# Linux命令

本章，我们将重点讨论学习CentOS的入门操作，基本命令，vi编辑器等功能。

## CentOS基本使用

如果安装的是CentOS的图形界面的Linux，那么你可以仔细熟悉一下Linux的桌面环境等。

登录`CentOS`图形界面后，熟悉`CentOS`的操作界面，文件管理器，系统配置等。

在桌面或者文件管理器中右键|打开终端，认识终端界面。

## 远程操作

由于在实际的工作环境中，普通用户很难直接接触到装有Linux系统的服务器或者主机，主要的工作方式便是远程登录Linux主机，使用Xshell，SecureCRT或者PuTTY这样的工具进行操作。

- SecureCRT：一个老牌的付费的远程工具
- PuTTY：免费简易的远程工具
- Xshell：一个新款的非商用免费的工具

在实际使用的过程中，一个很重要的按键：Tab键的使用尤其关键，主要用来自动补全。

- Tab键的作用
    - 自动补全命令
    - 自动补全文件名

推荐使用 Xshell进行远程操作

## CentOS终端操作

### 常用终端快捷键

| 功能 | 快捷键组合 | 备注 |
| --- | --- | --- |
| 终端窗口最大化 | `Alt F10` | |
| 放大字体 | `Ctrl Shift +` | |
| 减小字体 | `Ctrl -` | |
| 打开新的终端窗口 | `Ctrl Shift n` | |
| 打开新的终端标签 | `Ctrl Shift t` | |
| 切换到第n个终端标签 | `Alt n` | |
| 清屏 | `Ctrl l` | 也可以使用命令`clear` |
| 退出终端 | `Ctrl d` | 也可以使用命令`exit` |
| 删除光标到命令提示符之间的所有字符 | `Ctrl u` | |

## 命令行格式

```
[root@localhost ~]# command [-options] parameter1 parameter2 ...
```

| 项目 | 描述 | 备注 |
| --- | --- | --- |
| [root@localhost ~] | 命令行前面的提示符 | |
| root | 当前用户的用户名 | |
| @localhost | Linux主机名 | |
| ~ | 当前所在的目录 | |
| `#($)` | 代表当前用户是root（代表当前用户是普通用户） | |
| command | 命令，注意命令需要区分大小写 | |
| -options | 选项，一般可选 | |
| parameter1 | 参数 | |

ps：命令和选项之间、选项和参数之间以及参数之间需要空格隔开*

# 常见的命令

## date

查看当前的系统时间 `date`

```
[root@localhost ~]# date
Tue Dec 22 02:17:54 PST 2015
```

### uname

查看当前的系统版本号 `uname`

```
[root@localhost ~]# uname -r
2.6.18-194.el5
[root@localhost ~]# uname -m
i686
```

### cal

查看当前的系统日期 `cal`

```
[root@localhost ~]# cal
December 2015
Su Mo Tu We Th Fr Sa
1  2  3  4  5
6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

## 文件操作

**文件路径：**

绝对路径和相对路径

- 绝对路径：从根目录开始的路径，一定以`/`开头
- 相对路径：从当前目录开始的路径，一定不以`/`开头，而可以以`..`，`.`开头，其中`..`代表当前目录的上一层目录，`.`代表当前目录。

### 主要的目录

| 目录 | 描述 | |
| --- | --- | --- |
| `/bin` | 放的普通用户命令 | |
| `/sbin` | 放的管理员用户的命令 | |
| `/boot` | 放的是系统启动所需要的文件 | |
| `/dev` | 放的是linux系统下的设备管理文件，比如:cd-rom、u盘、磁盘。 | |
| `/etc` | 放的是系统里的所有配置文件 | |
| `/home` | 放的是各用户的用户信息，类似于windows操作系统下的"我的文档"。 | |
| `/lib` | 放的是动态链接库 | |
| `/lost+found` | 回收站 | |
| `/mnt` | 临时挂载目录 | |
| `/opt` | 用户软件安装目录，类似windows下program files。 | |
| `/tmp` | 临时目录 | |

## ls

查看目录下的文件 - list directory contents `ls`

- 不加选项和参数

```
[root@localhost ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog
```

- 只添加选项 `-l -a -t`

*显示详细文件格式 -l 或者 ll*

```
[root@localhost ~]# ls -l
total 52
-rw------- 1 root root   879 Dec 21 08:38 anaconda-ks.cfg
-rw-r--r-- 1 root root 28693 Dec 21 08:38 install.log
-rw-r--r-- 1 root root  3812 Dec 21 08:37 install.log.syslog
```

*ll等同ls -l*

```
[root@localhost ~]# ll
total 52
-rw------- 1 root root   879 Dec 21 08:38 anaconda-ks.cfg
-rw-r--r-- 1 root root 28693 Dec 21 08:38 install.log
-rw-r--r-- 1 root root  3812 Dec 21 08:37 install.log.syslog
```

*显示全部文件（包含隐藏文件） -a*

```
[root@localhost ~]# ls -a
..bash_logout   .cshrc  .tcshrc
..   .bash_profile  install.log .xauthT1oORi
anaconda-ks.cfg  .bashrcinstall.log.syslog
```

*照最后修改的日期倒序排列 -t*

```
[root@localhost ~]# ls -t
anaconda-ks.cfg  install.log  install.log.syslog
```

- 选项可以合并使用

