# Linux

# 一 Linux常用命令

## 1.1 常用命令

### 1.1.1重启网络

service network restart

### 1.1.2防火墙状态

service iptables status

### 1.1.3关闭防火墙

service iptables stop

### 1.1.4永久关闭防火墙

chkconfig iptables off

### 1.1.5修改主机名

hostname xxxx

### 1.1.6显示当前目录

pwd

### 1.1.7权限说明

-rw 以-开头的表示文件

Drw 以d开头的表示文件夹

### 1.1.8查看文件

Cat 【查看文件】 查看文件的部分，cat查看文件显示不全

More 【查看文件】 查看文件的全部

### 1.1.9创建文件

Touch 【文件名】

### 1.1.10创建文件夹

Mkdir 【文件名】

Mkdir –p 【path】递归的创建文件或者文件夹

### 1.1.11文件拷贝

Cp 【源文件】 【目标文件】

Cp –r 【源文件】 【目标文件】 –r 表示递归的拷贝

### 1.1.12删除文件

rm –f 【文件名】 删除文件

rm –r 【目录名】 删除目录

rm –rf 【目录，文件】 递归的删除所有的文件

### 1.1.13 查看文件的指定行

Head –number【文件名】 查看指定文件的头number行

Tail –number【文件名】 查看文件的尾部number行

### 1.1.14 修改文件名或者移动文件

Mv 【源文件名】 【新文件名】

Mv 【源文件名】【路径+新文件名】

### 1.1.15 创建链接文件

Ln –s 【源文件】【目标文件】 有-s表示软链接，没有-s表示硬链接。

软链接所有人都可以访问，并且软链接文件指向源文件。

硬链接类似于copy，硬链接的文件大小一样，并且同步更新。

硬链接不能跨系统的分区，但是软链接可以。

### 1.1.16 退出

Wq:保存退出

Q!:退出不保存

### 1.1.17 文件搜索

#### 1.1.1.1根据文件名搜索

**命令：find**

**语法：find [搜索范围路径] -name 【文件名称】（根据文件名查找）**

**find [搜索范围路径] -size [（+-）文件大小] （根据文件大小查找，大于+ 小于-）**

**find [搜索范围路径] -user（文件的所有者）**

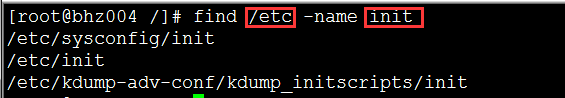
**find [时间查找] [以天为单位]**

**[以天为单位] 1 ctime、atime、mtime**

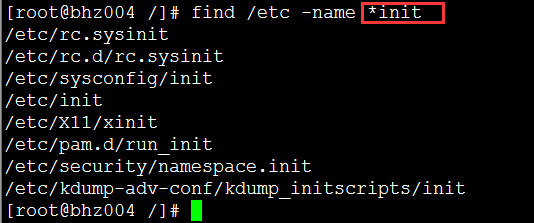
**[以分钟为单位] 2 cmin、amin、mmin**

**描述：查找任何文件或目录（所有）**

**find命令示例：find [搜索范围路径] –name**



**匹配任意字符 \***



**匹配指定字符 ？**



#### 1.1.1.2 根据文件大小搜索

**find [搜索范围路径] -size [+-文件大小] （根据文件大小查找，大于+小于-）**

**注意，对于文件的大小是以数据块为单位。数据块大小是512bit**

**形如：如需查找100M的文件，那么就要知道100M为多少block？**

**100M = ? block**

**1M = 1024K**

**100M = 102400K**

**1K = 2 block**

**100M = 2\*102400 block**

#### 1.1.1.3 根据指定的用户名查找

**find [搜索范围路径] -user（文件的所有者）**

**描述：查找隶属于某个用户的文件**



#### 1.1.1.4 根据改变时间来查找

**find [时间查找] [以天为单位]**

**天： ctime、atime、mtime**

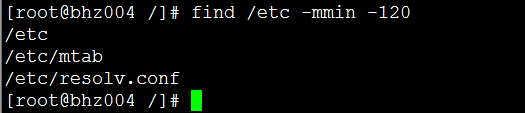
