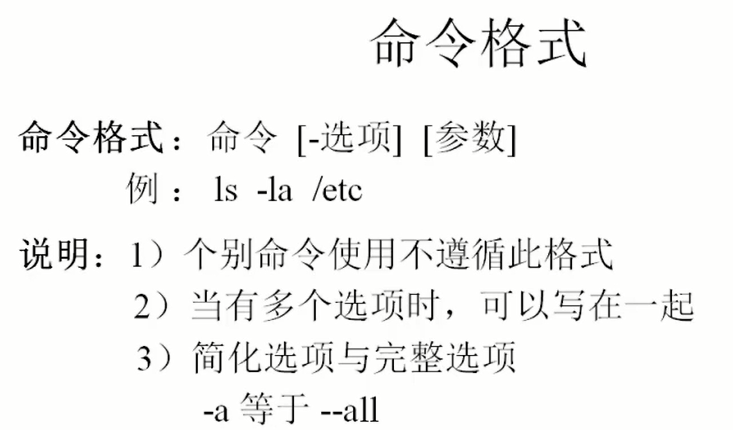
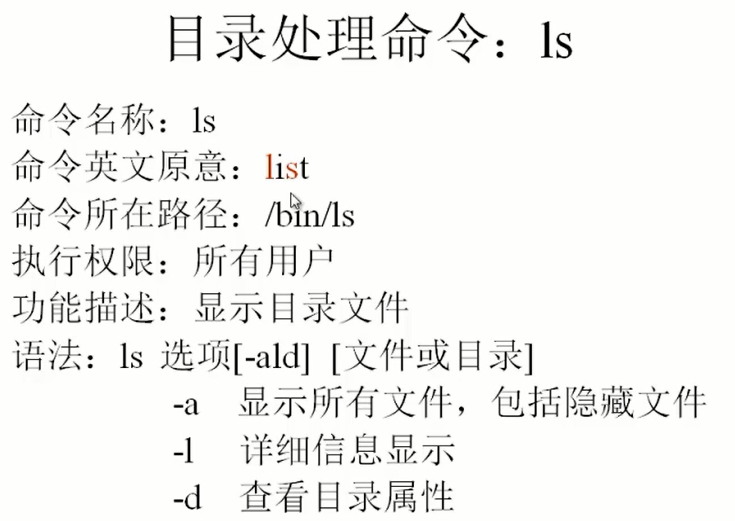
Linux

# Linux常用命令

## 文件处理命令

### Linux命令格式和规则 和目录处理命令ls





1. **基础使用**

[root@J2001 ~]# **ls**

anaconda-ks.cfg 公共 视频 文档 音乐

initial-setup-ks.cfg 模板 图片 下载 桌面

[root@J2001 ~]#

[root@J2001 ~]# **ls**

anaconda-ks.cfg 公共 视频 文档 音乐

initial-setup-ks.cfg 模板 图片 下载 桌面

[root@J2001 ~]# **ls -a**

. .bash\_profile .dbus .mozilla 图片

.. .bashrc .esd\_auth .tcshrc 文档

anaconda-ks.cfg .cache .ICEauthority 公共 下载

.bash\_history .config initial-setup-ks.cfg 模板 音乐

.bash\_logout .cshrc .local 视频 桌面

[root@J2001 ~]# ls / 查看指定目录的信息

bin dev home lib64 mnt proc run srv tmp var

boot etc lib media opt root sbin sys usr

[root@J2001 ~]# ls –l 详细查看文件信息

总用量 8

-rw-------. 1 root root 1582 3月 1 21:39 anaconda-ks.cfg

-rw-r--r--. 1 root root 1613 3月 1 21:49 initial-setup-ks.cfg

drwxr-xr-x. 2 root root 6 3月 1 21:54 公共

drwxr-xr-x. 2 root root 6 3月 1 21:54 模板

drwxr-xr-x. 2 root root 6 3月 1 21:54 视频

drwxr-xr-x. 2 root root 6 3月 1 21:54 图片

drwxr-xr-x. 2 root root 6 3月 1 21:54 文档

drwxr-xr-x. 2 root root 6 3月 1 21:54 下载

drwxr-xr-x. 2 root root 6 3月 1 21:54 音乐

drwxr-xr-x. 2 root root 6 3月 1 21:54 桌面

用户的所有者 只能有一个人 一般是创建者 所有者可以发生变化

所属组 定义了一组相同类型的用户

其他人

**2)文件参数的详解**

-rw-------.

-rw-r--r--.

drwxr-xr-x.

* 第一个参数 是文件类型 - 代表二进制文件 d代表目录 l 代表软连接文件、
* 第一个后 参数代表权限 三个为一组 分别表示 u所有者的权限 g所属组的权限 o其他人的权限 r可读 w可写 x可执行

rw------- 所有者rw- 所属组 --- 其他人 ---

rw-r--r— 所有者rw- 所属组r-- 其他人---

rwxr-xr-x 所有者rwx 所属组 r-x 其他人 r-x

* 第二个参数代表 硬链接数
* 第三个参数代表 所有者
* 第四个参数代表 所属组
* 第五个参数代表 文件大小
* 第六个参数 代表文件最后一个修改日期
* 第七个参数代表 文件的名称