```
[root@localhost ~]# ls -la
total 116
drwxr-x---  2 root root  4096 Dec 22 02:17 .
drwxr-xr-x 23 root root  4096 Dec 22 02:15 ..
-rw-------  1 root root   879 Dec 21 08:38 anaconda-ks.cfg
-rw-r--r--  1 root root24 Jul 12  2006 .bash_logout
-rw-r--r--  1 root root   191 Jul 12  2006 .bash_profile
-rw-r--r--  1 root root   176 Jul 12  2006 .bashrc
-rw-r--r--  1 root root   100 Jul 12  2006 .cshrc
-rw-r--r--  1 root root 28693 Dec 21 08:38 install.log
-rw-r--r--  1 root root  3812 Dec 21 08:37 install.log.syslog
-rw-r--r--  1 root root   129 Jul 12  2006 .tcshrc
-rw-------  1 root root59 Dec 22 02:17 .xauthT1oORi
```

```
[root@localhost ~]# ls -lt
total 52
-rw------- 1 root root   879 Dec 21 08:38 anaconda-ks.cfg
-rw-r--r-- 1 root root 28693 Dec 21 08:38 install.log
-rw-r--r-- 1 root root  3812 Dec 21 08:37 install.log.syslog
```

```
[root@localhost ~]# ls -at
. anaconda-ks.cfg .bash_logout   .cshrc
.xauthT1oORi  install.log .bash_profile  .tcshrc
..install.log.syslog  .bashrc
```

```
[root@localhost ~]# ls -lat
total 116
drwxr-x---  2 root root  4096 Dec 22 02:17 .
-rw-------  1 root root59 Dec 22 02:17 .xauthT1oORi
drwxr-xr-x 23 root root  4096 Dec 22 02:15 ..
-rw-------  1 root root   879 Dec 21 08:38 anaconda-ks.cfg
-rw-r--r--  1 root root 28693 Dec 21 08:38 install.log
-rw-r--r--  1 root root  3812 Dec 21 08:37 install.log.syslog
-rw-r--r--  1 root root24 Jul 12  2006 .bash_logout
-rw-r--r--  1 root root   191 Jul 12  2006 .bash_profile
-rw-r--r--  1 root root   176 Jul 12  2006 .bashrc
-rw-r--r--  1 root root   100 Jul 12  2006 .cshrc
-rw-r--r--  1 root root   129 Jul 12  2006 .tcshrc
```

- 只添加参数

```
* 查看指定目录下的文件
```

```
[root@localhost ~]# ls /boot/grub/
device.map grub.conf minix_stage1_5 stage2
e2fs_stage1_5  iso9660_stage1_5  reiserfs_stage1_5  ufs2_stage1_5
fat_stage1_5   jfs_stage1_5  splash.xpm.gz  vstafs_stage1_5
ffs_stage1_5   menu.lst  stage1 xfs_stage1_5
```

- 同时添加选项和参数

*以详细的格式查看指定目录下的文件*

```
[root@localhost ~]# ls -l /boot/grub/
total 234
-rw-r--r-- 1 root root 63 Dec 21 08:38 device.map
-rw-r--r-- 1 root root   7584 Dec 21 08:38 e2fs_stage1_5
-rw-r--r-- 1 root root   7456 Dec 21 08:38 fat_stage1_5
-rw-r--r-- 1 root root   6720 Dec 21 08:38 ffs_stage1_5
-rw------- 1 root root598 Dec 21 08:38 grub.conf
-rw-r--r-- 1 root root   6720 Dec 21 08:38 iso9660_stage1_5
-rw-r--r-- 1 root root   8192 Dec 21 08:38 jfs_stage1_5
lrwxrwxrwx 1 root root 11 Dec 21 08:38 menu.lst -> ./grub.conf
-rw-r--r-- 1 root root   6880 Dec 21 08:38 minix_stage1_5
-rw-r--r-- 1 root root   9248 Dec 21 08:38 reiserfs_stage1_5
-rw-r--r-- 1 root root  32428 Jan  4  2007 splash.xpm.gz
-rw-r--r-- 1 root root512 Dec 21 08:38 stage1
-rw-r--r-- 1 root root 104988 Dec 21 08:38 stage2
-rw-r--r-- 1 root root   7072 Dec 21 08:38 ufs2_stage1_5
-rw-r--r-- 1 root root   6272 Dec 21 08:38 vstafs_stage1_5
-rw-r--r-- 1 root root   8904 Dec 21 08:38 xfs_stage1_5
```

- 对于命令的帮助查询 man 或者 --help

  输入 man ls （需要再输入q去结束）

  或者 ls --help

## pwd

查看当前目录的绝对路径 - print working directory `pwd`

```
[root@localhost grub]# pwd
/boot/grub
```

```
[root@localhost grub]# cd ~
[root@localhost ~]# pwd
/root
```

## cd

切换当前工作目录 - change directory `cd`

*参数只能是目录的路径*

```
[root@localhost conf]# cd  /etc/ssh/sshd_config
bash: cd: /etc/ssh/sshd_config: Not a directory
```

*参数指定切换到哪个目录*

```
[root@localhost conf]# cd /etc/ssh/
[root@localhost ssh]# pwd
/etc/ssh
```

*不添加参数，则默认切换到当前用户家目录*

```
[root@localhost ssh]# cd
[root@localhost ~]#
```

*切换到上一次所在目录*

```
[root@localhost ~]# cd -
/etc/httpd
```

## mkdir

创建目录 - make directory `mkdir`

*在/usr下创建子目录，名字叫bwf*

```
[root@localhost ~]# mkdir /usr/bwf
```

*在当前目录下创建子目录bwf*

```
[root@localhost ~]# mkdir bwf
```

*-p 当路径中某些目录不存在时，先创建被依赖的目录*

```
[root@localhost ~]# mkdir -p  test/bwf
[root@localhost ~]# ls test/
bwf
```

## touch

创建文件或者更新文件最后修改时间(如果文件已经存在) `touch`

*在/下创建普通文件file1，在当前目录创建file2*

