

# linux 常见命令

---

## 删除命令

**rm** [选项] 文件名

选项:

- r 删除目录
- i 删除前逐一询问确定
- f 强制删除

## 复制命令

**cp** [选项] [源文件或目录] [目标目录]

命令或英文原意: **copy**

选项

- r 复制目录
- p 连带文件属性复制
- d 若源文件是链接文件，则复制链接属性
- a 相当于 -pdr

## cut命令

显示文件中每行的制定内容

**cut** [选项] [参数]

选项

- b<输出范围>: 设置输出的字节数或范围
- c<输出范文>: 设置输出的字符数或范围

## 剪切命令（可以给文件改名）

**mv** [原文件或目录] [目标目录]

命令英文原意: **move**

## find 查找命令

**find** [搜索范围] [搜索条件]

```

find / -name install.log
# 避免大范围搜索，会非常耗费系统资源
# find是在系统当中搜索符合条件的文件名。如需要匹配，使用通配符匹配，通配符是完全匹配

find /root -iname install.log    #不区分大小写
find /root -user root            # 按照所有者搜索
find /root -nouser               # 查找没有所有者的文件

find /var/log/ -mtime +10        #查找10天前修改的文件
-10      10天内修改的文件
 10      10天当天修改的文件
+10      10天前修改的文件
atime    文件访问时间
ctime    修改文件属性
mtime    修改文件内容

find ./ -size 25k                #查找当前文件夹下文件大小是25KB的文件
k        KB
M        MB
-25k     小于25KB的文件
 25k     等于25KB的文件
+25k     大于25KB的文件

find ./ -inum 262421            #查找i节点是262421的文件， ll -i可以查看i节点

find /etc/ -size +20k -a -size -50k
# 查找/etc/目录下，大于20KB并且小于50KB的文件
-a and 逻辑与，两个条件都满足
-o or  逻辑或，两个条件满足一个即可

find /etc/ -size +20k -a -size -50k -exec ls -lh {} \;
# 查找/etc/目录下，大于20KB并且小于50KB文件，并显示详细信息
# -exec /ok 命令 {} \; 对搜索结果执行操作
find /etc/ -inum 262421 -exec rm -rf {} \;
# 查找/etc/目录下，i节点为262421的文件，并删除

```

## 通配符

```

*      匹配任意内容
?      匹配任意一个字符
[]     匹配任意一个中括号内的字符

```

如：

```
find /root -name "install.log*"
```

```
find /root -name "[cd]"          #列出root目录下所有以c或d结尾的文件
```

## grep 字符串搜索命令

**grep** [选项] 字符串 文件名  
# 在文件当中匹配符合条件的字符串  
选项:  
    -i 忽略大小写  
    -v 排除制定字符串

**find**命令: 在系统当中搜索符合条件的文件名, 如需要匹配, 使用通配符匹配, 通配符是完全匹配  
**grep**命令: 在文件当中搜索符合条件的字符串, 如果需要匹配, 使用正则表达式进行匹配, 正则表达式为包含匹配

## 压缩与解压缩命令

windows常用压缩格式: .zip .rar .7z  
linux 常用压缩格式: .zip .gz .bz2 .tar.gz .tar.bz2

### .zip格式压缩

zip 压缩文件名 源文件  
#压缩文件  
zip -r 压缩文件名 源文件  
#压缩目录

### .zip解压缩

unzip 压缩文件名

### .gz格式压缩

gzip 源文件  
#压缩为.gz格式的压缩文件, 源文件会消失  
gzip -c 源文件 > 压缩文件  
# 压缩为.gz格式, 源文件保留  
如: gzip -c aa.txt > aa.gz  
gzip -r 目录  
#压缩目录下所有的子文件, 但是不能压缩目录