[root@J2001 ~]# **ls -lh**

总用量 8.0K

-rw-------. 1 root root 1.6K 3月 1 21:39 anaconda-ks.cfg

-rw-r--r--. 1 root root 1.6K 3月 1 21:49 initial-setup-ks.cfg

drwxr-xr-x. 2 root root 6 3月 1 21:54 公共

drwxr-xr-x. 2 root root 6 3月 1 21:54 模板

drwxr-xr-x. 2 root root 6 3月 1 21:54 视频

drwxr-xr-x. 2 root root 6 3月 1 21:54 图片

drwxr-xr-x. 2 root root 6 3月 1 21:54 文档

drwxr-xr-x. 2 root root 6 3月 1 21:54 下载

drwxr-xr-x. 2 root root 6 3月 1 21:54 音乐

drwxr-xr-x. 2 root root 6 3月 1 21:54 桌面

**3）其他用法**

**[root@J2001 ~]# ls /etc 查看目录之内的文件**

**[root@J2001 ~]# ls -l /etc**

**[root@J2001 ~]# ls -ld /etc 查看当前目录的属性**

**drwxr-xr-x. 138 root root 8192 3月 2 09:32 /etc**

### 目录处理命令



[root@J2001 ~]# **mkdir J2001**

[root@J2001 ~]# ls

[root@J2001 ~]# **mkdir -p /tmp/guigu/J2001**

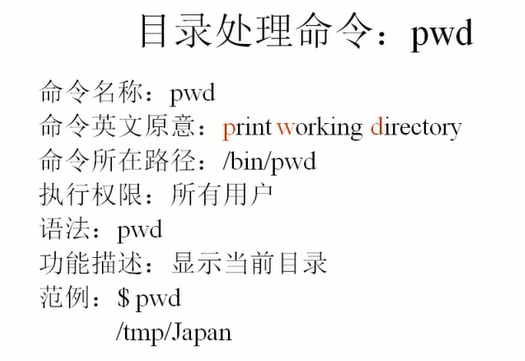
[root@J2001 ~]# **ls /tmp/guigu**

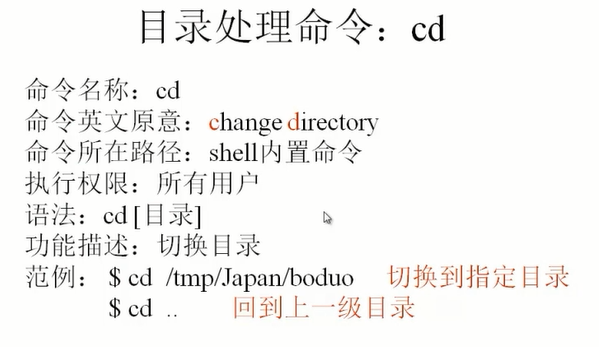
J2001

[root@J2001 ~]# **mkdir -p /tmp/guigu/J2002 /tmp/guigu/J2003 /tmp/guigu/J2004**

[root@J2001 ~]# **ls /tmp/guigu**

J2001 J2002 J2003 J2004





[root@J2001 ~]# pwd

/root

[root@J2001 ~]# cd /tmp/guigu

[root@J2001 guigu]# pwd

/tmp/guigu



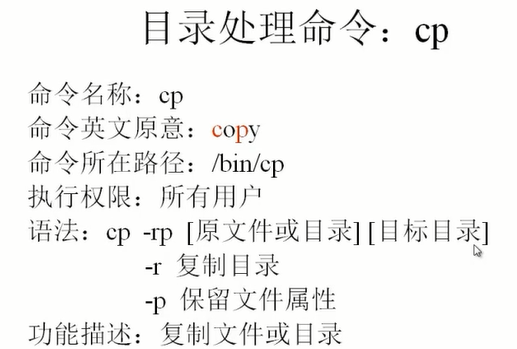
[root@J2001 guigu]# ls

J2001 J2002 J2003 J2004

[root@J2001 guigu]# rmdir J2004

[root@J2001 guigu]# ls

J2001 J2002 J2003



[root@J2001 ~]# **cp -r /tmp/guigu /root 拷贝文件夹到指定位置**

[root@J2001 ~]# ls

anaconda-ks.cfg initial-setup-ks.cfg 公共 视频 文档 音乐

guigu J2001 模板 图片 下载 桌面

[root@J2001 ~]# cd guigu

[root@J2001 guigu]# pwd

/root/guigu

[root@J2001 guigu]# ls

[root@J2001 guigu]# **cp /root/anaconda-ks.cfg /root/initial-setup-ks.cfg /tmp/guigu 拷贝多个文件到指定位置**

[root@J2001 guigu]# ls /tmp/guigu

anaconda-ks.cfg initial-setup-ks.cfg J2001 J2002 J2003

[root@J2001 guigu]# ls -ld /tmp/guigu

drwxr-xr-x. 5 root root 96 3月 2 10:29 /tmp/guigu

[root@J2001 guigu]# ls -ld /root/guigu

drwxr-xr-x. 5 root root 45 3月 2 10:27 /root/guigu

[root@J2001 guigu]# **cp -rp /root/initial-setup-ks.cfg /tmp/guigu/J2001 拷贝文件并且保留属性**

[root@J2001 guigu]# ls -lh /tmp/guigu/J2001



[root@J2001 guigu]# ls

J2001 J2002 J2003

[root@J2001 guigu]# pwd

/root/guigu

[root@J2001 guigu]# cd ..

[root@J2001 ~]# ls

anaconda-ks.cfg initial-setup-ks.cfg 公共 视频 文档 音乐

guigu J2001 模板 图片 下载 桌面

[root@J2001 ~]# cd guigu

[root@J2001 guigu]# ls

J2001 J2002 J2003

[root@J2001 guigu]# **mv J2002 /root/ 剪切文件到指定位置**

[root@J2001 guigu]# ls /root

[root@J2001 guigu]# ls

J2001 J2003



[root@J2001 ~]# rm -rf guigu

[root@J2001 ~]# ls

**总结：**

ls 列出 - a全部 -l详细 -h大小显示 -d目录

mkdir创建目录 -p 创建多级目录

pwd 显示当前目录

cd 进入当前目录 cd.. 退出 上级目录 cd 进入根目录

rmdir 删除空目录

cp 拷贝文件 -r 拷贝目录 -p 保留所有属性

mv 剪切 和重命名

rm删除文件或者目录 -r 删除目录 –f 强制删除

### 文件处理命令



[root@J2001 guigu]# **touch hellolinux.ini**

[root@J2001 guigu]# ls

hellolinux.ini J2001 J2002 J2003

[root@J2001 guigu]# **touch hzguigu.txt**

[root@J2001 guigu]# ls

hellolinux.ini hzguigu.txt J2001 J2002 J2003

[root@J2001 guigu]# **touch hzjava**



[root@J2001 guigu]# **cat /etc/issue 显示指定文件的内容**

\S

Kernel \r on an \m

[root@J2001 guigu]# **cat -n /etc/issue 显示指定文件的内容并且显示行号**

1 \S

2 Kernel \r on an \m

3

[root@J2001 guigu]# **cat -n /etc/service**

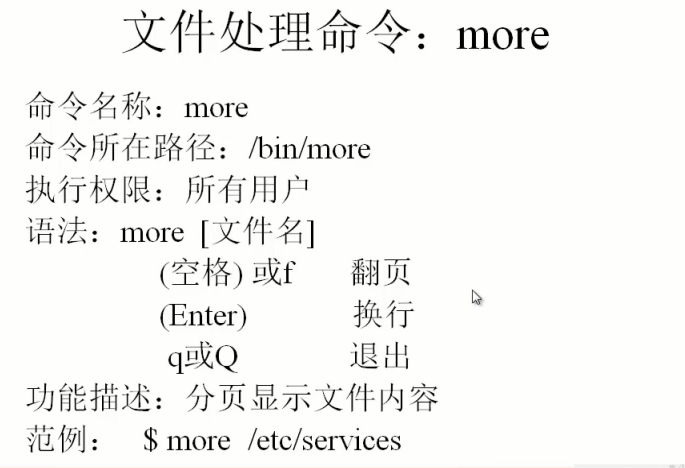
Cat不适合显示文件内容比较大的文件



**[root@J2001 guigu]# tac /etc/issue**

Kernel \r on an \m

\S



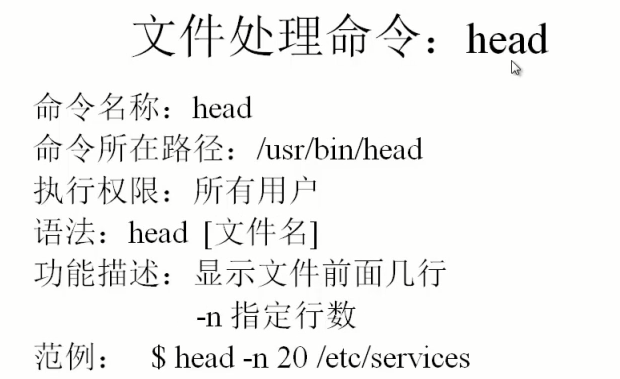
[root@J2001 guigu]# **more /etc/service**



