**tree老师：每天五分钟教你学linux基础命令V2版**

尊敬的学员：

感谢您报名参加Linux运维学院-云计算运维架构师在线直播课程，

本课程每周一到周六晚上20:30—22:30上课，白天全天答疑，由腾讯运维专家联合学神教育鼎力打造，学完考试合格由腾讯颁发《运维架构师高级证书》，我们的培训采用在线直播培训方式，如果你因故不能参加某天的培训，也不用担心，我们上课的时候会录制当期的视频，请联系讲师在任务群下载即可。

腾讯课堂的严格管理，运维部的精益求精，都竭诚为您提供高质量的在线培训课程，珍惜您对我们的信任，结合BAT各大公司的经验，发放和教授最具企业生产力的课程资料。

另外学习中或者在使用腾讯课堂的体验上有任何的建议都请直接告诉我们，我们会第一时间处理您的建议！

注：本套书籍每隔月更新一次版本，请联系讲师tree领取最新版本

讲师tree老师QQ： 3059942361

微信公众号：xinsz08

## man

格式：man [-afpM] [命令]

作用：用来提供在线帮助，使用权限是所有用户

选项：

* -a：在所有的man 帮助手册中搜索
* -f：等价于whatis 指令，显示给定关键字的简短描述信息
* -P：指定内容时使用分页程序
* -M：指定man 手册搜索的路径

示例一：使用man命令来查看ifconfig的使用方法

[root@vhxct67test1 ~]# man ifconfig|more

IFCONFIG(8) Linux Programmer’s Manual IFCONFIG(8)

NAME

ifconfig - configure a network interface

SYNOPSIS

ifconfig [interface]

ifconfig interface [aftype] options | address ...

NOTE

This program is obsolete! For replacement check ip addr and ip link.

For statistics use ip -s link.

## ls

格式：ls [-alicfbd] [文件或目录]

作用：显示目录列表

选项：

* -a：显示所有档案及目录
* -A：显示除影藏文件“.”和“..”以外的所有文件列表
* -c：多列显示输出结果。这是默认选项
* -f：在每个输出项后追加文件的类型标识符，其中

“\*”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

* -b：将文件中的不可输出的字符以反斜线加字符编码的方式输出
* -d：仅显示目录名，而不显示目录下的内容列表
* -i：显示文件索引节点号（inode）。一个索引节点代表一个文件
* -k：以KB（千字节）为单位显示文件大小
* -l：以长格式显示目录下的内容列表
* -m：用“,”号区隔每个文件和目录的名称
* -n：以用户识别码和群组识别码替代其名称
* -s：显示文件和目录的大小，以区块为单位
* -t：用文件和目录的更改时间排序
* -L：如果遇到性质为符号链接的文件或目录，直接列出该链接所指向的原始文件或目录
* -r：递归处理，将指定目录下的所有文件及子目录一并处理
* --full-time：列出完整的日期与时间
* --color[=WHEN]：使用不同的颜色高亮显示不同类型的

示例一：ls –a 显示当前目录中的所有文件，包含隐藏文件

[root@vhxct67test1 ~]# ls -a

. .bash\_profile Desktop .gconfd install.log Public .xsession-errors

.. .bashrc Documents .gnome2 install.log.syslog .pulse

.abrt .cache Downloads .gtk-bookmarks .local .pulse-cookie

anaconda-ks.cfg .config .esd\_auth .gvfs Music .tcshrc

.bash\_history .cshrc expect .ICEauthority .nautilus Templates

.bash\_logout .dbus .gconf .imsettings.log Pictures Videos

示例二：ls –l 显示文件及其详细信息

[root@vhxct67test1 ~]# ls -l /mnt

total 4

lrwxrwxrwx 1 root root 11 Nov 24 08:13 b.txt -> /test/b.txt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

## tree

格式：tree [-aACdDfFgilnNpqstux] [目录]

作用：tree以树状图样式列出目录的内容

选项：

* -a：显示所有文件和目录
* -A：使用ASNI 绘图字符显示树状图而非以ASCII 字符组合
* -C：在文件和目录清单加上色彩，便于区分各种类型
* -d：先是目录名称而非内容
* -D：列出文件或目录的更改时间
* -f：在每个文件或目录之前，显示完整的相对路径名称
* -F：在执行文件，目录，Socket，符号连接，管道名称名称，各自加上"\*"，"/"，"@"，"|"号
* -g：列出文件或目录的所属群组名称，没有对应的名称时，则显示群组识别码
* -i：不以阶梯状列出文件和目录名称
* -l：如遇到性质为符号连接的目录，直接列出该连接所指向的原始目录
* -n：不在文件和目录清单加上色彩
* -N：直接列出文件和目录名称，包括控制字符
* -p：列出权限标示
* -q：用“？”号取代控制字符，列出文件和目录名称
* -s：列出文件和目录大小
* -t：用文件和目录的更改时间排序
* -u：列出文件或目录的拥有者名称，没有对应的名称时，则显示用户识别码
* -x：将范围局限在现行的文件系统中，若指定目录下的某些子目录，其存放于另一个文件系统上，则将该
* 目录予以排除在寻找范围外

示例一：

在/home下创建test目录并在目录下创建一些测试目录和文件，用tree查看：

[root@vhxrhel7test ~]# tree /test

/test

├── a

│   ├── b

│   │   └── c

│   └── No.one.txt

├── aa.txt

├── bb.txt

├── cc.txt

├── dd.txt

└── test.txt

## pwd

格式：pwd

作用：命令以绝对路径的方式显示用户当前工作目录

示例一：显示当前目录

[root@vhxct67test1 ~]# pwd

/root

表示在root目录下。

示例二：cd到/home目录下并显示当前目录

[root@vhxct67test1 ~]# cd /home

[root@vhxct67test1 home]# pwd

/home

## mkdir

格式：mkdir (选项) (参数)

作用：创建文件夹

选项：

❒ -m 设置文件夹权限

❒ -p 如果父目录不存在，则连同父目录一起创建

❒ -v 显示详细

❒ -Z 设置创建的目录SELinux为默认类型

❒ --context

❒ --help 显示帮助并退出

❒ --version 现实版本信息并退出

参数：

目录：指定要创建的目录列表，多个目录之间用空格隔开  
示例：

在/home下创建test目录，设置权限为644

[root@localhost ~]# ls /home

hfx

[root@localhost ~]# mkdir -m 644 /home/test

[root@localhost ~]# ll /home/

total 0

drwx------. 3 hfx hfx 74 Oct 21 18:04 hfx

drw-r--r-- 2 root root 6 Nov 7 15:45 test

在/mnt目录下创建连续目录/mnt/test1/test2/test3

[root@localhost ~]# ls /mnt/

[root@localhost ~]# mkdir /mnt/test1/test2/test3

mkdir: cannot create directory ‘/mnt/test1/test2/test3’: No such file or directory

[root@localhost ~]# mkdir -p /mnt/test1/test2/test3

[root@localhost ~]# tree /mnt/

/mnt/

└── test1

└── test2

└── test3

3 directories, 0 files

显示mkdir的版本信息

[root@localhost ~]# mkdir --version

mkdir (GNU coreutils) 8.22

Copyright (C) 2013 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.

## rmdir

格式：rmdir (选项) (参数)

作用：用来删除空目录

选项：

❒ -p或—parents 删除指定目录后，若该目录的上层目录已变成空目录，则将其一并删除

❒ --ignore-fail-on-non-empty 此选项使rmdir命令忽略由于删除非空目录时导致的错误信息

❒ -v或-verboes 显示命令的详细执行过程

❒ --help 显示命令的帮助信息

❒ --version 显示命令的版本信息

参数：要删除的空目录列表。当删除多个空目录时，目录名之间使用空格隔开

示例：

删除空目录/mnt/test1/test2/test3

[root@localhost ~]# rmdir -p /mnt/test1/test2/test3/

rmdir: failed to remove directory ‘/’: Device or resource busy

[root@localhost ~]# ls /mnt

ls: cannot access /mnt: No such file or directory

◆可以看到上面把我们的/mnt目录也一起删掉了，这就是-p选项的威力。

删除/mnt/test/非空目录

[root@localhost ~]# mkdir /mnt/test/

[root@localhost ~]# echo "test" > /mnt/test/a.txt

[root@localhost ~]# ll /mnt/test/

total 4

-rw-r--r-- 1 root root 5 Nov 7 16:09 a.txt

[root@localhost ~]# rmdir /mnt/test/

rmdir: failed to remove ‘/mnt/test/’: Directory not empty

[root@localhost ~]# rmdir --ignore-fail-on-non-empty /mnt/test/

[root@localhost ~]# ll /mnt/test/

total 4

-rw-r--r-- 1 root root 5 Nov 7 16:09 a.txt

◆虽然不能删除但是也不会报错了

使用-v选项删除/mnt下./test1/test2/test3目录

[root@localhost ~]# mkdir -p /mnt/test1/test2/test3

[root@localhost ~]# rmdir -p -v /mnt/test1/test2/test3/

rmdir: removing directory, ‘/mnt/test1/test2/test3/’

rmdir: removing directory, ‘/mnt/test1/test2’

rmdir: removing directory, ‘/mnt/test1’

rmdir: removing directory, ‘/mnt’

rmdir: failed to remove directory ‘/mnt’: Directory not empty

[root@localhost ~]# ll /mnt/

total 0

drwxr-xr-x 2 root root 18 Nov 7 16:09 test

## cd

格式：cd (选项) （参数）

作用：切换工作目录

选项：

❒ -p 如果要切换到的目标目录是一个符号连接，直接切换到符号连接指向的目标目录

❒ -L 如果要切换的目标目录是一个符号的连接，直接切换到字符连接名代表的目录，而非符号连接所指向的目标目录。

❒ - 当仅实用"-"一个选项时，当前工作目录将被切换到环境变量"OLDPWD"所表示的目录。

参数：

需要切换到的工作目录，可以是绝对路径和可以是相对路径

❒ ~ 当前用户的家目录

❒ .. 当前目录的上级目录

❒ . 当前目录

示例：

切换到/etc/sys config/network-scripts/目录下

[root@localhost ~]# pwd

/root

[root@localhost ~]# cd /etc/sysconfig/network-scripts/

[root@localhost network-scripts]# pwd

/etc/sysconfig/network-scripts

返回上级目录

[root@localhost network-scripts]# pwd

/etc/sysconfig/network-scripts

[root@localhost network-scripts]# cd ..

[root@localhost sysconfig]# pwd

/etc/sysconfig

返回上两级目录

[root@localhost sysconfig]# pwd

/etc/sysconfig

[root@localhost sysconfig]# cd ../..

[root@localhost /]# pwd

/

进入当前用户主目录

[root@localhost /]# pwd

/

[root@localhost /]# cd ~

[root@localhost ~]# pwd

/root

或者用另一种写法直接cd

[root@localhost /]# pwd

/

[root@localhost /]# cd

[root@localhost ~]# pwd

/root

返回进入此目录之前所在的目录  
[root@localhost /]# pwd

/

[root@localhost /]# cd ~

[root@localhost ~]# pwd

/root

[root@localhost ~]#

[root@localhost ~]# cd -

/

## touch

格式：touch (选项) (参数)

作用：

一、是用来创建新的空文件

二、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；

选项：

❒ -a： 或--time=atime或--time=access或--time=use 只更改存取时间

❒ -c： 或--no-create 不建立任何文件

❒ -d： <时间日期> 使用指定的日期时间，而非现在的时间

❒ -f： 此参数将忽略不予处理，仅负责解决BSD版本touch指令的兼容性问题

❒ -m：或--time=mtime或--time=modify 只更该变动时间

❒ -r： <参考文件或目录> 把指定文件或目录的日期时间，统统设成和参考文件或目录的日期时间相同

❒ -t： <日期时间> 使用指定的日期时间，而非现在的时间； --help：在线帮助； --version：显示版本信息

参数：

一、指定要创建的空文件名称

二、指定要设置时间属性的文件名称

示例：  
在目录/mnt下创建空文件 a.txt b.txt

[root@localhost mnt]# touch a.txt b.txt

[root@localhost mnt]# ll

total 0

-rw-r--r-- 1 root root 0 Nov 7 16:59 a.txt

-rw-r--r-- 1 root root 0 Nov 7 16:59 b.txt

复制/etc/passwd文件到/mnt目录下,使用 touch –a 修改文件passwd的存取时间

[root@localhost mnt]# cp -a /etc/passwd .

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2016-11-07 18:36:43.441999782 +0800

Modify: 2016-10-21 18:15:07.082976185 +0800

Change: 2016-11-07 17:03:09.395040011 +0800

Birth: -

[root@localhost mnt]# touch -a passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2016-11-07 17:04:09.175035587 +0800

Modify: 2016-10-21 18:15:07.082976185 +0800

Change: 2016-11-07 17:04:09.175035587 +0800

Birth: -

使用touch –d 修改文件passwd  
[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2016-11-07 17:04:09.175035587 +0800

Modify: 2016-10-21 18:15:07.082976185 +0800

Change: 2016-11-07 17:04:09.175035587 +0800

Birth: -

[root@localhost mnt]# touch -d 2018-10-12 passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2018-10-12 00:00:00.000000000 +0800

Modify: 2018-10-12 00:00:00.000000000 +0800

Change: 2016-11-07 17:04:35.739033621 +0800

Birth: -  
使用touch –m 修改文件passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2018-10-12 00:00:00.000000000 +0800

Modify: 2018-10-12 00:00:00.000000000 +0800

Change: 2016-11-07 17:04:35.739033621 +0800

Birth: -

[root@localhost mnt]# touch -m passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2018-10-12 00:00:00.000000000 +0800

Modify: 2016-11-07 17:05:07.385031279 +0800

Change: 2016-11-07 17:05:07.385031279 +0800

Birth: -

使用touch –r修改文件passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2018-10-12 00:00:00.000000000 +0800

Modify: 2016-11-07 17:05:07.385031279 +0800

Change: 2016-11-07 17:05:07.385031279 +0800

Birth: -

[root@localhost mnt]# touch -r /etc/passwd passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2016-11-07 18:36:43.441999782 +0800

Modify: 2016-10-21 18:15:07.082976185 +0800

Change: 2016-11-07 17:06:45.478024019 +0800

Birth: -

[root@localhost mnt]# stat /etc/passwd

File: ‘/etc/passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 19973395 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2016-11-07 18:36:43.441999782 +0800

Modify: 2016-10-21 18:15:07.082976185 +0800

Change: 2016-10-21 18:15:07.084976185 +0800

Birth: -

使用touch –t修改文件passwd

[root@localhost mnt]# touch -t 201912221344.42 passwd

[root@localhost mnt]# stat passwd

File: ‘passwd’

Size: 2264 Blocks: 8 IO Block: 4096 regular file

Device: fd00h/64768d Inode: 4558657 Links: 1

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

Access: 2019-12-22 13:44:42.000000000 +0800

Modify: 2019-12-22 13:44:42.000000000 +0800

Change: 2016-11-07 17:18:55.136970019 +0800

Birth: -

◆Time 使用指定时间而不是当前时间。Time 变量以十进制形[[CC]YY]MMDDhhmm[.SS] 指定

## cp

格式：cp [-abdfilpPrRsuv] [源文件列表] [目的文件列表]

作用：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

参数选项：

❒ -a：此参数的效果和同时指定"-dpR"参数相同

* -d：当复制符号连接时，把目标文件或目录也建立为符号连接，并指向与源文件或目录连接的原始文件或目录
* -f：强行复制文件或目录，不论目标文件或目录是否已存在
* -i：覆盖既有文件之前先询问用户
* -l：对源文件建立硬连接，而非复制文件
* -p：保留源文件或目录的属性
* -R/r：递归处理，将指定目录下的所有文件与子目录一并处理
* -s：对源文件建立符号连接，而非复制文件
* -u：使用这项参数后只会在源文件的更改时间较目标文件更新时或是名称相互对应的目标文件并不存在时才复制文件
* -S：在备份文件时，用指定的后缀“SUFFIX”代替文件的默认后缀
* -b：覆盖已存在的文件目标前将目标文件备份
* -v：详细显示命令执行的操作

示例一：复制文件

[root@vhxct67test ~]# touch test.txt //创建一个测试文件

[root@vhxct67test ~]# cp test.txt /mnt //把test.txt文件复制到/mnt目录下

[root@vhxct67test ~]# ll /mnt //在/mnt目录下查看复制的文件

total 8

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

-rw------- 1 root root 2122 Nov 22 03:17 passwd

-rw------- 1 root root 0 Nov 24 03:14 test.txt

示例二：递归方式复制目录下的文件

[root@vhxct67test ~]# mkdir -p /cptest/a/c //创建测试目录

[root@vhxct67test ~]# touch /cptest/a/b.txt //在a目录下创建测试文件

[root@vhxct67test ~]# touch /cptest/a/c/cc.txt //在b目录下创建测试文件

[root@vhxct67test ~]# cp -rf /cptest/ /mnt //使用rf参数递归的方式把cptest目录的所以文件和目录都复制到mnt下

[root@vhxct67test ~]# ll /mnt /查看mnt目录下的cptest目录是否复制成功

total 12

drwx------ 3 root root 4096 Nov 24 03:22 cptest

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

-rw------- 1 root root 2122 Nov 22 03:17 passwd

-rw------- 1 root root 0 Nov 24 03:14 test.txt

[root@vhxct67test ~]# cd /mnt //进入mnt目录下

[root@vhxct67test mnt]# tree cptest/ //使用tree命令查看cptest目录下的文件和目录是否全部复制完成

cptest/

└── a

├── b.txt

└── c

└── cc.txt

## mv

格式：mv [-bfiuv] [源文件列表] [目标文件]

作用：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

选项：

* --backup=<备份模式>：若需覆盖文件，则覆盖前先行备份
* -b：当文件存在时，覆盖前，为其创建一个备份
* -f：若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录
* -i：交互式操作，覆盖前先行询问用户，如果源文件与目标文件或目标目录中的文件同名，则询问用户是否覆盖目标文件。用户输入”y”，表示将覆盖目标文件；输入”n”，表示取消对源文件的移动.这样可以避免误将文件覆盖
* --strip-trailing-slashes：删除源文件中的斜杠“/”
* -S<后缀>：为备份文件指定后缀，而不使用默认的后缀
* --target-directory=<目录>：指定源文件要移动到目标目录
* -u：当源文件比目标文件新或者目标文件不存在时，才执行移动操作

示例一：移动文件

[root@vhxct67test1 ~]# touch test.txt

[root@vhxct67test1 ~]# mv test.txt /mnt

[root@vhxct67test1 ~]# ll /mnt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

-rw------- 1 root root 0 Nov 24 06:01 test.txt

示例二：文件改名

[root@vhxct67test1 ~]# touch test1.txt

[root@vhxct67test1 ~]# mv test1.txt /mnt/test11.txt

[root@vhxct67test1 ~]# ll /mnt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

-rw------- 1 root root 0 Nov 24 06:03 test11.txt

-rw------- 1 root root 0 Nov 24 06:01 test.txt

示例三：移动多个文件到/mnt目录下

[root@vhxct67test1 ~]# touch test.txt test1.txt test2.txt

[root@vhxct67test1 ~]# mv test.txt test1.txt test2.txt /mnt/

[root@vhxct67test1 ~]# ll /mnt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

-rw------- 1 root root 0 Nov 24 06:06 test1.txt

-rw------- 1 root root 0 Nov 24 06:06 test2.txt

-rw------- 1 root root 0 Nov 24 06:06 test.txt

示例四：将文件file1改名为file2，如果file2已经存在，则询问是否覆盖

[root@vhxct67test1 ~]# touch test.txt

[root@vhxct67test1 ~]# mv -i test.txt /mnt/

mv: overwrite `/mnt/test.txt'? y

示例五：目录的移动

[root@vhxct67test1 ~]# mkdir -p /test/a

[root@vhxct67test1 ~]# touch /test/test1.txt

[root@vhxct67test1 ~]# touch /test/a/aa.txt

[root@vhxct67test1 ~]# mv /test /mnt

[root@vhxct67test1 ~]# ll /mnt/test/

drwx------ 2 root root 4096 Nov 24 06:12 a

-rw------- 1 root root 0 Nov 24 06:12 test1.txt

## rm

格式：rm [-dfirv] [文件]

作用：删除一个目录中的一个或多个文件或目录

选项：

* -d：直接把欲删除的目录的硬连接数据删除成0，删除该目录
* -f：强制删除文件或目录
* -i：删除已有文件或目录之前先询问用户
* -r 或-R：递归处理，将指定目录下的所有文件与子目录一并处理
* --preserve-root：不对根目录进行递归操作
* -v：显示指令的详细执行过程

示例一：删除指定文件

[root@vhxct67test1 ~]# cd /mnt

[root@vhxct67test1 mnt]# ls

hgfs test test1.txt test2.txt test.txt

[root@vhxct67test1 mnt]# rm test.txt

[root@vhxct67test1 mnt]# ls

hgfs test test1.txt test2.txt

示例二：**删除文件显示运行时详细信息**

[root@vhxct67test1 mnt]# rm -v test1.txt

removed `test1.txt'

[root@vhxct67test1 mnt]# ls

hgfs test test2.txt

示例三：**交互模式下删除文件**

[root@vhxct67test1 mnt]# ls

hgfs test test2.txt

[root@vhxct67test1 mnt]# rm -iv test2.txt

rm: remove regular empty file `test2.txt'? y

removed `test2.txt'

示例四：**递归删除某个目录下的文件和子目录**

[root@vhxct67test1 mnt]# ls

hgfs test

[root@vhxct67test1 mnt]# tree test

test

├── a

│   └── aa.txt

└── test1.txt

1 directory, 2 files

[root@vhxct67test1 mnt]# rm -rf test/

[root@vhxct67test1 mnt]# ls

hgfs

## ln

格式：ln [-bdfinsv] [源文件列表] [目标文件]

作用：为文件创建连接，连接类型分为硬链接和软连接（符号链接）两种，默认是硬链接，如果要创建软连接必须使用“-s”选项

选项：

* -b 或--backup：删除，覆盖目标文件之前的备份
* -d 或-F 或——directory：建立目录的硬连接
* -f 或——force：强行建立文件或目录的连接，不论文件或目录是否存在
* -i 或——interactive：覆盖既有文件之前先询问用户
* -n 或--no-dereference：把符号连接的目的目录视为一般文件
* -s 或——symbolic：对源文件建立符号连接，而非硬连接
* -S<字尾备份字符串>或--suffix=<字尾备份字符串>：用"-b"参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，预设的备份字符串是符号“~”，用户可通过“-S”参数来改变它
* -v 或——verbose：显示指令执行过程
* -V<备份方式>或--version-control=<备份方式>：用“-b”参数备份目标文件后，备份文件的字尾会被加上一个备份字符串，这个字符串不仅可用“-S”参数变更，当使用“-V”参数<备份方式>指定不同备份方式时，也会产生不同字尾的备份字符串

示例一：硬链接

将/test/a.txt的文件链接到/mnt下

[root@vhxct67test1 ~]# mkdir /test

[root@vhxct67test1 ~]# touch /test/a.txt

[root@vhxct67test1 ~]# ll /mnt

total 4

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

[root@vhxct67test1 ~]# ln /test/a.txt /mnt

[root@vhxct67test1 ~]# ll /mnt

total 4

-rw------- 2 root root 0 Nov 24 08:06 a.txt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

示例一说明：在执行ln命令之前，目录mnt中是不存在a.txt文件的，在执行ln命令后，目录mnt下才有a.txt这个文件，（注意，二者在物理上是同一文件）。

示例二：软链接

[root@vhxct67test1 ~]# touch /test/b.txt

[root@vhxct67test1 ~]# ln -s /test/b.txt /mnt

[root@vhxct67test1 ~]# ll /mnt

total 4

-rw------- 2 root root 0 Nov 24 08:06 a.txt

lrwxrwxrwx 1 root root 11 Nov 24 08:13 b.txt -> /test/b.txt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

## find

格式：find [选项] [参数]

作用：在指定目录下查找文件

选项：

* -daystart：从本日开始计算时间
* -mtime -n ,+n按文件更改时间来查找文件，-n指n天以内，+n指n天以前
* -atime     -n ,+n按文件访问时间来查, -n指n天以内，+n指n天以前
* -ctime     -n ,+n按文件创建时间来查找文件，-n指n天以内，+n指n天以前
* -depth：从指定目录下最深层的子目录开始查找
* -expty：寻找文件大小为0 Byte 的文件，或目录下没有任何子目录或文件的空目录
* -exec<执行指令>：假设find 指令的回传值为True，就执行该指令
* -fstype<文件系统类型>：只寻找该文件系统类型下的文件或目录
* -gid<群组识别码>：查找符合指定之群组识别码的文件或目录
* -group<群组名称>：查找符合指定之群组名称的文件或目录
* -inum：查找符合指定的inode 编号的文件或目录
* -links<连接数目>：查找符合指定的硬连接数目的文件或目录
* -iname<范本样式>：指定字符串作为寻找符号连接的范本样式
* -ls：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
* -maxdepth<目录层级>：设置最大目录层级
* -mindepth<目录层级>：设置最小目录层级
* name<范本样式>：指定字符串作为寻找文件或目录的范本样式
* -newer<参考文件或目录>：查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录
* -nouser：找出不属于本地主机用户识别码的文件或目录
* -path<范本样式>：指定字符串作为寻找目录的范本样式
* -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
* -size<文件大小>：查找符合指定的文件大小的文件
* -true：将find 指令的回传值皆设为True
* -typ<文件类型>：只寻找符合指定的文件类型的文件
* -uid<用户识别码>：查找符合指定的用户识别码的文件或目录
* -used<日数>：查找文件或目录被更改之后在指定时间曾被存取过的文件或目录，单位以日计算
* -user<拥有者名称>：查找符和指定的拥有者名称的文件或目录
* -xdev：将范围局限在先行的文件系统中

示例一：查找指定时间内修改过的文件

[root@vhxct67test1 ~]# find -atime -2

./Music

./Templates

./.bashrc

./Downloads

./.gvfs

./.bash\_profile

示例二：**根据关键字查找**

[root@vhxct67test1 ~]# find -name "\*.log"

./.imsettings.log

./.local/share/gvfs-metadata/home-36bb4135.log

./.local/share/gvfs-metadata/label-VMware\x20Tools-4078a08d.log

./install.log

示例三：按照目录或文件的权限来查找文件

[root@vhxct67test1 ~]# find /opt/ -perm 644

/opt/glibc-2.14/etc/ld.so.cache

/opt/glibc-2.14/etc/rpc

/opt/glibc-2.14/lib/librpcsvc.a

/opt/glibc-2.14/lib/crtn.o

/opt/glibc-2.14/lib/librt.a

/opt/glibc-2.14/lib/gconv/gconv-modules

/opt/glibc-2.14/lib/libc\_nonshared.a

/opt/glibc-2.14/lib/libutil.a

示例四：**按类型查找**

[root@vhxct67test1 ~]# find -type f -name "\*.log"

./.imsettings.log

./.local/share/gvfs-metadata/home-36bb4135.log

./.local/share/gvfs-metadata/label-VMware\x20Tools-4078a08d.log

./install.log

示例五：按大小查找文件

[root@vhxct67test1 ~]# find -size +10M -print

./Desktop/VMwareTools-9.4.15-2827462.tar.gz

示例六：在/mnt目录中查找文件并使用-exec选项删除它

[root@vhxct67test1 ~]# cd /mnt

[root@vhxct67test1 mnt]# ls

a.txt b.txt hgfs

[root@vhxct67test1 mnt]# cd

[root@vhxct67test1 ~]# find /mnt -name a.txt -exec rm -rf {} \; //注意{}和\;之间有空格

[root@vhxct67test1 ~]# ll /mnt

total 4

lrwxrwxrwx 1 root root 11 Nov 24 08:13 b.txt -> /test/b.txt

drwx------. 2 root root 4096 Aug 2 04:34 hgfs

## rename

语法：rename from to file

rename -V

作用：重命名文件

将a.txt，b.txt, c.txt批量修改为a.log,b.log,c.log

[root@xuegod66 rename]# touch a.txt b.txt c.txt

[root@xuegod66 rename]# ll

total 0

-rw-r--r--. 1 root root 0 Nov 16 21:39 a.txt

-rw-r--r--. 1 root root 0 Nov 16 21:34 b.txt

-rw-r--r--. 1 root root 0 Nov 16 21:34 c.txt

[root@xuegod66 rename]# rename .txt .log \*.txt

[root@xuegod66 rename]# ll -a

total 8

drwxr-xr-x. 2 root root 4096 Nov 16 21:39 .

drwxrwxrwt. 4 root root 4096 Nov 16 21:35 ..

