**操作规范化**

多用相对路径

命令分段执行，看执行对不对

修改之前先备份

修改之后查看结果

永久生效

# 四剑客

## awk命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| awk | 擅长取列 | 默认分隔符为空格或连续的空格 |
| **参数** | **参数说明** | **其他说明** |
| NR | 行号 | NR==数字 等于第几行 |
| && | 并且同时成立 |  |
| {print} | 显示文件内容 | ,(一个逗号代表空格) |
| $ 数字 | 取文件中的第几列 | $0 取整行内容 |
| NF | 一行有多少列 | $NF最后一列 |
| -F"[]" | 指定输入分隔符 |  |
| BEGIN |  |  |

### 实例：去文件的2到4行

命令：awk 'NR<=2 && NR>=4' /tmp/oldboy.txt

[root@oldboy-35 ~]# cat /tmp/oldboy.txt

1

2

3

4

5

[root@oldboy-35 ~]# awk 'NR<=2 && NR>=4' /tmp/oldboy.txt

2

3

4

### 实例：显示文件第一列并加行号

命令：awk '{print NR $1}' test.txt

[root@oldboy ~]# awk '{print NR $1}' test.txt

1tdu1

2tdu2

3tdu3

4tdu4

### 实例：从文件中过滤出“oldboy”和“31333741”字符串

命令：awk -F"[ ,]" '{print $3 " " $NF}' oldboy.txt

[root@oldboy38 tmp]# cat oldboy.txt

I am oldboy,myqq is 31333741

[root@oldboy38 tmp]# awk -F"[ ,]" '{print $3 " " $NF}' oldboy.txt

oldboy 31333741

### 实例：只取文件字母部分，去掉数字的部分。

命令：awk -F"-1" '{print $1}' test.txt

[root@oldboy38 tmp]# cat test.txt

amp-content-display-portlet-1.0.1-20120829.081044-11.war

amp-facebook-post-editor-1.0.1-20120829.084615-1.war

amp-services-portlet-1.0.2-20120829.085733-3.war

amp-facebook-account-editor-1.0.1-20120829.081353-15.war

amp-inbound-processor-hook-1.0.1-20120829.081546-9.war

amp-vcr-editor-portlet-1.0.1-20120829.081801-21.war

[root@oldboy38 tmp]# awk -F"-1" '{print $1".war"}' test.txt

amp-content-display-portlet.war

amp-facebook-post-editor.war

amp-services-portlet.war

amp-facebook-account-editor.war

amp-inbound-processor-hook.war

amp-vcr-editor-portlet.war

### 实例：取ip地址

命令：awk -F"[ :]+" 'NR==2 {print $4}'

[root@oldboy38 tmp]# ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:0C:29:8B:B4:9A

inet addr:192.168.56.128 Bcast:192.168.56.255 Mask:255.255.255.0

inet6 addr: fe80::20c:29ff:fe8b:b49a/64 Scope:Link

[root@oldboy38 tmp]# ifconfig eth0 |awk -F"[ :]+" 'NR==2 {print $4}'

192.168.56.128

### 实例：去除文件里的空格

命令：awk '!/^$/' test.txt

[root@oldboy38 tmp]# cat test.txt

hello

world

!

[root@oldboy38 tmp]# awk '!/^$/' test.txt

hello

world

!

## sed命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| sed | 擅长取行，查找替换 | sed默认输出整个输出 |
| **参数** | **参数说明** | **其他说明** |
| -n | 取消默认输出 |  |
| -i | 对文件内容进行修改 |  |
| -r | 支持扩展正则表达式 |  |
| ; |  |  |
| + | 表示后面的命令对所有没有被选定的行发生作用。 |  |
| **替换标记** | **说明** | **其他说明** |
| \1 | 子串匹配标记 |  |
| & | 保存搜索字符用来替换其他字符 |  |
| s | 替换指定字符 | 只匹配每行的第一个字段 |
| g | 表示行内全面替换 |  |
| d（delete） | 删除，删除选择的行。 |  |
| p（print） | 打印模板块的行。 |  |
| ! |  |  |
| **元字符集** | **说明** | **其他说明** |
| \(..\) | 匹配子串，保存匹配的字符 |  |
| /\s\*/ | 匹配任何空白字符,包括空格、制表符、换页符等等 |  |
|  |  |  |

### 实例：显示时删除/\*\*\*/中的内容，显示剩余内容。

命令：sed '/linux/d' /tmp/oldboy.txt

[root@oldboy-35 ~]# sed '/linux/d' /tmp/oldboy.txt

oldboy

test

### 实例：显示/\*\*\*/中的内容。（-n选项和p命令一起使用表示只打印那些发生替换的行）

命令：sed -n '/linux/p' /tmp/oldboy.txt

[root@oldboy-35 ~]# sed -n '/linux/p' /tmp/oldboy.txt

linux

### 实例：替换文件中内容

命令：sed -i 's#linux#inid#g' /tmp/oldboy.txt

[root@oldboy-35 ~]# sed 's#linux#inid#g' /tmp/oldboy.txt

oldboy

test

inid

[root@oldboy-35 ~]# sed -i 's#linux#inid#g' /tmp/oldboy.txt

[root@oldboy-35 ~]# cat /tmp/oldboy.txt

oldboy

test

inid

注意：先看看改的对不对，再写入。

### 实例：取iｐ地址

命令：sed -n '2s#^.\*r:\(.\*\) B.\*$#\1#p'

[root@oldboy38 tmp]# ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:0C:29:8B:B4:9A

inet addr:192.168.56.128 Bcast:192.168.56.255 Mask:255.255.255.0

inet6 addr: fe80::20c:29ff:fe8b:b49a/64 Scope:Link

[root@oldboy38 tmp]# ifconfig eth0 |sed -n '2s#^.\*r:\(.\*\) B.\*$#\1#p'

192.168.56.128

### 实例：替换s###和s###g的不同

命令：sed 's#oldboy#oldgril#' test.txt

sed 's#oldboy#oldgril#g' test.txt

[root@oldboy38 tmp]# cat test.txt #文件内容

oldboy oldboy

oldboyoldboy oldboy

[root@oldboy38 tmp]# sed 's#oldboy#oldgril#' test.txt #匹配每一行的行首开始匹配，匹配到第一个符合的字段，就会结束，跳到下一行。

oldgril oldboy

oldgriloldboy oldboy

[root@oldboy38 tmp]# sed 's#oldboy#oldgril#g' test.txt #全局替换

oldgril oldgril

oldgriloldgril oldgril

### 实例：去除文件中的空白部分

命令：sed 's/\s\*//g' test.txt

[root@oldboy38 ~]# cat test.txt

aaaaaaaaaa

bbbbbbbbbb

cccc ccccc ccccc

dddddd dddddd dddddd

yyyyyyyy yyyyyyyyyyyyy yyyyyyyyyyyyyy

ooooooo ooooooo oooooo ooooooo

[root@oldboy38 ~]# sed 's/\s\*//g' test.txt

aaaaaaaaaa

bbbbbbbbbb

cccccccccccccc

dddddddddddddddddd

yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy

ooooooooooooooooooooooooooo

### 实例：删掉文件里的空格

命令: sed '/^$/d' test.txt

[root@oldboy38 tmp]# sed '/^$/d' test.txt

hello

world

!

### 实例：修改内容并重定向到其他文件

命令：sed '/^$/d' test.txt >test1.txt

[root@oldboy38 tmp]# cat test.txt

hello

world

!

[root@oldboy38 tmp]# sed '/^$/d' test.txt

hello

world

!

[root@oldboy38 tmp]# sed '/^$/d' test.txt >test1.txt

[root@oldboy38 tmp]# cat test1.txt

hello

world

!

### 实例：给找出的内容添加<>

命令：echo 123456 |sed 's/[0-9]/<&>/g'

[root@oldboy38 ~]# echo 123456 |sed 's/[0-9]/<&>/g'

<1><2><3><4><5><6>

### 实例：备份要修改的文件

命令：sed -i.bak s#.\*#1#g test.txt

[root@oldboy38 tmp]# echo '123456' > test.txt

[root@oldboy38 tmp]# cat test.txt

123456

[root@oldboy38 tmp]# sed -i.bak s#.\*#1#g test.txt

[root@oldboy38 tmp]# cat test.txt

1

[root@oldboy38 tmp]# cat test.txt.bak

123456

### 实例：把修改的内容另存为

命令：sed '3s#1#0#3g' num.txt > num.txt.bak

[root@oldboy38 tmp]# sed '3s#1#0#3g' num.txt > num.txt.bak

[root@oldboy38 tmp]# cat num.txt.bak

1 1 1 1 1

1 1 1 1 1

1 1 0 0 0

1 1 1 1 1

### 实例：修改文件的行和列

命令：sed '3（行）s#1#0#3（列）g' num.txt

[root@oldboy38 tmp]# sed '2s#1#0#2g' num.txt

1 1 1 1 1

1 1 1 1 1

1 1 0 0 0

1 1 1 1 1

## grep命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| grep | 筛选，抓取，过滤 |  |
| **参数** | **参数说明** | **其他说明** |
| -v（invert-match）反向匹配 | 排除 |  |
| -A(after) | 之后 |  |
| -B(before) | 之前 |  |
| -C(context) | 中间 |  |
| -n（line-number） | 给找到的内容加行号 |  |
| -i--ignore-case | 忽略字母大小写 |  |
| egrep=grep -E | 用于在文件内查找多个指定的字符串 | | (或者) 正则 |
| -o | 只输出文件中匹配到的部分。 |  |
| -R |  |  |
| -c | 计数统计找出的行 |  |

### 实例：排除文件第三行只显示前两行

命令：grep -v 'linux' /tmp/oldboy.txt

[root@oldboy-35 ~]# cat /tmp/oldboy.txt

oldboy

test

linux

[root@oldboy-35 ~]# grep -v ' linux ' /tmp/oldboy.txt

oldboy

test

### 实例：显示文件的第3到5行

命令: grep -A 2 "3" /tmp/test/oldboy.txt

[root@oldboy-35 ~]# grep -A 2 "3" /tmp/test/oldboy.txt

3

4

5

命令: grep -B 2 "5" /tmp/test/oldboy.txt

[root@oldboy-35 ~]# grep -B 2 "5" /tmp/test/oldboy.txt

3

4

5

命令：grep -C 1 "4" /tmp/test/oldboy.txt

[root@oldboy-35 ~]# grep -C 1 "4" /tmp/test/oldboy.txt

3

4

5

### 实例：给文件test.txt加行号

命令：grep -n '.' test.txt

[root@oldboy ~]# grep -n '.' test.txt '.'正则表达式 取全部字符

1:tdu1

2:tdu2

3:tdu3

4:tdu4

5:tdu5

### 实例：从文件中过滤出“oldboy”和“31333741”字符串

命令：egrep -o 'oldboy|31333741' oldboy.txt

[root@oldboy38 tmp]# cat oldboy.txt

I am oldboy,myqq is 31333741

[root@oldboy38 tmp]# egrep -o 'oldboy|31333741' oldboy.txt

oldboy

31333741

### 实例：去除文件中的空格

命令：grep -v "^$" test.txt

[root@oldboy38 tmp]# grep -v "^$" test.txt

hello

world

!

## find命令

|  |  |  |
| --- | --- | --- |
| find命令用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则find命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。 | | |
| **参数** | **命令功能** | **其他功能** |
| -type（文件类型） | f：文本文件 |  |
| d：目录 |
| -name | 文件名 |  |
| -exec<执行指令> | 假设find指令的回传值为True，就执行该指令 | 一次一次执行，多次执行会覆盖前面的执行内容 |
| -size<cwbkMG> | 查找符合指定的文件大小的文件 | +(加号大于)-（减号小于） |
| -mtime | 文件最后一次修改时间 |  |
| -7搜索最近七天内被修改过的所有文件 | 7搜索恰好在第七天被修改过的所有文件 | +7搜索超过七天内被修改过的所有文件 |
| -maxdepth | 设置最大目录层级 |  |
| -inum | 查找符合指定的inode编号的文件或目录 |  |
| -user<拥有者名称> | 查找符和指定的拥有者名称的文件或目录 |  |
| -delete | 删除匹配文件 |  |
| -perm<权限数值> | 查找符合指定的权限数值的文件或目录 |  |
| -print0 | 假设find指令的回传值为Ture，就将文件或目录名称列出到标准输出。格式为全部的名称皆在同一行； |  |
| **逻辑运算符** | **功能** |  |
| ！ | 取反 |  |
| -a | 取交集and |  |
| -o | 取并集or |  |

### 实例：查找目录下所有文件

命令：find /tmp -type f

[root@oldboy-35 /]# find /tmp -type f

/tmp/yum.log

### 实例：查找所有目录

命令：find /tmp -type d

[root@oldboy-35 /]# find /tmp -type d

/tmp

/tmp/.ICE-unix

/tmp/data

### 实例：查找名字为“data”的目录

命令：find /tmp -type d -name "data"

[root@oldboy-35 /]# find /tmp -type d -name "data"

/tmp/data

### 实例：查找名字为“data”的目录并显示目录里详细信息

命令1：find /tmp/ -type d -name "data"|xargs ls

[root@oldboy-35 ~]# find /tmp/ -type d -name "data"|xargs ls

oldboy.txt

命令2：find /tmp/ -type d -name "data" -exec ls -l {} \;

[root@oldboy-35 ~]# find /tmp/ -type d -name "data" -exec ls -ld {} \;

drwxr-xr-x. 2 root root 4096 Mar 10 07:34 /tmp/data

命令3：ls -ld $(find /tmp/ -type d -name "data")或ls -ld `find /tmp/ -type d -name "data"`