[root@J2001 guigu]# less /etc/services

可以通过pageup 向上翻页 或者向上的箭头翻页 也可以通过/ 进行匹配查找的内容

可以通过/ \*\* 查找匹配的内容 然后通过n 进行下一个查找s

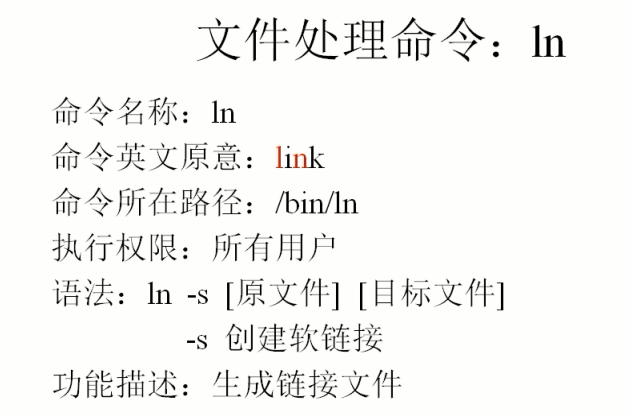


[root@J2001 guigu]# **head -n 30 /etc/services、**



[root@J2001 guigu]# **tail -n 10 /etc/services**

### 连接命令

****

**软连接 可以理解为快捷方式**

**硬链接 文件的拷贝+同步 源文件的一份拷贝 内容会同步更新**

**[root@J2001 ~]# ln -s /etc/issue /tmp/guigu/issue.soft**

**[root@J2001 ~]# ln /etc/issue /tmp/guigu/issue.hard**

**[root@J2001 ~]# cd /tmp/guigu**

**[root@J2001 guigu]# ls**

**hellolinux.ini hzguigu.txt hzjava issue.hard issue.soft J2001 J2002 J2003**

**[root@J2001 guigu]# ls -l**

**总用量 4**

**-rw-r--r--. 1 root root 0 3月 2 10:43 hellolinux.ini**

**-rw-r--r--. 1 root root 0 3月 2 10:43 hzguigu.txt**

**-rw-r--r--. 1 root root 0 3月 2 10:44 hzjava**

**-rw-r--r--. 2 root root 23 8月 30 2017 issue.hard**

**lrwxrwxrwx. 1 root root 10 3月 2 13:37 issue.soft -> /etc/issue**

**drwxr-xr-x. 2 root root 34 3月 2 10:32 J2001**

**drwxr-xr-x. 2 root root 6 3月 2 10:23 J2002**

**drwxr-xr-x. 2 root root 6 3月 2 10:23 J2003**

**[root@J2001 guigu]# echo "www.hzguigu.com">> /etc/issue 更新内容到指定的文件**

**[root@J2001 guigu]# cat /etc/issue 获取文件内容**

**\S**

**Kernel \r on an \m**

**www.hzguigu.com**

**[root@J2001 guigu]# cat /tmp/guigu/issue.hard**

**\S**

**Kernel \r on an \m**

**www.hzguigu.com**

**[root@J2001 guigu]# ls -i /etc/issue**

**16777322 /etc/issue**

**[root@J2001 guigu]# ls -i /tmp/guigu/issue.soft**

**18458950 /tmp/guigu/issue.soft**

**[root@J2001 guigu]# ls -i /tmp/guigu/issue.hard**

**16777322 /tmp/guigu/issue.hard**

**[root@J2001 guigu]# rm -rf /etc/issue**

**[root@J2001 guigu]# ls**

**hellolinux.ini hzguigu.txt hzjava issue.hard issue.soft J2001 J2002 J2003**

**[root@J2001 guigu]# ls -l**

## 权限管理命令

### 权限管理命令chmod

****

用户分为三种 U所有者 G 所有组 O其他人 a 所有用户

赋权和取消权限 + 添加新的权限 -取消某种权限 = 不管你之前什么权限现在设置的就是你的权限

R 可读 w可写 x可执行

**方式一赋权**

**[root@J2001 guigu]# ls -l /tmp/guigu/hellolinux.ini**

**-rw-r--r--. 1 root root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# chmod u+x /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# ls -l /tmp/guigu/hellolinux.ini**

**-rwxr--r--. 1 root root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# chmod g+w,o-r /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# ls -l /tmp/guigu/hellolinux.ini**

**-rwxrw----. 1 root root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# chmod g=rwx /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# ls -l /tmp/guigu/hellolinux.ini**

**-rwxrwx---. 1 root root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini**

**方式二赋权 主要**

**r -----4**

**w----2**

**x ----1**

**rwxrw-r—**

**764**

**[root@J2001 guigu]# chmod 760 /tmp/guigu/hellolinux.ini**

**[root@J2001 guigu]# ls -l /tmp/guigu/hellolinux.ini**

**-rwxrw----. 1 root root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini**

**更改目录的权限**

**[root@J2001 ~]# mkdir -p /tmp/guigu/java/J2001**

**[root@J2001 ~]# chmod 777 /tmp/guigu/java/ 仅仅更改本级目录的权限**

**[root@J2001 ~]# ls -ld /tmp/guigu/java 查看指定目前的信息**

**drwxrwxrwx. 3 root root 19 3月 2 13:59 /tmp/guigu/java**

**[root@J2001 ~]# ls -ld /tmp/guigu/java/J2001**

**drwxr-xr-x. 2 root root 6 3月 2 13:59 /tmp/guigu/java/J2001**

**[root@J2001 ~]# chmod 777 -R /tmp/guigu/java/ -R的递归设置权限**

**[root@J2001 ~]# ls -ld /tmp/guigu/java**

**drwxrwxrwx. 3 root root 19 3月 2 13:59 /tmp/guigu/java**

**[root@J2001 ~]# ls -ld /tmp/guigu/java/J2001**

**drwxrwxrwx. 2 root root 6 3月 2 13:59 /tmp/guigu/java/J2001**

****

**目录下有r的权限 一般都有x的权限 。是**

**只有当拥有w的权限 才可以进行创建和删除**

**文件的权限和目录的权限概念不同。**

**[root@J2001 ~]# touch /tmp/guigu/testFile.ini**

**[root@J2001 ~]# ls -ld /tmp/guigu**

**drwxrwxrwx. 6 root root 132 3月 2 14:09 /tmp/guigu**

**[root@J2001 ~]# chmod 777 -R /tmp/guigu 在hzguigu用户下正常删除没问题**

**[root@J2001 ~]# touch /tmp/guigu/testFile3.ini**

**[root@J2001 ~]# ls -ld /tmp/guigu**

**drwxrwxrwx. 6 root root 132 3月 2 14:12 /tmp/guigu**

**[root@J2001 ~]# chmod 754 /tmp/guigu 文件夹的权限是rwxr-xr—其他用户无法删除文件**

**[root@J2001 ~]# touch /tmp/guigu/testFile4.ini**

**[root@J2001 ~]# chmod 755 /tmp/guigu 件夹的权限是rwxr-xr-x其他用户无法删除文件**

**[hzguigu@J2001 ~]$ ls -l /tmp/guigu/**

**总用量 0**

**-rwxrwxrwx. 1 root root 0 3月 2 10:43 hellolinux.ini**

**-rwxrwxrwx. 1 root root 0 3月 2 10:43 hzguigu.txt**

**-rwxrwxrwx. 1 root root 0 3月 2 10:44 hzjava**

**drwxrwxrwx. 2 root root 34 3月 2 10:32 J2001**

**drwxrwxrwx. 2 root root 6 3月 2 10:23 J2002**

**drwxrwxrwx. 2 root root 6 3月 2 10:23 J2003**

**drwxrwxrwx. 3 root root 19 3月 2 13:59 java**

**-rw-r--r--. 1 root root 0 3月 2 14:14 testFile4.ini**

**-rwxrwxrwx. 1 root root 0 3月 2 14:09 testFile.ini**

**[hzguigu@J2001 ~]$ rm /tmp/guigu/testFile4.ini**

**rm：是否删除有写保护的普通空文件 "/tmp/guigu/testFile4.ini"？y**

**rm: 无法删除"/tmp/guigu/testFile4.ini": 权限不够**

### 其他权限管理命令



[root@J2001 ~]# chown hzguigu /tmp/guigu/hellolinux.ini

[root@J2001 ~]# ls -l /tmp/guigu/hellolinux.ini

-rwxrwxrwx. 1 hzguigu root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini



[root@J2001 ~]# **groupadd J2001\_hzguigu 新增一个组**

[hzguigu@J2001 ~]$ **ls -l /tmp/guigu/hellolinux.ini**  查看详细信息

-rwxrwxrwx. 1 hzguigu root 0 3月 2 10:43 /tmp/guigu/hellolinux.ini

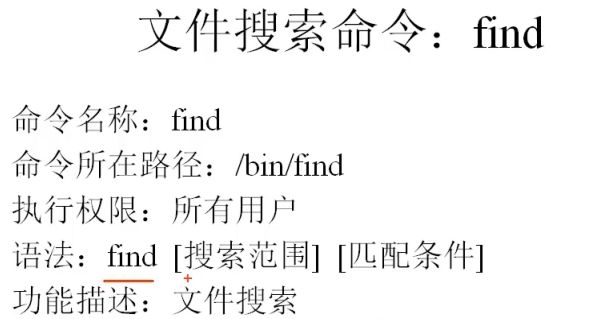
[root@J2001 ~]# **chgrp J2001\_hzguigu /tmp/guigu/hellolinux.ini 改变文件的所有组**