**分钟：cmin、amin、mmin**

**c表示：change 改变文件属性的意思（比如所有者、所属组、权限变更）。**

**a表示：access 表示被访问过的意思（比如被查看过等）。**

**m表示：modify 更改内容的意思。**

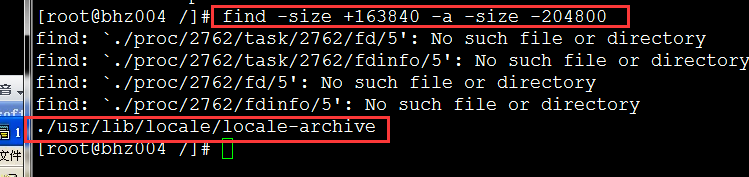
**在时间前面添加：-表示之内，+表示之外**

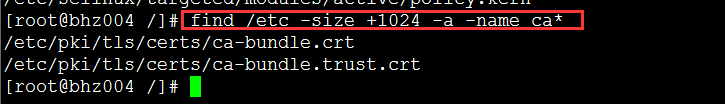


#### 1.1.1.5 find命令运用链接符

**-a （and的意思，逻辑与）**

**-o（or的意思，逻辑或）**

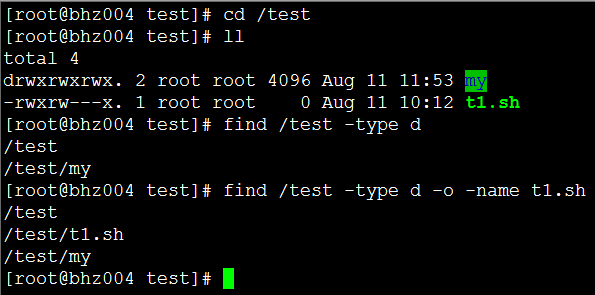


****

**1.1.1.6 根据文件类型查找文件**

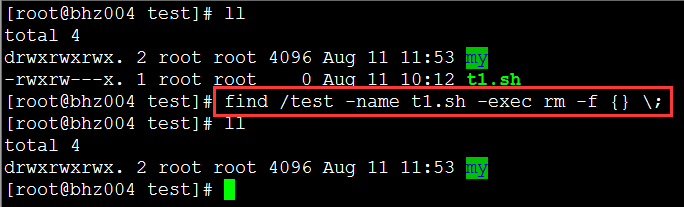
-type

其中：f表示二进制文件，l表示软连接文件 d表示目录

****

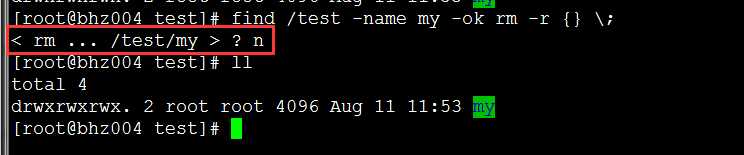
**1.1.1.6 查找并执行命令**

**find -exec [执行命令] {} \;**

****

**find -ok [执行命令] {} \;**

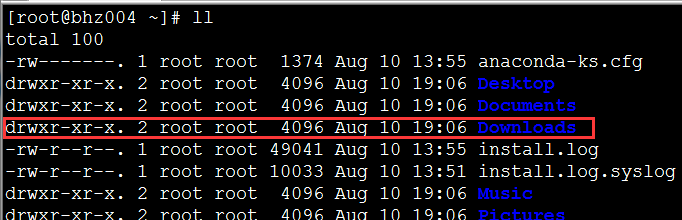
**ok和exec的区别就是ok有询问确认的意思。**

****

# 二 linux权限

## 2.1 权限讲解

### 2.1.1 目录权限



**d：开始为d，表示目录 directory**

**d: 表示文件夹**

**- : 开始为-，表示普通的二进制文件**

**l : 开始为l，表示软连接文件（link）**

**文件的rwx权限：**

**r：可以执行catch、more等读操作。**

**w：修改文件的内容等写操作，但是不代表你可以把这个文件删除。**

**x：对文件进行执行操作。**

**目录的rwx权限：**

**r：可读操作，可以列出目录的内容，比如ls命令。**

**w：表示可以在目录下创建或删除文件的权限。**

**x：表示可以进入这个目录（基本上所以的目录都会有rx权限）。**

**所以：删除文件的权限，是拥有你当前文件所在的目录的写权限。**

**Linux中的文件如：drwxr-xr-x 可以看做三个部分（rwx r-x r-x），**

**分别是文件的所有者rwx（user）的权限、**

**所属组r-x（group）的权限**

**其他人r-x（others）的权限。**

### 2.1.2 改变权限

Chmod 【{ugo}{+-=}{rwx}】 【文件或目录】

Chmod number 【文件或目录】

U 表示user

G 表示group

O 表示other

Number是二进制转十进制数775表示rwxrwxrw- 表示成111111110

### 2.1.3 切换用户

Su - 【用户名】

### 2.1.4 改变文件所有者

Chown 【新用户名】【文件名】