-rw-r--r--. 1 root root 0 Nov 16 21:39 a.log

-rw-r--r--. 1 root root 0 Nov 16 21:34 b.log

-rw-r--r--. 1 root root 0 Nov 16 21:34 c.log

示例2：查看版本

[root@xuegod66 rename]# rename -V

rename (util-linux-ng 2.17.2)

## cat

格式：cat (选项) (参数)

作用：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。

选项：

❒ -n或-number：有1开始对所有输出的行数编号；

❒ -b或--number-nonblank：和-n相似，只不过对于空白行不编号；

❒ -s或--squeeze-blank：当遇到有连续两行以上的空白行，就代换为一行的空白行；

❒ -A：显示不可打印字符，行尾显示“$”；

❒ -e：等价于"-vE"选项；

❒ -t：等价于"-vT"选项；

参数：指定要连接的文件列表

显示文件内容行号

[root@xuegod66 cat]# cat -n abc.txt

1 abc

2 def

3 aaaaaa

4 dddddd

5 bbbbbb

6 eee

查看两个文件内容

[root@redhat01 cat]# ll

total 8

-rw-r--r--. 1 root root 26 Oct 30 20:10 a.txt

-rw-r--r--. 1 root root 17 Oct 30 20:11 b.txt

[root@redhat01 cat]#

[root@redhat01 cat]# cat a.txt b.txt

kdjflaj

kfjdlafj

kkkkkkk

999999

00000

122

显示不可打印的字符

[root@redhat01 cat]# cat -A a.txt

kdjflaj$

kfjdlafj$

kkkkkkk$

$

## more

格式：more (选项)（参数）

作用：分页显示文本内容

选项：

❒ -<数字>：指定每屏显示的行数；

❒ -d：显示“[press space to continue,'q' to quit.]”和“[Press 'h' for instructions]”；

❒ -c：不进行滚屏操作。每次刷新这个屏幕；

❒ -s：将多个空行压缩成一行显示；

❒ -u：禁止下划线；

❒ +<数字>：从指定数字的行开始显示。

参数：

文件：指定分页显示内容的文件。

[root@redhat01 cat]# more a.txt

kdjflaj

kfjdlafj

kkkkkkk

more支持vi的很多选项，输入“？”就调出说明

[root@redhat01 tmp]# more event.sh

--More--(49%)

Most commands optionally preceded by integer argument k. Defaults in brackets.

Star (\*) indicates argument becomes new default.

-------------------------------------------------------------------------------

<space> Display next k lines of text [current screen size]

z Display next k lines of text [current screen size]\*

<return> Display next k lines of text [1]\*

d or ctrl-D Scroll k lines [current scroll size, initially 11]\*

q or Q or <interrupt> Exit from more

s Skip forward k lines of text [1]

f Skip forward k screenfuls of text [1]

b or ctrl-B Skip backwards k screenfuls of text [1]

' Go to place where previous search started

= Display current line number

/<regular expression> Search for kth occurrence of regular expression [1]

n Search for kth occurrence of last r.e [1]

!<cmd> or :!<cmd> Execute <cmd> in a subshell

v Start up /usr/bin/vi at current line

ctrl-L Redraw screen

:n Go to kth next file [1]

:p Go to kth previous file [1]

:f Display current file name and line number

. Repeat previous command

-------------------------------------------------------------------------------

## less

格式：less (选项) (参数)

作用：向前或向后浏览文件

选项：

❒ -e：文件内容显示完毕后，自动退出；

❒ -f：强制显示文件；

❒ -g：不加亮显示搜索到的所有关键词，仅显示当前显示的关键字，以提高显示速度；

❒ -l：搜索时忽略大小写的差异；

❒ -N：每一行行首显示行号；

❒ -s：将连续多个空行压缩成一行显示；

❒ -S：在单行显示较长的内容，而不换行显示；

❒ -x<数字>：将TAB字符显示为指定个数的空格字符。

参数：

文件：指定要分屏显示内容的文件。

[root@redhat01 tmp]# less -N event.sh

1 if [ -e /var/opt/resmon/log/event.log ]; then

2 MONTH=`date | awk '{print $2}'`

3 if [ $MONTH = Jan ] ; then

4 sed -n '/Dec 25/,/Dec 31/p' /var/opt/resmon/log/event.log

5 sed -n '/Jan 1/,/Jan 26/p' /var/opt/resmon/log/event.log

cat more less的区别：

cat 连续显示、查看文件内容

more 分页查看文件内容

less 分页可控制查看文件内容

cat一次性把文件内容全部显示出来，不能进行交互式操作，适合察看内容短小、不超过一屏的文件；

more比cat强大一点，支持分页显示，支持ctrl+B ctrl+F .....上下滚屏，但是不支持像shift+g（跳到文件尾）这种操作；

less比more更强大一点，支持各种命令，随便翻页、跳转、查找.....

## head

格式：mkdir (选项) (参数)

作用：文件内容查看用于显示文件的开头的内容

选项：

❒ -n 指定显示头部内容的行数

❒ -c 指定显示头部内容的字符数

❒ -v 总是显示文件名的头信息

❒ -q 不显示文件名的头信息

参数： 文件列表：指定显示头部内容的文件列表

示例：

在查看/etc/passwd用户文件的前十行内容

[root@localhost ~]# head –n 10 /etc/passwd

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

halt:x:7:0:halt:/sbin:/sbin/halt

mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin

## tail

格式：tail (选项) (参数)

作用：从指定点开始将文件写到标准输出

选项：

❒ -retry： 即是在tail命令启动时，文件不可访问或者文件稍后变得不可访问，都始终尝试打开文件。使用此选项时需要与选项“——follow=name”连用；

❒ -c或--bytes=：输出文件尾部的N（N为整数）个字节内容；

❒ -f或；--follow：显示文件最新追加的内容。“name”表示以文件名的方式监视文件的变化。“-f”与“-fdescriptor”等效；

❒ -F： 与选项“-follow=name”和“--retry"连用时功能相同；

❒ -n或--line=： 输出文件的尾部N（N位数字）行内容。   
❒ --pid=<进程号>：与“-f”选项连用，当指定的进程号的进程终止后，自动退出tail命令；

❒ -q或--quiet或--silent：当有多个文件参数时，不输出各个文件名；

❒ -s<秒数>或--sleep-interal=<秒数>：与“-f”选项连用，指定监视文件变化时间隔的秒数；

❒ -v或--verbose：当有多个文件参数时，总是输出各个文件名；

❒ --help： 显示指令的帮助信息；

❒ --version： 显示指令的版本信息。

参数：

文件列表：指定要显示尾部内容的文件列表。   
示例：

在/home下创建test目录，设置权限为644

[root@localhost ~]# tail -n 5 /etc/passwd

sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

tcpdump:x:72:72::/:/sbin/nologin

rget1:x:500:500::/home/rget1:/bin/bash

rput1:x:501:501::/home/rput1:/bin/bash

apache:x:48:48:Apache:/var/www:/sbin/nologin [root@localhost ~]# mkdir -m 644 /home/test

[root@localhost log]# tail -f fail2ban.log 可以实时监控日志

2016-11-17 17:46:32,773 fail2ban.database [1209]: INFO Connected to fail2ban persistent database '/var/lib/fail2ban/fail2ban.sqlite3'

2016-11-17 17:57:28,130 fail2ban.server [1209]: INFO Stopping all jails

2016-11-17 17:57:28,130 fail2ban.server [1209]: INFO Exiting Fail2ban

2016-11-18 06:45:54,509 fail2ban.server [1215]: INFO Changed logging target to /var/log/fail2ban.log for Fail2ban v0.9.4

2016-11-18 06:45:54,538 fail2ban.database [1215]: INFO Connected to fail2ban persistent database '/var/lib/fail2ban/fail2ban.sqlite3'

## cut

格式：cut (选项) (参数)

作用：用来显示行中的指定部分，删除文件中指定字段

选项：

❒ -b：仅显示行中指定直接范围的内容；

❒ -c：仅显示行中指定范围的字符；

❒ -d：指定字段的分隔符，默认的字段分隔符为“TAB”；

❒ -f：显示指定字段的内容；

❒ -n：与“-b”选项连用，不分割多字节字符；

❒ --complement：补足被选择的字节、字符或字段；

❒ --out-delimiter=<字段分隔符>：指定输出内容是的字段分割符；

❒ --help：显示指令的帮助信息；

❒ --version：显示指令的版本信息。

参数：

文件：指定要进行内容过滤的文件。

示例：

例如有一个学生报表信息，包含No、Name、Mark、Percent：

[root@localhost text]# cat test.txt

No Name Mark Percent

01 tom 80 91

02 jack3 81 87

03 alex 68 98

使用 -f 选项提取指定字段：

[root@localhost text]# cut -f 1 test.txt

No

01

02

03

[root@localhost text]# cut -f2,3 test.txt

Name Mark

tom 80   
jack 81

alex 68

使用 -d 选项指定字段分隔符：   
  
[root@localhost text]# cut -f2 -d";" test2.txt

Name

tom

jack

alex

## paste

格式：paste (选项) (参数)

作用：用于将多个文件按照列队列进行合并

选项：

❒ -d<间隔字符>或--delimiters=<间隔字符>：用指定的间隔字符取代跳格字符；

❒ -s或——serial串列进行而非平行处理。

参数：

文件列表：指定需要合并的文件列表

示例：

[root@localhost ~]# cat per1

1

2

3

4

[root@localhost ~]# cat per2

a

b

c

d

[root@localhost ~]# paste per1 per2

1 a

2 b

3 c

4 d

[root@localhost ~]# paste per1 per2

1 2 3 4

a b c d

[root@localhost ~]# ls | paste -d: - - - #用：分隔，且每行显示三个文件

anaconda-ks.cfg:a.txt:a.txxt

fail2ban-0.9.4:fail2ban-0.9.4.tar.gz:inotify.sh

inotify-tools-3.13.tar.gz:install.log:install.log.syslog

per1:per2:xuegod63

## sort

格式：sort (选项) (参数)

作用：sort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。

选项：

❒ -b： 忽略每行前面开始出的空格字符；

❒ -c： 检查文件是否已经按照顺序排序；

❒ -d: 排序时，处理英文字母、数字及空格字符外，忽略其他的字符；

❒ -f： 排序时，将小写字母视为大写字母；

❒ -i： 排序时，除了040至176之间的ASCII字符外，忽略其他的字符；

❒ -m：将几个排序号的文件进行合并；

❒ -M：将前面3个字母依照月份的缩写进行排序；

❒ -n： 依照数值的大小排序；

❒ -o<输出文件>：将排序后的结果存入制定的文件；

❒ -r：以相反的顺序来排序；

❒ -t<分隔字符>：指定排序时所用的栏位分隔字符；

❒ +<起始栏位>-<结束栏位>：以指定的栏位来排序，范围由起始栏位到结束栏位的前一栏位。

参数：

文件：指定待排序的文件列表  
示例：

sort将文件/文本的每一行作为一个单位，相互比较，比较原则是从首字符向后，依次按ASCII码值进行比较，最后将他们按升序输出。   
[root@localhost ~]# cat sort.txt

aaa:10:1.1

ccc:30:3.3

ddd:40:4.4

bbb:20:2.2

eee:50:5.5

eee:50:5.5

[root@localhost ~]# sort sort.txt

aaa:10:1.1

bbb:20:2.2

ccc:30:3.3

ddd:40:4.4

eee:50:5.5

eee:50:5.5

忽略相同行使用-u选项或者uniq：

[root@localhost ~]#sort -u sort.txt

aaa:10:1.1

bbb:20:2.2

ccc:30:3.3

ddd:40:4.4

eee:50:5.5

sort的-n、-r、-k、-t选项的使用：  
[root@localhost ~]#cat sort.txt

AAA:BB:CC

aaa:10:1.1

ccc:30:3.3

ddd:40:4.4

bbb:20:2.2

eee:50:5.5

eee:50:5.5

将BB列按照数字从小到大顺序排列：  
[root@localhost ~]# sort -nk 2 -t: sort.txt

AAA:BB:CC

aaa:10:1.1

bbb:20:2.2

ccc:30:3.3

ddd:40:4.4

eee:50:5.5

eee:50:5.5

将CC列数字从大到小顺序排列：  
[root@localhost ~]# sort -nk 2 -t: sort.txt

eee:50:5.5

eee:50:5.5

ddd:40:4.4

ccc:30:3.3

bbb:20:2.2

aaa:10:1.1

AAA:BB:CC

## uniq

格式：uniq (选项) (参数)

作用：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  
选项：

❒ -m 设置文件夹权限

❒ -c或——count：在每列旁边显示该行重复出现的次数；

❒ -d或--repeated：仅显示重复出现的行列；

❒ -f<栏位>或--skip-fields=<栏位>：忽略比较指定的栏位；

❒ -s<字符位置>或--skip-chars=<字符位置>：忽略比较指定的字符；

❒ -u或——unique：仅显示出一次的行列；

❒ -w<字符位置>或--check-chars=<字符位置>：指定要比较的字符。

参数：

输入文件：指定要去除的重复行文件。如果不指定此项，则从标准读取数据；

输出文件：指定要去除重复行后的内容要写入的输出文件。如果不指定此选项，

则将内容显示到标准输出设备（显示终端）。

示例：

删除重复行：

[root@localhost ~]#cat file.txt

AAA

aaa

AAA

aaa

BBB

bbb

BBB

bbb

[root@localhost ~]# sort file.txt | uniq

aaa

AAA

bbb

BBB

统计各行在文件中出现的次数：

[root@localhost ~]#sort file.txt | uniq –c

2 aaa

2 AAA

2 bbb

2 BBB

在文件中找出重复的行：   
[root@localhost ~]#sort file.txt | uniq -d

aaa

AAA

bbb

BBB

## wc

作用：

wc命令的功能为统计指定文件中的字节数、字数、行数, 并将统计结果显示输出。对每个文件输出行、单词、和字节统计数，如果指定了多于一个文件则还有一个行数的总计。没有指定文件或指定的文件是名为“-”，则读取标准输入。

wc命令类似与word中的字数统计功能，可以统计字数、行数、页数、段落等。

格式：

wc [选项列表]... [文件名列表]...

wc [-clw][--help][--version][文件...]

参数：

❒ -c, --bytes, --chars 输出字节统计数。

❒ -l, --lines 输出换行符统计数。

❒ -L, --max-line-length 输出最长的行的长度。

❒ -w, --words 输出单词统计数。

❒ --help 显示帮助并退出

❒ --version 输出版本信息并退出

示例：

统计passwd文件信息

[root@sail ~]# wc /etc/passwd

43 87 2267 /etc/passwd

第一列显示行数，第二列显示单词数，第三列显示字节数、第四列为文件名。

表示文件有43行、表示文件中有87个单词、2267表示文件字节数即文件的大小.

[root@sail ~]# ls -l /etc/passwd

-rw-r--r--. 1 root root 2267 Oct 23 01:26 /etc/passwd

[root@sail ~]# ls -lh /etc/passwd

-rw-r--r--. 1 root root 2.3K Oct 23 01:26 /etc/passwd

参数测试

[root@sail test]# wc -l /etc/passwd 统计行数

43 /etc/passwd

[root@sail test]# wc -w /etc/passwd 统计单词数

87 /etc/passwd

[root@sail test]# wc -c /etc/passwd 统计字节数

2267 /etc/passwd

[root@sail test]# wc -m /etc/passwd 统计字符数

2267 /etc/passwd

[root@sail test]# wc -L /etc/passwd 统计文件中最长行的长度（包含空格）

99 /etc/passwd

tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin

多个文件的统计

[root@sail test]# ls –l

-rw-r--r--. 1 root root 2267 Nov 8 21:23 passwd

-rw-r--r--. 1 root root 2267 Nov 8 21:23 passwd2

-rw-r--r--. 1 root root 2267 Nov 8 21:23 passwd3

上面3个测试文件其同样的内容。

[root@sail test]# wc -clw passwd passwd2 passwd3

43 87 2267 passwd

43 87 2267 passwd2

43 87 2267 passwd3

129 261 6801 total

多个文件一起统计时，按行显示每个文件的信息，输出结果最后一行就是将多个文件的统计结果汇总。

## iconv

作用：

对于给定文件把它的内容从一种编码转换成另一种编码。iconv 程序把文本 从一种编码转换为另一种编码，更准确一点,他是把 -f 指定的编码转换为-t指定的编码。这两种编码默认都是当前locale的编码, 所有 inputfile 都依次 进行转换。如果没有指定 inputfile 则使用标准输入. 转换后的内容会输出到 标准输出。

格式：

iconv [-f encoding] [-t encoding] [inputfile ...]

参数：

❒ -f encoding 把字符从encoding编码开始转换。

❒ -t encoding :把字符转换到encoding编码。

❒ -l 列出已知的编码字符集合 。

❒ -o file 指定输出文件 。

❒ -c 忽略输出的非法字符 。

❒ -s 禁止警告信息，但不是错误信息 。

❒ --verbose 显示进度信息 。

❒ -f和-t所能指定的合法字符在-l选项的命令里面都列出来了。

示例：

在windos用记事本编写的脚本文件，上传到linux服务器之后，经常会出现乱码，无法执行。而linux服务器上产生的脚本文件导出到windows常常也会发生乱码。乱码出现的原因就是linux和windows的编码不一致所产生的，可以使用iconv将文件转码输出成适合各自系统编码的文件，就不会出现乱码。

检查工具

[root@sail ~]# rpm -qf `which iconv` #查看工具是否安装

glibc-common-2.17-105.el7.x86\_64

如果没安装可以使用yum install –y glibc-common 进行安装。

查看系统中常见的几种编码

[root@sail ~]# iconv -l |grep GB

CN-GB//

CSGB2312//

CSISO58GB1988//

EBCDIC-CP-GB//

GB//

GB2312//

GB13000//

GB18030//

GBK//

GB\_1988-80//

GB\_198880//

ISO646-GB//

[root@sail ~]# iconv -l |grep UTF

ISO-10646/UTF-8/

ISO-10646/UTF8/

UTF-7//

UTF-8//

UTF-16//

UTF-16BE//

UTF-16LE//

UTF-32//

UTF-32BE//

UTF-32LE//

UTF7//

UTF8//

UTF16//

UTF16BE//

UTF16LE//

UTF32//

UTF32BE//

UTF32LE//

实战iconv转码输出

在windows上编写一个记事本文件内容如下，上传到linux服务器：

windows上的文件，上传到linux服务器，出现乱码。

原因是：windows和linux系统使用的编码不一致导致的。

解决方案：使用iconv转码工具，转化成适合系统的编码，再次查看文件就不会出现乱码了。

[root@sail ~]# ls -lt

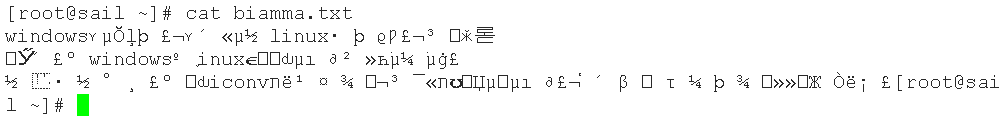
-rw-r--r--. 1 root root 181 11月 8 22:02 biamma.txt

[root@sail ~]# cat biamma.txt # 用cat查看文件内容显示乱码如下

windowsʏµŎļþ£¬ʏ´«µ½linux·þϱǷ£¬³󐗂ӂ롣

ԭӲˇ£ºwindowsº̬inuxϵͳʹԃµıძ²»һׂµ¼ׂµġ£

½⿶·½°¸£ºʹԃiconvתë¹¤¾ߣ¬ת»¯³ʊʺЏµͳµıძ£¬ՙ´β鿴τ¼þ¾Ͳ»»ᴶЖÒë¡£[root@sail ~]#



转码输出新的文件：

[root@sail ~]# iconv -f gb2312 biamma.txt -t utf-8 -o new.txt

参数解释：其中-f参数表示指出源文件biamma.txt的编码方式，其来自于windows所以编码为gb2312，-t指定输出文件的编码方式，-o是指定转码后输出文件的名称。

[root@sail ~]# ls –lt #转码输出文件new.txt

-rw-r--r--. 1 root root 255 11月 8 22:14 new.txt

-rw-r--r--. 1 root root 181 11月 8 22:02 biamma.txt

[root@sail ~]# file new.txt #用file查看文件

new.txt: UTF-8 Unicode text, with CRLF line terminators

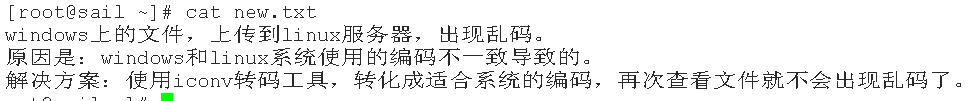
看到输出文件类型为UTF-8编码

[root@sail ~]# cat new.txt #再次用cat查看文件内容

windows上的文件，上传到linux服务器，出现乱码。

原因是：windows和linux系统使用的编码不一致导致的。

解决方案：使用iconv转码工具，转化成适合系统的编码，再次查看文件就不会出现乱码了。



查看到内容和之前在windows上的完全一致，且显示完全正常没有乱码。其原因是转码输出的文件已经是utf-8编码的文件，符合linux系统的编码方式。

## dos2unix

作用：

dos2unix是将Windows格式文件转换为Unix、Linux格式的实用命令。Windows格式文件的换行符为\r\n ,而Unix&Linux文件的换行符为\n. dos2unix命令其实就是将文件中的\r\n 转换为\n。 而unix2dos则是和dos2unix互为孪生的一个命令，它是将Linux&Unix格式文件转换为Windows格式文件的命令。

格式：

dos2unix [options] [-c convmode] [-o file ...] [-n infile outfile ...]

unix2dos [options] [-c convmode] [-o file ...] [-n infile outfile ...]

常用参数：

❒ -c, --convmode 转换模式

❒ -f, --force 强制转换

❒ -h, --help 查看帮助信息

❒ -k, --keepdate 保持文件时间戳不变

❒ -n, --newfile 保留原本的旧档，将转换后的内容输出到新档案.默认都会直接在原来的文件上修改

❒ -o, --oldfile 在源文件转换，默认参数

❒ -q, --quiet 静默模式，不输出转换结果信息等

❒ -V, --version 显示命令版本信息

示例：

查看工具是否安装

[root@sail ~]# rpm -qf `which dos2unix`

dos2unix-6.0.3-4.el7.x86\_64

将一个文件从dos格式转换成unix格式：

[root@sail ~]# dos2unix filename

一次转换多个文件：

[root@sail ~]# dos2unix filename1 filename2 filename3

保留原文件转码：

默认情况下会在源文件上进行转换，如果需要保留源文件，那么可以使用参数-n。

[root@sail ~]# dos2unix -n oldfilename newfilename

[root@sail ~]# dos2unix -n dosfile linuxfile

保持文件时间戳不变：

[root@sail ~]# dos2unix -k file

[root@sail ~]# dos2unix -k file1 file2 file3

静默模式格式化文件

[root@sail ~]# dos2unix -q file

[root@sail ~]# dos2unix -q file1 file2 file3

实战dos2unix转化格式

在Windows系统上编写的脚本文件，当上传到unix系统后无法直接来运行的。原因是Windows格式文件的换行符为\r\n ,而Unix&Linux文件的换行符为\n。

#!/bin/bash

#作者：sail

#时间：2016-11-08

#作用：输入要添加的用户数，然后批量添加用户

#--------------------------------------------------------------------------

read -p "请输入要添加的用户个数：" NUM

echo "系统将添加$NUM个用户"

a=`expr $NUM / 10`

for (( i=0;i<=$a;i++));

do

for (( j=0;j<=9;j++))

do

Ulist=$i$j

if [ $i$j -gt 00 -a $i$j -le $NUM ];then

useradd xuegod$i$j

echo "xuegod$i$j" |passwd --stdin xuegod$i$j &>/dev/null

echo "用户xuegod$Ulist已添加,初始密码为xuegod$i$j！"

fi

done

done

在windows上编写了一个添加用户的脚本文件，上传到unix系统，执行测试。

[root@sail scripts]# cat user.sh

#!/bin/bash

#作者：sail

#时间：2016-11-08

#-----------------------------------------------------------------------------------------------------

read -p "ȫˤɫҪͭ¼ӵœû§¸󋽣º" NUM

echo "ϵͳ½«ͭ¼ѤNUM¸󔃻§"

a=`expr $NUM / 10`

for (( i=0;i<=$a;i++));

do

for (( j=0;j<=9;j++))

do

Ulist=$i$j

if [ $i$j -gt 00 -a $i$j -le $NUM ];then

useradd xuegod$i$j

echo "xuegod$i$j" |passwd --stdin xuegod$i$j &>/dev/null

echo "ԃ»§xuegod$Ulistӑͭ¼Ѭ³򊽃݂쎪xuegod$i$j£¡"

fi

done

done

用cat查看文件内容首先是乱码，执行脚本测试

[root@sail scripts]# sh user.sh

ȫˤɫҪͭ¼ӵœû§¸󋽣º20

': not a valid identifierUM

ϵͳ½«ͭ¼Ӹ󔃻§

expr: syntax error

user.sh: line 9: syntax error near unexpected token `$'\r''

'ser.sh: line 9: `for (( i=0;i<=$a;i++));

执行脚本时报错，大概看到是格式换行问题，我们用dos2unix转码文件。

[root@sail ~]# yum install dos2unix-6.0.3-4.el7.x86\_64 #安装工具

[root@sail scripts]# dos2unix -n user.sh unix\_user.sh

dos2unix: converting file user.sh to file unix\_user.sh in Unix format ...

[root@sail scripts]# ls -lt

-rw-r--r-- 1 root root 620 Nov 9 15:57 unix\_user.sh

-rw-r--r-- 1 root root 645 Nov 9 14:49 user.sh

格式转化完成，再次执行脚本测试。

[root@sail scripts]# sh unix\_user.sh

ȫˤɫҪͭ¼ӵœû§¸󋽣º2

ϵͳ½«ͭ¼Ѳ¸󔃻§

ԃ»§xuegod01ӑͭ¼Ѭ³򊽃݂쎪xuegod01£¡

ԃ»§xuegod02ӑͭ¼Ѭ³򊽃݂쎪xuegod02£¡

这次没有报错，但是看到有乱码，其实就是windows上的文件在linux上编码不一致产生的。

用iconv进行转码输出新的文件，方法用前面写的iconv用法一样。

[root@sail scripts]# iconv -f gb2312 unix\_user.sh -t utf-8 -o utf8-unix-user.sh

[root@sail scripts]# ls -lt

-rw-r--r-- 1 root root 678 Nov 9 16:06 utf8-unix-user.sh

转码输出新的文件，再次执行脚本测试。

[root@sail scripts]# sh utf8-unix-user.sh

请输入要添加的用户个数：5

系统将添加5个用户

用户xuegod01已添加,初始密码为xuegod01！

用户xuegod02已添加,初始密码为xuegod02！

用户xuegod03已添加,初始密码为xuegod03！

用户xuegod04已添加,初始密码为xuegod04！

用户xuegod05已添加,初始密码为xuegod05！

[root@sail home]# ls -lt

drwx------ 3 xuegod05 xuegod05 74 Nov 9 16:08 xuegod05

drwx------ 3 xuegod04 xuegod04 74 Nov 9 16:08 xuegod04

drwx------ 3 xuegod03 xuegod03 74 Nov 9 16:08 xuegod03

drwx------ 3 xuegod02 xuegod02 74 Nov 9 16:08 xuegod02

drwx------ 3 xuegod01 xuegod01 74 Nov 9 16:08 xuegod01

脚本执行成功，完全实现了脚本的功能，批量创建5个用户。

## file

作用

file命令用来检测并显示给定文件的类型。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

格式

file [-beLvz][-f <名称文件>][-m <魔法数字文件>...][文件或目录...]