[root@oldboy-35 ~]# ls -ld $(find /tmp/ -type d -name "data")

drwxr-xr-x. 2 root root 4096 Mar 10 07:34 /tmp/data

[root@oldboy-35 ~]# ls -ld `find /tmp/ -type d -name "data"`

drwxr-xr-x. 2 root root 4096 Mar 10 07:34 /tmp/data

### 实例：找到tmp目录，把别的目录文件复制过来

命令：find /tmp -type d -name "tmp"|xargs cp /tmp/test/data/oldboy.txt

[root@oldboy-35 ~]# find /tmp -type d -name "tmp"|xargs cp /tmp/test/data/oldboy.txt

### 实例：找到所以名字为oldboy.txt的文件，并显示内容

命令：find /tmp -type f -name "oldboy.txt"|xargs cat

[root@oldboy-35 ~]# find /tmp -type f -name "oldboy.txt"|xargs cat

o,piup

o,piup

o,piup

### 实例：找出所有名字中带boy的文件并删除

命令：find /tmp -type f -name "\*boy\*"|xargs rm

[root@oldboy-35 ~]# find /tmp -type f -name "\*boy\*"|xargs rm

注意：删除没有提示

### 实例：只取目录的第一层目录

命令：find -maxdepth 1

[root@oldboy oldboy]# find

./xiaodong

./ext/oldboy

[root@oldboy oldboy]# find -maxdepth 1

./xiaodong

./ext

### 实例：查找7天前的文件

命令：find /app/logs/ -mtime +7

[root@oldboy oldboy]# ls -l /app/logs/

total 0

-rw-r--r-- 1 root root 0 Mar 1 00:00 access\_www\_2017-03-01.log

-rw-r--r-- 1 root root 0 Mar 2 00:00 access\_www\_2017-03-02.log

-rw-r--r-- 1 root root 0 Mar 3 00:00 access\_www\_2017-03-03.log

-rw-r--r-- 1 root root 0 Mar 4 00:00 access\_www\_2017-03-04.log

-rw-r--r-- 1 root root 0 Mar 5 00:00 access\_www\_2017-03-05.log

-rw-r--r-- 1 root root 0 Mar 6 00:00 access\_www\_2017-03-06.log

-rw-r--r-- 1 root root 0 Mar 7 00:00 access\_www\_2017-03-07.log

-rw-r--r-- 1 root root 0 Mar 8 00:00 access\_www\_2017-03-08.log

-rw-r--r-- 1 root root 0 Mar 9 00:00 access\_www\_2017-03-09.log

-rw-r--r-- 1 root root 0 Mar 10 00:00 access\_www\_2017-03-10.log

-rw-r--r-- 1 root root 0 Mar 11 00:00 access\_www\_2017-03-11.log

-rw-r--r-- 1 root root 0 Mar 12 00:00 access\_www\_2017-03-12.log

-rw-r--r-- 1 root root 0 Mar 13 00:00 access\_www\_2017-03-13.log

-rw-r--r-- 1 root root 0 Mar 14 00:00 access\_www\_2017-03-14.log

-rw-r--r-- 1 root root 0 Mar 15 00:00 access\_www\_2017-03-15.log

-rw-r--r-- 1 root root 0 Mar 16 00:00 access\_www\_2017-03-16.log

[root@oldboy oldboy]# find /app/logs/ -mtime +7

/app/logs/access\_www\_2017-03-03.log

/app/logs/access\_www\_2017-03-02.log

/app/logs/access\_www\_2017-03-01.log

/app/logs/access\_www\_2017-03-08.log

/app/logs/access\_www\_2017-03-05.log

/app/logs/access\_www\_2017-03-07.log

/app/logs/access\_www\_2017-03-04.log

/app/logs/access\_www\_2017-03-06.log

### 实例：用inode号查找文件

命令：find -inum "261857"

[root@oldboy ~]# ls -i

283154 anaconda-ks.cfg 259590 install.log 259592 install.log.syslog 261857 oldboy 261843 test.txt

[root@oldboy ~]# find -inum "261857"

./oldboy

### 实例：查找bin目录下大于50k小于100k文件

命令：find /bin -type f -size +50k -a -size -100k

[root@oldboy38 tmp]# find /bin -type f -size +50k -a -size -100k |xargs ls -lh

-rwxr-xr-x. 1 root root 55K May 11 2016 /bin/chgrp

-rwxr-xr-x. 1 root root 52K May 11 2016 /bin/chmod

-rwxr-xr-x. 1 root root 57K May 11 2016 /bin/chown

-rwxr-xr-x. 1 root root 58K May 11 2016 /bin/date

-rwxr-xr-x. 1 root root 54K May 11 2016 /bin/dd

-rwxr-xr-x. 1 root root 93K May 11 2016 /bin/df

-rwxr-xr-x. 1 root root 80K Nov 12 2010 /bin/dumpkeys

-rwxr-xr-x. 2 root root 53K Nov 11 2010 /bin/ed

-rwxr-xr-x. 1 root root 68K Mar 18 2014 /bin/gzip

-rwxr-xr-x. 1 root root 62K May 11 2016 /bin/lsblk

-rwsr-xr-x. 1 root root 76K May 11 2016 /bin/mount

-rwxr-xr-x. 1 root root 88K May 11 2016 /bin/ps

-rwxr-xr-x. 2 root root 53K Nov 11 2010 /bin/red

-rwxr-xr-x. 1 root root 57K May 11 2016 /bin/rm

-rwxr-xr-x. 1 root root 71K Jun 22 2012 /bin/sed

-rwxr-xr-x. 1 root root 97K May 11 2016 /bin/sort

# 文件处理

## dd命令

|  |  |  |
| --- | --- | --- |
| dd命令用于复制文件并对原文件的内容进行转换和格式化处理。dd命令功能很强大的，对于一些比较底层的问题，使用dd命令往往可以得到出人意料的效果。用的比较多的还是用dd来备份裸设备。但是不推荐，如果需要备份oracle裸设备，可以使用rman备份，或使用第三方软件备份，使用dd的话，管理起来不太方便。  建议在有需要的时候使用dd 对物理磁盘操作，如果是文件系统的话还是使用tar backup cpio等其他命令更加方便。另外，使用dd对磁盘操作时，最好使用块设备文件。 | | |
| **命令参数** | **命令功能** | **其他** |
| if | 代表输入文件。如果不指定if，默认就会从stdin中读取输入 |  |
| of | 代表输出文件。如果不指定of，默认就会将stdout作为默认输出。 |  |
| count | 代表被复制的块数。 |  |
| bs | 代表字节为单位的块大小。 |  |

### 实例：复制MBR主引导区第一扇区内容到/tmp

命令：dd if=/dev/sda of=/tmp/mbr.bak bs=512 count=1

[root@oldboy38 tmp]# dd if=/dev/sda of=/tmp/mbr.bak bs=512 count=1

1+0 records in

1+0 records out

512 bytes (512 B) copied, 0.000393117 s, 1.3 MB/s

## touch命令

|  |
| --- |
| touch命令有两个功能：一是用于把已存在文件的时间标签更新为系统当前的时间（默认方式），它们的数据将原封不动地保留下来；二是用来创建新的空文件 |

### 实例：创建文件

命令：touch test.txt

[root@oldboy-35 tmp]# touch test.txt

[root@oldboy-35 tmp]# ls

data oldboy.txt test.txt yum.log

## pwd命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| pwd | 显示当前位置 | print working directory |

### 实例：显示当前位置

命令：pwd

[root@oldboy-35 tmp]# pwd

/tmp

## tree命令需要yum安装

|  |  |  |
| --- | --- | --- |
| tree命令以树状图列出目录的内容。 | | |
| **参数** | **参数说明** | **其他说明** |
| -L(level) 数字 | 只显示几层目录 |  |
| -F | 在每个输出项后追加文件的类型标识符，具体含义：“\*”表示具有可执行权限的普通文件，“/”表示目录，“@”表示符号链接，“|”表示命令管道FIFO，“=”表示sockets套接字。当文件为普通文件时，不输出任何标识符； |  |
| -d | 只显示目录 |  |

### 实例：树形显示目录

命令：tree /tmp/

[root@oldboy-35 ~]# tree /tmp/

/tmp/

├── data

└── yum.log

注意：要单独安装yum install -y tree

### 实例：显示一层目录

命令：tree -L 1 /tmp/

[root@oldboy-35 ~]# tree -L 1 /tmp/

/tmp/

├── 01.log

├── 02.log

├── 03.log

### 实例：只显示目录下的目录

命令：tree -d

[root@oldboy oldboy]# tree

.

├── ext

│   └── oldboy

├── jeacen

├── oldboy

├── test

├── wodi.gz

├── xiaodong

├── xiaofan

├── xingfujie

└── yingsui.gz

6 directories, 4 files

[root@oldboy oldboy]# tree -d

.

├── ext

│   └── oldboy

├── test

├── xiaodong

├── xiaofan

└── xingfujie

6 directories

## mkdir命令

|  |  |  |
| --- | --- | --- |
| mkdir命令用来创建目录。该命令创建由dirname命名的目录。如果在目录名的前面没有加任何路径名，则在当前目录下创建由dirname指定的目录；如果给出了一个已经存在的路径，将会在该目录下创建一个指定的目录。在创建目录时，应保证新建的目录与它所在目录下的文件没有重名。  注意：在创建文件时，不要把所有的文件都存放在主目录中，可以创建子目录，通过它们来更有效地组织文件。最好采用前后一致的命名方式来区分文件和目录。例如，目录名可以以大写字母开头，这样，在目录列表中目录名就出现在前面。  在一个子目录中应包含类型相似或用途相近的文件。例如，应建立一个子目录，它包含所有的数据库文件，另有一个子目录应包含电子表格文件，还有一个子目录应包含文字处理文档，等等。目录也是文件，它们和普通文件一样遵循相同的命名规则，并且利用全路径可以唯一地指定一个目录。 | | |
| **参数** | **参数说明** | **其他说明** |
| -p= parents(父母复数) | 递归创建多个目录 |  |

### 实例：创建一个空目录

命令: mkdir test

### 实例：递归创建多个目录

命令：mkdir -p test1/test2

### 实例：一个目录下创建一系列文件

命令：mkdir /tmp/{1..5}

## ls命令

|  |  |  |
| --- | --- | --- |
| ls命令用来显示目标列表，在Linux中是使用率较高的命令。ls命令的输出信息可以进行彩色加亮显示，以分区不同类型的文件。 | | |
| **参数** | **参数说明** | **其他说明** |
| -l | 显示详细的信息，以长格式 |  |
| -d= directory（目录） | 显示目录本身的信息 |  |
| -h（human-readable） | 人类可读，显示目录大小 |  |
| -a | 显示所有文件 显示隐藏的文件或目录。 |  |
| -t | 用文件和目录的更改时间排序 |  |
| -r=reverse | 以文件名反序排列并输出目录内容列表 |  |
| -F | 在每个输出项后追加文件的类型标识符，具体含义：“\*”表示具有可执行权限的普通文件，“/”表示目录，“@”表示符号链接，“|”表示命令管道FIFO，“=”表示sockets套接字。当文件为普通文件时，不输出任何标识符； |  |
| -i | 显示文件索引节点号（inode）。一个索引节点代表一个文件 |  |

### 实例：显示目录里面的内容

命令：ls

[root@oldboy-35 ~]# ls

anaconda-ks.cfg install.log install.log.syslog test test.txt

### 实例：显示详细的信息，以长格式

命令：ls -l

[root@oldboy-35 ~]# ls -l

total 48

-rw-------. 1 root root 1073 Mar 8 12:08 anaconda-ks.cfg

-rw-r--r--. 1 root root 21764 Mar 8 12:08 install.log

-rw-r--r--. 1 root root 5890 Mar 8 12:06 install.log.syslog

drwxr-xr-x. 2 root root 4096 Mar 9 15:15 test

-rw-r--r--. 1 root root 18 Mar 9 10:12 test.txt

### 实例：仅显示目录名

命令：ls -d test

[root@oldboy-35 ~]# ls -d test

test

### 实例：显示目录的详细信息

命令：ls -ld test

[root@oldboy-35 ~]# ls -ld test

drwxr-xr-x. 2 root root 4096 Mar 9 15:15 test

### 实例：按照更改时间查看目录里文件

命令：ls -lt /tmp/

[root@oldboy ~]# ls -lt /tmp/

total 4

-rw-r--r--. 1 root root 0 Mar 14 15:40 test.txt

-rw-r--r--. 1 root root 20 Mar 14 15:21 oldboy.txt

-rw-------. 1 root root 0 Mar 8 18:56 yum.log

### 实例：反向显示目录里面文件

命令：ls -lr /tmp/

[root@oldboy ~]# ls -lr /tmp/

total 4

-rw-------. 1 root root 0 Mar 8 18:56 yum.log

-rw-r--r--. 1 root root 0 Mar 14 15:40 test.txt

-rw-r--r--. 1 root root 20 Mar 14 15:21 oldboy.txt

### 实例：给目录里的文件追加文件标类型识符

命令：ls -F oldboy/

[root@oldboy ~]# ls -F oldboy/

ext/ jeacen oldboy test/ wodi.gz xiaodong/ xiaofan/ xingfujie/ yingsui.gz

## md5sum

|  |  |  |
| --- | --- | --- |
| md5sum命令采用MD5报文摘要算法（128位）计算和检查文件的校验和。一般来说，安装了Linux后，就会有md5sum这个工具，直接在命令行终端直接运行。 | | |
| **命令参数** | **命令功能** | **其他** |
| -c |  |  |

### 实例：给文件oldboy.txt加密，再写入内容。加密码变化

命令：md5sum oldboy.txt

ef0d194c83a3f92c8ba62518fdbdb3fc oldboy.txt

[root@oldboy38 oldboy]# echo "I" >>oldboy.txt

[root@oldboy38 oldboy]# md5sum oldboy.txt

606313084a9e4d64b3e6c08d52389dfb oldboy.txt

## cd命令

|  |  |  |
| --- | --- | --- |
| cd命令用来切换工作目录至dirname。 其中dirName表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的home directory(也就是刚login时所在的目录)。另外，~也表示为home directory的意思，.则是表示目前所在的目录，..则表示目前目录位置的上一层目录。 | | |
| **参数** | **参数说明** | **其他说明** |
| .. | 返回上级目录 |  |
| - | 返回进入此目录之前所在的目录 |  |
| cd | 进入用户主目录 |  |

### 实例：切换目录

命令：cd /tmp/

[root@oldboy-35 ~]# cd /tmp/

[root@oldboy-35 tmp]#

### 实例：返回上一级目录

命令：cd ..