### 2.1.5 改变文件所属组

**chgrp 【组名】【文件名】**

### 2.1.6 帮助命令

Man 【命令或者配置文件】

帮助命令，非常有用。

Whatis 【命令】

查看命令描述。

--help

【命令】 --help

查看命令选项用法。

### 2.1.7 压缩与解压缩

**命令：gzip**

**语法：gzip [文件名称]**

**描述：压缩的时候不保留原文件，并且只能压缩文件不能压缩目录**

**命令：gunzip**

**语法：gunzip [已压缩的文件]**

**描述：解压缩文件，不不保留源文件**

**命令：tar**

**语法：tar [zcvf] [zxvf] [打包文件名.tar.gz] [源文件]**

**-c 产生tar打包文件（必选）**

**-x 产生的解压缩文件（必选）**

**-v 显示详细信息**

**-f 指定压缩后的文件名**

**-z 打包同时压缩**

**描述：打包目录 生成的后缀名 .tar.gz，或者进行解压**

**最后配置加-C 表示文件解压后存放的路径**

**file命令可以查看任何文件的类型**

**命令：zip**

**语法：zip 选项[-r] [压缩后文件名称] [源文件]**

**描述：zip的格式是windows和linux通用的格式，可以压缩文件和目录，压缩目录时需要选项-r。**

**命令：unzip**

**语法：unzip [解压缩的文件]**

**描述：进行解压缩**

**最后配置加-d 表示文件解压后存放的路径**

### 2.1.8 系统命令

**ping**

**（注意：ping 不通对方网络的原因有很多种，需要一步步详细排查）**

**（1）首先ping一下回环地址 127.0.0.1 检查自己本机的网络协议是否正确**

**（2）再ping一下本机ip 查看自己本机的网络是否正确**

**（3）然后检查对方网络设置、防火墙、插件等等**

**（4）如果发现丢包率里有丢失数据包，可能是网络、网线的原因**

**（5）ping 配置选项 ping -c 6 192.168.80.100（表示ping 6次之后断开）**

**（6）ping 配置选项 ping -s 60000 （最大65507）**

**查看网卡信息：ifconfig**

**关机：shutdown -h now**

**重启：reboot**

**ctrl + l 清屏。**

**ctrl + c 退出应用。**

**tab键，信息补全。**

### 2.1.9 管道与过滤

**过滤：grep，可以将指定内容进行过滤然后输出。**

**管道：**

**将一个命令的输出传送给另一个命令，作为另外一个命令的输入。管道可以连接N个命令。**

**ls -l /etc | more （表示将ls -l /etc的输出，当做more命令的输入，即more命令浏览的内容为前面命令的输出结果）**

**ls -l /etc | grep init（表示将ls -l /etc的输出结果进行过滤，显示为init的结果）**

**ls -l /etc | grep init | wc -l (最后进行统计显示的个数)**

### 2.1.10 逻辑命令符

**逻辑与（&&）**

**形如：ls && pwd（第一个命令如果执行成功。第二个命令才会执行）**

**逻辑或（||）**

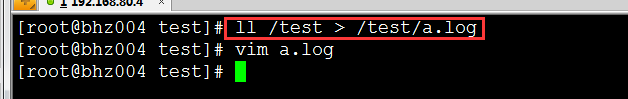
**形如：ls || pwd （第一个命令执行成功，则第二个不执行，第一个命令执行失败，则执行第二个）**