参数

file命令返回结果以及含义

❒ empty 空文件

❒ directory 目录文件

❒ English text 英文正式文件

❒ assembler program text 汇编语言程序的正文文件

❒ ascii text ASCII编码的文本文件

❒ command text 命令语言编写的命令正文程序

❒ c program C语言正文程序

❒ relocation text 用于连接的目标文件

❒ executable 可执行的目标代码文件

❒ data 数据文件

file参数

短选项 长选项 涵义

❒ -m --magic-file LIST 指定魔法数字名

❒ -z --uncompress 探测压缩过的文件类型

❒ -b --brief 列出辨识结果时，不显示文件名称

❒ -c --checking-printout 详细显示指令执行过程，便于排错或分析程序执行的情形

❒ -e --exclude TEST对文件列表排除TEST类型。

❒ -f --files-from FILE指定文件列表参数，获取该列表里面所有文件的类型

❒ -F --separator STRING 使用字符串作为分隔符而不是“:”

❒ -i --mime 显示MIME类别

--apple 显示Apple CREATOR/TYPE

--mime-type 显示MIME类别

--mime-encoding 显示MIME编码

❒ -k --keep-going 执行命令错误时不终止

❒ -l --list list magic strength

❒ -L --dereference跟随符号链接（默认）

❒ -h --no-dereference不跟随符号链接

❒ -n --no-buffer没有缓冲输出

❒ -N --no-pad do not pad output

❒ -0 --print0 terminate filenames with ASCII NUL

❒ -p --preserve-date preserve access times on files

❒ -r --raw don't translate unprintable chars to ooo

❒ -s --special-files treat special (block/char devices) files as ordinary ones

❒ -C --compile compile file specified by -m

❒ -d --debug print debugging messages

显示文件类型

[root@sail scripts]# file OK.sh

OK.sh: Bourne-Again shell script, UTF-8 Unicode text executable

[root@sail scripts]# file /etc/passwd

/etc/passwd: ASCII text

[root@sail scripts]# file /home/sail

/home/sail: directory

显示符号链接的文件类型

[root@sail etc]# ls -l /etc/grub2.cfg

lrwxrwxrwx. 1 root root 22 Oct 23 01:03 grub2.cfg -> ../boot/grub2/grub.cfg

[root@sail etc]# file /etc/grub2.cfg

/etc/grub2.cfg: symbolic link to `../boot/grub2/grub.cfg'

[root@sail etc]# file -L /etc/grub2.cfg #显示符号链接源文件的文件类型

/etc/grub2.cfg: ASCII text

查看目录下的文件类型

[root@sail home]# ls -l

-rw-r--r-- 1 root root 0 Nov 9 14:21 a.txt

drwx------. 14 sail sail 4096 Nov 7 13:46 sail

drwxr-xr-x 2 root root 6 Nov 9 14:21 test

[root@sail home]# file \* /home

a.txt: empty

sail: directory

test: directory

/home: directory

不显示文件名称，查看文件类型

[root@sail home]# file -b /etc/passwd

ASCII text

[root@sail home]# file -i /etc/passwd #显示文件MIME类别

/etc/passwd: text/plain; charset=us-ascii

file使用通配符，获取多个文件的类型

[root@sail home]# ls -lt

-rw-r--r-- 1 root root 0 Nov 9 14:35 iii.txt

-rw-r--r-- 1 root root 0 Nov 9 14:34 333.txt

-rw-r--r-- 1 root root 0 Nov 9 14:34 xx.txt

-rw-r--r-- 1 root root 0 Nov 9 14:21 a.txt

[root@sail home]# file \*.txt #一次可以获取多个文件的类型

333.txt: empty

a.txt: empty

iii.txt: empty

xx.txt: empty

探测压缩文件类型

[root@sail back]# ls -l

-rw-r--r-- 1 root root 1047 Nov 8 13:10 2016-11-08.tar.gz

[root@sail back]# file -z 2016-11-08.tar.gz

2016-11-08.tar.gz: POSIX tar archive (GNU) (bzip2 compressed data, block size = 900k)

列出所有文件类型

[root@sail /]# file \*

back: directory

bin: symbolic link to `usr/bin'

etc: directory

home: directory

lib: symbolic link to `usr/lib'

lib64: symbolic link to `usr/lib64'

linux-4.8.6.tar.xz: XZ compressed data

media: directory

mnt: directory

run: directory

sbin: symbolic link to `usr/sbin'

scripts: directory

sdb1: directory

tmp: sticky directory

## diff

格式：diff [-a] [-c] [文件1] [文件2]

作用：最简单的情况下，比较给定的两个文件的不同

参数选项：

❒ - <行数>：指定要显示多少行的文本。此参数必须与-c 或-u 参数一并使用

* -a 或——text：diff 预设只会逐行比较文本文件
* -b 或--ignore-space-change：不检查空格字符的不同
* -B 或--ignore-blank-lines：不检查空白行
* -c：显示全部内容，并标出不同之处
* -C：<行数>或--context<行数>：与执行“-c-<行数>”指令相同
* -d 或——minimal：使用不同的演算法，以小的单位来做比较
* -D<巨集名称>或ifdef<巨集名称>：此参数的输出格式可用于前置处理器巨集
* -e 或——ed：此参数的输出格式可用于ed 的script 文件
* -f 或-forward-ed：输出的格式类似ed 的script 文件，但按照原来文件的顺序来显示不同处
* -H 或--speed-large-files：比较大文件时，可加快速度
* -i 或--ignore-case：不检查大小写的不同
* -l 或——paginate：将结果交由pr 程序来分页
* -n 或——rcs：将比较结果以RCS 的格式来显示
* -N 或--new-file：在比较目录时，若文件A 仅出现在某个目录中，预设会显示：Only in 目录，文件A
* 若使用-N 参数，则diff 会将文件A 与一个空白的文件比较
* -p：若比较的文件为C 语言的程序码文件时，显示差异所在的函数名称
* -q 或--brief：仅显示有无差异，不显示详细的信息
* -r 或——recursive：比较子目录中的文件
* -s 或--report-identical-files：若没有发现任何差异，仍然显示信息
* -S<文件>或--starting-file<文件>：在比较目录时，从指定的文件开始比较
* -t 或--expand-tabs：在输出时，将tab 字符展开
* -T 或--initial-tab：在每行前面加上tab 字符以便对齐
* -u，-U<列数>或--unified=<列数>：以合并的方式来显示文件内容的不同
* -w 或--ignore-all-space：忽略全部的空格字符
* -W<宽度>或--width<宽度>：在使用-y 参数时，指定栏宽
* -x<文件名或目录>或--exclude<文件名或目录>：不比较选项中所指定的文件或目录
* -X<文件>或--exclude-from<文件>；您可以将文件或目录类型存成文本文件，然后在=<文件>中指定此文本文件
* -y 或--side-by-side：以并列的方式显示文件的异同之处
* --left-column：在使用-y 参数时，若两个文件某一行内容相同，则仅在左侧的栏位显示该行内容
* --suppress-common-lines：在使用-y 参数时，仅显示不同之处

示例：

编辑两个不同内容的文件，并用diff比较两个文件里面不相同的内容：

[root@vhxrhel7test ~]# cat 1.txt //查看1.txt 内容

hello1

aello2

cello3

cello3

[root@vhxrhel7test ~]# cat 2.txt //查看2.txt 内容

world1

world2

world3

[root@vhxrhel7test ~]# diff -c 1.txt 2.txt //比较两个文件不同之处

\*\*\* 1.txt 2016-11-21 03:41:27.193743595 +0100

--- 2.txt 2016-11-21 03:41:42.200804593 +0100

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 1,5 \*\*\*\*

! hello1

! aello2

! cello3

! cello3

--- 1,4 ----

! world1

! world2

! world3

## tree

格式：tree [-aACdDfFgilnNpqstux] [目录]

作用：tree以树状图样式列出目录的内容

选项：

* -a：显示所有文件和目录
* -A：使用ASNI 绘图字符显示树状图而非以ASCII 字符组合
* -C：在文件和目录清单加上色彩，便于区分各种类型
* -d：先是目录名称而非内容
* -D：列出文件或目录的更改时间
* -f：在每个文件或目录之前，显示完整的相对路径名称
* -F：在执行文件，目录，Socket，符号连接，管道名称名称，各自加上"\*"，"/"，"@"，"|"号
* -g：列出文件或目录的所属群组名称，没有对应的名称时，则显示群组识别码
* -i：不以阶梯状列出文件和目录名称
* -l：如遇到性质为符号连接的目录，直接列出该连接所指向的原始目录
* -n：不在文件和目录清单加上色彩
* -N：直接列出文件和目录名称，包括控制字符
* -p：列出权限标示
* -q：用“？”号取代控制字符，列出文件和目录名称
* -s：列出文件和目录大小
* -t：用文件和目录的更改时间排序
* -u：列出文件或目录的拥有者名称，没有对应的名称时，则显示用户识别码
* -x：将范围局限在现行的文件系统中，若指定目录下的某些子目录，其存放于另一个文件系统上，则将该
* 目录予以排除在寻找范围外

示例：

在/home下创建test目录并在目录下创建一些测试目录和文件，用tree查看：

[root@vhxrhel7test ~]# tree /test

/test

├── a

│   ├── b

│   │   └── c

│   └── No.one.txt

├── aa.txt

├── bb.txt

├── cc.txt

├── dd.txt

└── test.txt

## chattr

格式：chattr [选项] [参数]

作用：改变文件属性，这个属性不同于chmod修改的读写和执行属性，chattr修改的是隐藏属性

补充说明：此项指令可改变存放在ext2、ext3、ext4、xfs、ubifs、reiserfs、jfs等文件系统上的文件或目录属性，修改的属性共计一下八种模式。

a：让文件或目录仅供附加用途

b：不更新文件或目录的最后存放时间

c：将文件或目录压缩后存放

d：将文件或目录排除在倾倒操作之外

i：不得任意更动文件或目录

s：保密性删除文件或目录

S：即时更新文件或目录

u：预防意外删除

选项：

❒ -R 递归处理，将指定目录下的所有文件及子目录一并处理

❒ -v<版本编号> 设置文件或目录版本

❒ -V 显示指令执行过程

❒ +<属性> 开启文件或目录的该项属性

❒ -<属性> 关闭文件或目录的该项属性

❒ =<属性> 制定文件或目录的该项属性

参数：

目录或文件：指定要修改隐藏属性的目录或文件。如果是多个文件和目录要修改的话，各个名称之间要用空格隔开

示例一：

尝试在/tmp目录下创建一个文件，并加入i的参数，然后删除

[root@localhost ~]# cd /mnt

[root@localhost mnt]# touch test

[root@localhost mnt]# chattr +i test

[root@localhost mnt]# rm -f test

rm: cannot remove `test': Operation not permitted

示例：

将文件test的参数i移除

[root@localhost mnt]# chattr -i test

## lsattr

格式：lsattr [选项] [参数]

作用：显示文件属性，这里现实的是chattr修改的文件的隐藏属性

选项：

❒ -a 显示所有文件和目录，包括以“.”为名称开头字符的额外內建，现行目录“.”与上层目录“..”

❒ -d 显示目录名称，而非其内容

❒ -l 此参数目前没有任何作用

❒ -R 递归处理，将制定目录下的所有文件及子目录一并处理

❒ -v 显示文件或目录版本

❒ -V 显示版本信息

参数：

目录或文件：指定要查询隐藏属性的目录或文件。如果有多个目录和文件的隐藏属性要查询，各个名称之间用空格隔开

示例一：

在/mnt目录下创建一个test文件，查看其隐藏属性（chattr添加的属性），然后将test文件添加i属性，然后再次查看其隐藏属性

[root@localhost ~]# cd /mnt

[root@localhost mnt]# touch test

[root@localhost mnt]# lsattr test

-------------e- test

[root@localhost mnt]# chattr +i test

[root@localhost mnt]# lsattr test

----i--------e- test

示例二：

在/mnt目录下创建一个目录，然后查看这个目录的的隐藏属性，然后将这个目录添加i属性，再次查看该目录的隐藏属性，然后尝试删除目录。

[root@xuegod250 mnt]# mkdir test1

[root@xuegod250 mnt]# lsattr -d test1/

-------------e- test1/

[root@xuegod250 mnt]# chattr +i test1/

[root@xuegod250 mnt]# lsattr -d test1/

----i--------e- test1/

[root@xuegod250 mnt]# rm -rf test1/

rm: cannot remove `test1': Operation not permitted

示例三：

在示例二中的目录下创建一个文件，并且尝试删除这个文件

[root@xuegod250 mnt]# cd test1/

[root@xuegod250 test1]# touch test

touch: cannot touch `test': Permission denied

## rev

格式：rev [文件]

作用：将文件中的每行内容以字符为单位反序输出

选项：

* 文件：指定要反序显示内容的文件

示例：

[root@vhxrhel7test ~]# cat No.one.txt

xuegod1609

xuegod1607

VIP

redhat

CentOS

[root@vhxrhel7test ~]# rev No.one.txt

9061dogeux

7061dogeux

PIV

tahder

SOtneC

## vimdiff

格式：vimdiff [-do] [文件1] [文件2]

作用：实时反馈一个或者两个文件的比较结果

选项：

* -d：同时打开两个文件，并垂直在两栏显示
* -o：同时打开两个文件，并上下两栏显示

示例：

[root@vhxrhel7test ~]# vimdiff -d test.txt test1.txt //同时打开两个文件垂直在两栏显示



[root@vhxrhel7test ~]# vimdiff -o test.txt test1.txt //同时打开两个文件在上下两栏显示

## gzip

格式：gzip (选项)(参数)

作用：对文件压缩、解压命令

选项：

❒-c或--stdout 使用标准输出，保持原始文件不变；

❒-d或--decompress或----uncompress 解开压缩文件；

❒-f或--force 强行压缩文件。不理会文件名称或硬连接是否存在以及该文件是

否为符号连接；

❒-h或--help 在线帮助；

❒-l或--list 列出压缩文件的相关信息；

❒-L或--license 显示版本与版权信息；

❒-n或--no-name 压缩文件时，不保存原来的文件名称及时间戳记；

❒-N或--name 压缩文件时，保存原来的文件名称及时间戳记；

❒-q或--quiet 不显示警告信息；

❒-r或--recursive 递归处理，将指定目录下的所有文件及子目录一并处理；

❒-S或<压缩字尾字符串>或----suffix<压缩字尾字符串> 更改压缩字尾字符串；

❒-t或--test 测试压缩文件是否正确无误；

❒-v或--verbose 显示指令执行过程；

❒-V或--version 显示版本信息；

❒-<压缩效率> 压缩效率是一个介于1~9的数值，预设值为“6”，指定愈大的数

值，压缩效率就会愈高；

❒--best 此参数的效果和指定“-9”参数相同；

❒--fast 此参数的效果和指定“-1”参数相同。

参数：

文件列表：指定要压缩的文件列表。

示例：

准备下测试文件

[root@localhost mnt]# cp /etc/passwd .

[root@localhost mnt]# ll

total 8

-rw-r--r-- 1 root root 2264 Nov 24 13:41 passwd

打包当前目录下的passwd文件

[root@localhost mnt]# gzip passwd

[root@localhost mnt]# ll

total 8

-rw-r--r-- 1 root root 909 Nov 24 13:41 passwd.gz

显示passwd.gz中压缩的文件信息，但不解压

[root@localhost mnt]# gzip -l passwd.gz

compressed uncompressed ratio uncompressed\_name

909 2264 61.0% passwd

解压passwd.gz,并显示详细信息

[root@localhost mnt]# gzip -vd passwd.gz

passwd.gz: 61.0% -- replaced with passwd

压缩目录下的grub2目录

[root@localhost mnt]# ls

grub2 passwd

[root@localhost mnt]# gzip -rv grub2/

grub2//device.map: 25.0% -- replaced with grub2//device.map.gz

grub2//i386-pc/acpi.mod: 54.5% -- replaced with grub2//i386-pc/acpi.mod.gz

grub2//i386-pc/date.mod: 49.0% -- replaced with grub2//i386-pc/date.mod.gz

... ....

[root@localhost mnt]# ls grub2/\*.gz

grub2/device.map.gz grub2/grub.cfg.gz grub2/grubenv.gz

[root@localhost mnt]# ls grub2/i386-pc/\*.gz

grub2/i386-pc/acpi.mod.gz grub2/i386-pc/mdraid09\_be.mod.gz

grub2/i386-pc/adler32.mod.gz ... ....

提示：这里gzip是正对文件夹下的每个文件做压缩，如果有下级目录则进入下级目录继续压缩。

## tar

格式：tar（选项）（参数）

作用：为linux的文件和目录创建档案，把一大堆的文件和目录全部打包成一个文件

选项：

❒-A或--catenate 新增文件到以存在的备份文件；

❒-b 每个记录 BLOCKS x 512 字节

❒-B 读取时重新分块(只对 4.2BSD 管道有效)；

❒-c或--create 建立新的归档文件；

❒-C <目录> 这个选项用在解压缩，若要在特定目录解压缩，可以使用这个项。

❒-d 找出归档和文件系统的差异；

❒-x或--extract或--get 从归档文件中解出文件；

❒-t或--list 列出归档文件的内容；

❒-z或--gzip或--ungzip 通过gzip指令处理归档文件；

❒-j 支持bzip2解压文件；

❒-Z或--compress或--uncompress 通过compress指令处归档份文件；

❒-f<备份文件>或--file=<备份文件> 指定备份文件；

❒-v或--verbose 详细地列出处理的文件；

❒-r 添加文件到已经归档的文件尾；

❒-u 仅追加比归档中副本更新的文件；

❒-l 文件系统边界设置；

❒-k 保留原有文件不覆盖；

❒-m 不要解压文件的修改时间；

❒-W 确认压缩文件的正确性；

❒-p或--same-permissions 用原来的文件权限还原文件；

❒-P或--absolute-names 文件名使用绝对名称，不移除文件名称前的“/”号；

❒-N <日期格式>或--newer=<日期时间> 只将较指定日期更新的文件保存到归档文里；

❒-X或--exclude-from=FIL=<范本样式> 排除符合范本样式的文件。

参数：

文件或目录：指定要打包的文件或目录列表。

示例：

打包当前目录下的passwd文件

[root@localhost mnt]# tar -cvf passwd.tar ./passwd

./passwd

[root@localhost mnt]# ll

total 16

-rw-r--r-- 1 root root 2264 Nov 24 15:32 passwd

-rw-r--r-- 1 root root 10240 Nov 24 15:34 passwd.tar

打包当前目录下的boot目录

[root@localhost mnt]# tar -zcf boot.tar.gz boot/

[root@localhost mnt]# ll

total 167880

dr-xr-xr-x 4 root root 4096 Nov 24 15:55 boot

-rw-r--r-- 1 root root 171904278 Nov 24 16:14 boot.tar.gz

查看boot.tar.gz中包含的文件，但不解压。

[root@localhost mnt]# tar -ztf boot.tar.gz

boot/

boot/grub/

boot/grub/splash.xpm.gz

boot/grub2/

... ...

只解压boot.tar.gz文件中的img文件。

[root@localhost mnt]# tar -zxf boot.tar.gz \*.img

[root@localhost mnt]# ll boot/

total 148688

drwxr-xr-x 3 root root 20 Nov 24 16:17 grub2

-rw-r--r-- 1 root root 57594550 Nov 24 15:55 initramfs-0-rescue-08abcc7a0de842638b6353522bb74bad.img

-rw-r--r-- 1 root root 30135525 Nov 24 15:55 initramfs-3.10.0-327.el7.x86\_64.img

-rw-r--r-- 1 root root 54323169 Nov 24 15:55 initramfs-4.8.5.img

-rw-r--r-- 1 root root 10191972 Nov 24 15:55 initrd-plymouth.img

排除指定文件打包其余文件

[root@localhost mnt]# tar --exclude /etc/passwd -zcf etc.tar.gz /etc/

tar: Removing leading `/' from member names

[root@localhost mnt]# tar -ztf etc.tar.gz |grep passwd

etc/pam.d/passwd

etc/security/opasswd

## unzip

格式：unzip （选项）（参数）

作用：解压缩”.zip”格式的压缩包

选项：

❒-c 将解压缩的结果显示到屏幕上，并对字符做适当的转换；

❒-f 更新现有的文件；

❒-l 显示压缩文件内所包含的文件；

❒-p 与-c参数类似，会将解压缩的结果显示到屏幕上，但不会执行任何的转换；

❒-t 检查压缩文件是否正确；

❒-u 与-f参数类似，但是除了更新现有的文件外，也会将压缩文件中的其他文

件解压缩到目录中；

❒-v 执行时显示详细的信息；

❒-z 仅显示压缩文件的备注文字；

❒-a 对文本文件进行必要的字符转换；

❒-b 不要对文本文件进行字符转换；

❒-C 压缩文件中的文件名称区分大小写；

❒-j 不处理压缩文件中原有的目录路径；

❒-L 将压缩文件中的全部文件名改为小写；

❒-M 将输出结果送到more程序处理；

❒-n 解压缩时不要覆盖原有的文件；

❒-o 不必先询问用户，unzip执行后覆盖原有的文件；

❒-P<密码> 使用zip的密码选项；

❒-q 执行时不显示任何信息；

❒-s 将文件名中的空白字符转换为底线字符；

❒-V 保留VMS的文件版本信息；

❒-X 解压缩时同时回存文件原来的UID/GID；

❒-d<目录> 指定文件解压缩后所要存储的目录；

❒-x<文件> 指定不要处理.zip压缩文件中的哪些文件；

❒-Z unzip-Z等于执行zipinfo指令。

参数：

压缩包：指定要解压的“.zip”压缩包

示例：

显示压缩包boot.zip包含的所有文件

[root@localhost mnt]# unzip -l boot.zip

Archive: boot.zip

Length Date Time Name

--------- ---------- ----- ----

0 10-31-2016 17:26 boot/

0 10-21-2016 18:04 boot/grub/

... ....

3423099 10-31-2016 17:25 boot/System.map

54323169 10-31-2016 17:27 boot/initramfs-4.8.5.img

--------- -------

192066685 332 files

检查压缩文件boot.zip是否正确

[root@localhost mnt]# unzip -t boot.zip

Archive: boot.zip

testing: boot/ OK

... ...

testing: boot/initramfs-4.8.5.img OK

No errors detected in compressed data of boot.zip.

解压压缩文件boot.zip到当前目录

[root@localhost mnt]# unzip -d . boot.zip

Archive: boot.zip

creating: ./boot/

... ...

inflating: ./boot/System.map

inflating: ./boot/initramfs-4.8.5.img

## uname

格式：(选项)

作用：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）

选项：

❒ -a或--all 显示全部的信息；

❒ -m或--machine 显示电脑硬件架构名称；

❒ -n或-nodename 显示在网络上的主机名称；

❒ -r或--release 显示内核发行号；

❒ -s或--sysname 显示内核名称；

❒ -v 显示内核版本；

❒ -p或--processor 输出处理器类型或"unknown"；

❒ -i或--hardware-platform 输出硬件平台或"unknown"；

❒ -o或--operating-system 输出操作系统名称；

❒ --help 显示帮助并退出；

❒ --version 显示版本信息并退出。

示例：

[root@localhost ~]# uname

Linux

[root@localhost ~]# uname -a

Linux localhost 2.6.32-431.el6.x86\_64 #1 SMP Fri Nov 22 03:15:09 UTC 2013 x86\_64 x86\_64 x86\_64 GNU/Linux

[root@localhost ~]# uname -m

x86\_64

[root@localhost ~]# uname -n

localhost

[root@localhost ~]# uname -r

2.6.32-431.el6.x86\_64

[root@localhost ~]# uname -s

Linux

[root@localhost ~]# uname -v

#1 SMP Fri Nov 22 03:15:09 UTC 2013

[root@localhost ~]# uname -vp

#1 SMP Fri Nov 22 03:15:09 UTC 2013 x86\_64

[root@localhost ~]# uname -vpi

#1 SMP Fri Nov 22 03:15:09 UTC 2013 x86\_64 x86\_64

[root@localhost ~]# uname -p

x86\_64

[root@localhost ~]# uname -i

x86\_64

[root@localhost ~]# uname -o

GNU/Linux

## hostname

hostname：（选项）（参数）

作用：用于显示和设置系统的主机名称

选项：

❒ -V 详细信息模式；

❒ -a 显示主机别名；

❒ -d 显示DNS域名；

❒ -f 显示FQDN名称；

❒ -i 显示主机的ip地址；

❒ -s 显示短主机名称，在第一个点处截断；

❒ -y 显示NIS域名。

❒ --help 显示帮助并退出；

参数：

主机名：指定要设置的主机名。

示例：

[root@localhost ~]# hostname

localhost

[root@localhost ~]# hostname -V

hostname 3.13

[root@localhost ~]# hostname -a

localhost.localdomain localhost4 localhost4.localdomain4 localhost.localdomain localhost6 localhost6.localdomain6

[root@localhost ~]# hostname -f

localhost

[root@localhost ~]# hostname -i

127.0.0.1 127.0.0.1

[root@localhost ~]# hostname -s

localhost

[root@localhost ~]# hostname -y

(none)

## dmesg

格式：dmesg (选项)

作用：显示开机信息

选项：

❒ -c：显示信息后，清除ring buffer中的内容

❒-s<缓冲区大小>：预设置为8196，刚好等于ring buffer的大小

❒-n：设置记录信息的层级

❒-r：打印原始消息缓冲区，即不带前缀的日志级别  
关于 dmesg 的更多用法，你可以通过指定dmesg -h 选项或man dmesg查询

示例：

1 查看开机信息, 因信息量过大，可以通过 | more 分页查看，若要退出 ，直接输入 q 即可

[root@xuegod65 ~]# dmesg | more

Initializing cgroup subsys cpuset

Initializing cgroup subsys cpu

Linux version 2.6.32-431.el6.x86\_64 (mockbuild@c6b8.bsys.dev.centos.org) (gcc version 4.4.7 201

20313 (Red Hat 4.4.7-4) (GCC) ) #1 SMP Fri Nov 22 03:15:09 UTC 2013

Command line: ro root=UUID=45c89723-7ced-4218-b35d-e1077971681c rd\_NO\_LUKS rd\_NO\_LVM LANG=en\_US

.UTF-8 rd\_NO\_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_D

M rhgb quiet

KERNEL supported cpus:

Intel GenuineIntel

AMD AuthenticAMD

Centaur CentaurHauls

BIOS-provided physical RAM map:

BIOS-e820: 0000000000000000 - 000000000009ec00 (usable)

BIOS-e820: 000000000009ec00 - 00000000000a0000 (reserved)

BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)

BIOS-e820: 0000000000100000 - 0000000024ae0000 (usable)

BIOS-e820: 0000000024ae0000 - 0000000024aff000 (ACPI data)

BIOS-e820: 0000000024aff000 - 0000000024b00000 (ACPI NVS)

BIOS-e820: 0000000024b00000 - 0000000024c00000 (usable)

BIOS-e820: 00000000f0000000 - 00000000f8000000 (reserved) 。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

2 设置内核日志缓冲区大小，预设置为8196，刚好等于ring buffer的大小，也可以自定义，如设为：10000

[root@xuegod65 ~]# dmesg -s 10000

3. 设置记录信息的层级，即将消息日志记录到控制台的级别 ，如： 要将3级别的日志信息记录到控制台

[root@xuegod65 ~]# dmesg -n 3

4 显示并清除ring buffer中的内容

[root@xuegod65 ~]# dmesg -c

。。。。。。。。。。。。。。。。。。。。。。。

ci 0000:00:15.7: bridge window [mem 0xe5e00000-0xe5efffff 64bit pref]

pci 0000:00:16.0: PCI bridge to [bus 0b-0b]

pci 0000:00:16.0: bridge window [io 0x5000-0x5fff]

pci 0000:00:16.0: bridge window [mem 0xfd300000-0xfd3fffff]

pci 0000:00:16.0: bridge window [mem 0xe7900000-0xe79fffff 64bit pref]

pci 0000:00:16.1: PCI bridge to [bus 0c-0c]

pci 0000:00:16.1: bridge window [io 0x9000-0x9fff]

pci 0000:00:16.1: bridge window [mem 0xfcf00000-0xfcffffff]

pci 0000:00:16.1: bridge window [mem 0xe7500000-0xe75fffff 64bit pref]

pci 0000:00:16.2: PCI bridge to [bus 0d-0d]

pci 0000:00:16.2: bridge window [io 0xd000-0xdfff]

pci 0000:00:16.2: bridge window [mem 0xfcb00000-0xfcbfffff]

pci 0000:00:16.2: bridge window [mem 0xe7100000-0xe71fffff 64bit pref]

pci 0000:00:16.3: PCI bridge to [bus 0e-0e]

pci 0000:00:16.3: bridge window [io 0xf000-0x0000] (disabled)

pci 0000:00:16.3: bridge window [mem 0xfc700000-0xfc7fffff]

pci 0000:00:16.3: bridge window [mem 0xe6d00000-0xe6dfffff

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

5 打印原始消息缓冲区

[root@xuegod65 ~]# dmesg -r | more

<6>Initializing cgroup subsys cpuset

<6>Initializing cgroup subsys cpu

<5>Linux version 2.6.32-431.el6.x86\_64 (mockbuild@c6b8.bsys.dev.centos.org) (gcc version 4.4.7

20120313 (Red Hat 4.4.7-4) (GCC) ) #1 SMP Fri Nov 22 03:15:09 UTC 2013

<6>Command line: ro root=UUID=45c89723-7ced-4218-b35d-e1077971681c rd\_NO\_LUKS rd\_NO\_LVM LANG=en

\_US.UTF-8 rd\_NO\_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=us rd\_N

O\_DM rhgb quiet

<6>KERNEL supported cpus:

<6> Intel GenuineIntel

<6> AMD AuthenticAMD

<6> Centaur CentaurHauls

<6>BIOS-provided physical RAM map:

<6> BIOS-e820: 0000000000000000 - 000000000009ec00 (usable)

<6> BIOS-e820: 000000000009ec00 - 00000000000a0000 (reserved)

<6> BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)

<6> BIOS-e820: 0000000000100000 - 0000000024ae0000 (usable)

<6> BIOS-e820: 0000000024ae0000 - 0000000024aff000 (ACPI data)

<6> BIOS-e820: 0000000024aff000 - 0000000024b00000 (ACPI NVS)

<6> BIOS-e820: 0000000024b00000 - 0000000024c00000 (usable)

<6> BIOS-e820: 00000000f0000000 - 00000000f8000000 (reserved)

**扩展应用：**

1 查看显卡信息 **| grep** 表示从数据中将我们需要的数据过滤出来,**-i**并且不区分大小写

[root@xuegod65 ~]# lspci | grep -i vga

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

2查看 CPU 信息

[root@xuegod65 ~]# dmesg | grep -i cpu | more

I Initializing cgroup subsys cpuset

Initializing cgroup subsys cpu

KERNEL supported cpus:

x86 PAT enabled: cpu 0, old 0x7040600070406, new 0x7010600070106

SMP: Allowing 128 CPUs, 124 hotplug CPUs

NR\_CPUS:4096 nr\_cpumask\_bits:128 nr\_cpu\_ids:128 nr\_node\_ids:1

PERCPU: Embedded 31 pages/cpu @ffff880003000000 s94872 r8192 d23912 u131072

pcpu-alloc: s94872 r8192 d23912 u131072 alloc=1\*2097152

pcpu-alloc: [0] 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015

pcpu-alloc: [0] 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031

pcpu-alloc: [0] 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047

pcpu-alloc: [0] 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063

pcpu-alloc: [0] 064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079

pcpu-alloc: [0] 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095

pcpu-alloc: [0] 096 097 098 099 100 101 102 103 104 105 106 107 108 109 110 111

pcpu-alloc: [0] 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127

Initializing cgroup subsys cpuacct

CPU: Physical Processor ID: 0

CPU: Processor Core ID: 0

。。。。。。。。。。。。。。。。。。。。。。。。。。

3 查看内存信息

[root@xuegod65 ~]# dmesg | grep mem | more

initial memory mapped : 0 - 20000000

init\_memory\_mapping: 0000000000000000-0000000024c00000

NUMA: Allocated memnodemap from 10000 - 11840

Bootmem setup node 0 0000000000000000-0000000024c00000

(7 early reservations) ==> bootmem [0000000000 - 0024c00000]

crashkernel=auto resulted in zero bytes of reserved memory.