[hzguigu@J2001 ~]$ ls -l /tmp/guigu/hellolinux.ini 查看权限

-rwxrwxrwx. 1 hzguigu J2001\_hzguigu 0 3月 2 10:43 /tmp/guigu/hellolinux.ini

## 文件搜索的命令

### 文件搜索命令 find



1. 基础用法

1.在目录 /etc 下查找文件名为init d

[root@J2001 ~]# **find /etc -name init**

/etc/selinux/targeted/active/modules/100/init

/etc/sysconfig/init

2.在根目录下查找文件大小大于100m的文件

1个数据块 512字节 0.5k 所以100M=1024\*2\*100M

[root@J2001 ~]# f**ind / -size +204800**

**+n大于 -n小于 n等于**

3.在/tmp目录下查询 所有者是hzguigu的文件 -group 所属组

[root@J2001 ~]# find /tmp -user hzguigu

4、匹配查询 \* 匹配所有的文件 ？匹配单个字符

[root@J2001 ~]# **find /etc -name init???**

/etc/inittab

5、查找指定大小范围的文件

[root@J2001 ~]# find /etc -size +163840 -a -size -204800

-a 是两个条件都满足

-o 两条件满足一个即可

6.需要对查找到的文件再次执行操作

[root@J2001 ~]# **find /etc -name inittab -exec ls -l {} \;**

-rw-r--r--. 1 root root 511 8月 4 2017 /etc/inittab

-exec {} \; 是对查询结果进程执行操作

7.根据文件类型查找

-type 是根据类型查询 其中参数 –f 查找文件 -d是查询文件夹 -l是查询软连接

[root@J2001 ~]# **find -type l**

总结： find -name \* ？ -size + -

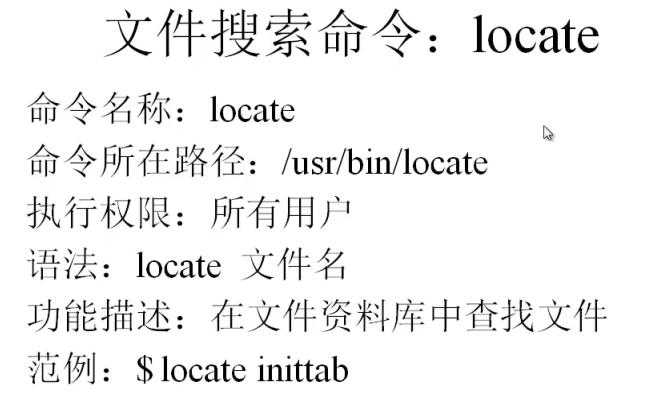
-user -group

-type f d l

-a -o

-exec {} \;

### 其他的搜索命令



Locate能快速查询和定位 但是新建的文件无法立即查询 需要更新资料库

Locate效率查询十分高

[root@J2001 ~]# **touch /root/hzguigu123**

[root@J2001 ~]# **locate hzguigu123**

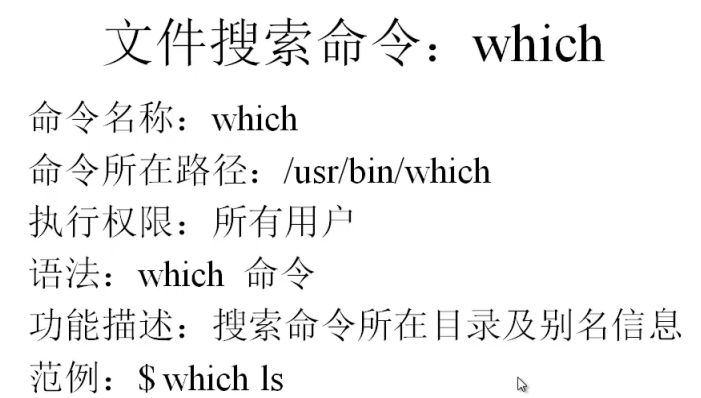
[root@J2001 ~]# **find /root -name hzguigu123**

/root/hzguigu123

[root@J2001 ~]# **updated 更新资料库**

[root@J2001 ~]# locate hzguigu123

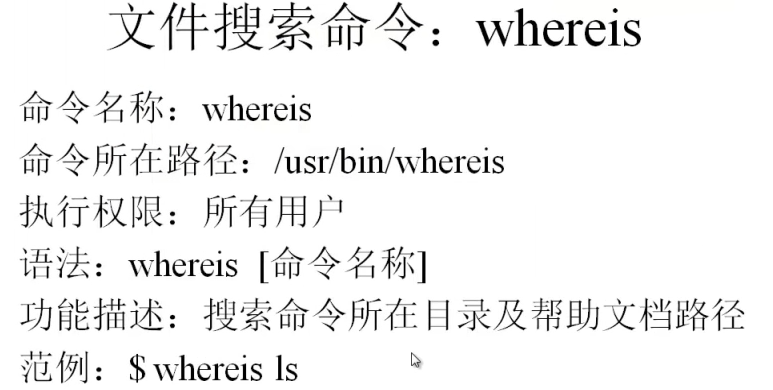
/root/hzguigu123



[root@J2001 ~]# **which cp**

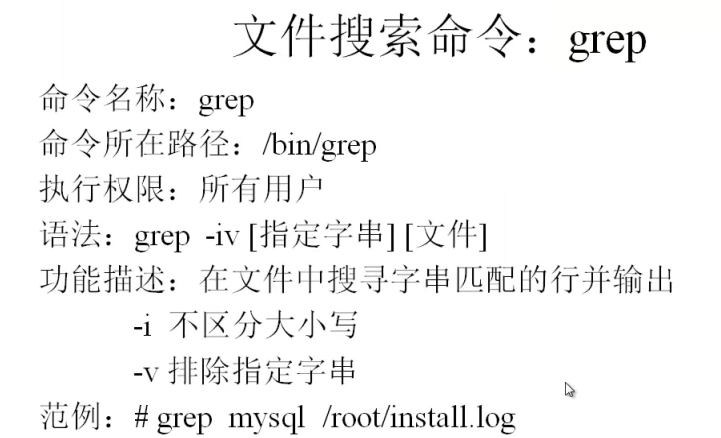
alias cp='cp -i'

/usr/bin/cp



[root@J2001 ~]# **whereis useradd**

useradd: /usr/sbin/useradd /usr/share/man/man8/useradd.8.gz



[root@J2001 ~]# **locate inittab**

/etc/inittab

/usr/share/augeas/lenses/dist/inittab.aug

/usr/share/man/zh\_CN/man5/inittab.5.gz

/usr/share/vim/vim74/syntax/inittab.vim

[root@J2001 ~]# **more /etc/inittab**

# inittab is no longer used when using systemd.

#

# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.