### 2.1.11输入输出重定向

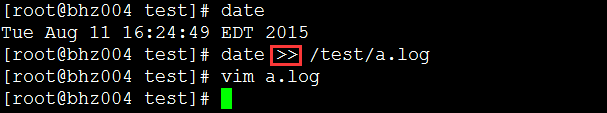
**Shell对每一个进程预先定义了3个文件描述字（0,1,2）**

**0 （stdin） 标准输入 1 （stdout）标准输出 2 （stderr）标准错误输出**

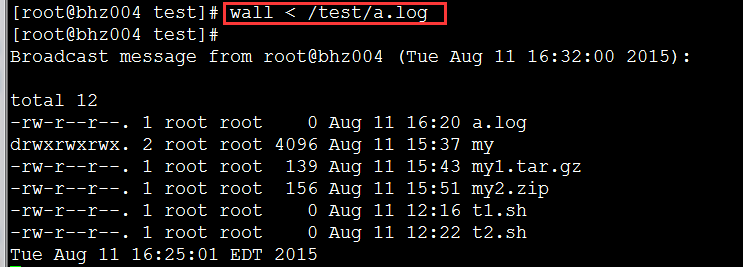
**输出重定向：就是把输出的结果显示到一个文件上 （>表示输出重定向）**



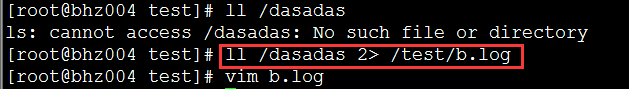
**如果想进行结果的追加，使用">>"**



**输入重定向：就是把输入的信息重定向，比如把一个文件里的内容，进行发出**



**错误重定向：一般是把程序执行的错误日志信息存放到指定的log日志中去。**



# 三 linux用户与文件

## 3.1 文本编辑

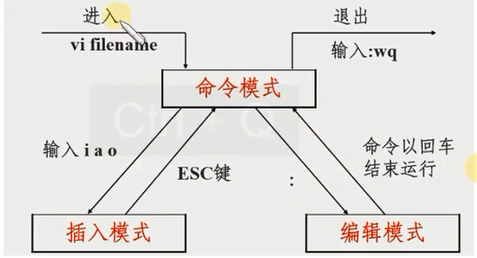
### 3.1.1 vi/vim

是一个功能强大的全屏幕文本编辑器，也是linux常用的文本编辑器，它的作用是建立，编辑，显示文本文件，没有菜单只有依靠命令去操作。

Vi/vim 【文件名】 进入编辑模式

使用insert 选择插入模式。

当使用文本编辑时，进入编辑模式。



输入a,I,o



### 3.1.2跳转到指定行

Set nu 设置行号

Set nonu 取消行号

如果想要跳转到指定行，直接进入编辑模式输入行号即可。



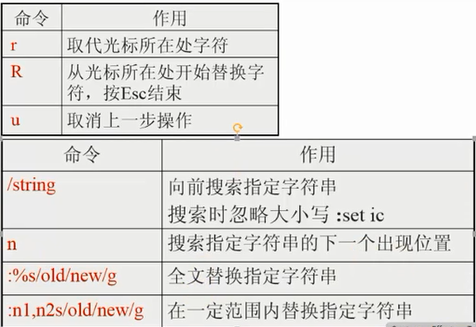
### 3.1.3 删除命令



### 3.1.4 复制与粘贴命令



如下图所示，在文本编辑模式下输入/hello 就可以全文检索hello字符串，按n可以一个一个依次切换匹配串。



如果替换的内容中含有/目录符，则需要用+进行替换

### 3.1.5 保存退出



### 3.1.6 一次性计划任务

At

At [-f 文件名] 时间

绝对计时方法: HH:MM YYYY-MM-DD

相对计时方法: now + n minutes now +n hours now +n days

描述:安排一个或多个命令在指定时间运行一次

## 3.2 用户管理

用户信息文件:/etc/passwd

密码文件: /etc/shadow

用户配置文件: /etc/login.defs ,/etc/default/useradd

新用户信息文件: /etc/skel

用户组文件: /etc/group

用户组密码文件: /etc/gshadow

/etc/passwd 文件中的字段标识符如下：



### 3.2.1 用户组管理

添加用户组 groupadd 【配置选项】【组名】

删除用户组 groupdel 【组名】

修改用户组 groupmod –n 【新组名】【就组名】

查看用户属于哪个组 groups u1

### 3.2.2 用户管理

添加用户 Useradd 【用户名】

U:uid

G:缺省所属的用户组名称

D:宿主目录

S:命令解释shell

C:描述信息

E:制定用户失效时间。

删除用户 userdel –r 【用户名】

修改用户 usermod –l 【新用户名】【旧用户名】

禁用与恢复

Usermod –L 【用户名】

Passwd –l 【用户名】

恢复

Usermod –U 【用户名】

Passwd –u 【用户名】

3.2.3 用户与用户组关联

Gpasswd

-a 添加用户到用户组中

-d 从用户组中删除用户

-A 设置用户组管理员

Gpasswd –a u1 webapps 添加用户u1到用户组webapps

Gpassws –d u1 root 从用户组root中删除用户u1

# 四 linux的shell编程

## 4.1 例子

### 4.1.1 demo1

#!/bin/sh

#This is to show what a example looks like.

echo "My First Shell!"

echo "This is current directory."