DMA zone: 56 pages used for memmap

DMA32 zone: 2002 pages used for memmap

PM: Registered nosave memory: 000000000009e000 - 000000000009f000

PM: Registered nosave memory: 000000000009f000 - 00000000000a0000

PM: Registered nosave memory: 00000000000a0000 - 00000000000dc000

PM: Registered nosave memory: 00000000000dc000 - 0000000000100000

PM: Registered nosave memory: 0000000024ae0000 - 0000000024aff000

PM: Registered nosave memory: 0000000024aff000 - 0000000024b00000

please try 'cgroup\_disable=memory' option if you don't want memory cgroups

Initializing cgroup subsys memory

pci\_root PNP0A03:00: host bridge window [mem 0x000a0000-0x000bffff]

pci\_root PNP0A03:00: host bridge window [mem 0x000cc000-0x000cffff]

pci\_root PNP0A03:00: host bridge window [mem 0x000d0000-0x000d3fff]

pci\_root PNP0A03:00: host bridge window [mem 0x000d4000-0x000d7fff]

。。。。。。。。。。。。。。。。。。。。。。。

4 查看硬盘信息

[root@xuegod65 ~]# dmesg | grep sda

sd 2:0:0:0: [sda] 83886080 512-byte logical blocks: (42.9 GB/40.0 GiB)

sd 2:0:0:0: [sda] Write Protect is off

sd 2:0:0:0: [sda] Mode Sense: 61 00 00 00

sd 2:0:0:0: [sda] Cache data unavailable

sd 2:0:0:0: [sda] Assuming drive cache: write through

sd 2:0:0:0: [sda] Cache data unavailable

sd 2:0:0:0: [sda] Assuming drive cache: write through

sda: sda1 sda2 sda3 sda4 <

sda5 sda6 >

sd 2:0:0:0: [sda] Cache data unavailable

sd 2:0:0:0: [sda] Assuming drive cache: write through

sd 2:0:0:0: [sda] Attached SCSI disk

EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts:

dracut: Mounted root filesystem /dev/sda2

EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:

EXT4-fs (sda5): mounted filesystem with ordered data mode. Opts:

5 查看网卡信息

[root@xuegod65 ~]# dmesg | grep eth

e1000 0000:02:01.0: eth0: (PCI:66MHz:32-bit) 00:0c:29:bb:2f:a6

e1000 0000:02:01.0: eth0: Intel(R) PRO/1000 Network Connection

e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None

eth0: no IPv6 routers present

## uptime

uptime(选项)

作用：打印系统总共运行了多长时间和系统的平均负载

选项：

❒ -V 显示指令的版本信息。

❒ -p 显示系统运行时间

❒ -h 显示帮助并退出

❒ -s 显示系统启动时间

示例：

[root@localhost ~]# uptime

09:32:15 up 17 min, 2 users, load average: 0.00, 0.02, 0.06

[root@localhost ~]# uptime -s

2016-11-20 09:14:20

[root@xuegod63 ~]# uptime -V

procps version 3.2.8

[root@localhost ~]# uptime -p

up 17 minutes

## Stat

格式：stat [-tfL] [文件]

作用:显示文件的状态信息；

主要参数说明：

❒ -L:支持符号链接

❒ -f:显示文件系统状态；

❒ -t: 以简洁方式输出信息

实例：

[root@hh-02 ~]# stat -f hh

File: "hh"

ID: 9e5bd8f809b7bf23 Namelen: 255 Type: ext2/ext3

Block size: 4096 Fundamental block size: 4096

Blocks: Total: 5283733 Free: 4640542 Available: 4370475

Inodes: Total: 1351680 Free: 1273221

[root@hh-02 ~]# stat -t hh

hh 4096 8 41ed 0 0 803 1049911 2 0 0 1479630401 1479630401 1479630401 4096

## du

格式：du (选项) (文件)

作用：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。

选项：

❒-a 显示目录中个别文件的大小。

❒-b 显示目录或文件大小时，以byte为单位。

❒-c 除了显示个别目录或文件的大小外，同时也显示所有目录或文件的总和。

❒-D 显示指定符号链接的源文件大小。

❒-k 以KB(1024bytes)为单位输出。

❒-l 重复计算硬件链接的文件。

❒-L<符号链接> 显示选项中所指定符号链接的源文件大小。

❒-m 以MB为单位输出。

❒-s 仅显示总计，只列出最后加总的值。

❒-h 以K，M，G为单位，提高信息的可读性。

❒-H 但是K，M，G是以1000为换算单位。

❒-S 显示个别目录的大小时，并不含其子目录的大小。

❒-x 以一开始处理时的文件系统为准，若遇上其它文件系统目录则略过。

❒-X<文件> 在<文件>指定目录或文件。

参数：

文件：指定要查看的文件或目录  
示例：

显示目录或者文件所占空间：

[root@localhost test]# du

4 ./test3

4 ./test1/test5

4 ./test1/test6

24 ./test1

4 ./test2

1400 .

只显示当前目录下面的子目录的目录大小和当前目录的总的大小，最下面的1400为当前目录的总大小

显示指定文件所占空间：

[root@localhost test]# du messages

900 messages

查看指定目录的所占空间：

[root@localhost test]# du test1/

4 test1/test5

4 test1/test6

24 test1/

显示多个文件所占空间：

[root@localhost test]# du messages yum.log

900 messages

4 yum.log

只显示总和的大小:

[root@localhost test]# du -s

1400 .

## df

格式：df (选项) (参数)

作用：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。

选项：

❒-a 包含全部的文件系统；

❒-h 以可读性较高的方式来显示信息；

❒-H 与-h参数相同，但在计算时是以1000 Bytes为换算单位；

❒-i 显示inode的信息；

❒-k 指定区块大小为1024字节；

❒-l 仅显示本地端的文件系统；

❒-m 指定区块大小为1048576字节；

❒-t<文件系统类型> 仅显示指定文件系统类型的磁盘信息；

❒ -T 显示文件系统的类型；

❒-x<文件系统类型> 不要显示指定文件系统类型的磁盘信息；

❒-P 使用POSIX的输出格式；

❒--sync： 在取得磁盘使用信息前，先执行sync指令；

❒--no-sync： 在取得磁盘使用信息前，不要执行sync指令，此为预设值；

❒--block-size= 以指定的区块大小来显示区块数目；

❒--help 显示帮助；

❒--version 显示版本信息。

参数：

文件：指定文件系统上的文件。

示例：

显示磁盘分区上的inode信息：

[root@localhost ~]# df -i

Filesystem Inodes IUsed IFree Use% Mounted on

/dev/sda3 1152816 55656 1097160 5% /

tmpfs 178799 1 178798 1% /dev/shm

/dev/sda1 128016 38 127978 1% /boot

查看全部文件系统：

[root@localhost ~]# df -a

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda3 18121032 1398148 15802360 9% /

proc 0 0 0 - /proc

sysfs 0 0 0 - /sys

devpts 0 0 0 - /dev/pts

tmpfs 715196 0 715196 0% /dev/shm

/dev/sda1 495844 34527 435717 8% /boot

none 0 0 0 - /proc/sys/fs/binfmt\_misc

显示当前磁盘上的分区类型和使用情况：

[root@localhost ~]# df -Th

Filesystem Type Size Used Avail Use% Mounted on

/dev/sda3 ext4 18G 1.4G 16G 9% /

tmpfs tmpfs 699M 0 699M 0% /dev/shm

/dev/sda1 ext4 485M 34M 426M 8% /boot

## top

格式：top (选项)

作用：可以实时动态地查看系统的整体运行情况，是一个综合了多方信息监测系统性能和运行信息的实用工具。

选项：

❒ -m 设置文件夹权限

❒ -b 以批处理模式操作；

❒ -c 显示完整的治命令；

❒ -d 屏幕刷新间隔时间；

❒ -I 忽略失效过程；

❒ -s 保密模式；

❒ -S 累积模式；

❒ -i<时间> 设置间隔时间；

❒ -u<用户名> 指定用户名；

❒ -p<进程号> 指定进程；

❒ -n<次数> 循环显示的次数。

示例：

显示进程信息

[root@localhost ~]# top

top - 14:06:23 up 70 days, 16:44,  2 users,  load average: 1.25, 1.32, 1.35

Tasks: 206 total,   1 running, 205 sleeping,   0 stopped,   0 zombie

Cpu(s):  5.9%us,  3.4%sy,  0.0%ni, 90.4%id,  0.0%wa,  0.0%hi,  0.2%si,  0.0%st

Mem:  32949016k total, 14411180k used, 18537836k free,   169884k buffers

Swap: 32764556k total,        0k used, 32764556k free,  3612636k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND

28894 root      22   0 1501m 405m  10m S 52.2  1.3   2534:16 java

## free

格式：free (选项)

作用：可以显示当前系统未使用的和已使用的内存数目，还可显示被内核使用的内存缓冲区

选项：

❒ -b 以Byte为单位显示内存使用情况；

❒ -k 以KB为单位显示内存使用情况；

❒ -m 以MB为单位显示内存使用情况；

❒ -o 不显示缓冲区调节列；

❒ -s<间隔秒数> 持续观察内存使用状况；

❒ -t 显示内存总和列；

❒ -V 显示版本信息。

示例：

显示内存的使用情况

[root@localhost ~]# free -m

total used free shared buff/cache available

Mem: 755 456 58 3 240 136

Swap: 2047 4 2043

## W

格式：w [-uhsf] [需指定查看的用户名]

作用：显示已经登陆系统的用户列表，并显示用户正在执行的指令

主要参数：

❒ -h：不打印头信息

❒ -u：当显示当前进程和cpu 时间时忽略用户名

❒ -s：使用短输出格式

❒ -f：显示用户从哪登录

实例：[root@hh-02 ~]# w -f root

16:55:55 up 3:13, 2 users, load average: 0.00, 0.00, 0.00

USER TTY LOGIN@ IDLE JCPU PCPU WHAT

root tty1 15:08 1:44m 0.46s 0.25s ssh 192.168.3.66

root pts/0 15:21 0.00s 0.20s 0.00s w -f root

## date

格式：data [-s 字符串] [-u]

作用：显示或设置系统时间与日期

主要参数：

❒ -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

❒ -s<字符串>：根据字符串来设置日期与时间。字符串前后必须加上双引号

❒ -u：显示GMT

实例：[root@hh-02 ~]# date -u

2016年 11月 20日 星期日 08:58:31 UTC

## find

格式：find （选项）（参数）

作用：在指定目录下查找文件

选项：

❒-nam 按照文件名查找文件。 “名称”

❒-perm 按照文件权限来查找文件。666 777 等

❒-prune 使用这一选项可以使find命令不在当前指定的目录中查找(排除) 如果同

时使用-depth选项，那么-prune将被find命令忽略

❒-depth 在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找

❒-user 按照文件属主来查找文件

❒-nouser 查找无有效属主的文件，即该文件的属主在/etc/passwd中不存在

❒-group 按照文件所属的组来查找文件

❒-nogroup 查找无有效所属组的文件，即该文件所属的组在/etc/groups中不存在

❒-mtime -n +n  按照文件的更改时间来查找文件，

- n 表示文件更改时间距现在n天以内

+ n 表示文件更改时间距现在n天以前

❒-type 查找某一类型的文件

b - 块设备文件

d - 目录

c - 字符设备文件

p - 管道文件

l- 符号链接文件

f - 普通文件

❒-size n [c] 查找文件长度为n块的文件，带有c时表示文件长度以字节计

❒-exec 对匹配的文件执行该参数所给出的其他linux命令， 相应命令的形式为'

命令 {} \;，注意{ }和 \；之间的空格，{}代表查到的内容

❒-fstype 查找位于某一类型文件系统中的文件，这些文件系统类型通常可以在配置

文件/etc/fstab中找到，该配置文件中包含了本系统中有关文件系统的息。

❒-mount 在查找文件时不跨越文件系统mount点。

❒-follow 如果find命令遇到符号链接文件，就跟踪至链接所指向的文件。

参数：

起始目录：查找文件的起始目录

示例：

查找root根目录下的所有文件

[root@localhost ~]# find .

.

./.bash\_logout

./.bash\_profile

... ...

./.bash\_history

./.viminfo

./.Xauthority

查找root目录下以txt结尾的所有文件

[root@localhost ~]# find /root -name "\*.txt"

/root/.cache/tracker/db-version.txt

/root/.cache/tracker/db-locale.txt

/root/.cache/tracker/miner-applications-locale.txt

/root/.cache/tracker/last-crawl.txt

/root/.cache/tracker/first-index.txt

/root/1.txt

查找权限是755的文件

[root@localhost ~]# find . -perm 755

./.config

./.config/imsettings

./.config/gnome-session

./.config/gnome-session/saved-session

./.config/dconf

... ...

查找/root目录下所有文件但忽略/root/.config目录

[root@localhost ~]# find /root -path "/root/.config" -prune -o -print

/root

/root/.bash\_logout

/root/.bash\_profile

/root/.bashrc

/root/.cshrc

/root/.tcshrc

/root/anaconda-ks.cfg

/root/.cache

/root/.cache/imsettings

查找/root目录下修改时间在5日以内的文件

[root@localhost ~]# find /root -mtime -5

/root

/root/.cache/abrt

/root/.cache/abrt/lastnotification

/root/.Xauthority

查找/home目录下的所有目录

[root@localhost ~]# find /home -type d

/home

/home/hfx

/home/hfx/.mozilla

/home/hfx/.mozilla/extensions

/home/hfx/.mozilla/plugins

... ...

查找/root目录下块小于1的

[root@localhost ~]# find /root -size -1

/root/.cache/gnome-shell/update-check-3.14

/root/.cache/abrt/applet\_dirlist

/root/.local/share/gnome-settings-daemon/input-sources-converted

/root/.local/share/tracker/data/.meta.isrunning

/root/.local/share/.converted-launchers

/root/test/1

/root/test/2

/root/test/3

## which

格式：which（选项）（参数）

作用：查找并显示给定命令的绝对路径

选项:

❒-n<文件名长度> 制定文件名长度，指定的长度必须大于或等于所有文件中最长的

文件名

❒-p<文件名长度> 与-n参数相同，但此处的<文件名长度>包含了文件的路径；

❒-w 指定输出时栏位的宽度；

❒-V 显示版本信息。

参数：

命令名称列表。

示例：

查找pwd的绝对路径

[root@localhost ~]# which pwd

/usr/bin/pwd

## whereis

格式：whereis （选项）（参数）

作用：用来定位指令的二进制程序、源代码文件和man手册页等相关文件的路径

选项：

❒-b 只查找二进制文件；

❒-B<目录> 只在设置的目录下查找二进制文件；

❒-f 不显示文件名前的路径名称；

❒-m 只搜索 man 手册；

❒-M<目录> 定义 man 手册查找路径；

❒-s 只查找原始代码文件；

❒-S<目录> 定义源代码查找路径；

❒-u 搜索不常见记录；

❒-l 输出有效查找路径；  
参数：

指令名：要查找的二进制程序、源文件和man手册页的指令名。

示例：

查找cat的相关路径

[root@localhost ~]# whereis cat

cat: /usr/bin/cat /usr/share/man/man1/cat.1.gz /usr/share/man/man1p/cat.1p.gz

查找cat的二进制文件

[root@localhost ~]# whereis -b cat

cat: /usr/bin/cat

查找cat的说明文件

[root@localhost ~]# whereis -m cat

cat: /usr/share/man/man1/cat.1.gz /usr/share/man/man1p/cat.1p.gz

## locate

格式：locate（选项）（参数）

作用：在数据库/var/lib/mlocate/mlocate.db中查找文件或目录

选项：

❒-A 只显示匹配所有模式的条目

❒-b 仅匹配路径名的基本名称

❒-c 只显示找到的条目数

❒-d<目录>或--database=<目录> 指定数据库所在的目录；

❒-i 在匹配模式时忽略大小写区分

❒-q 不报告有关读取数据库的错误消息

❒-u 更新slocate数据库；

❒-S 不搜索条目，打印有关每个数据库的统计信息

❒--help 显示帮助；

❒--version 显示版本信息。

参数：

查找字符串：要查找的文件名中含有的字符串。

示例：

查找/root目录下以a开头的文件

[root@localhost ~]# locate ~/a

/root/anaconda-ks.cfg

学神IT-Linux-1609-幽蓝整理

## useradd

格式：useradd (选项) (参数)

作用：建立用户账号

选项：

❒ -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

❒ -e<有效期限> 　指定帐号的有效期限

❒ -f<缓冲天数> 　指定在密码过期后多少天即关闭该帐号

❒ -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

❒ -m 　自动建立用户的登入目录

❒ -M 　不要自动建立用户的登入目录

❒ -n 　取消建立以用户名称为名的群组

❒ -r 　建立系统帐号

❒ -s<shell>　 　指定用户登入后所使用的shell

❒ -u<uid> 　指定用户ID

参数：

用户名：要创建的用户名

示例：

新建用户加入组

[root@localhost ~]# useradd –g sales jack –G company,employees

新建用户并设置ID

[root@localhost ~]# useradd sfit –u 544

## userdel

格式：userdel (选项) (参数)

作用：删除用户，以及与用户相关的文件

选项：

❒ -f：强制删除用户，即使用户当前已登录

❒ -r：删除用户的同时，删除与用户相关的所有文件

参数：

用户名：要删除的用户名

示例：

删除用户sfit

[root@localhost ~]# userdel sfit

删除用户sfit、其家目录及文件

[root@localhost ~]# userdel –r sfit

## passwd

格式：passwd（选项）（参数）

作用：设置用户的认证信息，包括用户密码、密码过期时间等

选项：

❒ -d：删除密码，仅有系统管理者才能使用

❒ -f：强制执行

❒ -k：设置只有在密码过期失效后，方能更新

❒ -l：锁住密码

❒ -s：列出密码的相关信息，仅有系统管理者才能使用

❒ -u：解开已上锁的帐号

参数：

用户名：需要设置密码的用户名

示例：

普通用户更改自己的密码

[sfit@localhost ~]$ passwd

Changing password for user sfit. //更改sfit用户的密码

(current) UNIX password: //请输入当前密码

New UNIX password: //请输入新密码

Retype new UNIX password: //确认新密码

passwd: all authentication tokens updated successfully. //更改成功

Root用户更改指定用户密码

[root@localhost ~]# passwd sfit

Changing password for user sfit. //更改sfit用户的密码

New UNIX password: //请输入新密码

Retype new UNIX password: //再输入一次

passwd: all authentication tokens updated successfully. //成功

## usermod

格式：usermod（选项）（参数）

作用：用于修改用户的基本信息

选项：

❒ -c<备注>：修改用户帐号的备注文字

❒ -d<登入目录>：修改用户登入时的目录

❒ -e<有效期限>：修改帐号的有效期限

❒ -f<缓冲天数>：修改在密码过期后多少天即关闭该帐号

❒ -g<群组>：修改用户所属的群组

❒ -G<群组>：修改用户所属的附加群组

❒ -l<帐号名称>：修改用户帐号名称

❒ -L：锁定用户密码，使密码无效

❒ -s：修改用户登入后所使用的shell

❒ -u：修改用户ID

❒ -U：解除密码锁定

参数：

登录名：指定要修改信息的用户登录名

示例：

将newuser2添加到组staff中

[root@localhost ~]# usermod -G staff newuser2

修改newuser的用户名为newuser1

[root@localhost ~]# usermod -l newuser1 newuser

锁定账号newuser1

[root@localhost ~]# usermod -L newuser1

解除对newuser1的锁定

[root@localhost ~]# usermod -U newuser1

## id

**功能说明：**变更用户身份

**语　　法：**su [-flmp][--help][--version][-][-c <指令>][-s <shell>][用户帐号]

**补充说明：**su可让用户暂时变更登入的身份。变更时须输入所要变更的用户帐号与密码。

**参　　数：**

❒-g或--group 　显示用户所属群组的ID。   
❒-G或--groups 　显示用户所属附加群组的ID。   
❒-n或--name 　显示用户，所属群组或附加群组的名称。   
❒-r或--real 　显示实际ID。   
❒-u或--user 　显示用户ID。   
❒-help 　显示帮助。   
❒-version 　显示版本信息。

实例1：查看当前root用户id信息

[root@xuegod65 ~]# id

uid=0(root) gid=0(root) groups=0(root)

实例2：输入特定用户信息，查看特定用户信息相关的UID和GID，只需在id命令后跟上用户：

[root@xuegod65 ~]# id xuzj

uid=504(xuzj) gid=504(xuzj) groups=504(xuzj)

实例3：输入特定用户信息有效的组ID，通过使用-g选项跟上特定用户

[root@xuegod65 ~]# id -g xuzj

504

## su

**语　　法：**id [-gGnru][--help][--version][用户名称]

**补充说明：**id会显示用户以及所属群组的实际与有效ID。若两个ID相同，则仅显示实际ID。若仅指定用户名称，则显示目前用户的ID。

**参　　数：**

❒-c<指令>或--command=<指令> 　执行完指定的指令后，即恢复原来的身份。   
❒-f或--fast 　适用于csh与tsch，使shell不用去读取启动文件。   
❒-.-l或--login 　改变身份时，也同时变更工作目录，以及HOME,SHELL,USER,LOGNAME。此外，也会变更PATH变量。   
❒-m,-p或--preserve-environment 　变更身份时，不要变更环境变量。   
❒-s<shell>或--shell=<shell> 　指定要执行的shell。   
❒--help 　显示帮助。   
❒--version 　显示版本信息。

[用户帐号] 　指定要变更的用户。若不指定此参数，则预设变更为root。

实例1：通过su – 切换用户

[root@xuegod65 ~]# su - xuzj

[xuzj@xuegod65 ~]$ pwd

/home/xuzj

说明：通过su – 切换用户，显示的是当前的用户环境变量。

实例2：通过su切换用户

[root@xuegod65 ~]# su xuzj

[xuzj@xuegod65 root]$ pwd

/root

说明：通过su 切换，用户是切换了，用户的环境变量还是在root下。

实例3：变更账号为root的使用者执行命令后再恢复原来的用户

[xuzj@xuegod65 /]$ su -c ls root

Password:

bin boot dev etc home lib lib64 lost+found media mnt opt proc root sbin selinux srv sys tmp usr var

[xuzj@xuegod65 /]$

## sudo

语法：sudo(选项)(参数)

作用：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。

选项

* -b：在后台执行指令；
* -h：显示帮助；
* -H：将HOME环境变量设为新身份的HOME环境变量；
* -k：结束密码的有效期限，也就是下次再执行sudo时便需要输入密码；。
* -l：列出目前用户可执行与无法执行的指令；
* -p：改变询问密码的提示符号；
* -s<shell>：执行指定的shell；
* -u<用户>：以指定的用户作为新的身份。若不加上此参数，则预设以root作为新的身份；
* -v：延长密码有效期限5分钟；
* -V ：显示版本信息。

参数

指令：需要运行的指令和对应的参数。

示例

配置sudo必须通过编辑/etc/sudoers文件，而且只有超级用户才可以修改它，还必须使用visudo编辑。之所以使用visudo有两个原因，一是它能够防止两个用户同时修改它；二是它也能进行有限的语法检查。所以，即使只有你一个超级用户，你也最好用visudo来检查一下语法。

visudo默认的是在[vi](http://man.linuxde.net/vi)m里打开配置文件，用vim来修改文件。我们可以在编译时修改这个默认项。visudo不会擅自保存带有语法错误的配置文件，它会提示你出现的问题，并询问该如何处理，就像：

>>> sudoers [file](http://man.linuxde.net/file): syntax error, line 22 <<

此时我们有三种选择：键入“e”是重新编辑，键入“x”是不保存退出，键入“Q”是退出并保存。如果真选择Q，那么sudo将不会再运行，直到错误被纠正。

现在，我们一起来看一下神秘的配置文件，学一下如何编写它。让我们从一个简单的例子开始：让用户Foobar可以通过sudo执行所有root可执行的命令。以root身份用visudo打开配置文件，可以看到类似下面几行：

[root@localhost ~]# Runas [alias](http://man.linuxde.net/alias) specification

[root@localhost ~]# User privilege specificationroot ALL=(ALL)ALL

我们一看就明白个差不多了，root有所有权限，只要仿照现有root的例子就行，我们在下面加一行（最好用tab作为空白）：

foobar ALL=(ALL) ALL

保存退出后，切换到foobar用户，我们用它的身份执行命令：

[foobar@localhost ~]$ [ls](http://man.linuxde.net/ls) /root

ls: /root: 权限不够

[foobar@localhost ~]$ sudo ls /root

PassWord:

anaconda-ks.cfg Desktop [install](http://man.linuxde.net/install).log install.log.syslog

好了，我们限制一下foobar的权利，不让他为所欲为。比如我们只想让他像root那样使用ls和[ifconfig](http://man.linuxde.net/ifconfig)，把那一行改为：

foobar localhost= /sbin/ifconfig, /bin/ls

再来执行命令：

[foobar@localhost ~]$ sudo [head](http://man.linuxde.net/head) -5 /etc/shadow

Password:

Sorry, user foobar is not allowed to execute '/usr/bin/head -5 /etc/shadow' [as](http://man.linuxde.net/as) root on localhost.localdomain.

[foobar@localhost ~]$ sudo /sbin/ifconfigeth0 Linkencap:Ethernet HWaddr 00:14:85:EC:E9:9B...