[root@oldboy ~]# cd /etc/sysconfig

[root@oldboy sysconfig]# cd ..

[root@oldboy etc]#

### 实例：返回进入此目录之前所在的目录

命令：cd -

[root@oldboy etc]# cd -

/etc/sysconfig

[root@oldboy sysconfig]# cd

### 实例：进入用户主目录

命令：cd

[root@oldboy sysconfig]# cd

[root@oldboy ~]#

## cp命令

|  |  |  |
| --- | --- | --- |
| cp命令用来将一个或多个源文件或者目录复制到指定的目的文件或目录。它可以将单个源文件复制成一个指定文件名的具体的文件（复制内容不改变文件名）或一个已经存在的目录下。cp命令还支持同时复制多个文件，当一次复制多个文件时，目标文件参数必须是一个已经存在的目录，否则将出现错误。 | | |
| **参数** | **参数说明** | **其他说明** |
| -r= recursive（循环） | 复制目录及目录里的内容 |  |
| -p | 复制的时候保持属性信息不变 |  |
| -a= archive（存档） | -pdr保持属性信息不变 复制目录 |  |
| -t=target-directory=DIRECTORY | 目录和源（SOURCE）调换位置，目录在前，源在后。 |  |

### 实例：复制文件和目录 默认不能拷贝目录

命令：cp oldboy.txt /tmp/data

[root@oldboy-35 tmp]# cp oldboy.txt /tmp/data

[root@oldboy-35 tmp]# ls data

oldboy.txt

### 实例：保持属性信息不变 复制目录

命令：cp -a data test

[root@oldboy-35 tmp]# cp -a data test

[root@oldboy-35 tmp]# ls

data oldboy.txt test test.txt yum.log

[root@oldboy-35 tmp]# ls test

data

### 实例：把文件用|xargs复制到其他目录

命令：find /tmp/ -type f -name "test.txt"|xargs cp -t /tmp/test

[root@oldboy ~]# find /tmp/ -type f -name "test.txt"|xargs cp -t /tmp/test

[root@oldboy ~]# ls /tmp/test

test.txt

### 实例：给打包的文件加上日期

命令：tar zcf tmp/etc-$(date +%F\_%H-%M-%S).tar.gz etc/

[root@oldboy38 /]# tar zcf tmp/etc-$(date +%F\_%H-%M-%S).tar.gz etc/

[root@oldboy38 /]# ll /tmp/

total 11788

-rw-r--r--. 1 root root 9674389 Apr 1 07:13 etc-2017-04-01\_07-13-03.tar.gz

## mv命令

|  |  |  |
| --- | --- | --- |
| mv命令用来对文件或目录重新命名，或者将文件从一个目录移到另一个目录中。source表示源文件或目录，target表示目标文件或目录。如果将一个文件移到一个已经存在的目标文件中，则目标文件的内容将被覆盖。 mv命令可以用来将源文件移至一个目标文件中，或将一组文件移至一个目标目录中。源文件被移至目标文件有两种不同的结果： 如果目标文件是到某一目录文件的路径，源文件会被移到此目录下，且文件名不变。 如果目标文件不是目录文件，则源文件名（只能有一个）会变为此目标文件名，并覆盖己存在的同名文件。如果源文件和目标文件在同一个目录下，mv的作用就是改文件名。当目标文件是目录文件时，源文件或目录参数可以有多个，则所有的源文件都会被移至目标文件中。所有移到该目录下的文件都将保留以前的文件名。  注意事项：mv与cp的结果不同，mv好像文件“搬家”，文件个数并未增加。而cp对文件进行复制，文件个数增加了。 | | |
| **参数** | **参数说明** | **其他说明** |
| -t | 指定源文件要移动到目标目录 |  |

### 实例：移动文件和目录

命令：mv /tmp/oldboy.txt /tmp/test

[root@oldboy-35 /]# mv /tmp/oldboy.txt /tmp/test

[root@oldboy-35 /]# ls /tmp/test

data oldboy.txt

### 实例：把test.txt改名tt.txt

命令：mv test.txt tt.txt

[root@oldboy ~]# mv test.txt tt.txt

[root@oldboy ~]# ls

anaconda-ks.cfg install.log install.log.syslog oldboy tt.txt

## rm命令

|  |  |  |
| --- | --- | --- |
| rm命令可以删除一个目录中的一个或多个文件或目录，也可以将某个目录及其下属的所有文件及其子目录均删除掉。对于链接文件，只是删除整个链接文件，而原有文件保持不变。  注意：使用rm命令要格外小心。因为一旦删除了一个文件，就无法再恢复它。所以，在删除文件之前，最好再看一下文件的内容，确定是否真要删除。rm命令可以用-i选项，这个选项在使用文件扩展名字符删除多个文件时特别有用。使用这个选项，系统会要求你逐一确定是否要删除。这时，必须输入y并按Enter键，才能删除文件。如果仅按Enter键或其他字符，文件不会被删除。 | | |
| 参数 | 参数说明 | 其他说明 |
| -r= recursive(递归) | (递归) 删除目录 |  |
| -f= force(强制) | (强制) 删除 |  |

### 实例：移除/删除文件或目录 默认只能删除文件

命令：rm /tmp/data/oldboy.txt

[root@oldboy-35 /]# rm /tmp/data/oldboy.txt

rm: remove regular file `/tmp/data/oldboy.txt'? y

[root@oldboy-35 /]#

### 实例：(强制) 删除

命令：rm -f oldboy.txt

[root@oldboy-35 test]# ls

data oldboy.txt

[root@oldboy-35 test]# rm -f oldboy.txt

[root@oldboy-35 test]# ls

data

### 实例：(递归) 删除目录

命令：rm -r data

[root@oldboy-35 test]# rm -r data

rm: descend into directory `data'? y

rm: remove regular file `data/oldboy.txt'? y

rm: remove regular file `data/test.txt'? y

rm: remove directory `data'? y

[root@oldboy-35 test]# ls

[root@oldboy-35 test]# cd ..

[root@oldboy-35 tmp]# ls

data test yum.log

# 查看文件及内容处理命令

## nl命令

|  |  |  |
| --- | --- | --- |
| nl命令读取 file 参数（缺省情况下标准输入），计算输入中的行号，将计算过的行号写入标准输出。在输出中，nl命令根据您在命令行中指定的标志来计算左边的行。输入文本必须写在逻辑页中。每个逻辑页有头、主体和页脚节（可以有空节）。除非使用-p选项，nl 命令在每个逻辑页开始的地方重新设置行号。可以单独为头、主体和页脚节设置行计算标志（例如，头和页脚行可以被计算然而文本行不能）。其默认的结果与cat -n有点不太一样， nl 可以将行号做比较多的显示设计，包括位数与是否自动补齐0等等的功能。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |
|  |  |  |

### 实例：给test.txt文件加行号

命令：nl test.txt

[root@oldboy ~]# cat test.txt

tdu1

tdu2

tdu3

tdu4

tdu5

[root@oldboy ~]# nl test.txt

1 tdu1

2 tdu2

3 tdu3

4 tdu4

5 tdu5

## seq命令

|  |  |  |
| --- | --- | --- |
| seq命令用于产生从某个数到另外一个数之间的所有整数。 | | |
| **命令参数** | **命令功能** | **其他** |
| -w | 在列前添加0 使得宽度相同 |  |

### 实例：生成从数字内容为1-3的文件

命令：seq 3 > test.txt

[root@CentOS6 ~]# seq 3 > test.txt

[root@CentOS6 ~]# cat test.txt

1

2

3

### 实例：在数字列表加0，只对个位数有作用。

命令：seq -w 10

[root@oldboy38 tmp]# seq -w 10

01

02

03

04

05

06

07

08

09

10

## less命令

|  |  |  |
| --- | --- | --- |
| less命令的作用与more十分相似，都可以用来浏览文字档案的内容，不同的是less命令允许用户向前或向后浏览文件，而more命令只能向前浏览。用less命令显示文件时，用PageUp键向上翻页，用PageDown键向下翻页。要退出less程序，应按Q键。 | | |
| 命令参数 | 命令功能 | 其他 |
| -N | 每一行行首显示行号 |  |
|  |  |  |

### 实例：让文件显示行号

命令：less -N test.txt

[root@oldboy ~]# cat test.txt

tdu1

tdu2

tdu3

tdu4

tdu5

[root@oldboy ~]# less -N test.txt

1 tdu1

2 tdu2

3 tdu3

4 tdu4

5 tdu5

## vim命令

|  |  |  |
| --- | --- | --- |
| 命令 | 命令功能 | 命令组合 |
| vim | 全屏幕纯文本编辑器 |  |

编辑模式：

|  |  |  |
| --- | --- | --- |
| 参数 | 参数说明 | 其他说明 |
| i= insert(插入；嵌入) | 写入/插入内容 |  |
| o | 在目前光标所在的下一行处插入新的一行 |  |
| O |  |  |
| ESC | 退出编辑模式 |  |

命令模式：

|  |  |  |
| --- | --- | --- |
| 参数 | 参数说明 | 其他说明 |
| :w | 保存当前文档，但不退出 |  |
| :w! | 强制保存当前文档，但不退出 |  |
| :q | 退出文档 |  |
| :q! | 强制退出 |  |
| :wq= write quit | 保存并退出 |  |
| :set nu | 显示行号 |  |
| :set nonu | 不显示行号 |  |
| :1,15copy(或co)15 | 复制1到15行到15行后 |  |
| :1,15move(或mo)15 | 移动1到15行到15行后 |  |
| :help | 查看帮助 | :help copy |
| / | 搜索 | n切换下一个，N切换上一个 |

普通模式：

|  |  |  |
| --- | --- | --- |
| 参数 | 命令功能 | 其他功能 |
| gg | 跳到该文本的行首 |  |
| G（shift+g） | 跳到该文本的行尾 |  |
| ngg | 跳到指定n行（n=数字） |  |
| 0 | 行首 |  |
| $ | 行尾 |  |
| yy | 复制所在一行 |  |
| nyy | 复制指定n行 |  |
| p | 粘帖下一行 |  |
| P | 粘帖上一行 |  |
| dd | 剪切所在一行 |  |
| ndd | 剪切指定n行 |  |
| dG | 剪切光标所在行，到文件的最后一行 |  |
| D | 剪切光标所在位置到行尾 |  |
| u | 恢复前一个操作 |  |
| ctrl+r | 恢复后一个操作 |  |
|  |  |  |
|  |  |  |

## cat命令

|  |  |  |
| --- | --- | --- |
| cat命令连接文件并打印到标准输出设备上，cat经常用来显示文件的内容，类似于下的type命令。 注意：当文件较大时，文本在屏幕上迅速闪过（滚屏），用户往往看不清所显示的内容。因此，一般用more等命令分屏显示。为了控制滚屏，可以按Ctrl+S键，停止滚屏；按Ctrl+Q键可以恢复滚屏。按Ctrl+C（中断）键可以终止该命令的执行，并且返回Shell提示符状态。 | | |
| 参数 | 参数说明 | 其他说明 |
| -n--number | 有1开始对所有输出的行数编号 |  |
| -b--number-nonblank | 和-n相似，只不过对于空白行不编号 |  |
| -A | 显示不可打印字符，行尾显示“$”； |  |

### 实例：显示文件内容

命令：cat test.txt

[root@oldboy-35 tmp]# cat test.txt

I am studying linux

### 实例：多行追加（先写好再复制）

命令： cat >>oldboy.txt <<eo

> iiii

> oooo

> jjjj

> eo

[root@oldboy-35 tmp]# cat oldboy.txt

iiii

oooo

jjjj

### 显示文件行号和内容，包括空格。

命令：cat -n test.txt

[root@oldboy38 ~]# cat -n test.txt

1

2 bbbbbbbbbb

3 cccccccccccccc

4 dddddd dddddd

5

6 ooooooo

### 显示文件行号和内容，不包括空格。

命令：cat -b test.txt

[root@oldboy38 ~]# cat -b test.txt

1 bbbbbbbbbb

2 cccccccccccccc

3 dddddd dddddd

4 ooooooo

## head命令

|  |  |  |
| --- | --- | --- |
| head命令用于显示文件的开头的内容。在默认情况下，head命令显示文件的头10行内容。 | | |
| 参数 | 参数说明 | 其他说明 |
| -n数字或-数字 | 取几行就写数字几 |  |
| -q | 不显示文件名的头信息。 |  |