#

# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target

#

# systemd uses 'targets' instead of runlevels. By default, there are two main targets:

#

# multi-user.target: analogous to runlevel 3

# graphical.target: analogous to runlevel 5

#

# To view current default target, run:

# systemctl get-default

#

# To set a default target, run:

# systemctl set-default TARGET.target

#

[root@J2001 ~]# **grep multi-user /etc/inittab 查询multi-user相关的数据**

# multi-user.target: analogous to runlevel 3

[root@J2001 ~]# g**rep multi-user -i /etc/inittab 不区分大小写查询**

# multi-user.target: analogous to runlevel 3

[

[root@J2001 ~]# **grep -v ^# /etc/inittab 查询去除注释的内容**

## 帮助命令

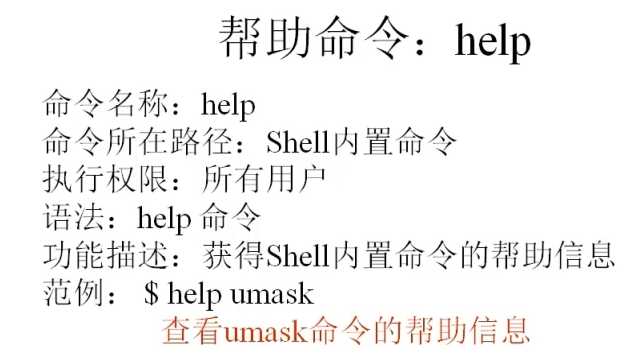
### Man



[root@J2001 ~]# man ls

[root@J2001 ~]# man passwd

### Help



ls --help 获取帮助 日月时分年秒

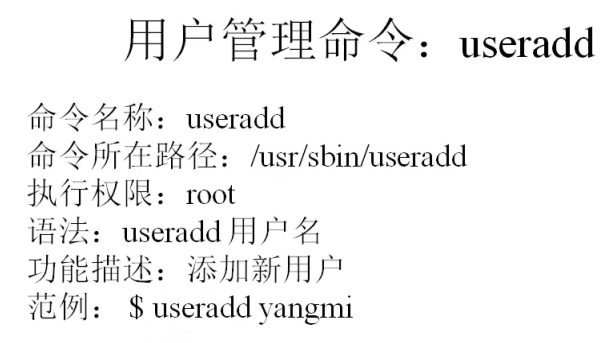
[root@J2001 ~]# date 030315512017.30

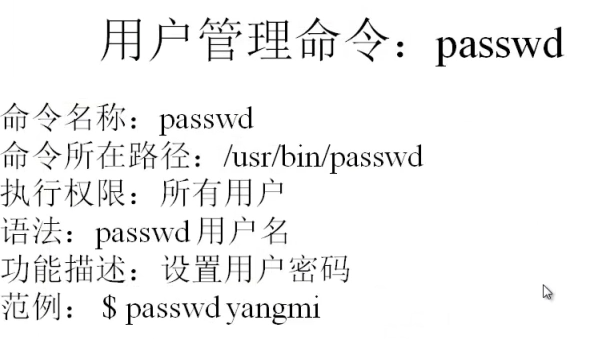
2017年 03月 03日 星期五 15:51:30 CST

[root@J2001 ~]# date 030215512017.30

2017年 03月 02日 星期四 15:51:30 CST

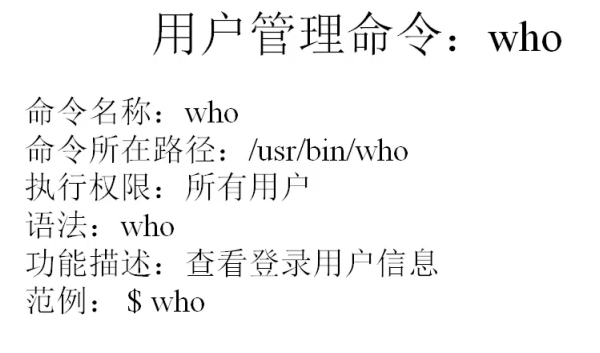
## 用户管理命令





[root@J2001 ~]# useradd zhangsan

[root@J2001 ~]# passwd zhangsan



[root@J2001 ~]# who

root :0 2018-03-02 12:35 (:0)

root pts/0 2018-03-03 08:33 (192.168.118.5)

zhangsan pts/1 2018-03-03 08:46 (192.168.118.5)

root pts/3 2018-03-02 14:36 (:0)

:0代表本地登录

第二个参数代表登录的终端 pts是代表远程登录

后方是登录的时间和ip地址

[root@J2001 ~]# **w 查看登录的详细信息**

08:50:04 up 3:37, 4 users, load average: 0.00, 0.06, 0.18

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

root :0 :0 五12 ?xdm? 12:17 0.61s /usr/libexec/gnome-session

root pts/0 192.168.118.5 08:33 4.00s 0.41s 0.07s w

zhangsan pts/1 192.168.118.5 08:46 1:40 0.11s 0.11s -bash

root pts/3 :0 五14 18:14m 0.09s 0.09s bash

up：代表服务器累计运行时间

load average 负载均衡信息

IDLE 累计空闲时间

JCPU 累计占用cpu的时间

PCUP代表占用cpu的时间

WHAT 执行的内容

## 压缩和解压命令

### 压缩文件





[root@J2001 lisi]# **gzip hellolinux**

[root@J2001 lisi]# ls

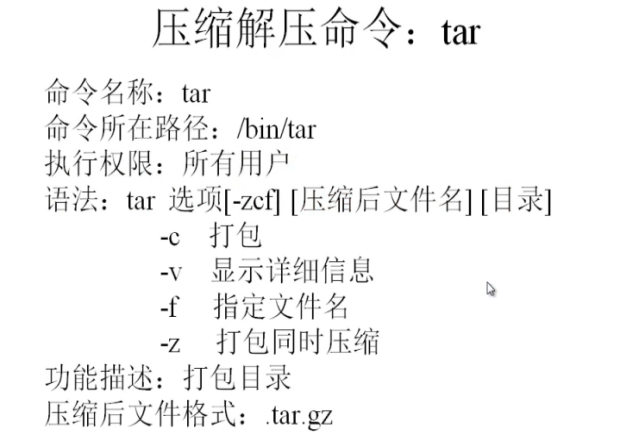
hellolinux.gz

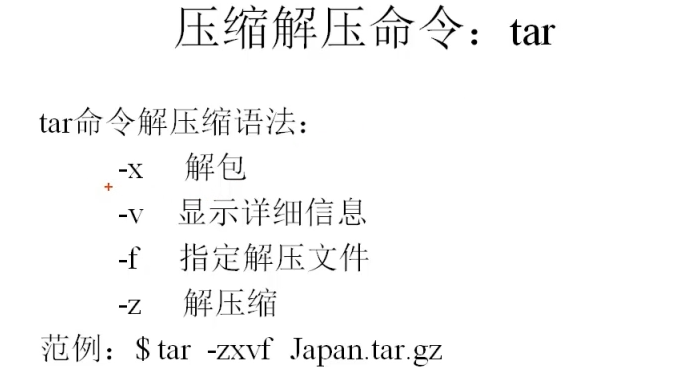
[root@J2001 lisi]# **gunzip hellolinux.gz**

[root@J2001 lisi]# ls

hellolinux

### 压缩文件夹





[root@J2001 lisi]# tar -cvf J2001.tar J2001 仅仅是把文件夹打包

J2001/

J2001/hellojava

[root@J2001 lisi]# **ls**

hellolinux J2001 J2001.tar

[root@J2001 lisi]# **ls -l**

总用量 12

-rw-r--r--. 1 root root 0 3月 3 09:06 hellolinux

drwxr-xr-x. 2 root root 23 3月 3 09:11 J2001

-rw-r--r--. 1 root root 10240 3月 3 09:14 J2001.tar

[root@J2001 lisi]# **ls -lh**

总用量 12K

-rw-r--r--. 1 root root 0 3月 3 09:06 hellolinux

drwxr-xr-x. 2 root root 23 3月 3 09:11 J2001

-rw-r--r--. 1 root root 10K 3月 3 09:14 J2001.tar

[root@J2001 lisi]# **gzip J2001.tar 压缩文件**

[root@J2001 lisi]# ls

hellolinux J2001 J2001.tar.gz

[root@J2001 lisi]# ls -lh

总用量 4.0K

-rw-r--r--. 1 root root 0 3月 3 09:06 hellolinux

drwxr-xr-x. 2 root root 23 3月 3 09:11 J2001

-rw-r--r--. 1 root root 157 3月 3 09:14 J2001.tar.gz

[root@J2001 lisi]# **tar -zxvf J2001.tar.gz 解压缩文件**

J2001/

J2001/hellojava

[root@J2001 lisi]# ls

hellolinux J2001 J2001.tar.gz

一次性打包并且压缩文件

[root@J2001 lisi]# **tar -zcvf J2001.tar.gz J2001**

J2001/

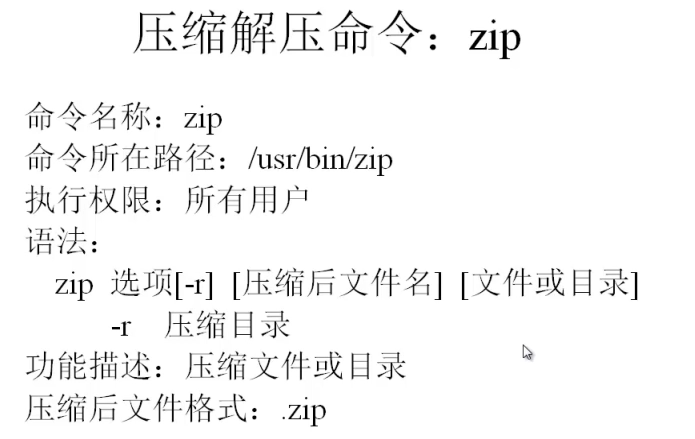
J2001/hellojava

[root@J2001 lisi]# ls

hellolinux J2001 J2001.tar.gz

### zip

linux压缩的zip可以直接到Windows上使用



[root@J2001 lisi]# **zip hellolinux.zip hellolinux**

adding: hellolinux (stored 0%)