现在让我们来看一下那三个ALL到底是什么意思。第一个ALL是指网络中的主机，我们后面把它改成了主机名，它指明foobar可以在此主机上执行后面的命令。第二个括号里的ALL是指目标用户，也就是以谁的身份去执行命令。最后一个ALL当然就是指命令名了。例如，我们想让foobar用户在linux主机上以jimmy或rene的身份执行[kill](http://man.linuxde.net/kill)命令，这样编写配置文件：

foobar linux=(jimmy,rene) /bin/kill

但这还有个问题，foobar到底以jimmy还是rene的身份执行？这时我们应该想到了sudo -u了，它正是用在这种时候。 foobar可以使用sudo -u jimmy kill PID或者sudo -u rene kill PID，但这样挺麻烦，其实我们可以不必每次加-u，把rene或jimmy设为默认的目标用户即可。再在上面加一行：

Defaults:foobar runas\_default=rene

Defaults后面如果有冒号，是对后面用户的默认，如果没有，则是对所有用户的默认。就像配置文件中自带的一行：

Defaults env\_reset

另一个问题是，很多时候，我们本来就登录了，每次使用sudo还要输入密码就显得烦琐了。我们可不可以不再输入密码呢？当然可以，我们这样修改配置文件：

foobar localhost=NOPASSWD: /bin/[cat](http://man.linuxde.net/cat), /bin/ls

再来sudo一下：

[foobar@localhost ~]$ sudo ls /rootanaconda-ks.cfg Desktop install.log

install.log.syslog

当然，你也可以说“某些命令用户foobar不可以运行”，通过使用!操作符，但这不是一个好主意。因为，用!操作符来从ALL中“剔出”一些命令一般是没什么效果的，一个用户完全可以把那个命令拷贝到别的地方，换一个名字后再来运行。

日志与安全

sudo为安全考虑得很周到，不仅可以记录日志，还能在有必要时向系统管理员报告。但是，sudo的日志功能不是自动的，必须由管理员开启。这样来做：

[root@localhost ~]# [touch](http://man.linuxde.net/touch) /var/log/sudo

[root@localhost ~]# vi m/etc/syslog.conf

在syslog.conf最后面加一行（必须用tab分割开）并保存：

local2.debug /var/log/sudo

重启日志守候进程，

[ps](http://man.linuxde.net/ps) aux [grep](http://man.linuxde.net/grep) syslogd

把得到的syslogd进程的PID（输出的第二列是PID）填入下面：

kill –HUP PID

这样，sudo就可以写日志了：

[foobar@localhost ~]$ sudo ls /rootanaconda-ks.cfg

Desktop install.log

install.log.syslog

$cat /var/log/sudoJul 28 22:52:54 localhost sudo: foobar :

TTY=pts/1 ; [pwd](http://man.linuxde.net/pwd)=/home/foobar ; USER=root ; [command](http://man.linuxde.net/command)=/bin/ls /root

不过，有一个小小的“缺陷”，sudo记录日志并不是很忠实：

[foobar@localhost ~]$ sudo cat /etc/shadow > /dev/null

cat /var/log/sudo...Jul 28 23:10:24 localhost sudo: foobar : TTY=pts/1 ;

PWD=/home/foobar ; USER=root ; COMMAND=/bin/cat /etc/shadow

重定向没有被记录在案！为什么？因为在命令运行之前，shell把重定向的工作做完了，sudo根本就没看到重定向。这也有个好处，下面的手段不会得逞：

[foobar@localhost ~]$ sudo ls /root > /etc/shadowbash: /etc/shadow: 权限不够

sudo 有自己的方式来保护安全。以root的身份执行sudo-V，查看一下sudo的设置。因为考虑到安全问题，一部分环境变量并没有传递给sudo后面的命令，或者被检查后再传递的，比如：PATH，HOME，SHELL等。当然，你也可以通过sudoers来配置这些环境变量。

## groupadd

1> 作用  
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

2> 格式

groupadd [-grKo] [用户组名]

主要参数说明

-g：指定新建工作组的id

-r：创建系统工作组，系统工作组的组ID小于500

-K：覆盖配置文件“/ect/login.defs”

-o：允许添加组ID号不唯一的工作组

## telnet

作用  
登录远程主机，对远程主机进行管理

格式

telnet [远程主机]

主要参数说明

-8：允许使用8位字符资料，包括输入与输出

-a：尝试自动登入远端系统

-b<主机别名>：使用别名指定远端主机名称

-c：不读取用户专属目录里的.telnetrc文件

-d：启动排错模式

-e<脱离字符>：设置脱离字符

-E：滤除脱离字符

-f：此参数的效果和指定"-F"参数相同

-F：使用Kerberos V5认证时，加上此参数可把本地主机的认证数据上传到远端主机

-k<域名>：使用Kerberos认证时，加上此参数让远端主机采用指定的领域名，而非该主机的域名

-K：不自动登入远端主机

-l<用户名称>：指定要登入远端主机的用户名称

-L：允许输出8位字符资料

-n<记录文件>：指定文件记录相关信息

-r：使用类似rlogin指令的用户界面

-S<服务类型>：设置telnet连线所需的ip TOS信息

-x：假设主机有支持数据加密的功能，就使用它

-X<认证形态>：关闭指定的认证形态

## ssh

作用：openssh套件中的客户端连接工具，可以给予ssh加密协议实现安全的远程登录服务器

格式

ssh [-4] [需要连接的远程ssh服务器]

主要参数说明

-1：强制使用ssh协议版本1

-2：强制使用ssh协议版本2

-4：强制使用IPv4地址

-6：强制使用IPv6地址

-A：开启认证代理连接转发功能

-a：关闭认证代理连接转发功能

-b：使用本机指定地址作为对应连接的源ip地址

-C：请求压缩所有数据

-F：指定ssh指令的配置文件

-f：后台执行ssh指令

-g：允许远程主机连接主机的转发端口

-i：指定身份文件

-l：指定连接远程服务器登录用户名

-N：不执行远程指令

-o：指定配置选项

-p：指定远程服务器上的端口

-q：静默模式

-X：开启X11转发功能

-x：关闭X11转发功能

-y：开启信任X11转发功能

## wget

格式：wget (选项) (参数)

作用：从网络上自动下载文件

选项：

1.启动:

❒ -V, --version 显示wget的版本后退出

❒ -h, --help 打印语法帮助

❒ -b, --background 启动后转入后台执行

❒ -e, --execute=COMMAND 执行`.wgetrc'格式的命令，wgetrc格式参见/etc/wgetrc或~/.wgetrc

wget默认会根据网站的robots.txt进行操作

使用-e robots=off参数即可绕过该限制

2.记录和输入文件:

❒ -o, --output-file=FILE 把记录写到FILE文件中

❒ -a, --append-output=FILE 把记录追加到FILE文件中

❒ -d, --debug 打印调试输出

❒ -q, --quiet 安静模式(没有输出)

❒ -v, --verbose 冗长模式(这是缺省设置)

❒ -nv, --non-verbose 关掉冗长模式，但不是安静模式

❒ -i, --input-file=FILE 下载在FILE文件中出现的URLs

❒ -F, --force-html 把输入文件当作HTML格式文件对待

❒ -B, --base=URL 将URL作为在-F -i参数指定的文件中出现的相对链接的前缀

❒ --sslcertfile=FILE 可选客户端证书

❒ --sslcertkey=KEYFILE 可选客户端证书的KEYFILE

❒ --egd-file=FILE 指定EGD socket的文件名

3.下载:

❒ --bind-address=ADDRESS 指定本地使用地址([主机名](http://baike.so.com/doc/6961799-7184310.html)或IP，当本地有多个IP或名字时使用)

❒ -t, --tries=NUMBER 设定最大尝试链接次数(0 表示无限制).

❒ -O --output-document=FILE 把文档写到FILE文件中

❒ -nc, --no-clobber 不要覆盖存在的文件或使用.#前缀

❒ -c, --continue 接着下载没下载完的文件

❒ --progress=TYPE 设定进程条标记

❒ -N, --timestamping 不要重新下载文件除非比本地文件新

❒ -S, --server-response[打印服务器](http://baike.so.com/doc/1455451-1538719.html)的回应

❒ --spider 不下载任何东西

❒ -T, --timeout=SECONDS 设定响应超时的秒数

❒ -w, --wait=SECONDS 两次尝试之间间隔SECONDS秒

❒ --waitretry=SECONDS 在重新链接之间等待1...SECONDS秒

❒ --random-wait 在下载之间等待0...2\*WAIT秒

❒ -Y, --proxy=on/off 打开或关闭代理

❒ -Q, --quota=NUMBER 设置下载的容量限制

❒ --limit-rate=RATE 限定下载输率

4.目录:

❒ -nd --no-directories 不创建目录

❒ -x, --force-directories 强制创建目录

❒ -nH, --no-host-directories 不创建主机目录

❒ -P, --directory-prefix=PREFIX 将文件保存到目录 PREFIX/...

❒ --cut-dirs=NUMBER 忽略 NUMBER层远程目录

5.HTTP 选项:

❒ --http-user=USER 设定HTTP用户名为 USER.

❒ --http-passwd=PASS 设定http密码为 PASS.

❒ -C, --cache=on/off 允许/不允许服务器端的[数据缓存](http://baike.so.com/doc/7049892-7272798.html)(一般情况下允许).

❒ -E, --html-extension 将所有text/html文档以.html扩展名保存

❒ --ignore-length 忽略 `Content-Length'头域

❒ --header=STRING 在headers中插入字符串 STRING

❒ --proxy-user=USER 设定代理的用户名为 USER

❒ --proxy-passwd=PASS 设定代理的密码为 PASS

❒ --referer=URL 在HTTP请求中包含 `Referer: URL'头

❒ -s, --save-headers 保存HTTP头到文件

❒ -U, --user-agent=AGENT 设定代理的名称为 AGENT而不是 Wget/VERSION.

❒ --no-http-keep-alive 关闭 HTTP活动链接 (永远链接).

❒ --cookies=off 不使用 cookies.

❒ --load-cookies=FILE 在开始会话前从文件 FILE中加载cookie

❒ --save-cookies=FILE 在会话结束后将 cookies保存到 FILE文件中

6.FTP 选项:

❒ -nr, --dont-remove-listing 不移走 `.listing'文件

❒ -g, --glob=on/off 打开或关闭文件名的 globbing机制

❒ --passive-ftp 使用被动传输模式 ([缺省值](http://baike.so.com/doc/329248-348712.html)).

❒ --active-ftp 使用主动传输模式

❒ --retr-symlinks 在[递归](http://baike.so.com/doc/5725016-5937752.html)的时候，将链接指向文件(而不是目录)

7.递归下载:

❒ -r, --recursive 递归下载--慎用!

❒ -l, --level=NUMBER 最大递归深度 (inf 或 0 代表无穷).

❒ --delete-after 在完毕后局部[删除文件](http://baike.so.com/doc/6970762-7193448.html)

❒ -k, --convert-links 转换非相对链接为相对链接

❒ -K, --backup-converted 在转换文件X之前，将之备份为 X.orig

❒ -m, --mirror 等价于 -r -N -l inf -nr.

❒ -p, --page-requisites 下载显示HTML文件的所有图片

## Ping

格式：ping (选项) (参数)

作用：检测主机

❒ -d：使用Socket的SO\_DEBUG功能；

❒ -c<完成次数>：设置完成要求回应的次数；

❒ -f：极限检测； -i<间隔秒数>：指定收发信息的间隔时间；

❒ -I<网络界面>：使用指定的网络界面送出数据包；

❒ -l<前置载入>：设置在送出要求信息之前，先行发出的数据包；

❒ -n：只输出数值；

❒ -p<范本样式>：设置填满数据包的范本样式；

❒ -q：不显示指令执行过程，开头和结尾的相关信息除外；

❒ -r：忽略普通的Routing Table，直接将数据包送到远端主机上；

❒ -R：记录路由过程；

❒ -s<数据包大小>：设置数据包的大小；

❒ -t<存活数值>：设置存活数值TTL的大小；

❒ -v：详细显示指令的执行过程。

举例：  
[root@xuegod63 ~]# ping -c 3 www.baidu.com

PING www.a.shifen.com (119.75.218.70) 56(84) bytes of data.

64 bytes from 119.75.218.70: icmp\_seq=1 ttl=51 time=24.2 ms

64 bytes from 119.75.218.70: icmp\_seq=2 ttl=51 time=23.3 ms

64 bytes from 119.75.218.70: icmp\_seq=3 ttl=51 time=24.1 ms

--- www.a.shifen.com ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2003ms

## Route

格式：route (选项) (参数)

作用：显示并设置linux内核中的网络路由表

选项：

❒ -A：设置地址类型；

❒ -C：打印将Linux核心的路由缓存；

❒ -v：详细信息模式；

❒ -n：不执行DNS反向查找，直接显示数字形式的IP地址；

❒ -e：netstat格式显示路由表；

❒ -net：到一个网络的路由表；

❒ -host：到一个主机的路由表。  
参数：

❒ Add：增加指定的路由记录；

❒ Del：删除指定的路由记录；

❒ Target：目的网络或目的主机；

❒ gw：设置默认网关；

❒ mss：设置TCP的最大区块长度（MSS），单位MB；

❒ window：指定通过路由表的TCP连接的TCP窗口大小；

❒ dev：路由记录所表示的网络接口。

举例：

[root@xuegod63 ~]# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

default 192.168.1.1 0.0.0.0 UG 100 0 0 eno16777736

192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 eno16777736

192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0

[root@xuegod63 ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 192.168.1.1 0.0.0.0 UG 100 0 0 eno16777736

192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 eno16777736

192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0

其中Flags为路由标志，标记当前网络节点的状态，Flags标志说明：

❒ U Up表示此路由当前为启动状态。

❒ H Host，表示此网关为一主机。

❒ G Gateway，表示此网关为一路由器。

❒ R Reinstate Route，使用动态路由重新初始化的路由。

❒ D Dynamically,此路由是动态性地写入。

❒ M Modified，此路由是由路由守护程序或导向器动态修改。

❒! 表示此路由当前为关闭状态。  
添加网关/设置网关  
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0 #增加一条到达244.0.0.0的路由。  
屏蔽一条路由：  
route add -net 224.0.0.0 netmask 240.0.0.0 reject #增加一条屏蔽的路由，目的地址为224.x.x.x将被拒绝。  
删除路由记录：  
route del -net 224.0.0.0 netmask 240.0.0.0

route del -net 224.0.0.0 netmask 240.0.0.0 reject  
删除和添加设置默认网关：  
route del default gw 192.168.120.240

route add default gw 192.168.120.240

## ifconfig

格式：ifconfig (网络设备) (选项)

作用：命令用来查看和配置网络设备。

选项：

❒ up 启动指定网络设备/网卡

❒ down 关闭指定网络设备/网卡

❒ arp 设置指定网卡是否支持ARP协议

❒ -promisc 设置是否支持网卡的promiscuous模式

❒ -allmulti 设置是否支持多播模式

❒ -s 显示摘要信息（类似于netstat –i）

❒add 给指定网卡配置IPv6地址

❒del 删除指定网卡配置IPv6地址

❒<硬件地址> 配置网卡最大的传输单元  
❒mtu<字节数> 设置网卡的最大传输单元 (bytes)  
❒netmask<子网掩码> 设置网卡的子网掩码  
❒tunel 建立隧道  
❒dstaddr 设定一个远端地址，建立点对点通信  
❒-broadcast<地址> 为指定网卡设置广播协议  
❒-pointtopoint<地址> 为网卡设置点对点通讯协议  
❒multicast 为网卡设置组播标志  
❒为网卡设置IPv4地址  
❒txqueuelen<长度> 为网卡设置传输列队的长度

[root@localhost~]#ifconfig#处于激活状态的网络接口

[root@localhost~]#ifconfig-a#所有配置的网络接口，不论其是否激活

[root@localhost~]#ifconfig eth0#显示eth0的网卡信息

[root@localhost~]#ifconfig eth0 down#关闭eth0网卡

[root@localhost~]#ifconfig eth0 add 33ffe:3240:800:1005::2/64#为网卡添加IPv6地址

[root@localhost~]#ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE#修改MAC地址

[root@localhost~]#ifconfig eth0 192.168.1.56#给eth0网卡配置IP地址

[root@localhost~]#ifconfig eth0 192.168.1.56 netmask 255.255.255.0 broadcast 192.168.1.255#给eth0网卡配置IP地址,加上子掩码,加上个广播地址

[root@localhost~]#ifconfig eth0 mtu 1500#设置能通过的最大数据包大小为1500 bytes

[root@localhost~]#ifconfig eth0 arp#开启arp功能

[root@localhost~]#ifconfig eth0-arp#关闭arp功能

## ifup

格式：ifup (interface)

作用：启动网络接口

示例：

启动eth0网络接口

[root@localhost ~]# ifup eth0

## ifdown

格式：ifdown (interface)

作用：关闭网络接口

示例：

启动eth0网络接口

[root@localhost ~]# ifdown eth0

## netstat

格式：netstat (选项)

作用：用来打印Linux中网络系统的状态信息

选项：

❒a或--all：显示所有连线中的Socket；

❒-A<网络类型>或--<网络类型>：列出该网络类型连线中的相关地址；

❒-c或--continuous：持续列出网络状态；

❒-C或--cache：显示路由器配置的快取信息；

❒-e或--extend：显示网络其他相关信息；

❒-F或--fib：显示FIB；

❒-g或--groups：显示多重广播功能群组组员名单；

❒-i或--interfaces：显示网络界面信息表单；

❒-l或--listening：显示监控中的服务器的Socket；

❒-M或--masquerade：显示伪装的网络连线；

❒-n或--numeric：直接使用ip地址，而不通过域名服务器；

❒-N或--netlink或--symbolic：显示网络硬件外围设备的符号连接名称；

❒-o或--timers：显示计时器；

❒-p或--programs：显示正在使用Socket的程序识别码和程序名称；

❒-r或--route：显示Routing Table；

❒-s或--statistice：显示网络工作信息统计表；

❒-t或--tcp：显示TCP传输协议的连线状况；

❒-u或--udp：显示UDP传输协议的连线状况；

❒-v或--verbose：显示指令执行过程；

❒-x或--unix：此参数的效果和指定"-A unix"参数相同；

❒--ip或--inet：此参数的效果和指定"-A inet"参数相同。

列出所有端口 (包括监听和未监听的)

netstat -a #列出所有端口

netstat -at #列出所有tcp端口

netstat -au #列出所有udp端口

列出所有处于监听状态的 Sockets

netstat -l #只显示监听端口

netstat -lt #只列出所有监听 tcp 端口

netstat -lu #只列出所有监听 udp 端口

netstat -lx #只列出所有监听 UNIX 端口

显示每个协议的统计信息

netstat -s 显示所有端口的统计信息

netstat -st 显示TCP端口的统计信息

netstat -su 显示UDP端口的统计信息

在netstat输出中显示 PID 和进程名称

netstat -pt

netstat -p可以与其它开关一起使用，就可以添加“PID/进程名称”到netstat输出中，这样debugging的时候可以很方便的发现特定端口运行的程序。

在netstat输出中不显示主机，端口和用户名(host, port or user)

当你不想让主机，端口和用户名显示，使用netstat -n。

netstat -an

如果只是不想让这三个名称中的一个被显示，使用以下命令:

netsat -a --numeric-ports

netsat -a --numeric-hosts

netsat -a --numeric-users

持续输出netstat信息

netstat -c #每隔一秒输出网络信息

显示系统不支持的地址族(Address Families)

netstat --verbose

在输出的末尾，会有如下的信息：

netstat: no support for `AF IPX' on this system.

显示核心路由信息

netstat -r

使用netstat -rn显示数字格式，不查询主机名称。

找出程序运行的端口

并不是所有的进程都能找到，没有权限的会不显示，使用 root 权限查看所有的信息。

netstat -ap | grep 80

找出运行在指定端口的进程：

netstat -an | grep ':80'

显示网络接口列表

netstat -i

## mutt

格式：mutt (选项) (参数)

作用：linux下的一个email程序

选项：

❒ alternates

这个变量应该设置为一个[正则表达式](http://baike.so.com/doc/4934015-5154299.html)，它是你有可能收到信件的地址。

这个变量有助于 Mutt 显示邮件的地址特征。一般 Mutt 会在index中显示几种不同的邮件地址特征，它们是由 $to\_chars 变量指定的。

❒ to\_chars

一般 $to\_chars 的定义为 " +TCFL".

\* " "(空): 表示这封邮件不是给你的，也就是说 To: 和 Cc: Bcc: 都没有你的地址，很多转发的邮件，未确认的[邮件列表](http://baike.so.com/doc/5567785-5782934.html)邮件，垃圾邮件都有这个特征。

\* +: 表示你是收件人(To: 是你的地址之一)，而且是唯一的收件人。

\* T: 表示你是收件人(To: 包括了你的地址之一)，但是你不是唯一的收件人。这是一封群体信件。

\* C: 表示你的地址出现在 CC:，但是你不是唯一的被抄送的人。

\* F: 表示这封邮件是你发出去的。

\* L: 表示这是一封你已经加入的[邮件列表](http://baike.so.com/doc/5567785-5782934.html)寄来的。

❒ index\_format

这是一个格式字符串，用来控制你的index的列表显示。它的缺省定义是:

set index\_format="%4C %Z %{%b %d} %-15.15L (%4l) %s"

显示出来就是这个样子:

index-original

这里有一个简单的对应关系，%4C表示4位数的序号，%Z是邮件状态，你看到那个 "r" 了吗?也就是说我已经回了这封信。%Z 还包含那个 "T", 也就是说这封信不只是寄给我一个人的，我爸爸每次寄信都会CC他自己一份 :) %{%b %d} 是日期格式，它是由 strftime() 函数的格式定义的。这里 %b 就是根据当前的 locale 设置的月份简称，"12月"。%d 就是十进制表示的每月的日号，"31"。后面的……你自己看看manual吧，我不啰嗦了。

❒ mailboxes

指定你有那些信箱文件。当你按 "c" 切换信箱时，再按 Tab 键，这些信箱就可供你选择。

mailboxes Mailbox sent-mail

❒ header

这是一个 bool 型变量。它表明你在回信时引用原文是否加入原文的邮件头。

set header=no

❒ quit

mutt-quit

可以设置为 yes, no, ask-yes, 或者 ask-no. 这是说，当你按q退出时，是否提示你(ask-yes,ask-no)，还是直接就退出了(yes)，还是根本不理你(no)。

set quit=ask-yes

❒ auto\_view

那些类型的附件是允许直接通过 .mailcap 浏览的?

auto\_view text/html

❒ move

这也是一个提示性变量。它是确定当你退出时，是否提示你把信件从 spool 移动到的 mbox 文件。

set move=no

❒ ascii\_chars

当 Mutt 用 thread 方式显示时，是否用纯 ascii 表示树状列表。

set ascii\_chars=yes

如果设置了就是这个样子:

mutt-ascii

❒ include

回信时是否包含原文。

set include

❒ indent\_str

回信的引文之前插入那个符号?

set indent\_str="> "

❒ my\_hdr

设置你自己的邮件头。

❒ 打分

新信件+4分，爸爸(w2r007)发来的 +2，主题包含"通知"的+2，主题包含 "Circulation" +3, 已经标记删除的 -5，上次没有读的 +1，包含 "believe"的 -10(垃圾广告!)。

score "~N" +4

score "~f w2r007" +2

score "~s 通知" +2

score "~s Circulation" +3

score "~D" -5

score "~O" +1

score "~s believe" -10

❒ sort

排序方式。可以是

date

date-sent

date-received

from

mailbox-order (unsorted)

score

size

subject

threads

to

几种方式，每种方式可以在前面加一个 "reverse-" 前缀表示反方向排序。我一般按分数排序:

set sort=score

❒ sort\_aux

当用 thread 排序方式时，我们对各个 thread 的相对排序顺序。

set sort\_aux=date

❒ pager\_stop

如果设置，那么你在pager(就是你看信的内容的窗口里)翻页时，如果翻到最后，再按往下翻也不会翻到下一封信件。如果不设置就会自动翻到下一封信。

set pager\_stop

❒ fast\_reply

如果设置，当你按 "r" 回信时，就不会再提示你输入回信地址和主题，直接进入编辑模式。

set fast\_reply

❒ resolve

当你按 "t" 或者 "D" 之类的标记操作时，是否自动把光标移动到下一封信件。

set resolve=yes

❒ alias\_file

当你在 index 里按 "a" 为来信者取别名时，使用哪一个别名文件保存这个别名。

set alias\_file=/home/wy/.mutt.alias

❒ record

你发出的邮件保存到那个信箱文件?比如可以像我这样每个月发出的信件放在不同的文件里。

set record="~/Mail/=sent-mail-`date +%Y-%m`"

❒ charset

你的终端支持哪一种编码的显示?这个必须和你的终端编码一样。

set charset="gb2312"

❒ send\_charset

发信时可以使用的字符集。只有当前面的字符集不能正确表示信件内容时才会使用后面的。比如，如果你像下面这样设置，你的信件如果全是英文，那么信件就会被设置为 us-ascii 编码，如果出现了法语字符，那么就会用 iso-8859-1，如果出现了汉字就会用 gb2312，如果以上都不是，那么就用 utf-8。

set send\_charset="us-ascii:iso-8859-1:gb2312:utf-8"

❒ wait\_key

当外部程序退出时，是否要求用户按一个键才返回。这在察看某些shell命令输出时是比要的，否则它们一下就消失了。

set wait\_key=yes

❒ confirmappend

当你要把信件存到另一个信箱，而这个信箱已经存在时，是否提示附加?

set noconfirmappend

❒ edit\_headers

是否把邮件头也放在编辑器里可以修改?

set edit\_headers=no

❒ pager\_index\_lines

当你在用 pager 看信件时，在 index 留出多少行显示[邮件列表](http://baike.so.com/doc/5567785-5782934.html)?

set pager\_index\_lines=4

比如我留下4行[显示列表](http://baike.so.com/doc/2344474-2479380.html)，其它的用来显示正文: mutt-indexpager

❒ subscribe

告诉 Mutt 你已经订阅了那些[邮件列表](http://baike.so.com/doc/5567785-5782934.html)(mailing-list). 这有助于 Mutt 判断那些信件是邮件列表转过来的，避免你错误的回复到别人的私人信箱。也避免别人回复到你的私人信箱

## Nslookup

格式：nslookup (选项) (参数)

作用：常用域名查询工具

选项：

-sil：不显示任何警告信息。

参数

域名：指定要查询的域名

举例：

[root@xuegod63 ~]# nslookup wwww.baidu.com

Server: 211.161.191.230

Address: 211.161.191.230#53

Non-authoritative answer:

wwww.baidu.com canonical name = ps\_other.a.shifen.com.