```
[root@localhost ~]# touch /file1
[root@localhost ~]# touch file2
```

## cp

复制文件和目录 - copy files and directories `cp`

```
格式：
cp [-选项] 源文件 目标文件
```

-f 强制拷贝，当目标文件存在时，直接覆盖

-r 针对目录，递归拷贝目录和里面的所有子对象

*把当前目录下的file复制到/usr下*

```
# cp file  /usr
```

*把当前目录下的file复制到/usr下，名字改为filex*

```
# cp file  /usr/filex
```

*把/usr下的filex拷贝到当前目录下*

```
# cp /usr/filex .
```

*把/usr下的filex拷贝到当前目录下,名字改为abc*

```
# cp /usr/filex ./abc
```

*把当前目录下的test目录和里面的所有子对象拷贝到/usr下*

```
# cp  -r test  /usr
```

*把当前目录下的test目录和里面的所有子对象拷贝到/usr下,名字改为test2*

```
# cp  -r test  /usr/test2
```

*cp -fr 通用选项，强制复制*

### rmdir

rmdir- remove empty directories

删除空目录

### rm

删除文件或目录 - remove files or directories `rm`

-f 强制删除

-r 删除目录和目录里的所有文件

*删除当前目录下的文件file*

---

```
# rm -f file
```

*删除当前目录下的test目录和里面的所有文件*

```
# rm -fr test
```

*rm -rf 通用选项*

### mv

移动文件或重命名文件 - move (rename) files `mv`

*把/usr下的filex移动到当前目录下*

```
# mv /usr/filex  .
```

*把/usr下的filex移动到当前目录下,名字改为file*

```
# mv /usr/filex ./file
```

*把当前目录下的file文件名字改为filex*

```
# mv file  filex
```

### cat

查看文件并打印到标准输出 - concatenate files and print on the standard output `cat`

```
[root@localhost ~]# cat filex
hello aaaaa xxxxx
test test bwf class

test test bwf class

fffffffffff fffffff
```

*显示行号*

```
[root@localhost ~]# cat -n filex
 1  hello aaaaa xxxxx
 2  test test bwf class
 3
 4  test test bwf class
 5
 6  fffffffffff fffffff
```

*不显示空白行并显示行号*

```
[root@localhost ~]# cat -b filex
 1  hello aaaaa xxxxx
 2  test test bwf class

 3  test test bwf class

 4  fffffffffff fffffff
```

## tac

从最后一行开始查看到第一行 - concatenate and print files in reverse `tac`

```
[root@localhost ~]# tac filex
fffffffffff fffffff

test test bwf class

test test bwf class
hello aaaaa xxxxx
```

## head

输出文件的第一部分 - output the first part of files `head`

*默认查看文件前10行*

```
[root@localhost ~]# head filex
1hello aaaaa xxxxx
test test bwf class

4test test bwf class

6fffffffffff fffffff
7hello aaaaa xxxxx
8est test bwf class

10test test bwf class
```

*查看文件前4行*

```
[root@localhost ~]# head -n 4 filex
1hello aaaaa xxxxx
test test bwf class

4test test bwf class
```

*查看文件除了最后4行以外的部分*

```
[root@localhost ~]# head -n -4 filex
1hello aaaaa xxxxx
test test bwf class

4test test bwf class

6fffffffffff fffffff
7hello aaaaa xxxxx
8est test bwf class
```

## tail

输出文件的最后部分 - output the last part of files `tail`

*默认查看最后10行*

```
[root@localhost ~]# tail filex
4test test bwf class

6fffffffffff fffffff
7hello aaaaa xxxxx
8est test bwf class

10test test bwf class

12fffffffffff fffffff
13 bwf java test ui
```

*查看文件最后2行*

```
[root@localhost ~]# tail -2 filex
12fffffffffff fffffff
13 bwf java test ui
```

```
[root@localhost ~]# tail -n 2 filex

12fffffffffff fffffff

13 bwf java test ui
```

## more

```
[root@localhost ~]# more filex
```

向下翻页 - file perusal filter for crt viewing `more`

只能向下翻页，翻到文档最后，命令结束

## less

向上翻页 opposite of more `less`

可以上下翻页，（j向上、k向下、q键退出）

## sed

流编辑器过滤和转换文本 - stream editor for filtering and transforming text `sed`

*查看文档第4到7行*

```
[root@localhost ~]# sed -n '4,7p' filex

4test test bwf class



6fffffffffff fffffff

7hello aaaaa xxxxx
```

*查看第4行和第7行*

```
[root@localhost ~]# sed -n -e '4p' -e '7p' filex

4test test bwf class

7hello aaaaa xxxxx
```

## tar

打包和解压 - The GNU version of the tar archiving utility `tar`

选项与参数：

| 选项 | 详情 |
| --- | --- |
| `-c` | 建立压缩文件，可以搭配`-v`来查看过程中被打包的文件名称 `create` |
| `-t` | 查看压缩文件的内容含有哪些文件，重点在查看文件名称 |
| `-x` | 解压压缩文件，可以搭配`-C`在特定目录解压缩 `extract` |
| `-j` | 通过bzip2的程序进行压缩和解压缩，此时的文件名称一般是`*.tar.bz2` |
| `-z` | 通过gzip的支持进行压缩和解压缩，此时的文件名称一般是`*.tar.gz` |
| `-v` | 在压缩和解压缩的过程中，将正在处理的文件名称显示出来。 |
| `-f` | 后面要立刻接要被处理的文件名称，建议`-f`单独写选项 |
| `-C` | 这个选项在解压缩的时候，指定特定的目录进行解压缩 |
| `-p` | 保留备份数据的原本权限和属性，常用来备份重要的配置文件 |

*压缩文件* `-cvf`或者`-cv -f`