### 实例：取文件前两行

命令：head -2 /tmp/oldboy.txt

[root@oldboy-35 ~]# head -2 /tmp/oldboy.txt

oldboy

test

### 实例：取多个文件头一行，不显示文件名。

命令：head -1 -q test.txt test1.txt

[root@oldboy38 tmp]# head -1 test.txt test1.txt

==> test.txt <==

123456

==> test1.txt <==

123456

[root@oldboy38 tmp]# head -1 -q test.txt test1.txt

123456

123456

## tail命令

|  |  |  |
| --- | --- | --- |
| ail命令用于输入文件中的尾部内容。tail命令默认在屏幕上显示指定文件的末尾10行。如果给定的文件不止一个，则在显示的每个文件前面加一个文件名标题。如果没有指定文件或者文件名为“-”，则读取标准输入。  注意：如果表示字节或行数的N值之前有一个”+”号，则从文件开头的第N项开始显示，而不是显示文件的最后N项。N值后面可以有后缀：b表示512，k表示1024，m表示1 048576(1M)。 | | |
| 命令 | 命令功能 | 其他 |
| -n数字 -数字 | 取几行写多少数字 |  |
| -f（可以用tailf代替整个命令） | 显示文件最新追加的内容 |  |

### 实例：取文件后两行

命令：tail -2 test.txt

[root@oldboy38 ~]# cat test.txt

test

liyao

oldboy

[root@oldboy38 ~]# tail -2 test.txt

liyao

oldboy

## ln命令

|  |  |  |
| --- | --- | --- |
| ln命令用来为文件创件连接，连接类型分为硬连接和符号连接两种，默认的连接类型是硬连接。如果要创建符号连接必须使用"-s"选项。  注意：符号链接文件不是一个独立的文件，它的许多属性依赖于源文件，所以给符号链接文件设置存取权限是没有意义的。 | | |
| **命令参数** | **命令功能** | **其他** |
| -s或——symbolic（象征的） | 对源文件建立符号连接，而非硬连接 | 符号链接=软链接 |

### 实例：建立文件oldboy.t的硬链接

命令：ln oldboy.t oldboy.t\_hard

[root@oldboy38 tmp]# ln oldboy.t oldboy.t\_hard

259588 -rw-r--r--. 2 root root 0 Mar 23 13:56 oldboy.t

259588 -rw-r--r--. 2 root root 0 Mar 23 13:56 oldboy.t\_hard

### 实例：建立文件oldboy.t的符号链接

命令：ln -s oldboy.t oldboy.t\_symbolic

[root@oldboy38 tmp]# ln -s oldboy.t oldboy.t\_symbolic

[root@oldboy38 tmp]# ll -i

total 0

259588 -rw-r--r--. 2 root root 0 Mar 23 13:56 oldboy.t

259588 -rw-r--r--. 2 root root 0 Mar 23 13:56 oldboy.t\_hard

262170 lrwxrwxrwx. 1 root root 8 Mar 23 13:58 oldboy.t\_symbolic -> oldboy.t

## cut

|  |  |  |
| --- | --- | --- |
| cut命令用来显示行中的指定部分，删除文件中指定字段。cut经常用来显示文件的内容，类似于下的type命令。  说明：该命令有两项功能，其一是用来显示文件的内容，它依次读取由参数file所指 明的文件，将它们的内容输出到标准输出上；其二是连接两个或多个文件，如cut fl f2 > f3将把文件fl和几的内容合并起来，然后通过输出重定向符“>”的作用，将它们放入文件f3中。  当文件较大时，文本在屏幕上迅速闪过（滚屏），用户往往看不清所显示的内容。因此，一般用more等命令分屏显示。为了控制滚屏，可以按Ctrl+S键，停止滚屏；按Ctrl+Q键可以恢复滚屏。按Ctrl+C（中断）键可以终止该命令的执行，并且返回Shell提示符状态。 | | |
| 命令参数 | 命令功能 | 其他 |
| -d--delimiter=DELIM | 仅显示行中指定直接范围的内容 |  |
| -f--fields=LIST | 显示指定字段的内容 | ,（和）-（到） |
| -c--characters=LIST | 仅显示行中指定范围的字符 |  |

### 实例：以空格为分隔，显示第1，2字段内容

命令：cut -d " " -f1,2 oldboy.txt

[root@oldboy /]# cat oldboy.txt

I am oldboy,myqq is 31333741

[root@oldboy /]# cut -d " " -f1,2 oldboy.txt

I am

### 实例：按照字符来显示指定部分

命令：cut -c '1-4' oldboy.txt

[root@oldboy /]# cut -c '1-4' oldboy.txt

I am

### 实例：取多个字符，要么单引号加空格。要么逗号隔开

[root@oldboy tmp]# cat old.txt

I am sutdying,linux

[root@oldboy tmp]# cut -c 1,15-19 old.txt

Ilinux

[root@oldboy tmp]# cut -c '1 15-19' old.txt

Ilinux

## wc

|  |  |  |
| --- | --- | --- |
| wc命令用来计算数字。利用wc指令我们可以计算文件的Byte数、字数或是列数，若不指定文件名称，或是所给予的文件名为“-”，则wc指令会从标准输入设备读取数据。 | | |
| 命令参数 | 命令功能 | 其他 |
| -c--bytes | 只显示Bytes数 |  |
| -w--chars | 只显示字数 |  |
| -l--lines | 只显示列数 |  |
| -L | 只显示最长字符的字符个数 |  |

### 实例：显示文件的行，字数，Byte数

命令：wc oldboy.txt

[root@oldboy /]# wc oldboy.txt

1 5 29 oldboy.txt 1行5个单词29个字符

# 文件压缩及解压命令

## tar命令

|  |  |  |
| --- | --- | --- |
| tar命令可以为linux的文件和目录创建档案。利用tar，可以为某一特定文件创建档案（备份文件），也可以在档案中改变文件，或者向档案中加入新的文件。tar最初被用来在磁带上创建档案，现在，用户可以在任何设备上创建档案。利用tar命令，可以把一大堆的文件和目录全部打包成一个文件，这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。  首先要弄清两个概念：打包和压缩。打包是指将一大堆文件或目录变成一个总的文件；压缩则是将一个大的文件通过一些压缩算法变成一个小文件。  为什么要区分这两个概念呢？这源于Linux中很多压缩程序只能针对一个文件进行压缩，这样当你想要压缩一大堆文件时，你得先将这一大堆文件先打成一个包（tar命令），然后再用压缩程序进行压缩（gzip bzip2命令）。 | | |
| 命令参数 | 命令功能 | 其他 |
| -z--gzip或--ungzip | 通过gzip指令处理备份文件 |  |
| -c--建立新的备份文件 | 建立新的备份文件 |  |
| -v--verbose | 显示指令执行过程 |  |
| -f--file | 指定备份文件 |  |
| -t--list | 列出备份文件的内容 |  |
| -x--extract | 从备份文件中还原文件 |  |
| --exclude=<文件位置> | 排除符合范本样式的文件 |  |
| --exclude-from=<txt文件> | 根据一个名单排除多个符合范本样式的文件 |  |
| -C <目录> | 这个选项用在解压缩，若要在特定目录解压缩，可以使用这个选项 |  |
| -N<日期格式> | 只将较指定日期更新的文件保存到备份文件里 |  |
| -r | 添加文件到已经压缩的文件 |  |

### 实例：把root目录打包放入tmp目录

命令：tar zcf tmp/tmp.tar.gz root/

[root@oldboy /]# tar zcf tmp/root.tar.gz root/

[root@oldboy /]# ls tmp/

oldboy.txt test.txt root.tar.gz yum.log

### 实例：排除root目录下的tt.txt文件并打包放入tmp目录

命令：tar zcf tmp/tmp.tar1.gz --exclude=tt.txt root

[root@oldboy /]# tar zcf tmp/root.tar1.gz --exclude=tt.txt root

[root@oldboy /]# ls /tmp/

oldboy.txt test.txt root.tar1.gz root.tar.gz yum.log

[root@oldboy /]# tar tf tmp/root.tar1.gz|grep 'tt.txt'

[root@oldboy /]#

### 实例：把root.tar.gz解压到oldboy目录下

命令：tar xf /tmp/tmp.tar.gz -C oldboy/

[root@oldboy /]# tar xf /tmp/tmp.tar.gz -C oldboy/

[root@oldboy /]# ls oldboy/

1. sh 02.sh 03.sh 04.sh 05.sh 06.sh 07.sh 08.sh 09.sh 10.sh 11.sh 12.sh 13.sh root test

# 文件权限属性设置

## chmod

|  |  |  |
| --- | --- | --- |
| chmod命令用来变更文件或目录的权限。在UNIX系统家族里，文件或目录权限的控制分别以读取、写入、执行3种一般权限来区分，另有3种特殊权限可供运用。用户可以使用chmod指令去变更文件与目录的权限，设置方式采用文字或数字代号皆可。符号连接的权限无法变更，如果用户对符号连接修改权限，其改变会作用在被连接的原始文件。 | | |
| **命令参数** | **命令功能** | **其他** |
| -R | 递归处理，将指令目录下的所有文件及子目录一并处理； |  |
|  |  |  |
| 权限范围 | 解释 |  |
| u--User | 即文件或目录的拥有者 |  |
| g--Group | 即文件或目录的所属群组 |  |
| o--Other | 除了文件或目录拥有者或所属群组之外，其他用户皆属于这个范围 |  |
| a-All | 即全部的用户，包含拥有者，所属群组以及其他用户 |  |
| r | 读取权限，数字代号为“4” |  |
| w | 写入权限，数字代号为“2” |  |
| x | 执行或切换权限，数字代号为“1” |  |
| - | 不具任何权限，数字代号为“0” |  |
| s | 特殊功能说明：变更文件或目录的权限 |  |
| 操作符号 | 意义 |  |
| + | 添加某个权限 |  |
| - | 取消某个权限 |  |
| = | 表示赋予给定的权限，同时取消文档以前的所有权限。 |  |

## chown

|  |  |  |
| --- | --- | --- |
| chown命令改变某个文件或目录的所有者和所属的组，该命令可以向某个用户授权，使该用户变成指定文件的所有者或者改变文件所属的组。用户可以是用户或者是用户D，用户组可以是组名或组id。文件名可以使由空格分开的文件列表，在文件名中可以包含通配符。 只有文件主和超级用户才可以便用该命令。 | | |
| **命令参数** | **命令功能** | **其他** |
| -R |  |  |

## chattr命令

|  |  |  |
| --- | --- | --- |
| chattr命令用来改变文件属性。这项指令可改变存放在ext2文件系统上的文件或目录属性，这些属性共有以下8种模式： | | |
| **命令参数** | **命令功能** | **其他** |
| i | 不得任意更动文件或目录 |  |
| a | 让文件或目录仅供附加用途； |  |
| 选项 | 含义 |  |
| + | 开启文件或目录的该项属性 |  |
| - | 关闭文件或目录的该项属性 |  |
| = | 指定文件或目录的该项属性 |  |
|  |  |  |
|  |  |  |

### 实例：锁定test.txt不能修改

命令：chattr +i test.txt

[root@oldboy38 tmp]# chattr +i test.txt

[root@oldboy38 tmp]# echo "11" > test.txt

-bash: test.txt: Permission denied

### 实例：只能追加内容到test.txt不能更改

命令：chattr +a test.txt

[root@oldboy38 tmp]# chattr +a test.txt

[root@oldboy38 tmp]# echo "11" >> test.txt

[root@oldboy38 tmp]# cat test.txt

11

11

[root@oldboy38 tmp]# echo "11" > test.txt

-bash: test.txt: Operation not permitted

# 信息显示命令

## file命令

|  |  |  |
| --- | --- | --- |
| file命令用来探测给定文件的类型。file命令对文件的检查分为文件系统、魔法幻数检查和语言检查3个过程。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：查看init文件类型

命令：file init

[root@CentOS6 etc]# file init

init: directory

## stat

|  |  |  |
| --- | --- | --- |
| stat命令用于显示文件的状态信息。stat命令的输出信息比ls命令的输出信息要更详细。 | | |
| **命令参数** | **命令功能** | **其他** |
| -c |  | %a |

### 实例：显示tmp目录具体状态信息

命令：stat /tmp/

[root@oldboy38 ~]# stat /tmp/

File: `/tmp/'

Size: 4096 Blocks: 8 IO Block: 4096 directory

Device: 803h/2051d Inode: 259587 Links: 7

Access: (1777/drwxrwxrwt) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2017-03-20 20:40:19.410187495 +0800

Modify: 2017-03-20 19:00:26.382200890 +0800

Change: 2017-03-20 19:00:26.382200890 +0800

## df命令

|  |  |  |
| --- | --- | --- |
| df命令用于显示磁盘分区上的可使用的磁盘空间。默认显示单位为KB。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。 | | |
| 命令参数 | 命令功能 | 其他 |
| -h | 以可读性较高的方式来显示信息 |  |
| -i | 显示inode的信息 |  |