Name: ps\_other.a.shifen.com

Address: 123.125.114.144

## Dig

格式：dig (选项) (参数)

作用：域名查询

选项：

❒ @<服务器地址>：指定进行域名解析的域名服务器；

❒ -b：当主机具有多个IP地址，指定使用本机的哪个IP地址向域名服务器发送域名查询请求；

❒ -f<文件名称>：指定dig以批处理的方式运行，指定的文件中保存着需要批处理查询的DNS任务信息；

❒ -P：指定域名服务器所使用端口号；

❒ -t<类型>：指定要查询的DNS数据类型；

❒ -x：执行逆向域名查询；

❒ -4：使用IPv4；

❒ -6：使用IPv6；

❒ -h：显示指令帮助信息。

参数

主机：指定要查询域名主机；

查询类型：指定DNS查询的类型；

查询类：指定查询DNS的class；

查询选项：指定查询选项。

举例：

[root@xuegod63 ~]# dig www.baidu.com

; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7 <<>> www.baidu.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43495

;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;www.baidu.com. IN A

;; ANSWER SECTION:

www.baidu.com. 463 IN CNAME www.a.shifen.com.

www.a.shifen.com. 127 IN A 119.75.218.70

www.a.shifen.com. 127 IN A 119.75.217.109

;; Query time: 4 msec

;; SERVER: 211.161.191.230#53(211.161.191.230)

;; WHEN: Thu Nov 24 20:13:24 CST 2016

;; MSG SIZE rcvd: 101

## mount

格式：mount (选项) (参数)

作用：用于加载文件系统到指定的加载点

选项：

❒ -V 显示程序版本；

❒ -l 显示已加载的文件系统列表；

❒ -h 显示帮助信息并退出；

❒ -v 冗长模式，输出指令执行的详细信息；

❒ -n 加载没有写入文件“/etc/mtab”中的文件系统；

❒ -r 将文件系统加载为只读模式；

❒ -a 加载文件“/etc/fstab”中描述的所有文件系统；

❒ -t 设置挂载类型，默认为auto

参数：

设备文件名：指定要加载的文件系统对应的设备名；

加载点：指定加载点目录。

示例：

将光驱挂载到/mnt/cdrom目录下

[root@localhost ~]# mkdir /mnt/cdrom

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

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

[root@localhost ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 18G 1.4G 16G 9% /

tmpfs 699M 0 699M 0% /dev/shm

/dev/sda1 485M 34M 426M 8% /boot

/dev/sr0 3.6G 3.6G 0 100% /mnt/cdrom

## umount

格式：umount (选项) (参数)

作用：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。

选项：

❒-a 卸除/etc/mtab中记录的所有文件系统；

❒-h 显示帮助；

❒-n 卸除时不要将信息存入/etc/mtab文件中；

❒-r 若无法成功卸除，则尝试以只读的方式重新挂入文件系统；

❒-t 仅卸除选项中所指定的文件系统；

❒-v 执行时显示详细的信息；

❒-V 显示版本信息。

参数：

文件系统：指定要卸载的文件系统或者其对应的设备文件名。  
示例：

卸载刚才挂载的光驱

[root@localhost ~]# umount /mnt/cdrom

[root@localhost ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 18G 1.4G 16G 9% /

tmpfs 699M 0 699M 0% /dev/shm

/dev/sda1 485M 34M 426M 8% /boot

## Df

格式：df [-a –h ] 文件

作用：显示磁盘分区上的可使用的磁盘空间

主要参数：

❒ -a 或--all：包含全部的文件系统

❒ --block-size=<区块大小>：以指定的区块大小来显示区块数目

❒ -h 或--human-readable：以可读性较高的方式来显示信息

❒ -H 或--si：与-h 参数相同，但在计算时是以1000 Bytes 为换算单位而非1024 Bytes

❒ -i 或--inodes：显示inode 的信息

❒ -k 或--kilobytes：指定区块大小为1024 字节

❒ -l 或--local：仅显示本地端的文件系统

❒ -m 或--megabytes：指定区块大小为1048576 字节

❒ --no-sync：在取得磁盘使用信息前，不要执行sync 指令，此为预设值

❒ -P 或--portability：使用POSIX 的输出格式

❒ --sync：在取得磁盘使用信息前，先执行sync 指令；

❒ -t<文件系统类型>或--type=<文件系统类型>：仅显示指定文件系统类型的磁盘信息

❒ -T 或--print-type：显示文件系统的类型

❒ -x<文件系统类型>或--exclude-type=<文件系统类型>：不要显示指定文件系统类型的磁盘信息

实例：

[root@hh-02 ~]# df -Th

Filesystem Type Size Used Avail Use% Mounted on

/dev/sda3 ext4 21G 2.5G 17G 13% /

tmpfs tmpfs 1.9G 0 1.9G 0% /dev/shm

/dev/sda1 ext4 477M 66M 386M 15% /boot

[root@hh-02 ~]# df

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda3 21134932 2572772 17481892 13% /

tmpfs 1954260 0 1954260 0% /dev/shm

/dev/sda1 487652 67479 394573 15% /boot

## du

格式：du [-a] [-b] [文件]

作用：du命令用来查看使用空间；

参数说明：

❒ -a 或-all 显示目录中个别文件的大小

❒ -b 或-bytes 显示目录或文件大小时，以byte 为单位

❒ -c 或--total 除了显示个别目录或文件的大小外，同时也显示所有目录或文件的总和

❒ -k 或--kilobytes 以KB(1024bytes)为单位输出

❒ -m 或--megabytes 以MB 为单位输出

❒ -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

❒ -x 或--one-file-xystem 以一开始处理时的文件系统为准，若遇上其它不同的文件系统目录则略过

❒ -L<符号链接>或--dereference<符号链接> 显示选项中所指定符号链接的源文件大小

❒ -S 或--separate-dirs 显示个别目录的大小时，并不含其子目录的大小

❒ -X<文件>或--exclude-from=<文件> 在<文件>指定目录或文件

❒ --exclude=<目录或文件> 略过指定的目录或文件

❒ -D 或--dereference-args 显示指定符号链接的源文件大小

❒ -H 或--si 与-h 参数相同，但是K，M，G 是以1000 为换算单位

❒ -l 或--count-links 重复计算硬件链接的文件。

参数说明：Du 命令与df 命令相似，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看

实例：

[root@hh-02 ~]# du -a

4 ./.bashrc

4 ./.cshrc

4 ./.Xauthority

4 ./.ssh/id\_rsa

4 ./.ssh/id\_rsa.pub

4 ./.ssh/known\_hosts

16 ./.ssh

4 ./anaconda-ks.cfg

12 ./install.log.syslog

4 ./.bash\_logout

4 ./.bash\_profile

4 ./hh

40 ./install.log

4 ./.tcshrc

4 ./.bash\_history

108 .

## fsck

格式：fsck (选项) (参数)

作用：用于检查并且试图修复文件系统中的错误。当文件系统发生错误时，可用fsck指令尝试加以修复。

选项：

❒-a 自动修复文件系统，不询问任何问题；

❒-A 依照/etc/fstab配置文件的内容，检查文件内所列的全部文件系统；

❒-N 不执行指令，仅列出实际执行会进行的动作；

❒-P 当搭配"-A"参数使用时，则会同时检查所有的文件系统；

❒-r 采用互动模式，在执行修复时询问问题，让用户得以确认并决定处理方式；

❒-R 当搭配"-A"参数使用时，则会略过/目录的文件系统不予检查；

❒-s 依序执行检查作业，而非同时执行；

❒-t 指定要检查的文件系统类型；

❒-T 执行fsck指令时，不显示标题信息；

❒-V 显示指令执行过程。

参数：

对应的磁盘

示例：

linux的文件系统损坏会导致linux不正常关机，出错的时候如果系统告诉你是哪一块硬盘的分区有问题，比如是/dev/hda2，就使用如下的命令修复：

fsck -y /dev/hda2

结束后使用reboot命令重启系统这样就好了！

如果不知道时哪个地方出了问题，可以直接输入：

fsck

在随后的多个确认对话框中输入:y 结束后同样使用reboot命令重启系统这样就好了！

## dd

格式：dd (选项)

作用：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。

选项：

❒ bs=<字节数> 将ibs（输入）与obs（输出）设成指定的字节数；

❒ cbs=<字节数> 转换时，每次只转换指定的字节数；

❒ conv=<关键字> 指定文件转换的方式；

❒ count=<区块数> 仅读取指定的区块数；

❒ if=<文件> 从 file 中读取；

❒ ibs=<字节数> 每次读取的字节数；

❒ obs=<字节数> 每次输出的字节数；

❒ of=<文件> 输出到文件；

❒ seek=<区块数> 一开始输出时，跳过指定的区块数；

❒ skip=<区块数> 一开始读取时，跳过指定的区块数；

❒ --help 显示帮助；

❒ --version 显示版本信息。

示例：

[root@localhost ~]# dd if=/dev/zero of=sun.txt bs=1M count=1

1+0 records in

1+0 records out

1048576 bytes (1.0 MB) copied, 0.00203259 seconds, 516 MB/s

该命令创建了一个1M大小的文件sun.txt，其中参数解释：

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。

bs 代表字节为单位的块大小。

count 代表被复制的块数。

/dev/zero 是一个字符设备，会不断返回0值字节（\0）。

块大小可以使用的计量单位表：

|  |  |
| --- | --- |
| 单元大小 | 代码 |
| 字节（1B） | c |
| 字节（2B） | w |
| 块（512B） | b |
| 千字节（1024B） | k |
| 兆字节（1024KB） | M |
| 吉字节（1024MB） | G |

## shutdown

格式：shutdown (选项) (参数)

作用：系统关机指令

选项：

❒ -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。

❒　-f 　重新启动时不执行fsck。

❒　-F 　重新启动时执行fsck。

❒　-h 　将系统关机。

❒　-k 　只是送出信息给所有用户，但不会实际关机。

❒ -n 　不调用init程序进行关机，而由shutdown自己进行。

❒　-r 　shutdown之后重新启动。

❒　-t<秒数> 　送出警告信息和删除信息之间要延迟多少秒。

参数：

[时间] 　设置多久时间后执行shutdown指令。

　[警告信息] 　要传送给所有登入用户的信息。  
示例：

指定现在立即关机：

[root@localhost ~]#shutdown -h now

指定5分钟后关机，同时送出警告信息给登入用户：

[root@localhost ~]#shutdown +5 "System will shutdown after 5 minutes"

## reboot

格式：reboot (选项)

作用：让系统停止运作，并重新开机。

选项：

❒ -d 重新开机时不把数据写入记录文件/var/tmp/wtmp。本参数具有"-n"参数的效果。

❒ -f 强制重新开机，不调用shutdown指令的功能。

❒ -i 在重开机之前，先关闭所有网络界面。

❒ -n 重开机之前不检查是否有未结束的程序。

❒ -w 仅做测试，并不真的将系统重新开机，只会把重开机的数据写入/var/log目录下的wtmp记录文件。

示例：

重启机器：

[root@localhost ~]#reboot

做个重启机器的模拟（只有纪录并不会真的重启）：

[root@localhost ~]#reboot –w

## ps

格式：ps [选项] [参数]

作用：报告程序状况

补充说明：ps是用来报告程序执行情况的指令，你可以搭配kill指令随时中断、删除不必要的程序。

选项与参数：

❒ -a 显示所有终端机下执行的程序，除了阶段作业领导者之外

❒ a 显示现行终端机下的所有程序，包括其他用户的程序

❒ -A 显示所有程序

❒ -c 显示CLS和PRI栏位

❒ c 列出程序时，显示每个程序真正的指令名称，而不包括路径，参数或常驻服务的标示。

❒ -C <指令名称> 指定执行指令的名称，并列出该指令的程序的状况

❒ -d 显示所有程序，但不包括阶段作业领导者的程序

❒ -e 此参数的效果和指定“A”参数相同

❒ e 列出程序时，显示每个程序所使用的环境变量

❒ -f 显示UID，PPIP，C与STIME栏位

❒ f 用ASCII字符显示树状结构，表达程序间的相互关系

❒ -g <群组名称> 此参数的效果和指定“-G”参数相同，当亦能使用阶段作业领导者的名称来指定

❒ g 显示现行终端机下的所有程序，包括群组领导者的程序

❒ -G <群组识别码> 列出属于该群组的程序的状况，也可使用群组名称来指定

❒ h 不显示标题列

❒ -H 显示树状结构，表示程序间的相互关系

❒ -j或j 采用工作控制的格式显示程序状况

❒ -l或l 采用详细的格式来显示程序状况

❒ L 列出栏位的相关信息

❒ -m或m 显示所有的执行绪

❒ n 以数字来表示USER和WCHAN栏位

❒ -N 显示所有的程序，除了执行ps指令终端机下的程序之外

❒ -p <程序识别码> 指定程序识别码，并列出该程序的状况

❒ p <程序识别码> 此参数的效果和指定“-p”参数相同，只在列表格式方面稍有差异

❒ r 只列出现行终端机正在执行中的程序

❒ -s <阶段作业> 指定阶段作业的程序识别码，并列出隶属该阶段作业的程序的状况

❒ s 采用程序信号的格式显示程序状况

❒ S 列出程序时，包括已中断的子程序资料

❒ -t <终端机编号> 指定终端机编号，并列出属于该终端机的程序的状况

❒ t <终端机编号> 此参数的效果和指定“-t”参数相同，只在列表格式方面稍有差异

❒ -T 显示现行终端机下的所有程序

❒ -u <用户识别码> 此参数的效果和指定“-U”参数相同

❒ u 以用户为主的格式来显示程序状况

❒ -U <用户识别码> 列出属于该用户的程序的状况，也可使用用户名称来指定

❒ U <用户识别码> 列出属于该用户的程序的状况

❒ v 采用虚拟内存的格式显示程序状况

❒ -V和V 显示版本信息

❒ -w或w 采用宽阔的格式来显示程序状况

❒ x 显示所有程序，不以终端机来区分

❒ X 采用旧式的Linux i386登陆格式显示程序状况

❒ -y 配合参数“-l”使用时，不显示F(flag)栏位，并以RSS栏位取代ADDR栏位

❒ -<程序识别码> 此参数的效果和指定“p”参数相同

❒ --cols<每列字符数> 设置每列的最大字符数

❒ --columns <每列字符数> 此参数的效果和指定“--cols”参数相同

❒ --cumulative 此参数的效果和指定“S”参数相同

❒ --deselect 此参数的效果和指定“-N”参数相同

❒ --forest 此参数的效果和指定“f”参数相同

❒ --headers 重复显示标题列

❒ --help 在线帮助

❒ --info 显示排错信息

❒ --lines <显示列数> 设置显示画面的列数

❒ --no-headers 此参数的效果和指定“h”参数相同，只在列表格式方面稍有差异

❒ --group <群组名称> 此参数的效果和指定“-G”参数相同

❒ --Group <群组识别码> 此参数的效果和指定“-G”参数相同

❒ --pid 此参数的效果和指定“-p”参数相同

❒ --rows <显示列数> 此参数的效果和指定“--lines”参数相同

❒ --sid <阶段作业> 此参数的效果和指定“-s”参数相同

❒ --tty <终端机编号> 此参数的效果和指定“-t”参数相同

❒ --user <用户名称> 此参数的效果和指定“-U”参数相同

❒ --User <用户识别码> 此参数的效果和指定“-U”参数相同

❒ --version 此参数的效果和指定“-V”参数相同

❒ --widty <每列字符数> 此参数的效果和指定“-cols”参数相同

注意：

这个命令选项和参数太多，而且不同选项还要配以不同的参数，所以直接推荐记住三个命令：ps -l，ps -el和ps aux

示例一：

将目前属于你自己这次登录的PID与相关信息显示出来（只与自己的bash有关）

[root@localhost mnt]# ps -l

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD

4 S 0 1667 1665 0 80 0 - 27085 wait pts/0 00:00:00 bash

4 R 0 1708 1667 0 80 0 - 27033 - pts/0 00:00:00 ps

示例二：

列出目前所有的正在内存当中的程序：

[root@localhost mnt]# ps aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.0 0.1 19356 1544 ? Ss 05:16 0:01 /sbin/init

root 2 0.0 0.0 0 0 ? S 05:16 0:00 [kthreadd]

……中间部分省略……

root 1283 0.0 0.1 251192 1672 ? Sl 05:17 0:00 /sbin/rsyslogd -i /var/run/syslogd.pid -c 5

dbus 1334 0.0 0.0 21432 928 ? Ss 05:17 0:00 dbus-daemon --system

root 1372 0.0 0.0 4080 660 ? Ss 05:17 0:00 /usr/sbin/acpid

68 1384 0.0 0.4 38188 4380 ? Ssl 05:17 0:00 hald

root 1385 0.0 0.1 20400 1160 ? S 05:17 0:00 hald-runner

root 1417 0.0 0.1 22520 1088 ? S 05:17 0:00 hald-addon-input: Listening on /dev/input/event2 /dev/i

68 1429 0.0 0.1 18008 1036 ? S 05:17 0:00 hald-addon-acpi: listening on acpid socket /var/run/acp

root 1470 0.0 0.1 66224 1228 ? Ss 05:17 0:00 /usr/sbin/sshd

root 1549 0.0 0.3 80864 3428 ? Ss 05:17 0:00 /usr/libexec/postfix/master

postfix 1576 0.0 0.3 80944 3384 ? S 05:17 0:00 pickup -l -t fifo -u

postfix 1577 0.0 0.3 81112 3432 ? S 05:17 0:00 qmgr -l -t fifo -u

root 1578 0.0 0.2 180900 2484 ? Ss 05:17 0:00 /usr/sbin/abrtd

……中间部分省略……

root 1665 0.0 0.4 102508 4520 ? Ss 05:19 0:00 sshd: root@pts/0

root 1667 0.0 0.1 108340 1800 pts/0 Ss 05:19 0:00 -bash

root 1709 0.0 0.1 110240 1128 pts/0 R+ 05:47 0:00 ps aux

范例三：

以范例一的显示内容，显示出所有的程序

[root@localhost mnt]# ps -Al

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD

4 S 0 1 0 0 80 0 - 4839 poll\_s ? 00:00:01 init

1 S 0 2 0 0 80 0 - 0 kthrea ? 00:00:00 kthreadd

1 S 0 3 2 0 -40 - - 0 migrat ? 00:00:00 migration/0

1 S 0 4 2 0 80 0 - 0 ksofti ? 00:00:00 ksoftirqd/0

1 S 0 5 2 0 -40 - - 0 cpu\_st ? 00:00:00 stopper/0

5 S 0 6 2 0 -40 - - 0 watchd ? 00:00:00 watchdog/0

1 S 0 7 2 0 80 0 - 0 worker ? 00:00:03 events/0

1 S 0 8 2 0 80 0 - 0 worker ? 00:00:00 events/0

……以下部分省略

范例四：

找出与crom和syslog两个服务有关的IPD号码

[root@localhost mnt]# ps aux | egrep '(cron|syslog)'

root 1283 0.0 0.1 251192 1672 ? Sl 05:17 0:00 /sbin/rsyslogd -i /var/run/syslogd.pid -c 5

root 1590 0.0 0.1 116860 1248 ? Ss 05:17 0:00 crond

root 1717 0.0 0.0 103308 880 pts/0 S+ 05:59 0:00 grep -E (cron|syslog)

## top

格式：top (参数)

作用：可显示目前正在系统中执行的程序，并通过它所提供的互动式界面，用热键加以管理。

参数：

❒ b 　使用批处理模式。

❒　c 　列出程序时，显示每个程序的完整指令，包括指令名称，路径和参数等相关信息。

❒　d<间隔秒数> 　设置top监控程序执行状况的间隔时间，单位以秒计算。

❒　i 　执行top指令时，忽略闲置或是已成为Zombie的程序。

❒　n<执行次数> 　设置监控信息的更新次数。

❒　q 　持续监控程序执行的状况。

❒　s 　使用保密模式，消除互动模式下的潜在危机。

❒　S 　使用累计模式，其效果类似ps指令的"-S"参数。

交互命令：

在top命令执行过程中可以使用的一些交互命令。这些命令都是单字母的，如果在命令行中使用了-s选项， 其中一些命令可能会被屏蔽。

❒ h：显示帮助画面，给出一些简短的命令总结说明；

❒ k：终止一个进程；

❒ i：忽略闲置和僵死进程，这是一个开关式命令；

❒ q：退出程序；

❒ r：重新安排一个进程的优先级别；

❒ S：切换到累计模式；

❒ s：改变两次刷新之间的延迟时间（单位为s），如果有小数，就换算成ms。输入0值则系统将不断刷新，默认值是5s；

❒ f或者F：从当前显示中添加或者删除项目；

❒ o或者O：改变显示项目的顺序；

❒ l：切换显示平均负载和启动时间信息；

❒ m：切换显示内存信息；

❒ t：切换显示进程和CPU状态信息；

❒ c：切换显示命令名称和完整命令行；

❒ M：根据驻留内存大小进行排序；

❒ P：根据CPU使用百分比大小进行排序；

❒ T：根据时间/累计时间进行排序；

❒ w：将当前设置写入~/.toprc文件中。

示例：

指定现在立即关机：

[root@localhost ~]#top

top - 09:44:56 up 16 days, 21:23, 1 user, load average: 1.59, 1.75, 1.92

Tasks: 145 total, 2 running, 143 sleeping, 0 stopped, 0 zombie

Cpu(s): 99.8%us, 0.1%sy, 0.0%ni, 0.2%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 4147888k total, 2493092k used, 1654796k free, 158188k buffers

Swap: 5144568k total, 56k used, 5144512k free, 2013180k cached

解释：

top - 09:44:56[当前系统时间],

16 days[系统已经运行了16天],

1 user[1个用户当前登录],

load average: 1.59, 1.75, 1.92[系统负载，即任务队列的平均长度],

Tasks: 145 total[总进程数],

2 running[正在运行的进程数], 143 sleeping[睡眠的进程数],

0 stopped[停止的进程数],

0 zombie[冻结进程数],

Cpu(s): 99.8%us[用户空间占用CPU百分比],

0.1%sy[内核空间占用CPU百分比],

0.0%ni[用户进程空间内改变过优先级的进程占用CPU百分比],

0.2%id[空闲CPU百分比],

0.0%wa[等待输入输出的CPU时间百分比],

0.0%hi[],

0.0%st[],

Mem: 4147888k total[物理内存总量],

2493092k used[使用的物理内存总量],

1654796k free[空闲内存总量],

158188k buffers[用作内核缓存的内存量]

Swap: 5144568k total[交换区总量],

56k used[使用的交换区总量],

5144512k free[空闲交换区总量],

2013180k cached[缓冲的交换区总量]

## free

格式：free [-bkmost]

作用：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

主要参数：

❒ -b：以Byte 为单位显示内存使用情况

❒ -k：以KB 为单位显示内存使用情况

❒ -m：以MB 为单位显示内存使用情况

❒ -o：不显示缓冲区调节列

❒ -s<间隔秒数>：持续观察内存使用状况

❒ -t：显示内存总和列

实例：查看当前内存使用状况：

[root@hh-02 ~]# free -m

total used free shared buffers cached

Mem: 3816 305 3511 0 17 111

-/+ buffers/cache: 177 3639

Swap: 3999 0 3999

## kill

格式：kill [ －s signal | －p ] [ －a ] pid …

作用：用来终止一个进程

选项：

❒-l 信号，若果不加信号的编号参数，则使用“-l”参数会列出全部的信号名称

❒-a 当处理当前进程时，不限制命令名和进程号的对应关系

❒-p 指定kill 命令只打印相关进程的进程号，而不发送任何信号

❒-s 指定发送信号

❒-u 指定用户

参数：

进程或者作业识别号：指定要删除的进程或作业  
示例：

1、列出所有信号名称

[root@B-MA /]# kill -l

1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP

6) SIGABRT 7) SIGBUS 8) SIGFPE 9) SIGKILL 10) SIGUSR1

11) SIGSEGV 12) SIGUSR2 13) SIGPIPE 14) SIGALRM 15) SIGTERM

16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP

21) SIGTTIN 22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ

26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO 30) SIGPWR

31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3

38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8

43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13

48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12

53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7

58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2

63) SIGRTMAX-1 64) SIGRTMAX

2、先用ps查找进程，然后用kill彻底杀掉该进程

[root@B-MA /]# vim &

[1] 4733

[root@B-MA /]# ps -ef |grep 4733

root 4733 2776 0 06:10 pts/0 00:00:00 vim

[1]+ Stopped vim

[root@B-MA /]# kill -9 4733

[1]+ Killed vim

## halt

**功能说明：**关闭系统。

**语　　法：**halt [-dfinpw]

**补充说明：**halt会先检测系统的runlevel。若runlevel为0或6，则关闭系统，否则即调用shutdown来关闭系统。

**参　　数：**

❒ -d 　不要在wtmp中记录。

❒ -f 　不论目前的runlevel为何，不调用shutdown即强制关闭系统。

❒ -i 　在halt之前，关闭全部的网络界面。

❒ -n 　halt前，不用先执行sync。

❒ -p 　halt之后，执行poweroff。

❒ -w 　仅在wtmp中记录，而不实际结束系统。

实例1：

关闭系统后关闭电源

[root@xuegod65 ~]# halt –p

Broadcast message from root@xuegod65

(/dev/pts/0) at 18:31 ...

The system is going down for power off NOW!

Connection closed by foreign host.

Disconnected from remote host(65) at 23:44:50.

实例2：

关闭系统后，但不留下记录

[root@xuegod65 ~]# halt –d

## chgrp

格式：chgrp（选项）（参数）

作用：改变文件或目录所属的组

选项：

❒ -c或 --changes：效果类似“-v”参数，但仅回报更改的部分

❒ -f或 --quiet或 --silent：不显示错误信息

❒ -h或 --no-dereference：只对符号连接的文件作修改，而不是该其他任何相关文件

❒ -R或 --recursive：递归处理，将指令目录下的所有文件及子目录一并处理

❒ -v或 --verbose：显示指令执行过程

❒ --reference=<参考文件或目录>：把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同；

参数：

组：指定新组名称

文件：指定要改变所属组的文件列表。多个文件或者目录之间使用空格隔开。

示例：

将/usr/test及其子目录下的所有文件的用户组改为sfit

[root@localhost test]# ll -a

total 12

drwxr-xr-x 2 root root 32 Nov 17 09:54 .

drwxr-xr-x. 17 root root 4096 Nov 17 09:53 ..

-rw-r--r-- 1 root root 1490 Nov 17 09:54 passwd

[root@localhost test]# chgrp –R sfit /usr/test

[root@localhost test]# ll –a

total 12

drwxr-xr-x 2 root sfit 32 Nov 17 09:54 .

drwxr-xr-x. 17 root root 4096 Nov 17 09:53 ..