```
[root@localhost tmp]# tar -cv -f bluetooth.tar bluetooth/
bluetooth/
bluetooth/rfcomm.conf
bluetooth/hcid.conf
```

*查看压缩文件内容* `-tvf` 或者 `-tv -f`

```
[root@localhost tmp]# tar -tv -f bluetooth.tar
drwxr-xr-x root/root 0 2015-12-23 03:06:04 bluetooth/
-rw-r--r-- root/root   297 2015-12-23 03:06:04 bluetooth/rfcomm.conf
-rw-r--r-- root/root  1231 2015-12-23 03:06:04 bluetooth/hcid.conf
```

*解压缩文件* `-xvf` 或者 `-xv -f`

```
[root@localhost tmp]# tar -xv -f bluetooth.tar -C testtar
bluetooth/
bluetooth/rfcomm.conf
bluetooth/hcid.conf
```

*使用gzip压缩* `-zcvf` 或者 `-zcv -f`

```
[root@localhost tmp]# tar -zcv -f bluetooth.tar.gz bluetooth/
bluetooth/
bluetooth/rfcomm.conf
bluetooth/hcid.conf
```

*使用gzip查看压缩文件* `-ztvf` 或者 `-ztv -f`

```
[root@localhost tmp]# tar -ztv -f bluetooth.tar.gz
drwxr-xr-x root/root 0 2015-12-23 03:06:04 bluetooth/
-rw-r--r-- root/root   297 2015-12-23 03:06:04 bluetooth/rfcomm.conf
-rw-r--r-- root/root  1231 2015-12-23 03:06:04 bluetooth/hcid.conf
```

*使用gzip解压* `-zxvf` 或者 `-zxv -f`

```
[root@localhost tmp]# tar -zxv -f bluetooth.tar.gz -C testtar
bluetooth/
bluetooth/rfcomm.conf
bluetooth/hcid.conf
```

*使用bzip2压缩* `-jcvf` 或者 `-jcv -f`

```
[root@localhost tmp]# tar -jcv -f bluetooth.tar.bz2 bluetooth/
bluetooth/
bluetooth/rfcomm.conf
bluetooth/hcid.conf
```

*使用bzip2查看压缩文件* `-jtvf`或者`-jtv -f`

```
[root@localhost tmp]# tar -jtv -f bluetooth.tar.bz2
drwxr-xr-x root/root 0 2015-12-23 03:06:04 bluetooth/
-rw-r--r-- root/root   297 2015-12-23 03:06:04 bluetooth/rfcomm.conf
-rw-r--r-- root/root  1231 2015-12-23 03:06:04 bluetooth/hcid.conf
```

*使用bzip2解压缩* `-jxvf`或者`-jxv -f`

```
[root@localhost tmp]# tar -jxv -f bluetooth.tar.bz2 -C testtar
bluetooth/
bluetooth/rfcomm.conf
bluetooth/hcid.conf
```

*比较三种压缩方式产生的压缩文件的大小*

```
[root@localhost tmp]# ll bluetooth*
-rw-r--r-- 1 root root 10240 Dec 23 03:07 bluetooth.tar
-rw-r--r-- 1 root root   944 Dec 23 03:46 bluetooth.tar.bz2
-rw-r--r-- 1 root root   868 Dec 23 03:42 bluetooth.tar.gz
```

可以看出来，.tar的文件要大很多，相比之下gzip和bzip2压缩的文件相对较小。

## zip / unzip

*zip*

打包和压缩文件 - package and compress (archive) files

不带选项，只能打包`首层`目录的文件

```
[root@'CentOS' testzip]# zip zipfiles1.zip zipfiles/
  adding: zipfiles/subzip1/ (stored 0%)
  adding: zipfiles/subzip2/ (stored 0%)
  adding: zipfiles/subzip3/ (stored 0%)
```

加上选项 `-r`，递归打包目标目录中的所有文件和子目录的文件

```
[root@'CentOS' testzip]# zip -r zipfiles2.zip zipfiles/
  adding: zipfiles/subzip1/ (stored 0%)
  adding: zipfiles/subzip1/subzip11/ (stored 0%)
  adding: zipfiles/subzip1/subzip11/sf12 (stored 0%)
  adding: zipfiles/subzip1/subzip11/sf13 (stored 0%)
  adding: zipfiles/subzip1/subzip11/sf11 (stored 0%)
  adding: zipfiles/subzip1/sub111 (stored 0%)
  adding: zipfiles/subzip2/ (stored 0%)
  adding: zipfiles/subzip2/s3 (stored 0%)
  adding: zipfiles/subzip2/s4 (stored 0%)
  adding: zipfiles/subzip2/s2 (stored 0%)
  adding: zipfiles/subzip3/ (stored 0%)
  adding: zipfiles/subzip3/s33 (stored 0%)
```

对比结果，可以看出没有使用选项-r的命令压缩的文件小很多。在下面的内容中我们可以查看压缩文件的内容

```
[root@'CentOS' testzip]# ll
total 12
drwxr-xr-x. 5 root root 4096 Dec 30 23:28 zipfiles
-rw-r--r--. 1 root root  508 Dec 30 23:33 zipfiles1.zip
-rw-r--r--. 1 root root 2092 Dec 30 23:34 zipfiles2.zip
```

*unzip*

列出、校验和解压缩zip文件 - list, test and extract compressed files in a ZIP archive

*查看已经压缩过的zip文件里面的内容*