### .gz格式解压缩

gzip -d 压缩文件 #解压缩文件  
gunzip 压缩文件 #解压缩文件

### .bzip2格式压缩

bzip2 源文件 # 压缩为.bz2格式, 不保留源文件  
bzip2 -k 源文件 # 压缩后保留源文件

注意: .bzip2命令不能压缩目录

### .bzip2解压缩

bzip2 -d 压缩文件 #解压缩, -k保留源文件  
bunzip2 压缩文件 #解压缩, -k保留源文件

### tar打包命令

tar -cvf 打包文件名 源文件  
选项:  
    -c: 打包  
    -v: 显示过程

-f: 指定打包后的文件名  
例如: `tar -cvf abc.tar abc`

**tar解打包命令**  
`tar -xvf 打包文件名`  
选项:  
-x: 解打包  
-t: 测试, 只查看  
例如: `tar -xvf abc.tar`  
`tar -xvf abc.tar -C 指定目录`  
#将abc解压到指定目录下

**.tar.gz压缩格式**  
其实质是先打包为.tar格式, 然后再压缩为.gz格式  
`tar -zcvf 压缩包名.tar.gz 源文件`  
选项:  
-z: 压缩为.tar.gz格式  
`tar -zxvf 压缩包名.tar.gz`  
选项:  
-x: 解压缩.tar.gz格式

**.tar.bz2压缩格式**  
`tar -jcvf 压缩包名.tar.bz2 源文件`  
选项:  
-z: 压缩为.tar.bz2格式  
`tar -jxvf 压缩包名.tar.bz2`  
选项:  
-x: 解压缩.tar.bz2格式  
-t: 测试, 只查看

## 关机与重启命令

`shutdown [选项] 时间`  
选项:  
-c: 取消前一个关机命令  
-h: 关机  
-r: 重启

其他关机命令(不太安全):

`halt`  
`poweroff`  
`init 0`

其他重启命令

`reboot` # 可以用  
`init 6` # 尽量不用

另外, 系统运行级别为:

0 关机  
1 单用户

```
2    不完全多用户，不含NFS服务
3    完全多用户
4    未分配
5    图像界面
6    重启
runlevel  查询系统所处的级别

logout    #退出登陆
```

## 挂载命令

## 脚本执行方式

```
echo 命令
echo -e "字符串"
控制字符：
\a      输出警告音
\b      退格键，也就是向左删除键
\n      换行符
\r      回车键
\t      制表符，也就是Tab键
\v      垂直制表符
\0nnn   按照八进制ASCII码表输出字符，其中0为数字零，nnn是三位八进制数
\xhh    按照十六进制ASCII码表输出字符，其中hh是两位十六进制数

颜色控制
echo -e "\e[1;31m需要输出的字符串\e[0m"
\e      调用颜色
[1      开启颜色选项
\e[0m   关闭颜色选项
#输出颜色
30m = 黑色，    31m = 红色，    32m = 绿色，    33m = 黄色
34m = 蓝色，    35m = 洋红，    36m = 青色，    37m = 白色

写一个脚本
#!/bin/bash
#The first program
echo -e "\e[1;31m这是我的第一个脚本\e[0m"
echo -e "\e[1;34m哈哈\e[0m"
```

## bash的基本功能

---

### 别名与快捷键

```
alias  #查看系统中所有的命令别名
alias 别名='命令'      #设定命令别名,例如: alias dd='ls'
```

```
vim ~/.bashrc    #写入环境变量配置文件中
source .bashrc   #配置文件生效
```

```
unalias 别名      #删除别名
```

命令生效顺序

第一顺位执行用绝对路径或相对路径执行的命令

第二顺位执行别名

第三顺位执行bash的内部命令

第四顺位执行按照\$PASH环境变量定义的目录查找到的第一条命令

常用快捷键：

ctrl+c 强制终止当前命令

ctrl+l 清屏

ctrl+a 光标移动到命令行首

ctrl+e 光标移动到命令行尾

ctrl+u 光标从当前位置删除到行首

ctrl+z 把命令放入后台

ctrl+r 在历史命令中搜索

## 历史命令

history [选项] [历史命令保存文件]