-rw-r--r-- 1 root sfit 1490 Nov 17 09:54 passwd

## umask

语法：umask(选项)(参数)

作用：umask命令用来设置限制新建文件权限的掩码。当新文件被创建时，其最初的权限由文件创建掩码决定。用户每次注册进入系统时，umask命令都被执行， 并自动设置掩码mode来限制新文件的权限。用户可以通过再次执行umask命令来改变默认值，新的权限将会把旧的覆盖掉。

选项

* -p：输出的权限掩码可直接作为指令来执行；
* -S：以符号方式输出权限掩码。

参数

权限掩码：指定权限掩码。

实例

利用umask命令可以指定哪些权限将在新文件的默认权限中被删除。例如，可以使用下面的命令创建掩码，使得组用户的写权限，其他用户的读、写和执行权限都被取消：

umask u=, g=[w](http://man.linuxde.net/w), o=rwx

执行该命令以后，对于下面创建的新文件，其文件主的权限未做任何改变，而组用户没有写权限，其他用户的所有权限都被取消。

应注意：操作符“=”在umask命令和[chmod](http://man.linuxde.net/chmod)命令中的作用恰恰相反。在chmod命令中，利用它来设置指定的权限，而其余权限则被删除；但是在umask命令中，它将在原有权限的基础上删除指定的权限。

不能直接利用umask命令创建一个可执行的文件，用户只能在其后利用chmod命令使它具有执行权限。假设执行了命令umask u=, g=w, o=rwx，虽然在命令行中，没有删去文件主和组用户的执行权限，但默认的文件权限还是640（即 rw-r-----），而不是750(rwxr-x---）。但是，如果创建的是目录或者通过编译程序创建的一个可执行文件，将不受此限制。在这种情况 下，会设置文件的执行权限。

也可以使用八进制数值来设置mode。由于在umask中所指定的权限是要从文件中删除的，所以，如果该文件原来的初始化权限是777，那么执行命令umask 022以后，该文件的权限将变为755：如果该文件原来的初始化权限是666，那么该文件的权限将变为644。

可以使用下面的命令检查新创建文件的默认权限：

[root@localhost ~]# umask -s

选项-s表示以字符形式显示当前的掩码。如果直接输入umask命令，不带任何参数，那么将以八进制形式显示当前的掩码。系统默认的掩码是0022。

## chgrp

语法：chgrp (选项) (参数)

作用：[chgrp](http://man.linuxde.net/chgrp)命令用来改变文件或目录所属的用户组。该命令用来改变指定文件所属的用户组。其中，组名可以是用户组的[id](http://man.linuxde.net/id)，也可以是用户组的组名。文件名可以 是由空格分开的要改变属组的文件列表，也可以是由通配符描述的文件集合。如果用户不是该文件的文件主或超级用户(root)，则不能改变该文件的组。

在UNIX系统家族里，文件或目录权限的掌控以拥有者及所属群组来管理。您可以使用chgrp指令去变更文件与目录的所属群组，设置方式采用群组名称或群组识别码皆可。

选项

* -c或——changes：效果类似“-v”参数，但仅回报更改的部分；
* -f或--quiet或——silent：不显示错误信息；
* -h或--no-dereference：只对符号连接的文件作修改，而不是该其他任何相关文件；
* -R或——recursive：递归处理，将指令目录下的所有文件及子目录一并处理；
* -v或——verbose：显示指令执行过程；
* --reference=<参考文件或目录>：把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同；

参数

组：指定新工作名称；

文件：指定要改变所属组的文件列表。多个文件或者目录之间使用空格隔开。

实例

将/usr/meng及其子目录下的所有文件的用户组改为mengxin

[root@localhost ~]# chgrp -R mengxin /usr/meng

## chmod

[chmod](http://man.linuxde.net/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。

权限范围的表示法如下：

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](http://man.linuxde.net/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。

语法

chmod(选项)(参数)

选项

* -c或——changes：效果类似“-v”参数，但仅回报更改的部分；
* -f或--quiet或——silent：不显示错误信息；
* -R或——recursive：递归处理，将指令目录下的所有文件及子目录一并处理；
* -v或——verbose：显示指令执行过程；
* --reference=<参考文件或目录>：把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同；

<权限范围>+<权限设置>：开启权限范围的文件或目录的该选项权限设置；

<权限范围>-<权限设置>：关闭权限范围的文件或目录的该选项权限设置；

<权限范围>=<权限设置>：指定权限范围的文件或目录的该选项权限设置；

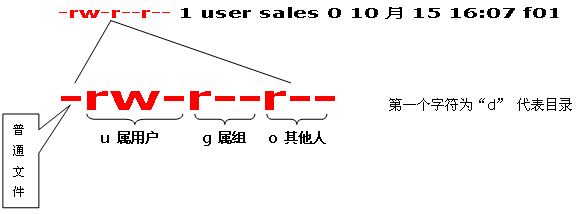
参数：

权限模式：指定文件的权限模式；  
 文件：要改变权限的文件。

知识扩展和实例

Linux用 户分为：拥有者、组群(Group)、其他（other），Linux系统中，预设的情況下，系统中所有的帐号与一般身份使用者，以及root的相关信 息， 都是记录在/etc/[passwd](http://man.linuxde.net/passwd)文件中。每个人的密码则是记录在/etc/shadow文件下。 此外，所有的组群名称记录在/etc/group內！

linux文件的用户权限的分析图



例：rwx　rw-　r--

r=读取属性　　//值＝4  
w=写入属性　　//值＝2  
x=执行属性　　//值＝1

chmod u+x,g+w f01　　//为文件f01设置自己可以执行，组员可以写入的权限

chmod u=rwx,g=rw,o=r f01

chmod 764 f01

chmod a+x f01　　//对文件f01的u,g,o都设置可执行属性

文件的属主和属组属性设置

[root@localhost ~]# [chown](http://man.linuxde.net/chown) user:market f01　　//把文件f01给uesr，添加到market组

ll -d f1 查看目录f1的属性

## chown

语法：chown(选项)(参数)

作用：[chown](http://man.linuxde.net/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](http://man.linuxde.net/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。

选项

* -c或——changes：效果类似“-v”参数，但仅回报更改的部分；
* -f或--quite或——silent：不显示错误信息；
* -h或--no-dereference：只对符号连接的文件作修改，而不更改其他任何相关文件；
* -R或——recursive：递归处理，将指定目录下的所有文件及子目录一并处理；
* -v或——version：显示指令执行过程；
* --dereference：效果和“-h”参数相同；
* --[help](http://man.linuxde.net/help)：在线帮助；
* --reference=<参考文件或目录>：把指定文件或目录的拥有者与所属群组全部设成和参考文件或目录的拥有者与所属群组相同；
* --version：显示版本信息。

参数：

用户：组：指定所有者和所属工作组。当省略“：组”，仅改变文件所有者；  
 文件：指定要改变所有者和工作组的文件列表。支持多个文件和目标，支持shell通配符。

示例

将目录/usr/meng及其下面的所有文件、子目录的文件主改成 liu：

[root@localhost ~]# chown -R liu /usr/meng

## whoami

格式：whoami [选项]

作用：显示当前用户名称，本指令相当于执行“id -un”指令

选项：

❒ --help 在线帮助

❒ --version 显示版本信息

示例：

显示自己登录系统的用户名称

[root@localhost mnt]# whoami

root

## route

格式：route (选项)

作用：打印和操作ip路由表

 选项：

❒ add 添加一条新路由

❒ del 删除一条路由

❒ -net 目标地址是一个网络

❒ -host 目标地址是一个主机

❒ netmask 当添加 一个网络路由时，需要使用网络掩码

❒ gw 路由数据包通过网关。注意：你指定的网关必须能够达到

❒ metric 设置路由条数

添加一条到达192.168.2.0网络的路由，指定网络掩码为255.255.255.0,数据包通过网络接口eth0。

[root@localhost ~]# route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0

添加一条到达192.57.66.0网络的路由，指定网络掩码为255.255.255.0,数据包通过网关地址192.168.2.1。

[root@localhost ~]#route add -net 192.57.66.0 netmask 255.255.255.0 gw 192.168.2.1

所有去往192.57.66.200主机的数据包发往网关地址192.168.2.1。

[root@localhost ~]#route add -host 192.57.66.200 gw 192.168.2.1

路由表内容说明：

查看路由

# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.57.66.200 192.168.1.1 255.255.255.255 UGH 0 0 0 eth0

192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

Destination：目标网络或主机。  
Gateway：网关地址。  
Genmask：目标网络的网络掩码。"255.255.255.255"表示一个主机。"0.0.0.0"表示网关。  
Flags：标记。  
       U、路由被启用。  
       H、目标是一个主机  
       G、使用网关。

## vmstat

格式：vmstat (选项) (参数)

作用：显示虚拟内存状态（“Viryual Memor Statics”），但是它可以报告关于进程、内存、I/O等系统整体运行状态。

选项：

❒ -a 显示活动内页；

❒ -f 显示启动后创建的进程总数；

❒ -m 显示slab信息；

❒ -n 头信息仅显示一次；

❒ -s 以表格方式显示事件计数器和内存状态；

❒ -d 报告磁盘状态；

❒ -p 显示指定的硬盘分区状态；

❒ -S 输出信息的单位。

参数：

事件间隔：状态信息刷新的时间间隔；

次数：显示报告的次数。

一般vmstat工具的使用是通过两个数字参数来完成的，第一个参数是采样的时间间隔数，单位是秒，第二个参数是采样的次数。

示例：

当mpstat不带参数时，输出为从系统启动以来的平均值。

[root@localhost ~]#mpstat

Linux 2.6.9-5.31AXsmp (builder.redflag-linux.com) 12/16/2005

09:38:46 AM CPU %user %[nice](http://man.linuxde.net/nice) %system %iowait %irq %soft %idle intr/s

09:38:48 AM all 23.28 0.00 1.75 0.50 0.00 0.00 74.47 1018.59

实际上，在应用过程中，我们会在一段时间内一直监控，不想监控直接结束vmstat就行了。

[root@localhost ~]# vmstat 2

procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----  
 r b swpd free buff cache si so bi bo in cs us sy id wa  
 1 0 0 3499840 315836 3819660 0 0 0 1 2 0 0 0 100 0  
 0 0 0 3499584 315836 3819660 0 0 0 0 88 158 0 0 100 0

## sar

格式：sar (选项) (参数)

作用：对系统的活动进行报告，包括：文件的读写情况、系统调用的使用情况、磁盘IO、CPU效率、内存使用状况、进程活动及IPC有关的活动等

选项：

❒ -A：所有报告的总和

❒ -u：输出CPU使用情况的统计信息

❒ -v：输出inode、文件和其他内核表的统计信息

❒ -d：输出每一个块设备的活动信息

❒ -r：输出内存和交换空间的统计信息

❒ -b：显示I/O和传送速率的统计信息

❒ -a：文件读写情况

❒ -c：输出进程统计信息，每秒创建的进程数

❒ -R：输出内存页面的统计信息

❒ -y：终端设备活动情况

❒ -w：输出系统交换活动信息

参数：

间隔时间：每次报告的间隔时间（秒）

次数：显示报告的次数  
示例：

1、每10秒采样一次，连续采样3次，观察CPU 的使用情况，并将采样结果以二进制形式存入当前目录下的文件test中

[root@B-MA ~]# sar -u -o test 10 3

Linux 2.6.32-431.el6.x86\_64 (B-MA.china.cs) 11/24/2016 \_x86\_64\_ (4 CPU)

04:26:10 AM CPU %user %nice %system %iowait %steal %idle

04:26:20 AM all 0.00 0.00 0.13 0.03 0.00 99.85

04:26:30 AM all 0.03 0.00 0.15 0.00 0.00 99.82

04:26:40 AM all 0.03 0.00 0.13 0.00 0.00 99.85

Average: all 0.02 0.00 0.13 0.01 0.00 99.84drw-r--r-- 2 root root 6 Nov 7 15:45 test

2、每10秒采样一次，连续采样3次，监控内存分页：

[root@B-MA ~]# sar -r 10 3

Linux 2.6.32-431.el6.x86\_64 (B-MA.china.cs) 11/24/2016 \_x86\_64\_ (4 CPU)

04:31:18 AM kbmemfree kbmemused %memused kbbuffers kbcached kbcommit %commit

04:31:28 AM 200348 804064 80.05 24652 256072 1048080 34.52

04:31:38 AM 200224 804188 80.07 24668 256072 1048080 34.52

04:31:48 AM 200100 804312 80.08 24732 256076 1048080 34.52

Average: 200224 804188 80.07 24684 256073 1048080 34.52

3、每10秒采样一次，连续采样3次，报告缓冲区的使用情况

[root@B-MA /]# sar -b 10 3

Linux 2.6.32-431.el6.x86\_64 (B-MA.china.cs) 11/24/2016 \_x86\_64\_ (4 CPU)

05:53:50 AM tps rtps wtps bread/s bwrtn/s

05:54:00 AM 4.40 0.00 4.40 0.00 70.40

05:54:10 AM 1.40 0.00 1.40 0.00 12.81

## chkconfig

格式：chkconfig [--add][--del][--list][系统服务] 或 chkconfig [--level <等级代号>][系统服务][on/off/reset]

作用：用来更新（启动或停止）和查询系统服务的运行级信息

选项：

❒--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

❒--del 　删除所指定的系统服务，不再由chkconfig指令管理，并同时在系统启动的叙述文件内删除相关数据

❒--level<等级代号> 　指定读系统服务要在哪一个执行等级中开启或关毕

参数：

等级0表示：表示关机

等级1表示：单用户模式

等级2表示：无网络连接的多用户命令行模式

等级3表示：有网络连接的多用户命令行模式

等级4表示：不可用

等级5表示：带图形界面的多用户模式

等级6表示：重新启动

示例：

1、设置httpd在运行级别为2、3、4、5的情况下都是on（开启）的状态

[root@B-MA /]# chkconfig --level 2345 httpd on

[root@B-MA /]# chkconfig --list httpd

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

2、列出系统所有的服务启动情况

[root@B-MA /]# chkconfig --list httpd

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

[root@B-MA /]# chkconfig --list

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

abrt-ccpp 0:off 1:off 2:off 3:on 4:off 5:on 6:off

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

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

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

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

blk-availability 0:off 1:on 2:on 3:on 4:on 5:on 6:off

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

cloudstack-management 0:off 1:off 2:on 3:on 4:on 5:on 6:off

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

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

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

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

dhcpd6 0:off 1:off 2:off 3:off 4:off 5:off 6:off

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

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

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

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

ip6tables 0:off 1:off 2:on 3:on 4:on 5:on 6:off

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

……..

## xargs

作用  
用作替换工具，读取输入数据重新格式化后输出

格式

xargs [-nd]

主要参数说明

-n：多行输出

-d：自定义一个定界符

## nc/netcat

作用  
netcat命令的简称，都是用来设置路由器

格式

nc [-gGhLnoprsuvzw] [主机/主机号]

主要参数说明

❒-g<网关>：设置路由器跃程通信网关，最多设置8个

❒-G<指向器数目>：设置来源路由指向器，其数值为4的倍数

❒-h：在线帮助； -i<延迟秒数>：设置时间间隔，以便传送信息及扫描通信端口

❒-l：使用监听模式，监控传入的资料

❒-n：直接使用ip地址，而不通过域名服务器

❒-o<输出文件>：指定文件名称，把往来传输的数据以16进制字码倾倒成该文件保存

❒-p<通信端口>：设置本地主机使用的通信端口

❒-r：指定源端口和目的端口都进行随机的选择

❒-s<来源位址>：设置本地主机送出数据包的IP地址

❒-u：使用UDP传输协议

❒-v：显示指令执行过程

❒-w<超时秒数>：设置等待连线的时间

❒-z：使用0输入/输出模式，只在扫描通信端口时使用

## nohup

作用  
将程序以忽略挂起信号的方式运行起来，被运行的程序的输出信息将不会显示到终端

格式

nohup [要运行的程序及选项]

## time

作用  
统计给定命令所花费的总时间格式

time [指定需要运行的额指令及其参数]

示例：

[root@XD Desktop]# time ls

a.txt

real 0m0.002s

user 0m0.000s

sys 0m0.002s

## eject

作用  
退出抽取式设备。若设备已挂入，则eject命令会先将该设备卸除再退出

格式

eject [-facdnqrst] [指定弹出的设备名称]

主要参数说明

-a<开关>或--auto<开关>：控制设备的自动退出功能

-c<光驱编号>或--changerslut<光驱编号> 选择光驱柜中的光驱

-d或--default：显示预设的设备，而不是实际执行动作

-f或--floppy：退出抽取式磁盘

-n或--noop：显示指定的设备

-q或--tape：退出磁带

-r或--cdrom：退出光盘

-s或--scsi：以SCSI指令来退出设备

-t或--trayclose：关闭光盘的托盘

## unalias

作用  
取消命令别名，是为shell内建命令

格式

unalias [-a] [指定要取消的命令别名]

主要参数说明

-a：取消所有命令别名

## alias

作用  
设置指令的别名

格式

alias [-p] [命令别名]

主要参数说明

-p：打印已经设置的命令别名

## watch

作用  
以周期性的方式执行给定的指令，指令输出以全屏方式显示

格式

watch [-ndt] [周期性执行指令]

主要参数说明

-n：指定指令执行的间隔时间（秒）

-d：高亮显示指令输出信息不同之处

-t：不显示标题

## yum

作用  
在Fedora和RedHat以及SUSE中基于rpm的软件包管理器

格式

yum [-cyqvdeRC] [install]

主要参数说明

❒-y：对所有的提问都回答“yes”

❒-c：指定配置文件

❒-q：安静模式

❒-v：详细模式

❒-d：设置调试等级（0-10）

❒-e：设置错误等级（0-10）

❒-R：设置yum处理一个命令的最大等待时间

❒-C：完全从缓存中运行，而不去下载或者更新任何头文件

install：安装rpm软件包

update：更新rpm软件包

check-update：检查是否有可用的更新rpm软件包

remove：删除指定的rpm软件包； list：显示软件包的信息

search：检查软件包的信息

info：显示指定的rpm软件包的描述信息和概要信息

clean：清理yum过期的缓存

shell：进入yum的shell提示符

resolvedep：显示rpm软件包的依赖关系

localinstall：安装本地的rpm软件包

localupdate：显示本地rpm软件包进行更新

deplist：显示rpm软件包的所有依赖关系

示例：

[root@XD Desktop]# yum install httpd

## last

格式：chkconfig [--add][--del][--list][系统服务] 或 chkconfig [--level <等级代号>][系统服务][on/off/reset]

作用：用于显示最近用户登录信息，它是读取/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。

参数：

用户名：显示用户登录列表

终端：显示从指定终端的登陆列表

选项：

❒-a：把从何处登入系统的主机名称或ip地址，显示在最后一行

❒-d：将IP地址转换成主机名称

❒-f <记录文件>：指定记录文件

❒-n <显示列数>或-<显示列数>：设置列出名单的显示列数

❒-R：不显示登入系统的主机名称或IP地址

❒-x：显示系统关机，重新开机，以及执行等级的改变等信息

示例：

1、显示用户登录情况

[root@B-MA ~]# last

root pts/0 192.168.138.1 Thu Nov 24 08:30 still logged in

reboot system boot 2.6.32-431.el6.x Thu Nov 24 08:18 - 08:31 (00:12)

root pts/0 192.168.138.1 Thu Nov 24 06:39 - crash (01:39)

root pts/1 192.168.138.1 Thu Nov 24 06:24 - 06:24 (00:00)

root pts/0 192.168.138.1 Thu Nov 24 04:24 - 06:24 (02:00)

root pts/0 192.168.138.1 Thu Nov 24 04:23 - 04:24 (00:00)

root tty1 Thu Nov 24 04:14 - crash (04:04)

reboot system boot 2.6.32-431.el6.x Thu Nov 24 04:13 - 08:31 (04:17)

root tty1 Wed Nov 23 10:02 - down (00:00)

reboot system boot 2.6.32-431.el6.x Wed Nov 23 05:59 - 10:02 (04:02)

2、把从何处登录系统的主机名称或者IP显示在最后一行

[root@B-MA ~]# last -a

root pts/0 Thu Nov 24 08:30 still logged in 192.168.138.1

reboot system boot Thu Nov 24 08:18 - 08:33 (00:14) 2.6.32-431.el6.x86\_64

root pts/0 Thu Nov 24 06:39 - crash (01:39) 192.168.138.1

root pts/1 Thu Nov 24 06:24 - 06:24 (00:00) 192.168.138.1

root pts/0 Thu Nov 24 04:24 - 06:24 (02:00) 192.168.138.1

root pts/0 Thu Nov 24 04:23 - 04:24 (00:00) 192.168.138.1

3、显示最近10次，且不显示登入系统的主机名称或IP地址

[root@B-MA ~]# last -10 -R

root pts/0 Thu Nov 24 08:30 still logged in

reboot system boot Thu Nov 24 08:18 - 08:34 (00:15)

root pts/0 Thu Nov 24 06:39 - crash (01:39)

root pts/1 Thu Nov 24 06:24 - 06:24 (00:00)

root pts/0 Thu Nov 24 04:24 - 06:24 (02:00)

root pts/0 Thu Nov 24 04:23 - 04:24 (00:00)

root tty1 Thu Nov 24 04:14 - crash (04:04)

reboot system boot Thu Nov 24 04:13 - 08:34 (04:20)

root tty1 Wed Nov 23 10:02 - down (00:00)

reboot system boot Wed Nov 23 05:59 - 10:02 (04:02)

## passwd

格式：passwd (选项) (参数)

作用：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。   
参数：

用户名：需要设置密码的用户名。

选项：

❒-d：删除密码，仅有系统管理者才能使用；

❒-f：强制执行；

❒-k：设置只有在密码过期失效后，方能更新；

❒-l：锁住密码；

❒-s：列出密码的相关信息，仅有系统管理者才能使用；

❒-u：解开已上锁的帐号。

示例：

1. 给新建的用户设置密码（需要使用root账户的权限来设置）

[root@B-MA ~]# passwd xuegod

Changing password for user xuegod.

New password:

BAD PASSWORD: it is too simplistic/systematic

BAD PASSWORD: is too simple

Retype new password:

passwd: all authentication tokens updated successfully.

1. 给自身设置密码，直接运行passwd

[xuegod@B-MA root]$ passwd

Changing password for user xuegod.

Changing password for xuegod.

(current) UNIX password:

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

1. 锁定普通用户不能更改密码

[root@B-MA ~]# passwd -l xuegod #锁定用户xuegod不能修改密码

Locking password for user xuegod.

passwd: Success

[root@B-MA ~]# su xuegod #切换到xuegod用户

[xuegod@B-MA root]$ passwd

Changing password for user xuegod.

Changing password for xuegod.

(current) UNIX password:

passwd: Authentication token manipulation error #提示无法更改密码

1. 清除账户密码

[root@B-MA ~]# passwd -S xuegod #查询xuegod账户密码状态

xuegod PS 2016-11-23 0 99999 7 -1 (Password set, MD5 crypt.)

[root@B-MA ~]# passwd -d xuegod #清除xuegod账户密码

Removing password for user xuegod.

passwd: Success #更改成功

[root@B-MA ~]# passwd -S xuegod #清除后再次查询该账户密码状态

xuegod NP 2016-11-23 0 99999 7 -1 (Empty password.) #提示空密码

**mpstat**

格式：mpstat (选项) (参数)

作用：指令主要用于多CPU环境下，它显示各个可用CPU的状态系你想。这些信息存放在/proc/[stat](http://man.linuxde.net/stat)文件中。在多CPUs系统里，其不但能查看所有CPU的平均状况信息，而且能够查看特定CPU的信息。

选项：

❒ -P 指定CPU编号。

参数：

间隔时间：每次报告的间隔时间（秒）；

次数：显示报告的次数。

示例：

1、当mpstat不带参数时，输出为从系统启动以来的平均值。

[root@localhost ~]# mpstat

Linux 2.6.9-5.31AXsmp (builder.redflag-linux.com) 12/16/2005

09:38:46 AM CPU %user %[nice](http://man.linuxde.net/nice) %system %iowait %irq %soft %idle intr/s

09:38:48 AM all 23.28 0.00 1.75 0.50 0.00 0.00 74.47 1018.59

2、每2秒产生了2个处理器的统计数据报告

下面的命令可以每2秒产生了2个处理器的统计数据报告，一共产生三个interval 的信息，然后再给出这三个interval的平均信息。

[root@localhost ~]# mpstat -P ALL 2 3

Linux 2.6.18-164.el5 (server.sys.com) 01/04/2010

09:34:20 PM CPU %user %nice %sys %iowait %irq %soft %steal %idle intr/s

09:34:22 PM all 0.00 0.00 0.00 0.00 0.00 0.00 0.00 100.00 1001.49

09:34:22 PM 0 0.00 0.00 0.50 0.00 0.00 0.00 0.00 99.50 1001.00

09:34:22 PM 1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 100.00 0.00

**iostat**

格式：iostat (选项) (参数)

作用：被用于监视系统输入输出设备和CPU的使用情况。它的特点是汇报磁盘活动统计情况，同时也会汇报出CPU使用情况。

选项：

❒ -c 仅显示CPU使用情况；

❒ -d 仅显示设备利用率；

❒ -k 显示状态以千字节每秒为单位，而不使用块每秒；

❒ m 显示状态以兆字节每秒为单位；

❒ -p 仅显示块设备和所有被使用的其他分区的状态；

❒ -t 显示每个报告产生时的时间；

❒ -V 显示版号并退出；

❒ -x 显示扩展状态。

参数：

间隔时间：每次报告的间隔时间（秒）；

次数：显示报告的次数。

示例：

用iostat -x /dev/sda1来观看磁盘I/O的详细情况：

[root@localhost ~]# iostat -x /dev/sda1

Linux 2.6.18-164.el5xen (localhost.localdomain)

2016年11月26日

avg-cpu: %user %[nice](http://man.linuxde.net/nice) %system %iowait

%steal %idle

0.11 0.02 0.18 0.35

0.03 99.31

Device: tps Blk\_read/s Blk\_wrtn/s

Blk\_read Blk\_wrtn

sda1 0.02 0.08

0.00 2014 4

## lastlog

格式：lastlog(选项)

作用：astlog命令用于显示系统中所有用户最近一次登录信息。  
选项：

❒-b<天数> 显示指定天数前的登录信息；

❒-h 显示召集令的帮助信息；

❒-t<天数> 显示指定天数以来的登录信息；

❒-u<用户名> 显示指定用户的最近登录信息。  
示例：

[root@haobo66 ~]# lastlog

Username Port From Latest

root pts/0 192.168.1.104 Thu Nov 17 17:33:22 +0800 2016

bin \*\*Never logged in\*\*

daemon \*\*Never logged in\*\*

adm \*\*Never logged in\*\*

lp \*\*Never logged in\*\*

sync \*\*Never logged in\*\*

shutdown \*\*Never logged in\*\*

halt \*\*Never logged in\*\*

mail \*\*Never logged in\*\*

uucp \*\*Never logged in\*\*

operator \*\*Never logged in\*\*

games \*\*Never logged in\*\*

gopher \*\*Never logged in\*\*

ftp \*\*Never logged in\*\*

nobody \*\*Never logged in\*\*

dbus \*\*Never logged in\*\*

vcsa \*\*Never logged in\*\*

rpc \*\*Never logged in\*\*

abrt \*\*Never logged in\*\*

saslauth \*\*Never logged in\*\*

postfix \*\*Never logged in\*\*

haldaemon \*\*Never logged in\*\*

ntp \*\*Never logged in\*\*

rpcuser \*\*Never logged in\*\*

nfsnobody \*\*Never logged in\*\*

sshd \*\*Never logged in\*\*

tcpdump \*\*Never logged in\*\*

## finger

格式：finger(选项)(参数)

作用：finger命令用于查找并显示用户信息。包括本地与远端主机的用户皆可，帐号名称没有大小写的差别。单独执行finger指令，它会显示本地主机现在所有的用户的登陆信息，包括帐号名称，真实姓名，登入终端机，闲置时间，登入时间以及地址和电话。

选项：

❒-l：列出该用户的帐号名称，真实姓名，用户专属目录，登入所用的Shell，登入时间，转信地址，电子邮件状态，还有计划文件和方案文件内容；

❒-m：排除查找用户的真实姓名；

❒-s：列出该用户的帐号名称，真实姓名，登入终端机，闲置时间，登入时间以及地址和电话；

❒-p：列出该用户的帐号名称，真实姓名，用户专属目录，登入所用的Shell，登入时间，转信地址，电子邮件状态，但不显示该用户的计划文件和方案文件内容。

参数：

用户名：指定要查询信息的用户。

示例：

[root@haobo66 ~]# finger

Login Name Tty Idle Login Time Office Office Phone

root root tty1 2 Nov 17 19:45

root root pts/0 Nov 17 19:45 (192.168.1.104)

## ethtool

格式：

ethtool [ -a | -c | -g | -i | -d | -k | -r | -S |] ethX

ethtool [-A] ethX [autoneg on|off] [rx on|off] [tx on|off]

ethtool [-C] ethX [adaptive-rx on|off] [adaptive-tx on|off] [rx-usecs N] [rx-frames N] [rx-usecs-irq N] [rx-frames-irq N] [tx-usecs N] [tx-frames N] [tx-usecs-irq N] [tx-frames-irq N] [stats-block-usecs N][pkt-rate-low N][rx-usecs-low N] [rx-frames-low N] [tx-usecs-low N] [tx-frames-lowN] [pkt-rate-high N] [rx-usecs-high N] [rx-frames-high N] [tx-usecs-high N] [tx-frames-high N] [sample-interval N]

ethtool [-G] ethX [rx N] [rx-mini N] [rx-jumbo N] [tx N]

ethtool [-e] ethX [raw on|off] [offset N] [length N]

ethtool [-E] ethX [magic N] [offset N] [value N]

ethtool [-K] ethX [rx on|off] [tx on|off] [sg on|off] [tso on|off]

ethtool [-p] ethX [N]

ethtool [-t] ethX [offline|online]

ethtool [-s] ethX [speed 10|100|1000] [duplex half|full] [autoneg on|off] [por tp|aui|bnc|mii] [phyad N] [xcvr internal|external]

[wol p|u|m|b|a|g|s|d...] [sopass xx:yy:zz:aa:bb:cc] [msglvl N]  
作用：ethtool命令用于获取以太网卡的配置信息，或者修改这些配置。这个命令比较复杂，

功能特别多。

选项：

❒ -a 查看网卡中 接收模块RX、发送模块TX和Autonegotiate模块的状态：启动on 或 停用off。

❒ -A 修改网卡中 接收模块RX、发送模块TX和Autonegotiate模块的状态：启动on 或 停用off。

❒ -c display the Coalesce information of the specified ethernet card。

❒ -C Change the Coalesce setting of the specified ethernet card。

❒ -g Display the rx/tx ring parameter information of the specified ethernet card。

❒ -G change the rx/tx ring setting of the specified ethernet card。

❒ -i 显示网卡驱动的信息，如驱动的名称、版本等。

❒ -d 显示register dump信息, 部分网卡驱动不支持该选项。

❒ -e 显示EEPROM dump信息，部分网卡驱动不支持该选项。

❒ -E 修改网卡EEPROM byte。

❒-k 显示网卡Offload参数的状态：on 或 off，包括rx-checksumming、tx-checksumming等。

❒ -K 修改网卡Offload参数的状态。

❒ -p 用于区别不同ethX对应网卡的物理位置，常用的方法是使网卡port上的led不断的闪；N指示了网卡闪的持续时间，以秒为单位。

❒ -r 如果auto-negotiation模块的状态为on，则restarts auto-negotiation。

❒ -S 显示NIC- and driver-specific 的统计参数，如网卡接收/发送的字节数、接收/发送的广播包个数等。

❒ -t 让网卡执行自我检测，有两种模式：offline or online。

❒ -s 修改网卡的部分配置，包括网卡速度、单工/全双工模式、mac地址等。

## mii-tool

格式：mii-tool [-VvRrwl] [-A media,... | -F media] [interface ...]  
作用：**linux下专门设置网卡工作模式的命令**

选项：

❒ -v 显示网络接口的信息；

❒-R 重设MII到开启状态；

❒-r 重启自动协商模式；

❒-w 查看网络接口连接的状态变化；

❒-l 写入事件到系统日志；

❒-F 更改网络接口协商方式；

❒-A 指令特定的网络接口；

❒ -V 显示网络接口的信息；

关于 mii-tool的更多用法，你可以通过指定mii-tool -h 选项或man mii-tool查询

示例：

1. 查看网卡的工作状态，

[root@xuegod65 ~]# mii-tool eth0

eth0: negotiated 100baseTx-FD, link ok

2显示网络接口信息，注意，这里是小写的 v

[root@xuegod65 ~]# mii-tool -v eth0

eth0: negotiated 100baseTx-FD, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: autonegotiation enabled

basic status: autonegotiation complete, link ok

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

3 重设MII到开启状态，重设之后，网卡的工作模式会变成半双工模式

重设前：

eth0: negotiated 100baseTx-FD, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: autonegotiation enabled

basic status: autonegotiation complete, link ok

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

重设

[root@xuegod65 ~]# mii-tool -R eth0

resetting the transceiver...