[root@J2001 lisi]# **ls**

hellolinux hellolinux.zip J2001 J2001.tar.gz

[root@J2001 lisi]# **zip -r J2001.zip J2001**

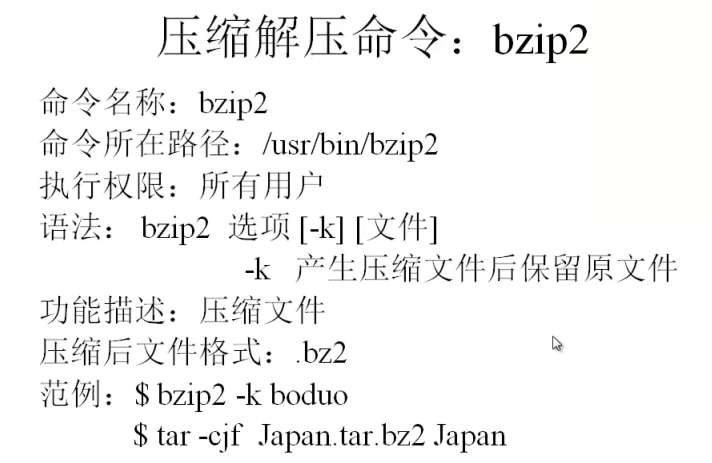
adding: J2001/ (stored 0%)

adding: J2001/hellojava (stored 0%)

[root@J2001 lisi]# **ls**

hellolinux hellolinux.zip J2001 J2001.tar.gz J2001.zip

### bzip2



Bzip2压缩效率比较高

[root@J2001 lisi]# **bzip2 -k hellolinux**

[root@J2001 lisi]# ls

hellolinux hellolinux.bz2 J2001

[root@J2001 lisi]# **tar -jcvf J2001.tar.bz2 J2001**

J2001/

J2001/hellojava

[root@J2001 lisi]# ls

hellolinux hellolinux.bz2 J2001 J2001.tar.bz2

[root@J2001 lisi]# rm -rf hellolinux

[root@J2001 lisi]# rm -rf J2001

[root@J2001 lisi]# ls

hellolinux.bz2 J2001.tar.bz2

[root@J2001 lisi]# **bunzip2 -k hellolinux.bz2**

[root@J2001 lisi]# ls

hellolinux hellolinux.bz2 J2001.tar.bz2

[root@J2001 lisi]# **tar -jxvf J2001.tar.bz2**

J2001/

J2001/hellojava

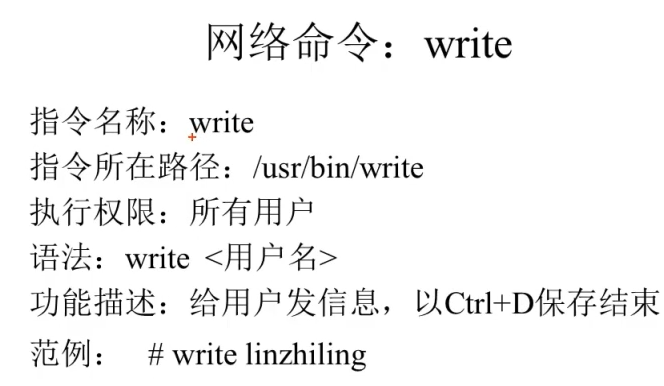
[root@J2001 lisi]# ls

hellolinux hellolinux.bz2 J2001 J2001.tar.bz2

### 总结

|  |  |  |  |
| --- | --- | --- | --- |
| 压缩后缀名 | 压缩命令 | 解压命令 | 说明 |
| .gz | gzip | gunzip | 对单个文件压缩 |
| .tar | Tar tar -cvf | ta r -xvf | 对文件夹打包 和解包 |
| .tar.gz | Tar -zcvf | Tar -zxvf | 对文件夹打包压缩和解包解压 |
| .zip | Zip  Zip –r 压缩文件夹 | Unzip |  |
| .bz2 | Bzip2 | Bunzip2 | 对zip压缩的升级 |
| .tar.bz2 | Tar -jcvf | Tar -jxvf | 对文件打包压缩和解包解压 |

## 网络命令



[root@J2001 ~]# write zhangsan

write: zhangsan is logged in more than once; writing to pts/5

hello zhangsan this is tudou

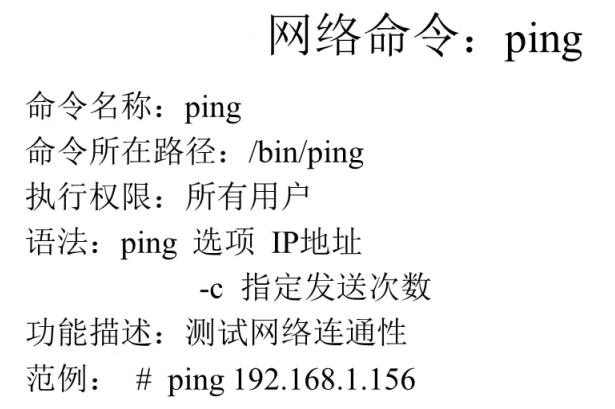


[root@J2001 ~]# wall hello aini l love you

[root@J2001 ~]#

Broadcast message from root@J2001 (pts/4) (Sat Mar 3 10:25:46 2018):

hello aini l love you



[root@J2001 ~]# ping 192.168.118.5 -c 6



[root@J2001 ~]# **ifconfig**

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.118.6 netmask 255.255.255.0 broadcast 192.168.118.255

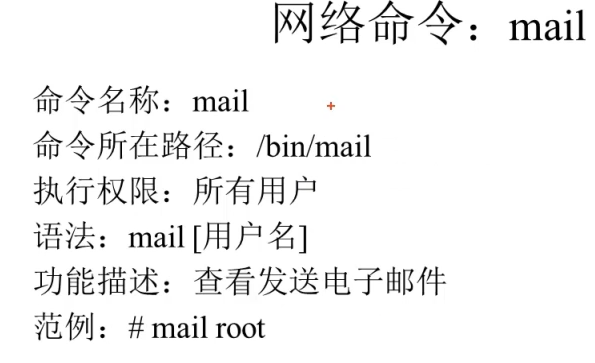
inet6 fe80::685c:383b:a2f0:6392 prefixlen 64 scopeid 0x20<link>

ether 00:50:56:2c:1f:88 txqueuelen 1000 (Ethernet)

RX packets 30922 bytes 4455804 (4.2 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 18757 bytes 27175255 (25.9 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  


发送邮件

[root@guigu ~]# **mail ccc**

Subject: qichuangxuexila

linux haha is so esay

EOT

[root@guigu ~]# **mail ccc**

Subject: hello1

hello1

EOT

[root@guigu ~]# mail ccc

Subject: hello2

hello22222

EOT

接受邮件

[ccc@guigu ~]$ mail

Heirloom Mail version 12.5 7/5/10. Type ? for help.

"/var/spool/mail/ccc": 2 messages 2 new

>N 1 root Sat Mar 3 10:34 18/576 "hello1"

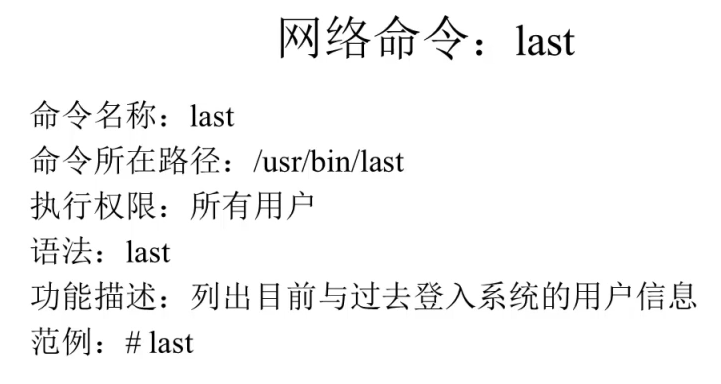
N 2 root Sat Mar 3 10:35 18/580 "hello2"