```
[root@'CentOS' testzip]# unzip -v zipfiles1.zip
Archive:  zipfiles1.zip
 Length   MethodSize  Cmpr    Date    Time   CRC-32   Name
--------  ------  ------- ---- ---------- ----- --------  ----
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip2/
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip3/
--------      -------  ---                            -------
   0               0   0%                            3 files
```

```
[root@'CentOS' testzip]# unzip -v zipfiles2.zip
Archive:  zipfiles2.zip
 Length   MethodSize  Cmpr    Date    Time   CRC-32   Name
--------  ------  ------- ---- ---------- ----- --------  ----
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/subzip11/
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/subzip11/sf12
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/subzip11/sf13
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/subzip11/sf11
   0  Stored        0   0% 12-30-2015 23:30 00000000  zipfiles/subzip1/sub111
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip2/
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip2/s3
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip2/s4
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip2/s2
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip3/
   0  Stored        0   0% 12-30-2015 23:29 00000000  zipfiles/subzip3/s33
--------      -------  ---                            -------
   0                0   0%                            12 files
```

*-d 选项，指定文件解压后的目录*

*-n 选项，如果指定的目录已经存在，将不覆盖*

```
[root@'CentOS' testzip]# unzip -n zipfiles1.zip -d output/
Archive:  zipfiles1.zip
   creating: output/zipfiles/subzip1/
   creating: output/zipfiles/subzip2/
   creating: output/zipfiles/subzip3/
```

```
[root@'CentOS' testzip]# ls output/
zipfiles
```

```
[root@'CentOS' testzip]# unzip -n zipfiles1.zip -d output/
Archive:  zipfiles1.zip
```

*-o 选项，如果指定的目录已经存在，将不覆盖*

```
[root@'CentOS' testzip]# unzip -o zipfiles2.zip -d output/
Archive:  zipfiles2.zip
   creating: output/zipfiles/subzip1/subzip11/
 extracting: output/zipfiles/subzip1/subzip11/sf12
 extracting: output/zipfiles/subzip1/subzip11/sf13
 extracting: output/zipfiles/subzip1/subzip11/sf11
 extracting: output/zipfiles/subzip1/sub111
 extracting: output/zipfiles/subzip2/s3
 extracting: output/zipfiles/subzip2/s4
 extracting: output/zipfiles/subzip2/s2
 extracting: output/zipfiles/subzip3/s33
```

```
[root@'CentOS' testzip]# ls output/
zipfiles
```

```
[root@'CentOS' testzip]# unzip -o zipfiles2.zip -d output/
Archive:  zipfiles2.zip
 extracting: output/zipfiles/subzip1/subzip11/sf12
 extracting: output/zipfiles/subzip1/subzip11/sf13
 extracting: output/zipfiles/subzip1/subzip11/sf11
 extracting: output/zipfiles/subzip1/sub111
 extracting: output/zipfiles/subzip2/s3
 extracting: output/zipfiles/subzip2/s4
 extracting: output/zipfiles/subzip2/s2
 extracting: output/zipfiles/subzip3/s33
```

**WC**

计算文件行数，字节数，字符数，单词数

print newline, word, and byte counts for each file `wc`

| 选项 | 描述 |
| --- | --- |
| -l | 列出行数 |
| -c | 列出字节数 |
| -m | 列出字符数 |
| -w | 列出单词数 |

*列出文件的行数，字符数，和单词数字（无选项）*

```
[root@localhost tmp]# wc man.config
141  722 4617 man.config
```

*列出文件的行数*

```
[root@localhost tmp]# wc -l man.config
141 man.config
```

*列出文件的单词数*

```
[root@localhost tmp]# wc -w man.config
722 man.config
```

*列出文件的字符数*

```
[root@localhost tmp]# wc -m man.config
4617 man.config
```

*列出文件的字节数*

```
[root@localhost tmp]# wc -c man.config
4617 man.config
```

## find

查找目录层次中的文件 - search for files in a directory hierarchy `find`

命令格式：

```
find [PATH] [options] action
若 PATH 缺省，查找当前目录和子目录
```

*与时间有关的选项* `-mtime`

`-mtime n` ： 列出在n天前的“一日内”被变动过内容的文件

`-mtime +n`： 列出在n天前被变动过内容的文件（不包括n天本身）

`-mtime -n`： 列出在n天之内被变动过内容的文件（包括n天本身）

*与使用者或者群组有关的参数*

`-uid n` ： 列出文件主人（拥有者）是所填写id的文件

`-gid n` ： 列出文件群组是所填写id的文件

*与档案权限及名称有关的参数*

`-name` ： 按照文件名搜索，支持通配符`*`和`?`

`-size` ： 按照文件大小搜索，用`+`和`-`来标识大于和小于，c是字符，k是1024字符，M是1024k。

```
[root@localhost tmp]# find /tmp/ -name "*gz"
/tmp/man.config.gz
/tmp/man2.config.gz
/tmp/bluetooth.tar.gz
```

```
[root@localhost tmp]# find /tmp/ -size -100c
[root@localhost tmp]# find /tmp/ -size +100c
[root@localhost tmp]# find /tmp/ -size -100k
[root@localhost tmp]# find /tmp/ -size -100M
```

## locate

根据文件名查找文件 - find files by name `locate`

```
[root@localhost tmp]# locate passwd
/etc/passwd
/etc/passwd-
/etc/passwd.OLD
/etc/pam.d/passwd
/etc/security/opasswd
#以下省略……
```

和find相比，locate比较快，这是因为locate寻找f的数据是由“已建立的数据库`/var/lib/mlocate/mlocate.db`”里面的数据搜索的。而数据库的建立默认是每天执行一次，不同的发行版本还不一样，`CentOS` 5.x是每天数据库更新一次。所以，今天新建的文件，是不能直接用locate查找到的，必须要先`updatedb`，然后才可以。