/bin/pwd

echo

echo "This is files."

/bin/ls

### 4.1.2 demo2

#!/bin/sh

/bin/date +%F >> /test/shelldir/ex2.info

echo "disk info:" >> /test/shelldir/ex2.info

/bin/df -h >> /test/shelldir/ex2.info

echo >> /test/shelldir/ex2.info

echo "online users:" >> /test/shelldir/ex2.info

/usr/bin/who | /bin/grep -v root >> /test/shelldir/ex2.info

echo "memory info:" >> /test/shelldir/ex2.info

/usr/bin/free -m >> /test/shelldir/ex2.info

echo >> /test/shelldir/ex2.info

#write root

/usr/bin/write root < /test/shelldir/ex2.info && /bin/rm /test/shelldir/ex2.info

#crontab -e

#0 9 \* \* 1-5 /bin /sh /test/ex2.sh

### 4.1.3 demo3

#!/bin/sh

DATE=`/bin/date +%Y%m%d`

echo "TODAY IS $DATE"

/bin/ls -l $1

/bin/ls -l $2

/bin/ls -l $3

执行上面的demo 需要输入三个参数，所以$1-$3分别表示三个输入参数,$date表示引用上下文变量。

特殊变量:

$\* 这个程序的所有参数

$# 这个程序的参数个数

$$ 这个程序的PID

$! 执行上一个后台命令的pid

$? 执行上一个命令的返回值

$(0-9) 显示位置变量

#!/bin/sh

DATE=`/bin/date +%F`

echo "today is $DATE"

echo '$# :' $#

echo '$\* :' $\*

echo '$? :' $?

echo '$$ :' $$

echo '$0 :' $0

### 4.1.4 demo4

#!/bin/sh

read f s t

echo "the first is $f"

echo "the second is $s"

echo "the third is $t"

### 4.1.5 demo5

分支语句语法 If test –d $1 then … else … fI

分支语句语法 if [-d $1] then … else ….fi

#!/bin/sh

# if test $1 then ... else ... fi

if [ -d $1 ]

then

echo "this is a directory!"

else

echo "this is not a directory!"

fi

### 4.1.6 demo6

#!/bin/sh

# if test then ... elif test then ... else ... fi

if [ -d $1 ]

then

echo "is a directory!"

elif [ -f $1 ]

then

echo "is a file!"

else

echo "error!"

fi

### 4.1.7 demo7

#!/bin/sh

# -a -o

if [ $1 -eq $2 -a $1 = 1 ]

then

echo "param1 == param2 and param1 = 1"

elif [ $1 -ne $2 -o $1 = 2 ]

then

echo "param1 != param2 or param1 = 2"

else

echo "others"

fi

-a 逻辑与

-o 逻辑或

### 4.1.8 demo8

#!/bin/sh

# for var in [params] do ... done

for var in 1 2 3 4 5 6 7 8 9 10

do

echo "number is $var"

done

### 4.1.9 demo9

#!/bin/sh

# select var in [params] do ... done

select var in "java" "c++" "php" "linux" "python" "ruby" "c#"

do

break

done

echo "you selected $var"

### 4.1.10 demo10

#!/bin/sh

read op

case $op in

a)

echo "you selected a";;

b)

echo "you selected b";;

c)

echo "you selected c";;

\*)

echo "error"

esac

### 4.1.11 demo11

#!/bin/sh

#while test do ... done

num=1

sum=0

while [ $num -le 100 ]

do

sum=`expr $sum + $num`

num=`expr $num + 1`

done

#sleep 5

echo $sum

### 4.1.12 demo12

#!/bin/sh

i=0

while [ $i -le 100 ]

do

i=`expr $i + 1`

if [ $i -eq 5 -o $i -eq 10 ]

then continue;

else

echo "this number is $i"

fi

if [ $i -eq 15 ]

then break;

fi

done