输入前方的数字 即可查看对应的邮件

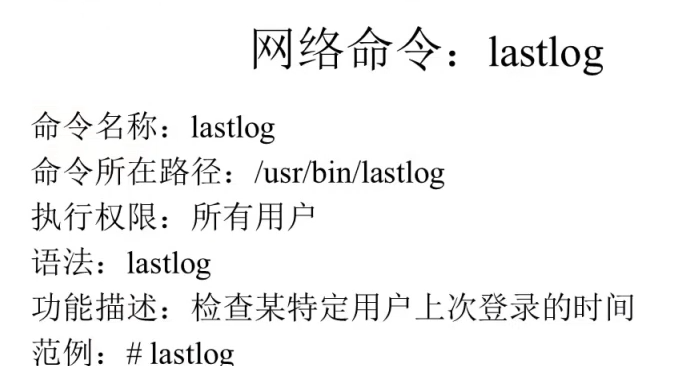
h查看邮件列表

d 删除 d 1 删除第一个邮件 d 2

q退出邮件



[root@guigu ~]# **last**



检查所有用户最后一次登录时间 即使创建了用户从未登录也会记录

[root@J2001 ~]# **lastlog**



[root@J2001 ~]# **traceroute www.baidu.com**

traceroute to www.baidu.com (115.239.211.112), 30 hops max, 60 byte packets

1 \* \* \*

2 192.168.100.254 (192.168.100.254) 0.541 ms 0.411 ms 0.313 ms

3 \* \* \*

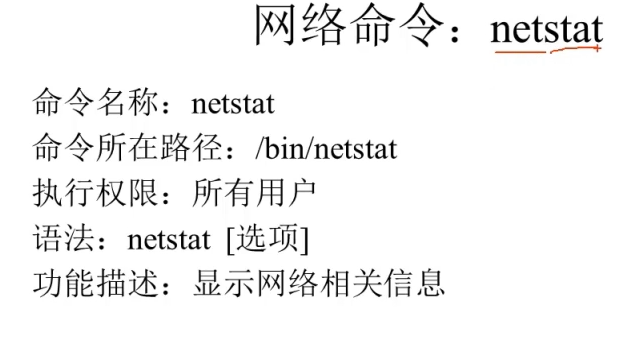
4 183.129.152.225 (183.129.152.225) 2.594 ms 2.486 ms 2.344 ms

5 61.130.127.22 (61.130.127.22) 2.560 ms 61.130.127.10 (61.130.127.10) 2.451 ms 61.130.127.8 (61.130.127.8) 3.629 ms

6 \* 61.164.31.206 (61.164.31.206) 2.486 ms 61.164.31.202 (61.164.31.202) 5.044 ms

7 \* \* \*

8 115.239.209.18 (115.239.209.18) 3.834 ms 115.239.209.30 (115.239.209.30) 5.779 ms 115.239.209.26 (115.239.209.26) 5.844 ms



-t 显示所有的tcp协议

-u 显示所有的udp协议

-l 显示所有的监听

-r 显示所有的路由

-n 显示ip地址和端口号

--举例 查看本机监听的端口号

[root@J2001 ~]#

[root@J2001 ~]# netstat -tuln

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN

tcp 0 0 192.168.122.1:53 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:631 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN

tcp6 0 0 :::111 :::\* LISTEN

tcp6 0 0 :::22 :::\* LISTEN

tcp6 0 0 ::1:631 :::\* LISTEN

tcp6 0 0 ::1:25 :::\* LISTEN

udp 0 0 192.168.122.1:53 0.0.0.0:\*

udp 0 0 0.0.0.0:67 0.0.0.0:\*

udp 0 0 0.0.0.0:5353 0.0.0.0:\*

udp 0 0 0.0.0.0:40260 0.0.0.0:\*

**nmtui**

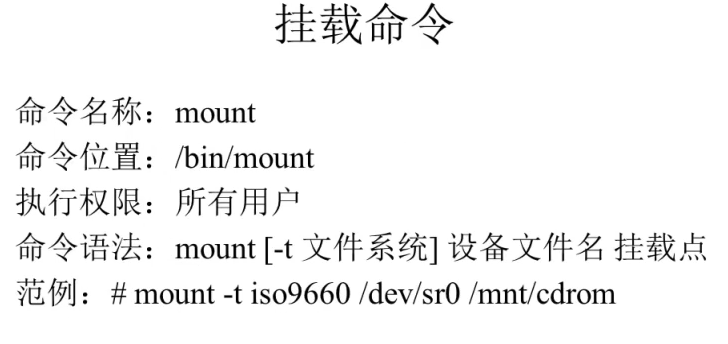
**设置网络连接和主机名称**

[root@J2001 ~]# nmtui

如果修改了ip网络连接后 必须重启网络才可以使用

ssservice network restart

## 挂载磁盘



[root@J2001 ~]# **mkdir /mnt/cdrom 创建挂载目录**

[root@J2001 ~]# **ll /dev/cdrom 查看软连接**

lrwxrwxrwx. 1 root root 3 3月 2 12:33 /dev/cdrom -> sr0

[root@J2001 ~]# **mount -t iso9660 /dev/sr0 /mnt/cdrom 挂载磁盘 方式一**

**[root@J2001 ~]# umount /dev/sr0 卸载挂载**

**[root@J2001 ~]# mount -t auto /dev/sr0 /mnt/cdrom**

**mount: /dev/sr0 写保护，将以只读方式挂载**

[root@J2001 ~]# **ls /mnt/cdrom/**

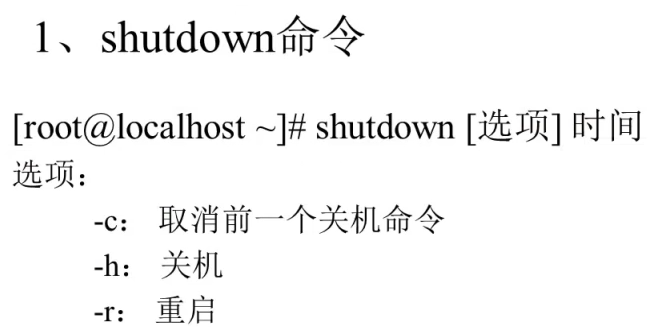
CentOS\_BuildTag GPL LiveOS RPM-GPG-KEY-CentOS-7