```
[root@localhost ~]# locate bluetooth.tar
[root@localhost ~]# updatedb
[root@localhost ~]# locate bluetooth.tar
/root/bluetooth.tar
```

## whereis

查找命令的二进制文件、源文件以及帮助文件 - locate the binary, source, and manual page files for a command `whereis`

-b :查找binary格式的文件-m :查找在说明文件manual路径下的文件

-s :查找source文件

```
[root@localhost ~]# whereis locate
locate: /usr/bin/locate /usr/share/man/man1/locate.1.gz
[root@localhost ~]# whereis find
find: /usr/bin/find /usr/share/man/man1p/find.1p.gz /usr/share/man/man1/find.1.gz
```

# 权限管理

Linux的用户管理，是依靠**用户标识符**来进行的。虽然我们登录Linux主机的时候，使用的是我们的账号，但是其实Linux主机不会直接来判断、认识我们的账号名称的，他们认识的只是我们的ID。这里的ID是数字组成的。

在Linux系统中，每个文件都拥有主人和群组的两个属性的。所以每个登录的用户都至少拥有两个ID，一个是使用者的ID（User ID，用户ID，简称UID），另一个是群组的ID（Group ID，主人所在的组的ID，群组ID，简称GID）。

每一个文件就是依靠UID和GID来判断所属的主人和群组的。

*查看当前系统的所有用户*

```
[root@localhost tmp]# cat /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
# 省略……
linty:x:500:500:'CentOS'5:/home/linty:/bin/bash
```

## useradd

创建用户 `create a new user or update default new user information` `useradd`

*不添加选项*

```
[root@localhost tmp]# useradd john
[root@localhost tmp]# grep john /etc/passwd
john:x:501:501::/home/john:/bin/bash
```

*添加选项* `-u`

```
[root@localhost tmp]# useradd johnson -u 503
[root@localhost tmp]# grep johnson /etc/passwd
johnson:x:503:503::/home/johnson:/bin/bash
```

指定了UID为503

添加选项 `-g`

指定组群，可以选择组名和gid

```
[root@localhost tmp]# useradd wison -g users
[root@localhost tmp]# grep wison /etc/passwd
wison:x:504:100::/home/wison:/bin/bash
[root@localhost tmp]# useradd wison2 -g 100
wison:x:505:100::/home/wison:/bin/bash
```

指定了GID为100

## passwd

更新用户的密码 - update a user' s authentication tokens(s) `passwd`

```
[root@localhost tmp]# passwd john
Changing password for user john.
New UNIX password:
BAD PASSWORD: it is too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

## usermod

修改账户- modify a user account `usermod`

```
[root@localhost tmp]# usermod -c "Jonh's son account" johnson
[root@localhost tmp]# grep johnson /etc/passwd
johnson:x:503:503:Jonh's son account:/home/johnson:/bin/bash
```

## userdel

删除用户和相关文件 - delete a user account and related files `userdel`

删除用户和用户的家目录 `-r`

```
[root@localhost tmp]# userdel wison
[root@localhost tmp]# ls /home/
john  johnson  linty  wison
[root@localhost tmp]# userdel -r john
[root@localhost tmp]# ls /home/
johnson  linty  wison
```

## groupadd

创建一个新的群组 - create a new group `groupadd`

```
[root@localhost tmp]# groupadd group1
[root@localhost tmp]# grep group1 /etc/group
group1:x:504:
```

## groupmod

修改群组 - modify a group `groupmod`

```
[root@localhost tmp]# groupmod -n newgroup1 group1
[root@localhost tmp]# grep 504 /etc/group
newgroup1:x:504:
```

## groupdel

删除群组 - delete a group `groupdel`

```
[root@localhost tmp]# groupdel newgroup1
[root@localhost tmp]# grep 504 /etc/group
[root@localhost tmp]#
```

## gpasswd

管理群组 - administer the /etc/group file `gpasswd`

向组里添加用户 -a

```
[root@localhost tmp]# groupadd qagroup
[root@localhost tmp]# gpasswd -a johnson qagroup
Adding user johnson to group qagroup
[root@localhost tmp]# grep qagroup /etc/group
qagroup:x:504:johnson
```

从组里面移除用户 -d

```
[root@localhost tmp]# gpasswd -d johnson qagroup
Removing user johnson from group qagroup
[root@localhost tmp]# grep qagroup /etc/group
qagroup:x:504:
```

## su

切换用户 - run a shell with substitute user and group IDs `su`

```
[linty@localhost ~]$ su - root
Password:
```

## 文件权限

在Linux中，每个文件都有相当多的属性和权限，其中最重要的就是文件的主人（拥有者）和群组的概念了。其实还有一种个，就是其他人。

- 文件的主人（拥有者）
- 群组（拥有者所在的群组的成员，除去主人）
- 其他人

*Linux的文件属性*

```
[root@localhost tmp]# ls -al /etc/
total 3184
drwxr-xr-x 89 root root   12288 Dec 23 06:14 .
drwxr-xr-x 23 root root 4096 Dec 22 02:15 ..
drwxr-xr-x  4 root root 4096 Dec 21 08:33 acpi
-rw-r--r--  1 root root  44 Dec 21 08:43 adjtime
-rw-r--r--  1 root root 1512 Apr 25  2005 aliases
-rw-r--r--  1 root root   10793 Jul 12  2006 gnome-vfs-mime-magic
-rw-r--r--  1 root root 1756 Jul 12  2006 gpm-root.conf
drwxr-xr-x  2 root root 4096 Dec 21 08:36 gre.d
-rw-r--r--  1 root root 682 Dec 23 06:14 group
-rw-------  1 root root 689 Dec 23 06:12 group-
lrwxrwxrwx  1 root root  22 Dec 21 08:38 grub.conf -> ../boot/grub/grub.conf
-r--------  1 root root 562 Dec 23 06:14 gshadow
-rw-r--r--  1 root root 177 Dec 15  2009 idmapd.conf
lrwxrwxrwx  1 root root  11 Dec 21 08:30 init.d -> rc.d/init.d
-rw-r--r--  1 root root 658 Jul 13  2009 initlog.conf
```