重设后：

[root@xuegod65 ~]# mii-tool -v eth0

eth0: 10 Mbit, half duplex, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode:  **10 Mbit, half duplex**

basic status: **link ok**

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

注意加深颜色部分，发现工作模式已经改变。

4 恢复网卡的自适应模式

恢复前：

[root@xuegod65 ~]# mii-tool -v eth0

eth0: 10 Mbit, half duplex, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: **10 Mbit, half duplex**

basic status: **link ok**

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

恢复：

[root@xuegod65 ~]# mii-tool -r eth0

restarting autonegotiation...

恢复后：

[root@xuegod65 ~]# mii-tool -v eth0

eth0: negotiated 100baseTx-FD, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: **autonegotiation enabled**

basic status: **autonegotiation complete, link ok**

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

注意加深颜色部分，发现状态和工作模式都已经改变。

5 时时动态的查看网络接口连接的状态变化

[root@xuegod65 ~]# mii-tool -w eth0

14:52:50 eth0: negotiated 100baseTx-FD, link ok

6 将当前的连接状态写到系统日志中

[root@xuegod65 ~]# mii-tool -l eth0

eth0: negotiated 100baseTx-FD, link ok

7.若要将 eht0 的工作模式由100M全双工改为10M半双工模式如何修改

更改前，发现网卡工作在100M全双工模式下

[root@xuegod65 ~]# mii-tool -v eth0

eth0: negotiated **100baseTx-FD,** link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: autonegotiation enabled

basic status: autonegotiation complete, link ok

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

现将网卡更改为 10M双半双工

[root@xuegod65 ~]# mii-tool -F 10baseT-HD eth0

更改后，发现工作模式已经改变过来

[root@xuegod65 ~]# mii-tool -v eth0

eth0: **10 Mbit, half duplex**, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: 10 Mbit, half duplex

basic status: link ok

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

8 指令特定的网络接口，如：将网卡的自适应模式开启并启用。

启用前：

[root@xuegod65 ~]# mii-tool -v eth0

eth0: **10 Mbit, half duplex**, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode:  **10 Mbit, half duplex**

basic status: link ok

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

将网卡的自适应模式开启：

root@xuegod65 ~]# mii-tool -A 100baseTx-FD eth0

restarting autonegotiation...

开启后，再次查看，发现网卡的工作模式已经变为100M全双工的工作模式

[root@xuegod65 ~]# mii-tool -v eth0

eth0: **negotiated 100baseTx-FD**, link ok

product info: vendor 00:50:43, model 2 rev 3

basic mode: **autonegotiation enabled**

basic status: autonegotiation complete, link ok

capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD

9 显示版本信息  
root@xuegod65 ~]# mii-tool -V eth0

mii-tool.c 1.9 2000/04/28 00:56:08 (David Hinds)

eth0: negotiated 100baseTx-FD, link ok

更多示例，可以通过网上搜索获取

## dmidecode

格式：dmidecode (选项)

作用：在Linux系统下获取有关硬件方面的信息  
选项：

❒-h：显示帮助信息。

❒-s：只显示指定DMI字符串的信息。(string)

❒-t：只显示指定条目的信息。(type)

❒-V：显示版本信息。

❒ -d：(default:/dev/mem)从设备文件读取信息，输出内容与不加参数标准输出相同。

❒--dump-bin file：将DMI数据转储到一个二进制文件中。

❒--from-dump FILE：从一个二进制文件读取DMI数据。

更多选项可以通过dmidecode --help 或 man dmidecode 获取

示例：

1 直接命令，后面不带选项，则默认查看所有硬件信息,因信息量过大，可以通过 |more 分页查看,若要退出 ，直接输入q 即可

[root@xuegod65 ~]# dmidecode | more

# dmidecode 2.11

SMBIOS 2.7 present.

620 structures occupying 29956 bytes.

Table at 0x000E0010.

Handle 0x0000, DMI type 0, 24 bytes

BIOS Information

Vendor: Phoenix Technologies LTD

Version: 6.00

Release Date: 07/02/2015

Address: 0xEA5E0

Runtime Size: 88608 bytes

ROM Size: 64 kB

Characteristics:

ISA is supported

PCI is supported

PC Card (PCMCIA) is supported

PNP is supported

APM is supported

BIOS is upgradeable

。。。。。。。。。。。。。。。。。。。。。。。。。。

2.获取帮助信息

[root@xuegod65 ~]# dmidecode -h

Usage: dmidecode [OPTIONS]

Options are:

-d, --dev-mem FILE Read memory from device FILE (default: /dev/mem)

-h, --help Display this help text and exit

-q, --quiet More verbose output

-s, --string KEYWORD Only display the value of the given DMI string

-t, --type TYPE Only display the entries of given type

-u, --dump Do not decode the entries

--dump-bin FILE Dump the DMI data to a binary file

--from-dump FILE Read the DMI data from a binary file

-V, --version Display the version and exit

3显示指定DMI字符串的信息

先查看可以查看哪些DMI字符串信息

[root@xuegod65 ~]# dmidecode -s

dmidecode: option requires an argument -- 's'

String keyword expected

Valid string keywords are:

bios-vendor

bios-version

bios-release-date

system-manufacturer

system-product-name

system-version

system-serial-number

system-uuid

baseboard-manufacturer

baseboard-product-name

baseboard-version

baseboard-serial-number

baseboard-asset-tag

chassis-manufacturer

chassis-type

chassis-version

chassis-serial-number

chassis-asset-tag

processor-family

processor-manufacturer

processor-version

processor-frequency

如：要查看 BIOS 的版本信息，后面跟上对应的名称即可

BIOS 的版本信息

[root@xuegod65 ~]# dmidecode -s bios-version

6.00

系统序列号等各种信息

系统序列号

[root@xuegod65 ~]# dmidecode -s system-serial-number

VMware-56 4d 01 01 f1 6d 07 4b-a3 e2 54 84 0e bb 2f a6

4 显示指定条目的信息

先查看可以查看哪些条目信息

[root@xuegod65 ~]# dmidecode -t

dmidecode: option requires an argument -- 't'

Type number or keyword expected

Valid type keywords are:

bios

system

baseboard

chassis

processor

memory

cache

connector

slot

如：要查看 BIOS 的详细信息，后面跟上对应的名称即可

BIOS 的详细信息

[root@xuegod65 ~]# dmidecode -t bios | more

# dmidecode 2.11

SMBIOS 2.7 present.

Handle 0x0000, DMI type 0, 24 bytes

BIOS Information

Vendor: Phoenix Technologies LTD

Version: 6.00

Release Date: 07/02/2015

Address: 0xEA5E0

Runtime Size: 88608 bytes

ROM Size: 64 kB

Characteristics:

ISA is supported

PCI is supported

PC Card (PCMCIA) is supported

PNP is supported

APM is supported

BIOS is upgradeable

BIOS shadowing is allowed

ESCD support is available系统序列号

。。。。。。。。。。。。。。。。。。。。。。。。。。

内存的详细信息等各种信息

[root@xuegod65 ~]# dmidecode -t memory | more

# dmidecode 2.11

SMBIOS 2.7 present.

Handle 0x0084, DMI type 5, 46 bytes

Memory Controller Information

Error Detecting Method: None

Error Correcting Capabilities:

None

Supported Interleave: One-way Interleave

Current Interleave: One-way Interleave

Maximum Memory Module Size: 32768 MB

Maximum Total Memory Size: 491520 MB

Supported Speeds:

70 ns

60 ns

Supported Memory Types:

FPM

EDO

DIMM

SDRAM

。。。。。。。。。。。。。。。。。。。。。。。。。。。

5 从设备文件读取信息，输出内容与不加参数标准输出相同

[root@xuegod65 ~]# dmidecode -d /dev/mem | more

# dmidecode 2.11

SMBIOS 2.7 present.

620 structures occupying 29956 bytes.

Table at 0x000E0010.

Handle 0x0000, DMI type 0, 24 bytes

BIOS Information

Vendor: Phoenix Technologies LTD

Version: 6.00

Release Date: 07/02/2015

Address: 0xEA5E0

Runtime Size: 88608 bytes

ROM Size: 64 kB

Characteristics:

ISA is supported

PCI is supported

PC Card (PCMCIA) is supported

PNP is supported

APM is supported

BIOS is upgradeable

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

6 将DMI数据转储到一个二进制文件中，如要将数据存储到Dmidate.bin 这个文件中

[root@xuegod65 ~]# dmidecode --dump-bin Dmidate.bin

# dmidecode 2.11

SMBIOS 2.7 present.

# Writing 29956 bytes to Dmidate.bin.

# Writing 31 bytes to Dmidate.bin.

7. 从一个二进制文件读取DMI数据，如从Dmidate.bin这个文件中读取数据

[root@xuegod65 ~]# dmidecode --from-dump Dmidate.bin | more

# dmidecode 2.11

Reading SMBIOS/DMI data from file Dmidate.bin.

SMBIOS 2.7 present.

620 structures occupying 29956 bytes.

Handle 0x0000, DMI type 0, 24 bytes

BIOS Information

Vendor: Phoenix Technologies LTD

Version: 6.00

Release Date: 07/02/2015

Address: 0xEA5E0

Runtime Size: 88608 bytes

ROM Size: 64 kB

Characteristics:

ISA is supported

PCI is supported

PC Card (PCMCIA) is supported

PNP is supported

APM is supported

BIOS is upgradeable

8显示版本信息

[root@xuegod65 ~]# dmidecode -V

2.11

更多示例，可以通过网上搜索获取

## dmesg

dmesg(选项)

作用：检查和控制内核的环形缓冲区。

选项：

❒ -c 显示信息后，清除ring buffer中的内容；

❒ -s <缓冲区大小> 预设置为8196，刚好等于ring buffer的大小；

❒ -n 设置记录信息的层级。

❒ -C 清除内核环形缓冲区(ring butter)

❒ -D 禁止向终端打印消息

❒ -d 显示打印消息之间的时间差

❒ -e 以易读格式显示本地时间和时间差

❒ -E 启用向终端打印消息

❒ -F 用文件代替内核日志缓冲区

❒ -f <列表> 将输出限制为定义的设施

❒ -H 易读格式输出

❒ -k 显示内核消息

❒ -L<列表> 显示彩色消息

❒ -l <列表> 限制输出级别

❒ -P 不将输出通过管道传递给分页程序

❒ -r 打印原生消息缓冲区

❒ -S 强制使用 syslog(2) 而非 /dev/kmsg

❒ -T 显示易读的时间戳(如果您使用了SUSPEND/RESUME

则可能不准)

❒ -t 不打印消息时间戳

❒ -u 显示用户空间消息

❒ -w 等待新消息

❒ -x 将设施和级别解码为可读的字符串

❒ -h 显示此帮助并退出

❒ -V 输出版本信息并退出

示例：

[root@xuegod63 ~]# dmesg | head

Initializing cgroup subsys cpuset

Initializing cgroup subsys cpu

Linux version 2.6.32-431.el6.x86\_64 (mockbuild@c6b8.bsys.dev.centos.org) (gcc version 4.4.7 20120313 (Red Hat 4.4.7-4) (GCC) ) #1 SMP Fri Nov 22 03:15:09 UTC 2013

Command line: ro root=UUID=aa300141-a4bc-46e0-b46c-54f5bc6a714d rd\_NO\_LUKS rd\_NO\_LVM LANG=en\_US.UTF-8 rd\_NO\_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_DM rhgb quiet

KERNEL supported cpus:

Intel GenuineIntel

AMD AuthenticAMD

Centaur CentaurHauls

BIOS-provided physical RAM map:

BIOS-e820: 0000000000000000 - 000000000009ec00 (usable)

## lspci

格式：lspci (选项)

作用：用于显示当前主机的所有PCI总线信息，以及所有已连接的PCI设备信息

选项：

❒ -n：以数字方式显示PCI厂商和设备代码  
❒ -b：以总线为中心的视图  
❒ -x：以十六进制显示 PCI 配置空间 (configuration space) 的前64个字节映像 (标准头部信息)

❒ -v：以冗余模式显示所有设备的详细信息

❒ -m：以机器可读方式显示

❒ -t：以树状结构显示PCI设备的层次关系，包括所有的总线、桥、设备以及它们之间的联接

❒-k：显示内核驱动程序处理每个设备和内核模块的能力

❒-q：通过DNS 查询未知ID的PCI

❒-G启用PCI的访问调试

❒ -d ：[<vendor>]:[<device>] 只显示指定生产厂商和设备 ID 的设备。 这两个 ID 都以十六进制表示，可以忽略或者以「\*」代替显示PCI设备信息

❒ -i <file> ：使用 <file> 作为 PCI ID 数据库而不是使用预设的 /usr/share/hwdata/pci.ids。

❒-p <dir> ：使用 <dir> 作为包含 PCI 总线信息的目录而不是使用预设的目录 /proc/bus/pci

❒ -s [[<bus>]:][<slot>][.[<func>]] 仅显示指定总线、插槽上的设备或设备上的功能块信

关于 lspci 的更多用法，你可以通过指定 lspci -h 选项或man lspci查询

示例：

1 以数字方式显示PCI厂商和设备代码, 因信息量过大，可以通过 | more 分页查看,若要退出，直接输入q 即可

[root@xuegod65 ~]# lspci –n | more

00:00.0 0600: 8086:7190 (rev 01)

00:01.0 0604: 8086:7191 (rev 01)

00:07.0 0601: 8086:7110 (rev 08)

00:07.1 0101: 8086:7111 (rev 01)

00:07.3 0680: 8086:7113 (rev 08)

00:07.7 0880: 15ad:0740 (rev 10)

00:0f.0 0300: 15ad:0405

00:10.0 0100: 1000:0030 (rev 01)

00:11.0 0604: 15ad:0790 (rev 02)

00:15.0 0604: 15ad:07a0 (rev 01)

00:15.1 0604: 15ad:07a0 (rev 01)

00:15.2 0604: 15ad:07a0 (rev 01)

00:15.3 0604: 15ad:07a0 (rev 01)

00:15.4 0604: 15ad:07a0 (rev 01)

00:15.5 0604: 15ad:07a0 (rev 01)

00:15.6 0604: 15ad:07a0 (rev 01)

00:15.7 0604: 15ad:07a0 (rev 01)

00:16.0 0604: 15ad:07a0 (rev 01)

00:16.1 0604: 15ad:07a0 (rev 01)

00:16.2 0604: 15ad:07a0 (rev 01)

。。。。。。。。。。。。。。。。。。

2显示以总线为中心的视图

[root@xuegod65 ~]# lspci –b | more

0 00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual Ultra3

20 SCSI (rev 01)

00:11.0 PCI bridge: VMware PCI bridge (rev 02)

00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)

。。。。。。。。。。。。。。。。。。。。。。。。。。

3 以十六进制显示 PCI 配置空间 (configuration space) 的前64个字节映像 (标准头部信

息)

root@xuegod65 ~]# lspci –x | more

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

00: 86 80 90 71 06 00 00 02 01 00 00 06 00 00 00 00

10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

20: 00 00 00 00 00 00 00 00 00 00 00 00 ad 15 76 19

30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)

00: 86 80 91 71 1f 01 20 02 01 00 04 06 00 00 01 00

10: 00 00 00 00 00 00 00 00 00 01 01 40 f0 00 a0 02

20: f0 ff 00 00 f0 ff 00 00 00 00 00 00 00 00 00 00

30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 00

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

00: 86 80 10 71 07 00 80 02 08 00 01 06 00 00 80 00

10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

20: 00 00 00 00 00 00 00 00 00 00 00 00 ad 15 76 19

30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)

00: 86 80 11 71 05 00 80 02 01 8a 01 01 00 40 00 00

。。。。。。。。。。。。。。。。。。。

4 以冗余模式显示所有设备的详细信息

[root@xuegod65 ~]# lspci –v | more

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

Subsystem: VMware Virtual Machine Chipset

Flags: bus master, medium devsel, latency 0

Kernel driver in use: agpgart-intel

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01) (prog-if

00 [Normal decode])

Flags: bus master, 66MHz, medium devsel, latency 0

Bus: primary=00, secondary=01, subordinate=01, sec-latency=64

Kernel modules: shpchp

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

Subsystem: VMware Virtual Machine Chipset

Flags: bus master, medium devsel, latency 0

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01) (prog-if 8a [Master S

ecP PriP])

Subsystem: VMware Virtual Machine Chipset

Flags: bus master, medium devsel, latency 64

[virtual] Memory at 000001f0 (32-bit, non-prefetchable) [size=8]

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

5.以机器可读方式显示PCI设备信息

[root@xuegod65 ~]# lspci –m | more

00:00.0 "Host bridge" "Intel Corporation" "440BX/ZX/DX - 82443BX/ZX/DX Host bridge" -r01 "VMwar

e" "Virtual Machine Chipset"

00:01.0 "PCI bridge" "Intel Corporation" "440BX/ZX/DX - 82443BX/ZX/DX AGP bridge" -r01 "" ""

00:07.0 "ISA bridge" "Intel Corporation" "82371AB/EB/MB PIIX4 ISA" -r08 "VMware" "Virtual Machi

ne Chipset"

00:07.1 "IDE interface" "Intel Corporation" "82371AB/EB/MB PIIX4 IDE" -r01 -p8a "VMware" "Virtu

al Machine Chipset"

00:07.3 "Bridge" "Intel Corporation" "82371AB/EB/MB PIIX4 ACPI" -r08 "VMware" "Virtual Machine

Chipset"

00:07.7 "System peripheral" "VMware" "Virtual Machine Communication Interface" -r10 "VMware" "V

irtual Machine Communication Interface"

00:0f.0 "VGA compatible controller" "VMware" "SVGA II Adapter" "VMware" "SVGA II Adapter"

00:10.0 "SCSI storage controller" "LSI Logic / Symbios Logic" "53c1030 PCI-X Fusion-MPT Dual Ul

tra320 SCSI" -r01 "VMware" "LSI Logic Parallel SCSI Controller"

00:11.0 "PCI bridge" "VMware" "PCI bridge" -r02 -p01 "" ""

00:15.0 "PCI bridge" "VMware" "PCI Express Root Port" -r01 "" ""

00:15.1 "PCI bridge" "VMware" "PCI Express Root Port" -r01 "" ""

00:15.2 "PCI bridge" "VMware" "PCI Express Root Port" -r01 "" ""

00:15.3 "PCI bridge" "VMware" "PCI Express Root Port" -r01 "" ""

00:15.4 "PCI bridge" "VMware" "PCI Express Root Port" -r01 "" ""。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

6 以树状结构显示PCI设备的层次关系，包括所有的总线、桥、设备以及它们之间的联接

[root@xuegod65 ~]# lspci –t | more

-[0000:00]-+-00.0

+-01.0-[01]--

+-07.0

+-07.1

+-07.3

+-07.7

+-0f.0

+-10.0

+-11.0-[02]--+-00.0

| +-01.0

| +-02.0

| \-03.0

+-15.0-[03]--

+-15.1-[04]--

+-15.2-[05]--

+-15.3-[06]--

+-15.4-[07]--

+-15.5-[08]--

+-15.6-[09]--

+-15.7-[0a]—

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

7 显示内核驱动程序处理每个设备和内核模块的能力

[root@xuegod65 ~]# lspci -k | more

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

Subsystem: VMware Virtual Machine Chipset

Kernel driver in use: agpgart-intel

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)

Kernel modules: shpchp

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

Subsystem: VMware Virtual Machine Chipset

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)

Subsystem: VMware Virtual Machine Chipset

Kernel driver in use: ata\_piix

Kernel modules: ata\_generic, pata\_acpi, ata\_piix

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

Subsystem: VMware Virtual Machine Chipset

Kernel modules: i2c-piix4

00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)

Subsystem: VMware Virtual Machine Communication Interface

Kernel driver in use: vmci

Kernel modules: vmci

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

Subsystem: VMware SVGA II Adapter

8 通过DNS 查询未知ID的PCI

[root@xuegod65 ~]# lspci -q | more

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual Ultra3

20 SCSI (rev 01)

00:11.0 PCI bridge: VMware PCI bridge (rev 02)

00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

9 启用PCI的访问调试功能

[root@xuegod65 ~]# lspci -G | more

Trying method 1......using /sys/bus/pci...OK

Decided to use linux-sysfs

00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)

00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)

00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)

00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)

00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)

00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual Ultra3

20 SCSI (rev 01)

00:11.0 PCI bridge: VMware PCI bridge (rev 02)

00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)

00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)

**扩展应用**

查看指定的PCI设备信息，**| grep** 表示从数据中将我们需要的数据过滤出来, **-i**不区分大小写。如：

查看网卡信息

[root@xuegod65 ~]# lspci | grep -i eth

02:01.0 Ethernet controller: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) (rev 01)

查看显卡信息

[root@xuegod65 ~]# lspci | grep -i vga

00:0f.0 VGA compatible controller: VMware SVGA II Adapter

更多示例，可以通过网上搜索获取

## tac

格式：tac(选项)(参数)

作用：tac 将文件内容的反序输出，即第一行最后显示，最后一行先显示。

选项：

❒ -a或--append： 将内容追加到文件的末尾；

❒ -i或--ignore-interrupts：忽略中断信号。

参数：

文件列表：指定要保存内容的文件列表。  
示例：

将/etc/passwd 前五行内容生成到/root/a.txt 文件中

[root@localhost ~]# head -n 5 /etc/passwd > ~/a.txt

[root@localhost ~]# cat a.txt

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

[root@localhost ~]# tac a.txt

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

bin:x:1:1:bin:/bin:/sbin/nologin

root:x:0:0:root:/root:/bin/bash

## chattr

格式：chattr [-RuV] [文件]

作用：改变文件属性

说明：

文件属性分8 种模式：

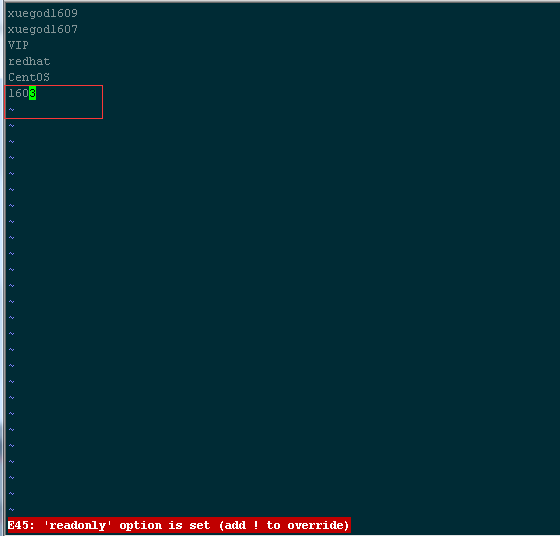
* a：让文件或目录仅供附加用途
* b：不更新文件或目录的最后存取时间
* c：将文件或目录压缩后存放
* d：将文件或目录排除在倾倒操作之外
* i：不得任意更动文件或目录
* s：保密性删除文件或目录
* S：即时更新文件或目录
* u：预防意外删除

选项：

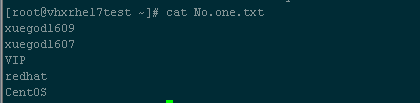
* -R：递归处理，将指令目录下的所有文件及子目录一并处理
* -V：显示指令执行过程
* +<属性>：开启文件或目录的该项属性
* -<属性>：关闭文件或目录的该项属性
* =<属性>：指定文件或目录的该项属性

示例：

[root@vhxrhel7test ~]# chattr +i No.one.txt //添加不得任意更动文件或目录



提示无法向文件中增加内容。



[root@vhxrhel7test ~]# chattr -i No.one.txt //移除不得任意更动文件或目录

## who

格式：who [选项] [参数]

作用：显示目前登录系统的用户信息，执行这项指令可得知目前登录系统的用户，单独执行who指令会列出登入系统的账号，使用的终端，登录的时间以及从何处登录或正在使用那个X显示器

选项：

❒ -H或--heading 显示各栏位的标题信息列

❒ -i或-u或--idle 显示闲置时间，若该用户在前一分钟之内有进行任何动作，将标示成“.”号，如果该用户已超过24小时没有任何动作，则标示出“old”字符串

❒ -m 此参数的效果和指定“am i”字符串相同

❒ -q或--count 只显示登入系统的帐号名称和总人数

❒ -s 此参数将忽略不予处理，仅负责解决who指令其他版本的兼容性问题

❒ -w或T或--mess或--message或--writable 显示用户的信息状态栏

❒ --help 在线帮助

❒ --version 显示版本信息

注意：

示例一：

显示登录系统的用户和登录的地点，还有终端类型

[root@localhost mnt]# who

root pts/0 2016-11-26 05:19 (192.168.0.103)

示例二：

显示系统的运行级别

[root@localhost ~]# who -r

run-level 3 2016-11-26 05:17

示例三：

显示系统当前登录的用户总数及用户名称

[root@localhost ~]# who -q

yeyan root root

# users=3

示例四：

模拟三个用户登录主机，分别用whoami，w，who命令进行输出，对比whoami，w，who三个命令输出的不同

[root@localhost ~]# whoami

root

[root@localhost ~]# w

08:31:35 up 3:14, 3 users, load average: 0.00, 0.00, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

yeyan tty1 - 08:26 5:01 0.00s 0.00s -bash

root pts/0 192.168.0.103 05:19 5:17 0.15s 0.15s -bash

root pts/1 192.168.0.103 08:27 0.00s 0.01s 0.00s w

[root@localhost ~]# who

yeyan tty1 2016-11-26 08:26

root pts/0 2016-11-26 05:19 (192.168.0.103)

root pts/1 2016-11-26 08:27 (1

## users

格式：users(选项)

作用：users命令用于显示当前登录系统的所有用户的用户列表。每个显示的用户名对应一

个登录会话。如果一个用户有不止一个登录会话，那他的用户名将显示相同的次数。

选项：

❒--help：显示命令的帮助信息；

❒--version：显示命令的版本信息。  
示例：

查看并且打印当前登录系统所有用户。

[root@haobo66 ~]# users

root root

## echo

作用  
在shell中打印shell变量的值，或者直接输出指定的字符串

格式

echo [-e] [指定要打印的变量]

主要参数说明

-e：激活转义字符。

## file

file(选项)(参数)

作用：探测给定文件的类型

选项：

❒ -b 列出辨识结果时，不显示文件名称；

❒ -c 详细显示指令执行过程，便于排错或分析程序执行的情形；

❒ -f<名称文件> 指定名称文件，其内容有一个或多个文件名称时，让file依序辨

识这些文件，格式为每列一个文件名称；

❒ -F 使用指定分隔符号替换输出文件名后的默认的“:”分隔符

❒ -L 直接显示符号连接所指向的文件类别；

❒ -h 不跟随符号链接

❒ -i--mime 输出mime类型的字符串

--apple 输出Apple CREATOR/TYPE

--mime-type 输出MIME类别

--mime-encoding 输出MIME类别

❒ -k 执行错误命令式不终止

❒ -m<魔法数字文件> 指定魔法数字文件；

❒ -v 显示版本信息；

❒ -z 尝试去解读压缩文件的内容。

❒ --help 显示帮助并退出；

参数：要确定类型的文件列表，多个文件之间使用空格分开，可以使用shell通配符匹配多

个文件

示例：

[root@localhost ~]# file install.log

install.log: ASCII ext

[root@localhost ~]# file -b install.log

ASCII tex

[root@localhost ~]# file -z boot.zip

boot.zip: ASCII English text (Zip archive data, at least v2.0 to extract)

[root@localhost ~]# file -z fail2ban-0.8.14.tar.gz

fail2ban-0.8.14.tar.gz: POSIX tar archive (gzip compressed data, from Unix)

[root@localhost ~]# file -L /bin/awk

/bin/awk: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18, stripped

[root@localhost ~]# file /bin/gawk

/bin/gawk: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18, stripped

[root@localhost test]# ls

1 2 3

[root@localhost test]# ls >a

[root@localhost test]# ls

1 2 3 a

[root@localhost test]# file -f a

1: empty

2: empty

3: empty

a: ASCII text