文件已经存在时，在覆盖时会先询问操作的进行
-r : 递归持续复制，用于目录的复制行为 //经常使用递归复制
-u : 目标文件与源文件有差异时才会复制
```

mv命令，用于移动文件、目录或更名，move之意

```
-f : force强制的意思，如果目标文件已经存在，不会询问而直接覆盖
-i : 若目标文件已经存在，就会询问是否覆盖
-u : 若目标文件已经存在，且比目标文件新，才会更新
```

rm命令，用于删除文件或目录，remove之意

```
-f : 就是force的意思，忽略不存在的文件，不会出现警告消息
-i : 互动模式，在删除前会询问用户是否操作
-r : 递归删除，最常用于目录删除，它是一个非常危险的参数
```

二、查看文件内容

cat命令，用于查看文本文件的内容，后接要查看的文件名，通常可用管道与more和less一起使用

```
cat file1 从第一个字节开始正向查看文件的内容
tac file1 从最后一行开始反向查看一个文件的内容
cat -n file1 标示文件的行数
more file1 查看一个长文件的内容

head -n 2 file1 查看一个文件的前两行
tail -n 2 file1 查看一个文件的最后两行
tail -n +1000 file1 从1000行开始显示，显示1000行以后的
cat filename | head -n 3000 | tail -n +1000 显示1000行到3000行
cat filename | tail -n +3000 | head -n 1000 从第3000行开始，显示1000(即显示3000~3999行)
```

三.文件搜索

find命令，用来查找系统的

```
find / -name file1 从 '/' 开始进入根文件系统搜索文件和目录
find / -user user1 搜索属于用户 'user1' 的文件和目录
find /usr/bin -type f -atime +100 搜索在过去100天内未被使用过的执行文件
find /usr/bin -type f -mtime -10 搜索在10天内被创建或者修改过的文件
whereis halt 显示一个二进制文件、源码或man的位置
which halt 显示一个二进制文件或可执行文件的完整路径
```

删除大于50M的文件：

```
find /var/mail/ -size +50M -exec rm {} \;
```

四.文件的权限 - 使用 "+" 设置权限，使用 "-" 用于取消

chmod命令，改变文件/文件夹权限

```
ls -lh 显示权限
chmod ugo+rw directory1 设置目录的所有人(u)、群组(g)以及其他(o)以读(r, 4)、写(w, 2)和执行(x, 1)的权限
chmod go-rwx directory1 删除群组(g)与其他人(o)对目录的读写执行权限
```

chown命令，改变文件的所有者

```
chown user1 file1 改变一个文件的所有人属性
chown -R user1 directory1 改变一个目录的所有人属性并同时改变该目录下所有文件的属性
chown user1:group1 file1 改变一个文件的所有人和群组属性
```

chgrp命令，改变文件所属用户组

```
chgrp group1 file1 改变文件的群组
```

五.文本处理

grep命令，分析一行的信息，若当中有我们所需要的信息，就将该行显示出来，该命令通常与管道命令一起使用，用于对一些命令的输出进行筛选加工等等

```
grep Aug /var/log/messages 在文件 '/var/log/messages' 中查找关键词"Aug"

grep ^Aug /var/log/messages 在文件 '/var/log/messages' 中查找以"Aug"开始的词汇
grep [0-9] /var/log/messages 选择 '/var/log/messages' 文件中所有包含数字的行

grep Aug -R /var/log/* 在目录 '/var/log' 及随后的目录中搜索字符串"Aug"

sed 's/stringa1/stringa2/g' example.txt 将example.txt文件中的 "string1" 替换成
"string2"

sed '/^$/d' example.txt 从example.txt文件中删除所有空白行
```

paste命令

```
paste file1 file2 合并两个文件或两栏的内容
paste -d '+' file1 file2 合并两个文件或两栏的内容，中间用"+"区分
```

sort命令

```
sort file1 file2 排序两个文件的内容
sort file1 file2 | uniq 取出两个文件的并集(重复的行只保留一份)
sort file1 file2 | uniq -u 删除交集，留下其他的行
sort file1 file2 | uniq -d 取出两个文件的交集(只留下同时存在于两个文件中的文件)
```

comm命令

```
comm -1 file1 file2 比较两个文件的内容只删除 'file1' 所包含的内容
comm -2 file1 file2 比较两个文件的内容只删除 'file2' 所包含的内容
comm -3 file1 file2 比较两个文件的内容只删除两个文件共有的部分
```

六、打包和压缩文件

tar命令，对文件进行打包，默认情况并不会压缩，如果指定了相应的参数，它还会调用相应的压缩程序（如gzip和bzip等）进行压缩和解压

```
-c : 新建打包文件
-t : 查看打包文件的内容含有哪些文件名
-x : 解打包或解压缩的功能，可以搭配-C（大写）指定解压的目录，注意-c,-t,-x不能同时出现在同一条命令中
-j : 通过bzip2的支持进行压缩/解压缩
-z : 通过gzip的支持进行压缩/解压缩
-v : 在压缩/解压缩过程中，将正在处理的文件名显示出来
-f filename : filename为要处理的文件
-C dir : 指定压缩/解压缩的目录dir
```

压缩: tar -jcv -f filename.tar.bz2 要被处理的文件或目录名称 查询: tar -jtv -f filename.tar.bz2 解压:
tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