我们来关注每一个文件的前面十个字符：`drwxr-xr-x`

这个就是文件的属性描述，

*文件类型*

| 文件类型 | 主人权限 user | 群组权限 group | 其他人权限 other |
| --- | --- | --- | --- |
| `d` | `rwx` | `r-x` | `r-x` |

第一位代表了文件类型，主要有一下几种类型：

| 字符 | 类型 | 英文名 |
| --- | --- | --- |
| `-` | 普通文件 | `-- file` |
| `d` | 目录 | `-- directory` |
| `l` | 链接文件 | `-- link` |
| `c` | 字符设备 | `-- character` |
| `b` | 块设备 | `-- block` |
| `s` | 套接字 | `-- socket` |
| `p` | 管道 | `-- pipe` |

*文件访问权限*

| 字符 | 类型 | 英文名 |
| --- | --- | --- |
| `r` | 读权限 | `read` |
| `w` | 写权限 | `write` |
| `x` | 执行权限 | `execute` |

事实上文件的权限来源于二进制的标识

| 权限名 | 十进制 | 二进制 |
| --- | --- | --- |
| `---` | `0` | `000` |
| `--x` | `1` | `001` |
| `-w-` | `2` | `010` |
| `-wx` | `3` | `011` |
| `r--` | `4` | `100` |
| `r-x` | `5` | `101` |
| `rw-` | `6` | `110` |
| `rwx` | `7` | `111` |

## chmod

修改文件的访问权限 `- change file access permissions` `chmod`

*数字类型改变文件权限*

---

```
[root@localhost tmp]# ll bluetooth.tar
-rw-r--r-- 1 root  root  10240 Dec 23 03:07 bluetooth.tar
[root@localhost tmp]# chmod 777 bluetooth.tar
[root@localhost tmp]# ll bluetooth.tar
-rwxrwxrwx 1 root root 10240 Dec 23 03:07 bluetooth.tar
```

修改后，文件的权限变成 777

*递归方式用数字类型改变文件权限* `-R`

```
[root@localhost tmp]# ll bluetooth
total 16
-rw-r--r-- 1 root root 1231 Dec 23 03:06 hcid.conf
-rw-r--r-- 1 root root  297 Dec 23 03:06 rfcomm.conf
[root@localhost tmp]# chmod -R 777 bluetooth
[root@localhost tmp]# ll bluetooth
total 16
-rwxrwxrwx 1 root root 1231 Dec 23 03:06 hcid.conf
-rwxrwxrwx 1 root root  297 Dec 23 03:06 rfcomm.conf
```

修改后，目录bluetooth中的所有文件的权限

*符号类型改变档案权限*

命令格式

| 命令 | 选项 | 表达式 | 权限 | 参数 |
| --- | --- | --- | --- | --- |
| `chmod` | `u` / `g` / `o` / `a` | `+` / `-` / `=` | `r` / `w` / `x` | `file` or `directory` |

```
chmod u/g/o/a +/-/= r/w/x file_or_directory
```

```
[root@localhost tmp]# chmod u=rw,g=r,o=x bluetooth.tar
[root@localhost tmp]# ll bluetooth.tar
-rw-r----x 1 root root 10240 Dec 23 03:07 bluetooth.tar
[root@localhost tmp]# chmod u+x,g-r,g+w,o-x,o+r bluetooth.tar
[root@localhost tmp]# ll bluetooth.tar
-rwx-w-r-- 1 root root 10240 Dec 23 03:07 bluetooth.tar
```

## vi编辑器

> vi是所有Unix Like系统都会内建的文本编辑器，Linux的大多数命令都会默认调用vi的接口。是上古神器。

基本上vi一共有三种模式：一般模式、编辑模式和末行模式

- 一般模式：

  以vi打开或新建一个文件，就直接进入了一般模式。这个也是vi的默认的模式。在这个模式中，我们使用方向键（上下左右）按键来移动光标，使用各种命令按键来删除字符、删除整行、复制、粘贴等处理文件。