### 实例：查询磁盘剩余空间

命令：df -h

[root@CentOS6 ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 6.9G 1.5G 5.1G 22% /

tmpfs 238M 0 238M 0% /dev/shm

/dev/sda1 190M 33M 147M 19% /boot

/dev/sr0 3.7G 3.7G 0 100% /mnt

### 实例：查询inode使用量

命令：df -i

[root@oldboy38 tmp]# df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/sda3 462384 55337 407047 12% /

tmpfs 60751 1 60750 1% /dev/shm

/dev/sda1 51200 39 51161 1% /boot

## uname命令

|  |  |  |
| --- | --- | --- |
| uname命令用于打印当前系统相关信息（内核版本号、硬件架构、主机名称和操作系统类型等）。 | | |
| **命令参数** | **命令功能** | **其他** |
| -r（kernel） | 显示内核信息 |  |
| -m | 显示32或64位 |  |

### 实例 现实linux内核

* 命令：uname
* 命令：uname -r
* 命令：uname -m

[root@CentOS6 ~]# uname

Linux

[root@CentOS6 ~]# uname -r

2.6.32-642.el6.x86\_64

[root@CentOS6 ~]# uname -m

x86\_64

useradd

## hostname

|  |  |  |
| --- | --- | --- |
| hostname命令用于显示和设置系统的主机名称。环境变量HOSTNAME也保存了当前的主机名。在使用hostname命令设置主机名后，系统并不会永久保存新的主机名，重新启动机器之后还是原来的主机名。如果需要永久修改主机名，需要同时修改/etc/hosts和/etc/sysconfig/network的相关内容。 | | |
| **命令参数** | **命令功能** | **其他** |
| -i | 显示主机的ip地址 |  |
| -d | 显示DNS域名 |  |

### 实例：显示主机名

[root@oldboy /]# hostname

oldboy

## date

|  |  |  |
| --- | --- | --- |
| date命令是显示或设置系统时间与日期 | | |
| **命令参数** | **命令功能** | **其他** |
| +%F |  |  |
| -s<字符串> | 显示字符串所指的日期与时间。字符串前后必须加上双引号； |  |
| -d<字符串> | 显示字符串所指的日期与时间。字符串前后必须加上双引号； |  |
| %m | 月份（1-12） |  |
| %w | 一个星期的第几天（0代表星期天） |  |
| %d | 一个月的第几天（01~31） |  |
| %M | 分钟（00~59） |  |
| %S | 显示秒（00~59） |  |
| %H | 小时，24小时制（00~23） |  |
| year  month  day  hour  min[ute]  sec[ond] | 年  月  日  小时  分  秒 |  |

### 实例：显示主机时间

命令：date

[root@oldboy /]# date

Thu Mar 16 16:56:28 CST 2017

### 实例：按照年月日显示日期

命令：date +%F

[root@CentOS6 ~]# date +%F

2017-03-27

### 实例：只取年月日

命令：date +%H-%m-%d

[root@CentOS6 ~]# date +%H-%m-%d

18-03-27

### 实例：修改时间

命令：date -s '20170301' +%F

[root@CentOS6 ~]# date -s '20170301' +%F

2017-03-01

### 实例：显示前3天或后3天的时间

命令：date -d '3 day' 后3天，date -d '-3 day'前3天

[root@CentOS6 ~]# date -d '3 day'

Fri Mar 31 15:09:57 CST 2017

[root@CentOS6 ~]# date -d '-3 day'

Sat Mar 25 15:10:19 CST 2017

### 实例：显示小时分钟秒

命令：date +%H-%M-%S

[root@oldboy38 tmp]# date +%H-%M-%S

16-23-13

### 实例：给打包的文件加日期

命令：tar zcf tmp/etc-`date +%F`.tar.gz etc/

[root@oldboy38 /]# tar zcf tmp/etc-`date +%F`.tar.gz etc/

[root@oldboy38 tmp]# ll

total 28356

drwxr-xr-x. 78 root root 4096 Mar 26 05:33 etc

-rw-r--r--. 1 root root 9674202 Mar 28 22:35 etc-2017-03-28.tar.gz

# 搜索文件命令

## which

|  |  |  |
| --- | --- | --- |
| which命令用于查找并显示给定命令的绝对路径，环境变量PATH中保存了查找命令时需要遍历的目录。which指令会在环境变量$PATH设置的目录里查找符合条件的文件。也就是说，s使用which命令，就可以看到某个系统命令是否存在，以及执行的到底是哪一个位置的命令。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：查找mv的位置

命令：which mv

[root@oldboy /]# which mv

alias mv='mv -i'

/bin/mv

# 用户管理命令

## useradd命令

|  |  |  |
| --- | --- | --- |
| useradd命令用于Linux中创建的新的系统用户。useradd可用来建立用户帐号。帐号建好之后，再用passwd设定帐号的密码．而可用userdel删除帐号。使用useradd指令所建立的帐号，实际上是保存在/etc/passwd文本文件中。  在Slackware中，adduser指令是个script程序，利用交谈的方式取得输入的用户帐号资料，然后再交由真正建立帐号的useradd命令建立新用户，如此可方便管理员建立用户帐号。在Red Hat Linux中，adduser命令则是useradd命令的符号连接，两者实际上是同一个指令。 | | |
| **命令参数** | **命令功能** | **其他** |
| -s<shell> | 指定用户登入后所使用的shell； |  |
| -u | 指定用户id |  |
| -g<群组> | 指定用户所属的群组 |  |
| -G<群组> | 指定用户所属的附加群组 |  |
| -c<备注> | 加上备注文字。备注文字会保存在passwd的备注栏位中 |  |
| -d<登入目录> | 指定用户登入时的启始目录 |  |
| -D | 变更预设值 | 编辑/etc/default/useradd |
| -M | 不要自动建立用户的登入目录 |  |

### 实例：建立一个名字为hellhappy的普通用户

命令：useradd hellhappy

[root@CentOS6 ~]# useradd hellhappy

### 实例：添加用户alex666，UID指定为999，归属为用户组 root、oldboy、sa成员，并设置其用户注释信息为HandsomeBoy，设置家目录为/alex666，其shell类型为/bin/sh

命令：useradd -u 999 -G root,oldboy,sa -c "HandsomeBoy" -d /alex666 -s /bin/sh alex666

[root@oldboy38 ~]# useradd -u 999 -G root,oldboy,sa -c "HandsomeBoy" -d /alex666 -s /bin/sh alex666

[root@oldboy38 ~]# id alex666

uid=999(alex666) gid=1006(alex666) groups=1006(alex666),0(root),1004(sa),1005(oldboy)

[root@oldboy38 ~]# tail -1 /etc/passwd

alex666:x:999:1006:HandsomeBoy:/alex666:/bin/sh

### 实例：创建一个新用户hell100指定UID为556，指定组为hell100

命令：useradd -u 556 -g hell100 hell100

[root@oldboy38 ~]# useradd -u 556 -g hell100 hell100

[root@oldboy38 ~]# id hell100

uid=556(hell100) gid=886(hell100) groups=886(hell100)

### 实例：更用户shell命令解释器

命令：useradd -D -s /bin/sh

[root@oldboy38 ~]# useradd -D -s /bin/sh

[root@oldboy38 ~]# useradd -D

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/sh

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes

## groupadd

|  |  |  |
| --- | --- | --- |
| groupadd命令用于创建一个新的工作组，新工作组的信息将被添加到系统文件中。 | | |
| **命令参数** | **命令功能** | **其他** |
| -g | 指定新建工作组的id |  |

### 实例：增加一个叫hellhome的组

命令：groupadd hellhome

[root@oldboy38 tmp]# groupadd hellhome

### 实例：创建一个叫hell100的组，GID指定为886

命令：groupadd -g 886 hell100

[root@oldboy38 ~]# groupadd -g 886 hell100

[root@oldboy38 ~]# tail -1 /etc/group

hell100:x:886:

## usermod

|  |  |  |
| --- | --- | --- |
| usermod命令用于修改用户的基本信息。usermod命令不允许你改变正在线上的使用者帐号名称。当usermod命令用来改变user id，必须确认这名user没在电脑上执行任何程序。你需手动更改使用者的crontab档。也需手动更改使用者的at工作档。采用NIS server须在server上更动相关的NIS设定. | | |
| **命令参数** | **命令功能** | **其他** |
| -g<群组> | 修改用户所属的群组 |  |

### 实例：更改用户hellhappy的组

命令：usermod -g hellhome hellhappy

[root@oldboy38 tmp]# usermod -g hellhome hellhappy

[root@oldboy38 tmp]# id hellhappy

uid=500(hellhappy) gid=504(hellhome) groups=504(hellhome)

## userdel

|  |  |  |
| --- | --- | --- |
| userdel命令用于删除给定的用户，以及与用户相关的文件。若不加选项，则仅删除用户帐号，而不删除相关文件。 | | |
| **命令参数** | **命令功能** | **其他** |
| -r | 删除用户的同时，删除与用户相关的所有文件。 |  |

### 实例：删除用户test

命令：userdel test

[root@oldboy38 tmp]# userdel test

## passwd命令

|  |  |  |
| --- | --- | --- |
| passwd命令用于设置用户的认证信息，包括用户密码、密码过期时间等。系统管理者则能用它管理系统用户的密码。只有管理者可以指定用户名称，一般用户只能变更自己的密码。 | | |
| **命令参数** | 命令功能 | 其他 |
| --stdin | 非交互式设置密码 |  |

### 实例：设置普通用户hellhappy的密码

命令：passwd hellhppy

[root@CentOS6 ~]# passwd hellhappy

Changing password for user hellhappy.

New password:

BAD PASSWORD: it is too simplistic/systematic

BAD PASSWORD: is too simple

Retype new password:

passwd: all authentication tokens updated successfully.

### 非交互式修改hellhappy密码并删除历史记录

命令：echo 123456 |passwd --stdin hellhappy

[root@CentOS6 ~]# echo 123456 |passwd --stdin hellhappy && history -c

Changing password for user hellhappy.

passwd: all authentication tokens updated successfully.

## chage命令

|  |  |  |
| --- | --- | --- |
| chage命令是用来修改帐号和密码的有效期限。 | | |
| **命令参数** | **命令功能** | **其他** |
| -l | 例出当前的设置。由非特权用户来确定他们的密码或帐号何时过期。 |  |
| -m | 密码可更改的最小天数。为零时代表任何时候都可以更改密码。 |  |
| -M | 密码保持有效的最大天数。 |  |
| -W | 用户密码到期前，提前收到警告信息的天数。 |  |
| -E | 帐号到期的日期。过了这天，此帐号将不可用。 |  |
| -d | 上一次更改的日期。 |  |
| -i | 停滞时期。如果一个密码已过期这些天，那么此帐号将不可用。 |  |

### 实例：查看oldboy用户当前设置

命令：chage -l oldboy

[root@oldboy38 ~]# chage -l oldboy

Last password change : Apr 05, 2017 最后一次密码改变时间

Password expires : never 密码过期时间

Password inactive : never 密码失效时间

Account expires : never 账户过期时间

Minimum number of days between password change : 0 两次改变密码之间相距的最小天数

Maximum number of days between password change : 99999 两次改变密码之间相距的最大天数

Number of days of warning before password expires : 7 在密码过期之前警告的天数

### 实例：下面要求oldboy用户7天内不能更改密码，60天以后必须修改密码，过期前10天通知oldboy用户，过期后30天后禁止用户登陆。

命令：chage -m 7 -M 60 -W 10 -E 30 oldboy

[root@oldboy38 ~]# chage -m 7 -M 60 -W 10 -E 30 oldboy

[root@oldboy38 ~]# chage -l oldboy

Last password change : Apr 05, 2017

Password expires : Jun 04, 2017

Password inactive : never

Account expires : Jan 31, 1970

Minimum number of days between password change : 7

Maximum number of days between password change : 60

Number of days of warning before password expires : 10

## id

|  |  |  |
| --- | --- | --- |
| id命令可以显示真实有效的用户ID(UID)和组ID(GID)。UID 是对一个用户的单一身份标识。组ID（GID）则对应多个UID。id命令已经默认预装在大多数Linux系统中。要使用它，只需要在你的控制台输入id。不带选项输入id会显示如下。结果会使用活跃用户。 当我们想知道某个用户的UID和GID时id命令是非常有用的。一些程序可能需要UID/GID来运行。id使我们更加容易地找出用户的UID以GID而不必在/etc/group文件中搜寻。如往常一样，你可以在控制台输入man id进入id的手册页来获取更多的详情。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：查询用户信息

命令：id oldboy

[root@oldboy38 tmp]# id oldboy

uid=501(oldboy) gid=501(incahome) groups=501(incahome)

## su

|  |  |  |
| --- | --- | --- |
| su命令用于切换当前用户身份到其他用户身份，变更时须输入所要变更的用户帐号与密码。 | | |
| **命令参数** | **命令功能** | **其他** |
| - | 会切换shell环境变量 |  |
| -c<指令>或--command=<指令> | 执行完指定的指令后，即恢复原来的身份； |  |

### 实例：只切换用户身份，不变化shell环境

命令：su hellhappy

[root@oldboy ~]# echo $PATH

/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

[root@oldboy ~]# su hellhappy

[hellhappy@oldboy root]$ echo $PATH

/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

### 实例：即切换用户身份，又切换shell环境

命令：su -hellhappy

[root@oldboy ~]# echo $PATH

/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

[root@oldboy ~]# su - hellhappy

[hellhappy@oldboy ~]$ echo $PATH

/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/hellhappy/bin

### 实例：在root身份下用test身份执行whoami命令

命令：su - test -c whoami

[root@oldboy38 ~]# su - test -c whoami

test

## sudo命令

|  |  |  |
| --- | --- | --- |
| sudo命令用来以其他身份来执行命令，预设的身份为root。在/etc/sudoers中设置了可执行sudo指令的用户。若其未经授权的用户企图使用sudo，则会发出警告的邮件给管理员。用户使用sudo时，必须先输入密码，之后有5分钟的有效期限，超过期限则必须重新输入密码。  来自: http://man.linuxde.net/sudo | | |
| **命令参数** | **命令功能** | **其他** |
| -l | 列出目前用户可执行与无法执行的指令； |  |
|  |  |  |

### 实例：看用户test下有什么可执行的命令

命令：sudo -l

[test@oldboy38 ~]$ sudo -l

[sudo] password for test:

Matching Defaults entries for test on this host:

!visiblepw, always\_set\_home, env\_reset, env\_keep="COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR

LS\_COLORS", env\_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC\_ADDRESS LC\_CTYPE", env\_keep+="LC\_COLLATE

LC\_IDENTIFICATION LC\_MEASUREMENT LC\_MESSAGES", env\_keep+="LC\_MONETARY LC\_NAME LC\_NUMERIC LC\_PAPER

LC\_TELEPHONE", env\_keep+="LC\_TIME LC\_ALL LANGUAGE LINGUAS \_XKB\_CHARSET XAUTHORITY",

secure\_path=/sbin\:/bin\:/usr/sbin\:/usr/bin, logfile=/var/log/sudo.log

User test may run the following commands on this host:

(ALL) /usr/sbin/useradd

# 基本网络操作命令

## ifup和ifdown

|  |
| --- |
| ifdown命令用于禁用指定的网络接口。 |
| ifup命令用于激活指定的网络接口。 |

### 实例：关闭网卡后再开启网卡

命令：ifdown eth0 && ifup eth0

[root@CentOS6 ~]# ifdown eth0 && ifup eth0

Connection closed by foreign host.