选项：

-c: 清空历史命令

-w: 把缓存中的历史命令写入历史命令文件中， ~/.bash\_history

!n 重复执行第n条历史命令

!! 重复执行上一条命令

!字符串 重复执行最后一条以该字符串开头的命令

## 输出重定向

类型	符号	作用
标准输出重定向	命令 > 文件	覆盖写入文件或设备中
标准输出重定向	命令 >> 文件	追加写入文件或设备中
标准错误输出重定向	错误命令 2>文件	覆盖方式，把错误命令输出到文件或设备中
标准错误输出重定向	错误命令 2>>文件	追加方式，把错误命令输出到文件或设备中

正确输出和错误输出通常保存：

```
命令 > 文件 2>&1    #以覆盖的方式，把正确输出和错误输出都保存到同一文件当中
命令 >> 文件 2>&1    #以追加的方式，把正确输出和错误输出都保存到同一文件当中
命令 &> 文件        #以覆盖的方式，把正确输出和错误输出都保存到同一文件当中
```

```
命令 &>> 文件          #以追加的方式，把正确输出和错误输出都保存到同一文件当中
命令 >>文件1 2>>文件2  #把正确的输出追加到文件1中，把错误的输出追加到文件2中

命令 &>/dev/null        #将命令丢到黑洞，即不输出到屏幕，也不保存到系统
```

## 输入重定向

```
wc [选项] [文件名]
选项:
  -c: 统计字节数
  -w: 统计单词数
  -l: 统计行数
命令<文件  把文件作为命令的输入
命令<<标识符
...
标识符 把标识符之间的内容作为命令的输入
```

## 管道符

### 多命令顺序执行

```
;  命令1 ; 命令2      多个命令顺序执行，没有逻辑关系
&& 命令1 && 命令2     逻辑与，当命令1正确执行时，则命令2才会执行，
                        当命令1执行不正确时，则命令2不会执行
||  命令1 || 命令2     逻辑或，当命令1执行正确时，命令2不执行
                        当命令1执行错误时，命令2才执行

命令1 && echo yes || echo no
```

### 管道符

```
命令1 | 命令2          #命令1的正确输出作为命令2的操作对象

netstat -an | grep ESTABLISHED      #查看远程连接的详细情况
netstat -an | grep ESTABLISHED | wc -l  #查看远程连接的详细个数
```

## 通配符

### 通配符

```
?  # 匹配一个任意字符
*  # 匹配0个或任意多个字符，也就是可以匹配任意内容
```

[ ] # 匹配中括号中任意一个字符。例如[abc]代表一定匹配一个字符，为a,或b,或c  
[-] # 匹配中括号中任意一个字符，-代表一定范围。例如：[a-z]代表匹配一个小写字母。  
[^] # 逻辑非，表示匹配不是中括号内的一个字符。例如：[^0-9]代表匹配一个不是数字的字符

## 其他特殊命令

' ' # 单引号。在单引号中的所有特殊符号，如 \$ 和 ` 都没有特殊含义  
" " # 双引号。在双引号中特殊符号都没有特殊含义，但是 \$ 和 ` 和 \ 是例外，拥有 调用变量的值 、 引用命令 和 转义符 的特殊含义  
` ` # 反引号。反引号括起来的内容是系统命令，在bash中会先执行它，和 \$( ) 作用一样，不过推荐使用 \$( ) ,反引号非常容易看错  
\$( ) # 和反引号作用一样，用来引用系统命令  
# # 在shell脚本中，#开头的行代表注释  
\$ # 用于调用变量的值，如需要调用变量name的值时，需要用 \$name 的方式得到变量的值。  
\ # 转义符。跟着 \ 之后的特殊符号将失去特殊含义，变为普通字符。  
如 \\$ 将输出 \$ 符号，而不当做变量引用。