```
bunzip2 file1.bz2 解压一个叫做 'file1.bz2'的文件
bzip2 file1 压缩一个叫做 'file1' 的文件
gunzip file1.gz 解压一个叫做 'file1.gz'的文件
gzip file1 压缩一个叫做 'file1'的文件
gzip -9 file1 最大程度压缩
rar a file1.rar test_file 创建一个叫做 'file1.rar' 的包
rar a file1.rar file1 file2 dir1 同时压缩 'file1', 'file2' 以及目录 'dir1'
rar x file1.rar 解压rar包

zip file1.zip file1 创建一个zip格式的压缩包
unzip file1.zip 解压一个zip格式压缩包
zip -r file1.zip file1 file2 dir1 将几个文件和目录同时压缩成一个zip格式的压缩包
```

七.系统和关机（关机、重启和登出）

```
shutdown -h now 关闭系统(1)
init 0 关闭系统(2)
telinit 0 关闭系统(3)
shutdown -h hours:minutes & 按预定时间关闭系统
shutdown -c 取消按预定时间关闭系统
shutdown -r now 重启(1)
reboot 重启(2)
logout 注销
time 测算一个命令（即程序）的执行时间
```

八、进程相关的命令

jps命令，显示当前系统的java进程情况，及其id号

jps(Java Virtual Machine Process Status Tool)是JDK 1.5提供的一个显示当前所有java进程pid的命令，简单实用，非常适合在linux/unix平台上简单察看当前java进程的一些简单情况。

ps命令，用于将某个时间点的进程运行情况选取下来并输出，process之意

```
-A : 所有的进程均显示出来
-a : 不与terminal有关的所有进程
-u : 有效用户的相关进程
-x : 一般与a参数一起使用，可列出较完整的信息
-l : 较长，较详细地将PID的信息列出

ps aux # 查看系统所有的进程数据
ps ax # 查看不与terminal有关的所有进程
ps -lA # 查看系统所有的进程数据
ps axjf # 查看连同一部分进程树状态
```

kill命令,用于向某个工作（%jobnumber）或者是某个PID（数字）传送一个信号，它通常与ps和jobs命令一起使用

命令格式: kill[命令参数][进程id]

命令参数:

- l 信号，若不加信号的编号参数，则使用“-l”参数会列出全部的信号名称
- a 当处理当前进程时，不限制命令名和进程号的对应关系
- p 指定kill 命令只打印相关进程的进程号，而不发送任何信号
- s 指定发送信号
- u 指定用户

实例1：列出所有信号名称 命令：kill -l 输出：

```
[root@localhost test6]# kill -l
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL
5) SIGTRAP 6) SIGABRT 7) SIGBUS 8) SIGFPE
9) SIGKILL 10) SIGUSR1 11) SIGSEGV 12) SIGUSR2
13) SIGPIPE 14) SIGALRM 15) SIGTERM 16) SIGSTKFLT
17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU
25) SIGXFSZ 26) SIGVTALRM 27) SIGPROF 28) SIGWINCH
29) SIGIO 30) SIGPWR 31) SIGSYS 34) SIGRTMIN
35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3 38) SIGRTMIN+4
39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7 58) SIGRTMAX-6
59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

说明：

只有第9种信号(SIGKILL)才可以无条件终止进程，其他信号进程都有权利忽略。 下面是常用的信号：

```
HUP 1  终端断线
INT 2  中断（同 Ctrl + C）
QUIT 3 退出（同 Ctrl + \）
TERM 15 终止
KILL    9    强制终止
CONT 18    继续（与STOP相反， fg/bg命令）
STOP    19    暂停（同 Ctrl + Z）
```

实例2：得到指定信号的数值

```
[root@localhost test6]# kill -l KILL
[root@localhost test6]# kill -l SIGKILL
[root@localhost test6]# kill -l TERM
[root@localhost test6]# kill -l SIGTERM
[root@localhost test6]#
```

实例3：先用ps查找进程，然后用kill杀掉

```
命令：kill 3268
[root@localhost test6]# ps -ef|grep vim
root 3268  2884  0 16:21 pts/1    00:00:00 vim install.log
root 3370  2822  0 16:21 pts/0    00:00:00 grep vim
[root@localhost test6]# kill 3268
```

实例4：彻底杀死进程

```
命令: kill -9 3268 // -9 强制杀掉进程
```

killall命令，向一个命令启动的进程发送一个信号，用于杀死指定名字的进程

命令格式：killall[命令参数][进程名]

命令参数：

- Z 只杀死拥有scontext 的进程
- e 要求匹配进程名称
- I 忽略小写
- g 杀死进程组而不是进程
- i 交互模式，杀死进程前先询问用户
- l 列出所有的已知信号名称
- q 不输出警告信息
- s 发送指定的信号
- v 报告信号是否成功发送
- w 等待进程死亡
- help 显示帮助信息
- version 显示版本显示

示例

1: 杀死所有同名进程

```
killall nginx  
killall -9 bash
```

2. 向进程发送指定信号

```
killall -TERM nginx 或者 killall -KILL nginx
```

top命令，是Linux下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，类似于Windows的任务管理器。

如何杀死进程：

- (1) 图形化界面的方式
- (2) kill -9 pid （-9表示强制关闭）
- (3) killall -9 程序的名字
- (4) pkill 程序的名字

查看进程端口号：

```
netstat -tunlp|grep 端口号
```