Disconnected from remote host(CenOS6.8-3) at 14:14:36.

Type `help' to learn how to use Xshell prompt.

[e:\~]$

## ping

|  |  |  |
| --- | --- | --- |
| ping命令用来测试主机之间网络的连通性。执行ping指令会使用ICMP传输协议，发出要求回应的信息，若远端主机的网络功能没有问题，就会回应该信息，因而得知该主机运作正常。 | | |
| **命令参数** | **命令功能** | **其他** |
| -c<完成次数> | 设置完成要求回应的次数； | -c 3 发送3次icmp包 |
| -i<间隔秒数> | 指定收发信息的间隔时间； | -i 3 每次发包时间间隔为3s; |
| -s<数据包大小> | 设置数据包的大小； | -s 1024 设置发送的数据包大小为1024字节 |
| -t<存活数值> | 设置存活数值TTL的大小； | -t 255 设置发送数据包的ttl值为255 |
|  |  |  |

存活時間（英语：Time To Live，簡寫TTL）是電腦網路技術的一個術語，指一個封包在经过一個網路時，可传递的最長距离（跃点数）。

## 实例：测试主机远程连接通不通

命令：ping 192.168.56.128

[root@oldboy ~]# ping 192.168.56.128

PING 192.168.56.128 (192.168.56.128) 56(84) bytes of data.

64 bytes from 192.168.56.128: icmp\_seq=1 ttl=64 time=0.041 ms

64 bytes from 192.168.56.128: icmp\_seq=2 ttl=64 time=0.048 ms

## ifconfig

|  |  |  |
| --- | --- | --- |
| ifconfig命令被用于配置和显示Linux内核中网络接口的网络参数。用ifconfig命令配置的网卡信息，在网卡重启后机器重启后，配置就不存在。要想将上述的配置信息永远的存的电脑里，那就要修改网卡的配置文件了。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：查看远程连接机器网络信息

命令：ifconfig

[root@oldboy ~]$ ifconfig

eth0 Link encap:Ethernet HWaddr 00:0C:29:8B:B4:9A

inet addr:192.168.56.128 Bcast:192.168.56.255 Mask:255.255.255.0

# 深入网络操作命令

## ss命令

|  |  |  |
| --- | --- | --- |
| ss命令用来显示处于活动状态的套接字信息。ss命令可以用来获取socket统计信息，它可以显示和netstat类似的内容。但ss的优势在于它能够显示更多更详细的有关TCP和连接状态的信息，而且比netstat更快速更高效。  当服务器的socket连接数量变得非常大时，无论是使用netstat命令还是直接cat /proc/net/tcp，执行速度都会很慢。可能你不会有切身的感受，但请相信我，当服务器维持的连接达到上万个的时候，使用netstat等于浪费 生命，而用ss才是节省时间。  天下武功唯快不破。ss快的秘诀在于，它利用到了TCP协议栈中tcp\_diag。tcp\_diag是一个用于分析统计的模块，可以获得Linux 内核中第一手的信息，这就确保了ss的快捷高效。当然，如果你的系统中没有tcp\_diag，ss也可以正常运行，只是效率会变得稍慢。 | | |
| **命令参数** | **命令功能** | **其他** |
| -l | 显示处于监听状态的套接字 |  |
| -n | 不解析服务名称，以数字方式显示 |  |
| -t | 只显示tcp套接字 |  |
| -u | 只显示udp套接字 |  |
| -p | 显示使用套接字的进程信息 |  |
| - |  |  |

### 实例：查看Linux开放端口

命令：ss -lntup

[root@oldboy38 tmp]# ss -lntup

Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port

udp UNCONN 0 0 \*:68 \*:\* users:(("dhclient",26493,6))

tcp LISTEN 0 128 :::22 :::\* users:(("sshd",1466,4))

tcp LISTEN 0 128 \*:22 \*:\* users:(("sshd",1466,3))

tcp LISTEN 0 100 ::1:25 :::\* users:(("master",1545,13))

tcp LISTEN 0 100 127.0.0.1:25 \*:\* users:(("master",1545,12))

## telnet

|  |  |  |
| --- | --- | --- |
| telnet命令用于登录远程主机，对远程主机进行管理。telnet因为采用明文传送报文，安全性不好，很多Linux服务器都不开放telnet服务，而改用更安全的ssh方式了。但仍然有很多别的系统可能采用了telnet方式来提供远程登录，因此弄清楚telnet客户端的使用方式仍是很有必要的。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：window下查看能否连接远程主机

命令：telnet 192.168.56.128 22

[e:\~]$ telnet 192.168.56.128 22

Connecting to 192.168.56.128:22...

Connection established.

To escape to local shell, press 'Ctrl+Alt+]'.

SSH-2.0-OpenSSH\_5.3

# 磁盘管理

## fdisk命令

|  |  |  |
| --- | --- | --- |
| fdisk命令磁盘管理 fdisk命令用于观察硬盘实体使用情况，也可对硬盘分区。 | | |
| **命令参数** | **命令功能** | **其他** |
| -l | 查看系统中所有磁盘的信息 |  |
| -c | 关闭模式DOS兼容模式 |  |
| -u | 以扇区的形式进行分区/查看 默认是柱面。 |  |
|  |  |  |

### 实例：查看当先系统磁盘分区表

命令：fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes

255 heads, 63 sectors/track, 1044 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x0001ed28

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 26 204800 83 Linux

Partition 1 does not end on cylinder boundary.

/dev/sda2 26 124 786432 82 Linux swap / Solaris

Partition 2 does not end on cylinder boundary.

/dev/sda3 124 1045 7396352 83 Linux

### 实例：查看新建的磁盘sdb和sdc

命令：fdisk -l |egrep "sd(b|c)"

[root@CentOS6 ~]# fdisk -l |egrep "sd(b|c)"

Disk /dev/sdb: 106 MB, 106954752 bytes

Disk /dev/sdc: 106 MB, 106954752 bytes

实例：给磁盘sdb分区

命令：fdisk -cu /dev/sdb

[root@CentOS6 ~]# fdisk -cu /dev/sdb 关闭模式DOS兼容模式并按照扇区数目显示磁盘大小

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0xdeafa3e8.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n 创建新分区 d:删除分区

Command action

e extended 逻辑分区

p primary partition (1-4) 主分区

p

Partition number (1-4): 1 第一个分区

First sector (2048-208895, default 2048):

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-208895, default 208895): +10M 分区大小（怎么也不填就是全部大小）

Command (m for help): p 查看分区结果

Disk /dev/sdb: 106 MB, 106954752 bytes

64 heads, 32 sectors/track, 102 cylinders, total 208896 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0xdeafa3e8

Device Boot Start End Blocks Id System

/dev/sdb1 2048 22527 10240 83 Linux

Command (m for help): w 保存分区并退出

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

## partprobe命令

|  |  |  |
| --- | --- | --- |
| partprobe命令用于重读分区表，当出现删除文件后，出现仍然占用空间。可以partprobe在不重启的情况下重读分区。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

实例：

命令：

## du

|  |  |  |
| --- | --- | --- |
| du命令也是查看使用空间的，但是与df命令不同的是Linux du命令是对文件和目录磁盘使用的空间的查看，还是和df命令有一些区别的。 | | |
| **命令参数** | **命令功能** | **其他** |
| -s或--summarize | 仅显示总计，只列出最后加总的值。 |  |
| -h或--human-readable | 以K，M，G为单位，提高信息的可读性。 |  |
|  |  |  |

### 实例：查看磁盘/目录下的目录大小

命令：du -sh /\*

[root@oldboy38 ~]# du -sh /\*

7.3M /bin

37M /boot

224K /dev

30M /etc

4.0K /home

206M /lib

27M /lib64

16K /lost+found

4.0K /media

4.0K /mnt

8.0K /opt

## dumpe2fs

|  |  |  |
| --- | --- | --- |
| dumpe2fs命令用于打印“ext2/ext3/ext4”文件系统的超级块和快组信息。 | | |
| **命令参数** | **命令功能** | **其他** |
| -h | 仅显示超级块信息； |  |

### 实例：查看磁盘inode size大小

命令：dumpe2fs /dev/sda3 |grep -i 'inode size'

[root@oldboy38 tmp]# dumpe2fs /dev/sda3 |grep -i 'inode size'

dumpe2fs 1.41.12 (17-May-2010)

Inode size: 256

实例：只显示磁盘分区sdb1超级块信息

命令：dumpe2fs -h /dev/sdb1

[root@CentOS6 ~]# dumpe2fs -h /dev/sdb1

dumpe2fs 1.41.12 (17-May-2010)

Filesystem volume name: <none>

Last mounted on: <not available>

Filesystem UUID: a4213662-af15-49db-9870-bef0e5d81b86

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem features: has\_journal ext\_attr resize\_inode dir\_index filetype needs\_recovery extent flex\_bg sparse\_super huge\_file uninit\_bg dir\_nlink extra\_isize

Filesystem flags: signed\_directory\_hash

Default mount options: (none)

Filesystem state: clean

Errors behavior: Continue

Filesystem OS type: Linux

Inode count: 2560

Block count: 10240

Reserved block count: 512

Free blocks: 8795

Free inodes: 2549

First block: 1

Block size: 1024

Fragment size: 1024

Reserved GDT blocks: 39

Blocks per group: 8192

Fragments per group: 8192

Inodes per group: 1280

Inode blocks per group: 160

Flex block group size: 16

Filesystem created: Sun Apr 9 06:13:37 2017

Last mount time: Sun Apr 9 06:19:13 2017

Last write time: Sun Apr 9 06:19:13 2017

Mount count: 1

Maximum mount count: -1

Last checked: Sun Apr 9 06:13:37 2017

Check interval: 4294880896 (1656 months, 4 weeks, 1 day, 6:28:16)

Next check after: Sat Apr 8 06:13:37 2017

Lifetime writes: 1482 kB

Reserved blocks uid: 0 (user root)

Reserved blocks gid: 0 (group root)

First inode: 11

Inode size: 128

Journal inode: 8

Default directory hash: half\_md4

Directory Hash Seed: fb814c7c-ab9c-4c2e-8275-ef9e669004d3

Journal backup: inode blocks

Journal features: (none)

Journal size: 1024k

Journal length: 1024

Journal sequence: 0x00000001

Journal start: 0

## mount和umount命令

|  |  |  |
| --- | --- | --- |
| mount命令用于加载文件系统到指定的加载点。此命令的最常用于挂载cdrom，使我们可以访问cdrom中的数据，因为你将光盘插入cdrom中，Linux并不会自动挂载，必须使用Linux mount命令来手动完成挂载。 | | |
| umount命令用于卸载已经加载的文件系统。利用设备名或挂载点都能umount文件系统，不过最好还是通过挂载点卸载，以免使用绑定挂载（一个设备，多个挂载点）时产生混乱。 | | |
| **命令参数** | **命令功能** | **其他** |
| -t | 指定挂载文件类型 | 如：ext4 |
| -lf | 强制卸载 |  |

### 实例：加载虚拟光盘

命令：mount //mwohism /mnt/

[root@CentOS6 ~]# mount /dev/cdrom /mnt/

mount: block device /dev/sr0 is write-protected, mounting read-only

# 关机和查看系统信息的命令

## init命令

|  |
| --- |
| init命令是Linux下的进程初始化工具，init进程是所有Linux进程的父进程，它的进程号为1。init命令是Linux操作系统中不可缺少的程序之一，init进程是Linux内核引导运行的，是系统中的第一个进程。 |

# 系统管理相关命令

## chkconfig命令

|  |  |  |
| --- | --- | --- |
| chkconfig命令检查、设置系统的各种服务。这是Red Hat公司遵循GPL规则所开发的程序，它可查询操作系统在每一个执行等级中会执行哪些系统服务，其中包括各类常驻服务。谨记chkconfig不是立即自动禁止或激活一个服务，它只是简单的改变了符号连接。 | | |
| **命令参数** | **命令功能** | **其他** |
| --level type name | 选择运行级别 |  |
| --add | 增加所指定的系统服务，让chkconfig指令得以管理它，并同时在系统启动的叙述文件内增加相关数据 |  |