EFI images Packages RPM-GPG-KEY-CentOS-Testing-7

EULA isolinux repodata TRANS.TBL

[root@J2001 ~]# **cd /mnt/cdrom/**

## 关机和重启命令



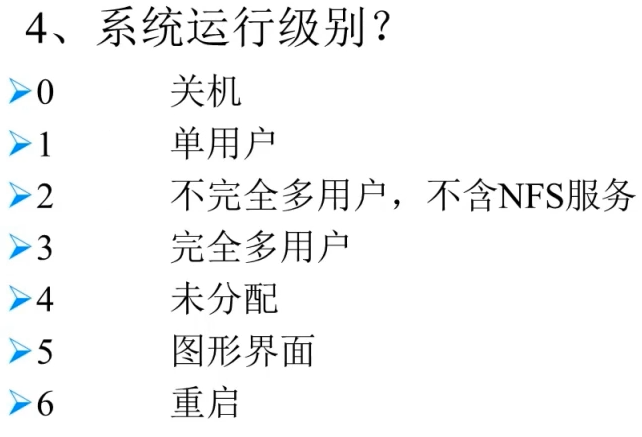
Shutdown -h now 立马关机

Shutdown –h 20:30 在指定时间进行关机

Shutdown –r now 立即重启







Runlevel查看当前运行级别

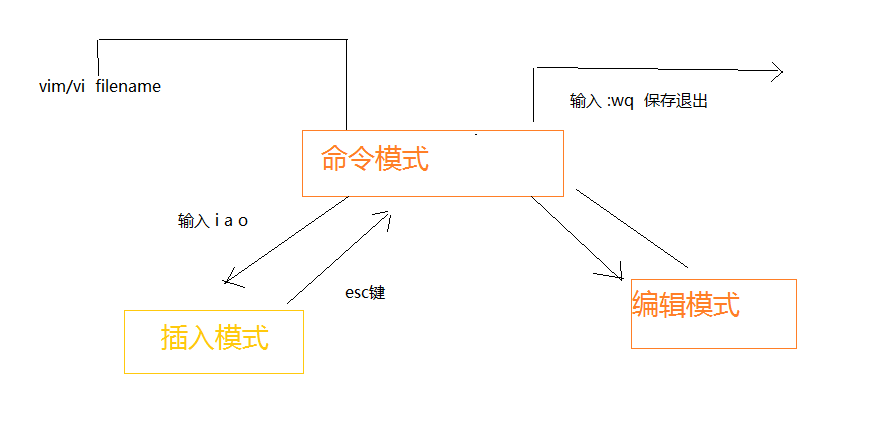
可以通过 init 3 init5 改变不同的级别

# VIM 文本编辑器

## VIM基础操作

Vim是一个功能强大的文本编辑器

Vim没有菜单 只有命令



1. **Vim的基本操作**

a i o 是进入到编辑模式 就可以进行文件的修改

a 是在光标所在字符之后插入

i 是光标所在字符前插入

o 是在光标的 插入新行

esc 是退出编辑模式进入到命令模式

进入到命令格可以通过

:wq 保存并且退出文件

:q 是退出

:q! 是强制退出不保存修改

:wq! 是强制退出并保存

都是命令模式下

:set nu 设置显示行号信

:set nonu 设置取消行号

1. **定位命令**

gg 到第一行

G 到最后一行

nG 100G 到指定的100行数

:n :500 到指定的500行

$ 移至行尾

0是移至行首

1. **删除**

[root@J2001 ~]# cp /etc/services /tmp/guigu 拷贝一份文件

x 删除光标所在的字符

nx 删除光标之后的n个字符 5x就是 删除光标所在位置之后的5个字符

dd 删除光标所在的行

dG 删除光标所在位置到行末尾的所有数据

D 删除光标所在处到当前行末尾的位置

:n1,n2d :700,703d 是删除700-703行

1. **复制和剪切的命令**

yy 复制当前行

nyy 复制当前行以及往下的n行

dd 剪切当前行

ndd 剪切当前行以及往下的n行

p 粘贴

1. **替换和取消命令**

u 取消上一步的操作

r 取代大当前光标所在处的字符

安装jdk

[root@J2001 sofeware]# mkdir -p /usr/local/sofeware/

上传文件到指定目录

[root@J2001 sofeware]# **cd /usr/local/sofeware/**

[root@J2001 sofeware]# **tar -zxvf jdk-8u161-linux-x64.tar.gz -C /usr/local/sofeware/**

[root@J2001 sofeware]# **vim /etc/profile**

配置一下内容

export JAVA\_HOME=/usr/local/sofeware/jdk1.8.0\_161

export PATH=$JAVA\_HOME/bin:$PATH

[root@J2001 sofeware]# source /etc/profile **让配置文件生效**

[root@J2001 sofeware]# **java –version**

**安装tomcat**

[root@J2001 bin]# **tar -zxvf apache-tomcat-8.5.28.tar.gz -C /usr/local/sofeware/**

[root@J2001 sofeware]# ls

apache-tomcat-8.5.28 httpd-2.4.29.tar.bz2 jdk1.8.0\_161

apache-tomcat-8.5.28.tar.gz httpd-2.4.29.tar.gz jdk-8u161-linux-x64.tar.gz

[root@J2001 sofeware]# **cd apache-tomcat-8.5.28/conf/**

[root@J2001 conf]# ls

catalina.policy jaspic-providers.xml server.xml web.xml

catalina.properties jaspic-providers.xsd tomcat-users.xml

context.xml logging.properties tomcat-users.xsd

[root@J2001 conf]# **vim server.xml**

[root@J2001 conf]# cd ..

[root@J2001 apache-tomcat-8.5.28]# cd bin

[root@J2001 bin]# .**/startup.sh 开启服务**

Using CATALINA\_BASE: /usr/local/sofeware/apache-tomcat-8.5.28

Using CATALINA\_HOME: /usr/local/sofeware/apache-tomcat-8.5.28

Using CATALINA\_TMPDIR: /usr/local/sofeware/apache-tomcat-8.5.28/temp

Using JRE\_HOME: /usr/local/sofeware/jdk1.8.0\_161

Using CLASSPATH: /usr/local/sofeware/apache-tomcat-8.5.28/bin/bootstrap.jar:/usr/local/sofeware/apache-tomcat-8.5.28/bin/tomcat-juli.jar

Tomcat started.

[root@J2001 bin]# **systemctl stop firewalld 关闭防火墙**

[root@J2001 bin]# **systemctl disable firewalld** 永久禁用防火墙

**[root@J2001 bin]# ./shutdown.sh 关闭服务**

# Linux软件包管理

## 软件包管理简介

1. **软件包的分类**

源码包 🡪脚本安装包

二进制包---> rpm包 系统默认包

**2、源码包的优点和缺点**

源码包的优点： 开源的 如果有足够能力 可以修改源代码适合自己和工程

可以自由选择功能

软件是编辑安装 所以更加适合系统

卸载更加方便

缺点：

安装过程步骤比较多比较繁琐。 尤其安装大的软件 容易出现错误

编辑过程比较常 一旦安装过程中出现错误 很难手动解决

**3、RMP包的优缺点**

优点： 包的管理比较简单 只需要几个简单的命令

实现包的 安装升级查询和卸载都比较简单

安装速度比安装源码包快很多

缺点：

经过编译 不能在看到源码

功能不能灵活选择

有很强的依赖性

## RMP包管理

**1）RPM包的命名规则**

httpd-2.4.6-67.el7.centos.x86\_64.rpm

第一个是文件的名称 软件包名

2.4.6 软件版本号

67 软件发布的次数

el7.centos 适合的linux平台

x86\_64

rpm 扩展名

**2）RMP包依赖性**

A->B-->C

www.rpmfind.net

1. **RPM安装**

Rpm –ivh 包全名

-i 安装

-v 显示详细信息

-h 显示进度

root@J2001 ~]# **cd /mnt/cdrom/**

[root@J2001 cdrom]# **cd Packages/**

[root@J2001 Packages]# **rpm -ivh httpd-**

httpd-2.4.6-67.el7.centos.x86\_64.rpm httpd-manual-2.4.6-67.el7.centos.noarch.rpm

httpd-devel-2.4.6-67.el7.centos.x86\_64.rpm httpd-tools-2.4.6-67.el7.centos.x86\_64.rpm

[root@J2001 Packages]# **rmp -ivh httpd-2.4.6-67.el7.centos.x86\_64.rpm**

[root@guigu\_student1 Packages]# **rpm -ivh httpd-2.4.6-67.el7.centos.x86\_64.rpm**

**[root@guigu\_student1 Packages]# rpm -ivh httpd-tools-2.4.6-67.el7.centos.x86\_64.rpm**

Libapr-1.so.0 去rmp查找依赖 安装即可

[root@guigu\_student1 Packages]# **rpm -ivh apr-util-1.5.2-6.el7.x86\_64.rpm**

[root@guigu\_student1 Packages]# rpm -ivh mailcap-2.1.41-2.el7.noarch.rpm

[root@guigu\_student1 Packages]# rpm -ivh httpd-2.4.6-67.el7.centos.x86\_64.rpm 安装httpd软件 需要依赖其他软件 需要先安装其他软件

1. **RPM包的升级**

[root@J2001 Packages]# **rpm -Uvh httpd-2.4.6-67.el7.centos.x86\_64.rpm**

1. **RPM包的卸载**

[root@guigu\_student1 Packages]# **rpm -e httpd**

1. **RPM包的查询**

[root@J2001 Packages]# **rpm -q httpd 查询指定的软件包是否安装**

httpd-2.4.6-67.el7.centos.x86\_64

[root@J2001 Packages]# **rpm –qa 查询所有的安装软件包**

[root@J2001 Packages]# **rpm -qip zip-3.0-11.el7.x86\_64.rpm 查询未安装包的详细信息**

[root@J2001 Packages]# **rpm -qR httpd 查询当前软件包的依赖性**

## RMP包管理 --yum在线管理

必须保证虚拟机能够正常联网

**1）常用的yum命令**

[root@localhost ~]# yum search httpd 查询软件是否存在

[root@localhost ~]# yum -y install httpd 安装软件httpd

[root@localhost ~]# yum -y install gcc 安装软件gcc

**2）升级**

[root@J2001 Packages]# yum -y update gcc 更新软件

**2）卸载**

[root@localhost ~]# yum -y remove httpd

尽量避免使用yum卸载 会把依赖一并卸载