### 实例：让netfs服务在345系统级别下不自启动

命令：chkconfig --level 345 netfs off

[root@oldboy ~]# chkconfig | grep netfs

netfs 0:off 1:off 2:off 3:on 4:on 5:on 6:off

[root@oldboy ~]# chkconfig --level 345 netfs off

[root@oldboy ~]# chkconfig | grep netfs

netfs 0:off 1:off 2:off 3:off 4:off 5:off 6:off

### 实例：把执行脚本加入到chkconfig开机自启动

命令：chkconfig --add oldboy.sh

[root@oldboy38 init.d]# chkconfig --add oldboy.sh

[root@oldboy38 init.d]# chkconfig |grep oldboy.sh

oldboy.sh 0:off 1:off 2:on 3:on 4:on 5:on 6:off

## free命令

|  |  |  |
| --- | --- | --- |
| free命令可以显示当前系统未使用的和已使用的内存数目，还可以显示被内核使用的内存缓冲区。 | | |
| **命令参数** | **命令功能** | **其他** |
| -m | 以MB为单位显示 |  |
| -h | 人类可读 |  |

### 实例：人类可读

命令：free -h

[root@oldboy oldboy]# free -h

total used free shared buffers cached

Mem: 474M 148M 326M 240K 19M 42M

-/+ buffers/cache: 86M 388M

Swap: 767M 0B 767M

### 实例：以MB为单位显示

命令：free -m

[root@oldboy oldboy]# free -m

total used free shared buffers cached

Mem: 474 148 326 0 19 42

-/+ buffers/cache: 86 388

Swap: 767 0 767

# 系统安全相关命令

# 查看系统用户登录信息的命令

## last命令

|  |  |  |
| --- | --- | --- |
| 单独执行last命令，它会读取/var/log/wtmp的文件，并把该给文件的内容记录的登入系统的用户名单全部显示出来 | | |
| **命令** | **命令功能** | **命令组合** |
| last | 用于显示用户最近登录信息 |  |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：查找登入系统的用户名

命令：last

[root@CentOS6 etc]# last

root pts/0 192.168.56.1 Thu Mar 16 18:58 still logged in

root pts/0 192.168.56.1 Thu Mar 16 18:11 - 18:58 (00:47)

reboot system boot 2.6.32-642.el6.x Thu Mar 16 18:10 - 19:06 (00:56)

reboot system boot 2.6.32-642.el6.x Sat Mar 11 15:44 - 19:06 (5+03:22)

root tty1 Sat Mar 11 15:39 - crash (00:05)

reboot system boot 2.6.32-642.el6.x Sat Mar 11 15:39 - 19:06 (5+03:27)

## whoami命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| whoami | 显示当前用户的用户名字 |  |

### 实例 查询当前用户

命令：whoami

[root@CentOS6 ~]# whoami

root

# 其他

## echo命令

|  |  |  |
| --- | --- | --- |
| echo命令用于在shell中打印shell变量的值，或者直接输出指定的字符串。linux的echo命令，在shell编程中极为常用, 在终端下打印变量value的时候也是常常用到的，因此有必要了解下echo的用法echo命令的功能是在显示器上显示一段文字，一般起到一个提示的作用。 | | |
| **参数** | **参数说明** | **其他说明** |
| $？ | 打印上一条命令返回的结果 | 0代表true 非0代表fail |
| {数字..数字} | 建立一系列文件 |  |
| -e | 激活转义字符。 |  |
| -n | 取消换行 |  |

### 实例：显示打印内容

命令：echo "I am studying linux"

[root@oldboy-35 tmp]# echo "I am studying linux"

I am studying linux

## alias命令

|  |
| --- |
| alias命令用来设置指令的别名。我们可以使用该命令可以将一些较长的命令进行简化。使用alias时，用户必须使用单引号''将原来的命令引起来，防止特殊字符导致错误。  alias命令的作用只局限于该次登入的操作。若要每次登入都能够使用这些命令别名，则可将相应的alias命令存放到bash的初始化文件/etc/bashrc中。 |

### 实例：临时修改rm别名

命令：alias rm='echo Do not used command'

[root@CentOS6 ~]# alias

alias cp='cp -i'

alias l.='ls -d .\* --color=auto'

alias ll='ls -l --color=auto'

alias ls='ls --color=auto'

alias mv='mv -i'

alias rm='rm -i'

alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

[root@CentOS6 ~]# alias rm='echo Do not used command'

[root@CentOS6 ~]# alias

alias cp='cp -i'

alias l.='ls -d .\* --color=auto'

alias ll='ls -l --color=auto'

alias ls='ls --color=auto'

alias mv='mv -i'

alias rm='echo Do not used command'

alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

### 实例：把打印内容输入到文件里

命令：echo "I am studying linux">>test.txt

[root@oldboy-35 tmp]# echo "I am studying linux">>test.txt

### 实例：看命令是否执行成功

命令：echo $?

[root@oldboy ~]# echo "oldboy">>test.txt

[root@oldboy ~]# echo $?

0

### 实例：显示stu1 stu2 stu3 stu4 stu5

命令：echo stu{1..5}

echo stu{1..5}

stu1 stu2 stu3 stu4 stu5

## xargs命令

|  |  |  |
| --- | --- | --- |
| xargs命令是给其他命令传递参数的一个过滤器，也是组合多个命令的一个工具。它擅长将标准输入数据转换成命令行参数，xargs能够处理管道或者stdin并将其转换成特定命令的命令参数。xargs也可以将单行或多行文本输入转换为其他格式，例如多行变单行，单行变多行。xargs的默认命令是echo，空格是默认定界符。这意味着通过管道传递给xargs的输入将会包含换行和空白，不过通过xargs的处理，换行和空白将被空格取代。xargs是构建单行命令的重要组件之一。 | | |
| **参数** | **参数说明** | **其他说明** |
| -n 数字 | 把文件内容分组 |  |
| -i | 使用-I指定一个替换字符串{}，这个字符串在xargs扩展时会被替换掉，当-I与xargs结合使用，每一个参数命令都会被执行一次 |  |

### 实例：把文件内容（1 2 3 4 5 6）分为2组或3组

命令：xargs -n 2 < /tmp/oldboy.txt

[root@oldboy-35 ~]# echo "1 2 3 4 5 6 7 8">/tmp/oldboy.txt

[root@oldboy-35 ~]# xargs -n 2 < /tmp/oldboy.txt

1 2

3 4

5 6

7 8

### 把所有带log结尾的文件复制到目录中

命令：find /tmp/data -type f -name "\*.log"|xargs -i cp {} /tmp/data/a

[root@oldboy-35 ~]# ls /tmp/data/a

[root@oldboy-35 ~]# find /tmp/data -type f -name "\*.log"|xargs -i cp {} /tmp/data/a

[root@oldboy-35 ~]# ls /tmp/data/a

log 02.log 03.log 04.log 05.log 06.log 07.log 08.log 09.log 10.log 11.log 12.log 13.log 14.log

## history命令

|  |  |  |
| --- | --- | --- |
| history命令用于显示指定数目的指令命令，读取历史命令文件中的目录到历史命令缓冲区和将历史命令缓冲区中的目录写入命令文件。  该命令单独使用时，仅显示历史命令，在命令行中，可以使用符号!执行指定序号的历史命令。例如，要执行第2个历史命令，则输入!2。  历史命令是被保存在内存中的，当退出或者登录shell时，会自动保存或读取。在内存中，历史命令仅能够存储1000条历史命令，该数量是由环境变量HISTSIZE进行控制。 | | |
| **命令参数** | **命令功能** | **其他** |
| -c | 清空全部历史命令 |  |
|  |  |  |

### 实例：查询历史命令

命令：history

[root@oldboy ~]# history

cat /etc/selinux/config

2 sed 's#SELINUX=enforcing##g'

3 sed 's#SELINUX=enforcing#SELINUX=enforcing#g'

### 实例：清除历史命令

命令：history -c

[root@oldboy ~]# history -c

[root@oldboy ~]# history

history

补充：一般用于非交互密码设置并清除命令历史

## yum命令

|  |
| --- |
| yum命令是在Fedora和RedHat以及SUSE中基于rpm的软件包管理器，它可以使系统管理人员交互和自动化地更细与管理RPM软件包，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载、安装。  yum提供了查找、安装、删除某一个、一组甚至全部软件包的命令，而且命令简洁而又好记。 |

### 实例：安装tree工具

命令：yum install -y tree

[root@CentOS6 ~]# yum install -y tree

Loaded plugins: fastestmirror, security

……忽略  
Installed:

tree.x86\_64 0:1.5.3-3.el6

Complete!

## getenforce和setenforce命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| getenforce | 查看selinux的状态 |  |
| setenforc | 临时修改selinux的状态 |  |

### 实例 临时关闭selinux

#### 命令：getenforce

#### 命令：setenforce

#### 命令：setenforce 0

[root@CentOS6 ~]# getenforce

Enforcing

[root@CentOS6 ~]# setenforce

usage: setenforce [ Enforcing | Permissive | 1 | 0 ]

[root@CentOS6 ~]# setenforce 0

[root@CentOS6 ~]# getenforce

Permissive

## tr

|  |  |  |
| --- | --- | --- |
| tr命令可以对来自标准输入的字符进行替换、压缩和删除。它可以将一组字符变成另一组字符，经常用来编写优美的单行命令，作用很强大。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

### 实例：将输入字符由大写转换为小写

命令：tr 'A-Z' 'a-z' hello world

echo "HELLO WORLD" | tr 'A-Z' 'a-z' hello world

hello world

## source

|  |  |  |
| --- | --- | --- |
| source命令也称为“点命令”，也就是一个点符号（.）。source命令通常用于重新执行刚修改的初始化文件，使之立即生效，而不必注销并重新登录。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## crontab

|  |  |  |
| --- | --- | --- |
| crontab命令被用来提交和管理用户的需要周期性执行的任务，与windows下的计划任务类似，当安装完成操作系统后，默认会安装此服务工具，并且会自动启动crond进程，crond进程每分钟会定期检查是否有要执行的任务，如果有要执行的任务，则自动执行该任务。 | | |
| **命令参数** | **命令功能** | **示例** |
| -e | 编辑该用户的计时器设置； |  |
| -l | 列出该用户的计时器设置； |  |
| -u<用户名称> | 指定要设定计时器的用户名称。 |  |

### 实例：编辑定时任务

命令：crontab -e

[root@oldboy38 /]# crontab -e

#lock in time zhangdi 20170331

#\*/5 \* \* \* \* /usr/sbin/ntpdate > /del/null 2>&1

### 实例：浏览定时任务

命令：crontab -l

[root@oldboy38 /]# crontab -l

#lock in time zhangdi 20170331

\*/5 \* \* \* \* /usr/sbin/ntpdate ntp1.aliyun.com > /del/null 2>&1

## nc命令需要yum安装

## lrzsz命令需要yum安装

## nmap命令需要yum安装

## dos2unix命令需要yum安装

## unix2dos命令需要yum安装

## sl命令需要yum安装

## telnet命令需要yum安装

## bc命令

|  |  |  |
| --- | --- | --- |
| bc命令是一种支持任意精度的交互执行的计算器语言。bash内置了对整数四则运算的支持，但是并不支持浮点运算，而bc命令可以很方便的进行浮点运算，当然整数运算也不再话下。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## od命令

|  |  |  |
| --- | --- | --- |
| od命令用于输出文件的八进制、十六进制或其它格式编码的字节，通常用于显示或查看文件中不能直接显示在终端的字符。 常见的文件为文本文件和二进制文件。此命令主要用来查看保存在二进制文件中的值。比如，程序可能输出大量的数据记录，每个数据是一个单精度浮点数。这些数据记录存放在一个文件中，如果想查看下这个数据，这时候od命令就派上用场了。在我看来，od命令主要用来格式化输出文件数据，即对文件中的数据进行无二义性的解释。不管是IEEE754格式的浮点数还是ASCII码，od命令都能按照需求输出它们的值。 | | |
| **命令参数** | **命令功能** | **其他** |
| -x |  |  |
| -a |  |  |

### 实例：查看MBR主引导备份文件

命令：od -xa mbr.bak

[root@oldboy38 tmp]# od -xa mbr.bak

0000000 48eb 1090 d08e 00bc b8b0 0000 d88e c08e

## sort

|  |  |  |
| --- | --- | --- |
| sort命令是在Linux里非常有用，它将文件进行排序，并将排序结果标准输出。sort命令既可以从特定的文件，也可以从stdin中获取输入。 | | |
| **命令参数** | **命令功能** | **其他** |
| -n | 依照数值的大小排序 |  |
| -r | 以相反的顺序来排序 |  |
| -k | 指定从第几列开始排序 |  |

## uniq

|  |  |  |
| --- | --- | --- |
| uniq命令用于报告或忽略文件中的重复行，一般与sort命令结合使用。 | | |
| **命令参数** | **命令功能** | **其他** |
| -c或——count | 在每列旁边显示该行重复出现的次数。 |  |

## column -t

|  |  |  |
| --- | --- | --- |
|  | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## tee

|  |  |  |
| --- | --- | --- |
|  | | |
| **命令参数** | **命令功能** | **其他** |
| -a |  |  |

## mkpasswd命令需要yum安装

|  |  |  |
| --- | --- | --- |
|  | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## rpm

|  |  |  |
| --- | --- | --- |
| rpm命令是RPM软件包的管理工具。rpm原本是Red Hat Linux发行版专门用来管理Linux各项套件的程序，由于它遵循GPL规则且功能强大方便，因而广受欢迎。逐渐受到其他发行版的采用。RPM套件管理方式的出现，让Linux易于安装，升级，间接提升了Linux的适用度。 | | |
| **命令参数** | **命令功能** | **其他** |
| -a | 查询所有套件 |  |
| -q 绝对路径 | 使用询问模式，当遇到任何问题时，rpm指令会先询问用户 |  |
| -l | 显示套件的文件列表 |  |
| -f<文件>+ | 查询拥有指定文件的套件 |  |
| --nodeps | 不检查依赖关系 |  |
| -ivh | 安装一个rpm包 |  |

## kill命令

|  |  |  |
| --- | --- | --- |
| kill命令用来删除执行中的程序或工作。kill可将指定的信息送至程序。预设的信息为SIGTERM(15),可将指定程序终止。若仍无法终止该程序，可使用SIGKILL(9)信息尝试强制删除程序。程序或工作的编号可利用ps指令或job指令查看。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## killall

|  |  |  |
| --- | --- | --- |
| killall命令使用进程的名称来杀死进程，使用此指令可以杀死一组同名进程。我们可以使用kill命令杀死指定进程PID的进程，如果要找到我们需要杀死的进程，我们还需要在之前使用ps等命令再配合grep来查找进程，而killall把这两个过程合二为一，是一个很好用的命令。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## pkill

|  |  |  |
| --- | --- | --- |
| pkill命令可以按照进程名杀死进程。pkill和killall应用方法差不多，也是直接杀死运行中的程序；如果您想杀掉单个进程，请用kill来杀掉。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

## rsync

|  |  |  |
| --- | --- | --- |
| rsync命令是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件。rsync使用所谓的“rsync算法”来使本地和远程两个主机之间的文件达到同步，这个算法只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。 rsync是一个功能非常强大的工具，其命令也有很多功能特色选项，我们下面就对它的选项一一进行分析说明 | | |
| **命令参数** | **命令功能** | **其他** |
| -a |  |  |
| -v |  |  |
| -z |  |  |
| -r |  |  |
| -p |  |  |
| -t |  |  |
| -P |  |  |
| -g |  |  |

## dirname

## basename

## nfs命令需要yum安装

## rpcbind命令需要yum安装

## route命令

|  |  |  |
| --- | --- | --- |
| 当前位置：首页 » 网络管理 » route route命令网络配置 route命令用来显示并设置Linux内核中的网络路由表，route命令设置的路由主要是静态路由。要实现两个不同的子网之间的通信，需要一台连接两个网络的路由器，或者同时位于两个网络的网关来实现。  在Linux系统中设置路由通常是为了解决以下问题：该Linux系统在一个局域网中，局域网中有一个网关，能够让机器访问Internet，那么就需要将这台机器的ip地址设置为Linux机器的默认路由。要注意的是，直接在命令行下执行route命令来添加路由，不会永久保存，当网卡重启或者机器重启之后，该路由就失效了；可以在/etc/rc.local中添加route命令来保证该路由设置永久有效。 | | |
| **命令选项** | **选项功能** | **其他** |
| -n | 不执行DNS反向查找，直接显示数字形式的IP地址 |  |
| -net | 到一个网络的路由表 |  |
| -host | 到一个主机的路由表。 |  |
| **命令参数** | **命令功能** | **其他** |
| add | 增加指定的路由记录 |  |
| del | 删除指定的路由记录 |  |
| gw | 设置默认网关 |  |
| dev | 路由记录所表示的网络接口。 |  |

## ip命令

## nmap

## nohup命令

## curl命令

|  |  |  |
| --- | --- | --- |
| curl命令是一个利用URL规则在命令行下工作的文件传输工具。它支持文件的上传和下载，所以是综合传输工具，但按传统，习惯称curl为下载工具。作为一款强力工具，curl支持包括HTTP、HTTPS、ftp等众多协议，还支持POST、cookies、认证、从指定偏移处下载部分文件、用户代理字符串、限速、文件大小、进度条等特征。做网页处理流程和数据检索自动化，curl可以祝一臂之力。 | | |
| **命令参数** | **命令功能** | **其他** |
| I/--head | 只显示请求头信息 |  |
| -s/--silent | 静默模式。不输出任何东西 |  |
| -w/--write-out [format] | 什么输出完成后 |  |
| -o/--output | 把输出写到该文件中 |  |
|  |  |  |

实例：查看http响应的数字状态码

命令：curl -I www.etiantian.org

[root@web02-7 ~]# curl -I www.etiantian.org

HTTP/1.1 200 OK

Server: nginx/1.10.2

Date: Fri, 09 Jun 2017 09:45:03 GMT

Content-Type: text/html; charset=utf-8

Connection: keep-alive

X-Powered-By: PHP/5.5.32

命令2：curl -I -s -w %{http\_code} -o /dev/null www.etiantian.org

[root@web02-7 ~]# echo `curl -I -s -w %{http\_code} -o /dev/null www.etiantian.org`

200

# 系统性能监视

|  |  |  |
| --- | --- | --- |
|  | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

# 高级命令

## w

|  |  |  |
| --- | --- | --- |
|  | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

# 关机/重启/注销命令

## shutdown

|  |  |  |
| --- | --- | --- |
| shutdown命令用来系统关机命令。shutdown指令可以关闭所有程序，并依用户的需要，进行重新开机或关机的动作。 | | |
| **命令参数** | **命令功能** | **其他** |
| -c | 当执行“shutdown -h 11:50”指令时，只要按+键就可以中断关机的指令； |  |
| -h | 将系统关机 |  |
| -r | shutdown之后重新启动 |  |

### 实例：五分后关机

命令：shutdown +5 "System will shutdown after 5 minutes"

[root@oldboy38 tmp]# shutdown +5 "System will shutdown after 5 minutes"

Broadcast message from root@oldboy38

(/dev/pts/0) at 16:31 ...

The system is going down for maintenance in 5 minutes!

System will shutdown after 5 minutes

## reboot

|  |  |  |
| --- | --- | --- |
| reboot命令用来重新启动正在运行的Linux操作系统。 | | |
| **命令参数** | **命令功能** | **其他** |
|  |  |  |

# 进程管理

## lsof命令

|  |  |  |
| --- | --- | --- |
| lsof命令用于查看你进程打开的文件，打开文件的进程，进程打开的端口(TCP、UDP)。找回/恢复删除的文件。是十分方便的系统监视工具，因为lsof命令需要访问核心内存和各种文件，所以需要root用户执行。 | | |
| **命令参数** | **命令功能** | **其他** |
| -i |  | -i:22 |
| lsof输出各列信息 | 意义 |  |
| COMMAND | 进程的名称 |  |
| PID | 进程标识符 |  |
| PPID | 父进程标识符（需要指定-R参数） |  |
| USER | 进程所有者 |  |
| PGID | 进程所属组 |  |
| FD | 文件描述符，应用程序通过文件描述符识别该文件。 |  |

### 实例：查看系统进程打开的文件

命令：lsof

[root@oldboy38 tmp]# lsof

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

init 1 root cwd DIR 8,3 4096 2 /

init 1 root rtd DIR 8,3 4096 2 /

init 1 root txt REG 8,3 150352 7364 /sbin/init

## ps命令

|  |  |  |
| --- | --- | --- |
| ps命令用于报告当前系统的进程状态。可以搭配kill指令随时中断、删除不必要的程序。ps命令是最基本同时也是非常强大的进程查看命令，使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵死、哪些进程占用了过多的资源等等，总之大部分信息都是可以通过执行该命令得到的。 | | |
| **命令参数** | **命令功能** | **其他** |
| -e |  |  |
| -f | 显示UID,PPIP,C与STIME栏位。 |  |

### 实例：查看sshd进程

命令：ps -ef |grep sshd

[root@oldboy38 tmp]# ps -ef |grep sshd

root 1466 1 0 09:28 ? 00:00:00 /usr/sbin/sshd

## runlevel命令

|  |  |  |
| --- | --- | --- |
| **命令** | **命令功能** | **命令组合** |
| runlevel | 用于打印当前Linux系统的运行等级 |  |

### 实例：查看当前Linux系统运行级别

命令：runlevel

[root@oldboy38 tmp]# runlevel

N 3

# 非常危险的系统命令

# 快捷键

|  |  |
| --- | --- |
| 快捷键 | 功能 |
| Tab | 命令或路径等的补全键 |
| ctrl+l | 清空屏幕 |
| ctrl+c | 取消当前的操作 |
| ctrl+a | 光标回到命令行首 |
| ctrl+e | 光标回到命令行尾 |
| shift+Insert | 粘贴命令行内容 |
| ctrl+u | 从光标的位置到命令行首剪切内容 |
| ctrl+k | 从光标的位置到命令行尾剪切内容 |
| ctrl+y | 复制剪切内容 |
| Esc+. | 获取上一条命令的最后的部分（空格分隔） |
| ctrl+d | 退出当前shell命令行 |
| ctrl+z | 暂停执行在终端运行的任务 |
| ctrl+r | 搜索命令行使用过的历史命令记录命令或路径等的补全键 |
|  |  |

# 特殊符号

|  |  |
| --- | --- |
| 符号 | 作用 |
| > | 输出重定向-会清除文件原有内容，再输入内容。 |
| >> | 追加输出重定向-直接追加内容。 |
| \* | 表示所有文本，字符组成的文件名字。 |
| | | 把前一个命令的结果，传递给后面处理。 |
| ``=$() | 先执行里面的命令，结果留下。 |
| && | 前一个命令执行成功，然后执行后面一个命令 |
| # | 井号 linux下面表示 注释 linux会忽略 后面的内容 |
| { }（大括号） | 生成一系列序列 |
| ''（单引号） |  |
| ""（双引号） |  |

# 正则表达式

|  |  |
| --- | --- |
| 基础正则表达式 | 作用 |
| ^word | 搜索以字符（word）开头的行。 |
| 例：grep -v '^#' oldboy.log  过滤掉以#行首（开始）的行，一般查看配置文件时会用到。 |
| word$ | 搜索以字符（word）结尾的行。 |
| 例：grep -n '!$' oldboy.log  搜索以!（叹号）结尾的行 |
| . | 代表切只能代表任意一个字符 |
| 例：grep -n 'e.e' oldboy.log  搜索的字符串可以是（eve）,(eae),(eee),(e e),但不能仅有(ee)即e与e之间一定且仅有一个字符，而空格符也是字符 |
| | | 交替匹配|两边的任意一项 |
| 例：egrep "grep|egrep" /etc/profile  找出含有（grep）和（egrep）等的字符串。 |
| \* | 重复零个或多个的前一个重复字符 |
| 例：grep -n 'ess\*' oldboy.log  找出含有（es）,(ess),(esss)等的字符串，注意，因为(\*)可以是0个，所以es 也是符合搜索字符串。另外，因为（\*）为重复前一个字符的符号，因此，在（\*）之前必须要紧接一个重复字符。任意字符则为（.\*） |
| .\* | 根据前面的单个字符，我们知道.\*匹配所有字符串 |
| ^$ | 空行 |
| [abc] | 匹配字符集合内任意一个字符[a-z] |
| 例：coo[kl]可以匹配cook或cool |
| [^abc] | ^再中括号里面表示非，不包含a或b或c。 |
| 例：123[^45]不可以匹配1234或1235，1236、1237都可以 |
| 扩展正则表达式 | 作用 |
| {n} | 匹配之前的项n次，n是可以为0的正整数 |
| 例：[0-9]{3}匹配任意一个三位数，可以扩展为[0-9][0-9][0-9] |
| {n,} | 之前的项至少需要匹配n次 |
| 例：[0-9]{2,}匹配任意一个两位数或更多位数 |
| {n,m} | 重复n到m次，前一个字符。  {n,} 至少N次，多了不限。  {n} N次  {,m} 至多m次，少了不限。 |
| 例：[0-9]{2,5}匹配从两位数到五位数之间的任意一个数字 |
| + | 匹配之前的项1次或者多次 |
| 例：sa-6+匹配sa-6、sa-666，不能匹配sa- |
| ( ) | 匹配表达式，创建一个用于匹配的子串 |
| 例：ab(c|d)匹配abc或abd |

注意：grep 要对{转义} \{\},egrep（grep -E）不需要转义，sed用扩展正则加参数-r

# 别名

## 查询别名

命令：which

[root@oldboy-35 ~]# which cp

alias cp='cp -i'

/bin/cp

## 查询全部别名

命令：alias

[root@oldboy-35 ~]# alias

alias cp='cp -i'

alias l.='ls -d .\* --color=auto'

alias ll='ls -l --color=auto'

alias ls='ls --color=auto'

alias mv='mv -i'

alias rm='echo do not use rm command'

alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'

## 修改别名

命令：alias rm='echo Do not use rm command'

[root@oldboy-35 ~]# alias rm='echo Do not use rm command'

注意：rm后直接跟等号，用单引号，等会后面必须是命令。

## 永久生效1

命令：vim /etc/profile

在最后加上alias rm='echo Do not use rm command'保存并退出：wq

## 永远生效2

命令：[root@oldboy-35 ~]# vim /root/.bashrc

alias rm='rm -i'在前加#及忽略此行#alias rm='rm -i'

## 重新登录xshell生效