- 编辑模式：

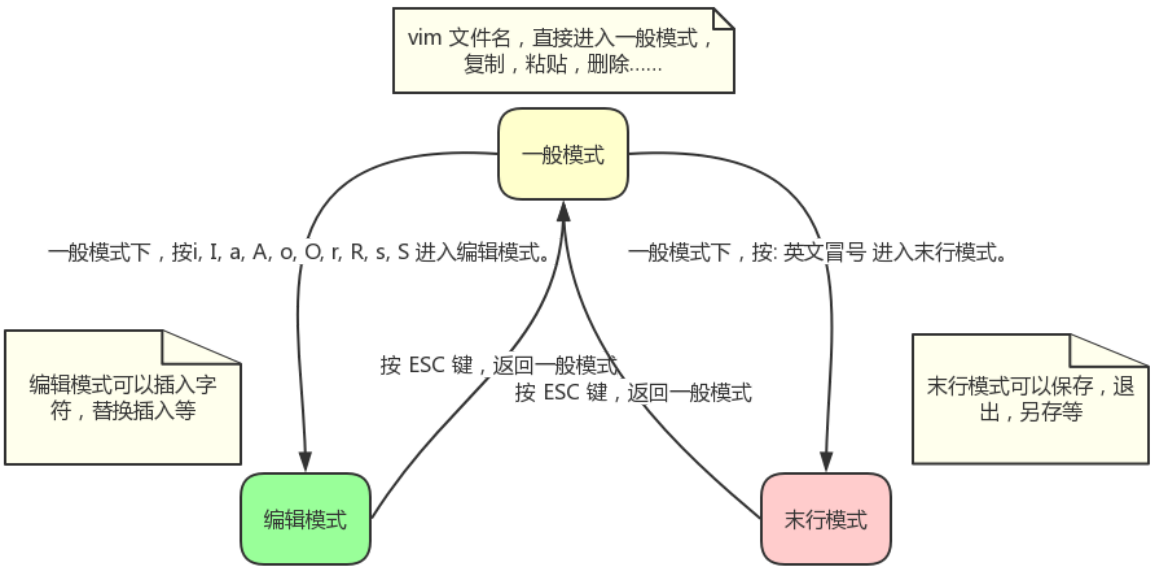
  在一般模式中，我们可以复制、删除、粘贴等处理文件，但是我们无法编辑文件内容。只有进入编辑模式以后，才可以编辑文件的内容。我们在一般模式下，通过按`i`/`I`/`o`/`O`/`a`/`A`/`r`/`R`等任何一个按键，进入编辑模式。请注意，在Linux中，通常情况下我们按下这些按键以后，在左下角我们可以看到“INSERT”或“REPLACE”的字样，此时我们才可以编辑。如果要再次返回一般模式，需要按“Esc”按键。

- 末行模式：

  在一般模式中，按下`:`就可以将光标移动到最底的那一行，这样便进入了末行模式。在这个模式中，我们可以进行读取、存储、替换字符、退出vi等操作。如果要再次返回一般模式，需要按“Esc”按键。

三种模式的切换，都必须先切换到一般模式，然后再切换到指定的模式中。编辑模式和末行模式之间不可以直接切换。

- 三种模式的切换关系图




## 一般模式的命令

| 命令 | 描述 |
| --- | --- |
| `nyy` | 从当前行开始向下复制n行 |
| `np` | 在当前行下粘贴n次 |
| `ndd` | 从当前行开始向下删除n行 |
| `u` | 撤销 |
| `Ctrl r` | 恢复撤销的操作 |
| `gg` | 移动光标到第一行 |
| `G` | 移动光标到最后一行 |
| `nG` | 移动光标到第n行 |
| `H` | 光标移动到屏幕上方(high) |
| `M` | 光标移动到屏幕中间(middle) |
| `L` | 光标移动到屏幕下方(low) |
| `Ctrl f` | 向下翻页(forward) |
| `Ctrl b` | 向上翻页(backward) |
| `0` | 光标移动到行首 |
| `$` | 光标移动到行尾 |
| `d0` | 删除光标之前当前行的所有字符 |
| `d$` | 删除光标到行尾的所有字符 |
| `dd` | 删除光标当前行的所有字符 |
| `x` | 向后删除光标所在行的一个字符 |
| `X` | 向前删除光标所在行的一个字符 |
| `w` | 以单词为单位移动光标到下一个单词的第一位 |
| `b` | 以单词为单位移动光标到上一个单词的第一位 |
| `e` | 以单词为单位移动光标移动到下一个单词的最后一位 |
| `n[space]` | 光标向右侧移动n个字符 |
| `n[enter]` | 光标向下移动n行 |
| `/[string]` | 向下查找字符为string的字符 |
| `?[string]` | 向上查找字符为string的字符 |

| 命令 | 描述 |
| --- | --- |
| `n` | 重复上一个查找操作 |
| `N` | 反向重复上一个查找操作 |
| `:%s/test/java/g` | 把文档的所有test替换成java |
| `:m,ns/test/java/g` | 把文档第m到n行的test替换成java |
| `:1,$s/test/java/g` | 把文档第1到最后1行的test替换成java |
| `ZZ` | 若文件没有改动，则不储存离开，若档案已经被改动过，则储存后离开！ |

## 编辑模式的命令

| 命令 | 描述 |
| --- | --- |
| `i` | 在当前字符前插入 -- insert |
| `I` | 在当前行行首或者第一个非空的字符处插入 -- |
| `a` | 在当前字符后插入 -- after |
| `A` | 在当前行行尾插入 -- |
| `o` | 在当前行下另起一行插入 |
| `O` | 在当前行上另起一行插入 |
| `s` | 删除当前字符插入 -- |
| `S` | 删除当前行插入-- |
| `r` | 替换当前字符插入一次 --replace |
| `R` | 替换所有的字符插入 |
| `Esc` | 返回到一般模式 |

## 末行模式的命令

| 命令 | 描述 |
| --- | --- |
| `:w` | 保存 |
| `:q` | 退出 |
| `:q!` | 强制退出 |
| `:wq` | 保存退出 |
| `:w [filename]` | 另存为 |
| `:n` | 移动光标到第n行 |
| `:set nu[mber]` | 显示行号 |
| `:set nonu[mber]` | 不显示行号 |
| `Esc` | 返回到一般模式 |

# 系统管理

## ifconfig

配置网络 - configure a network interface `ifconfig`

```
[root@localhost ~]# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:0C:29:A1:D0:89
      inet addr:192.168.220.129  Bcast:192.168.220.255  Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fea1:d089/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:53469 errors:0 dropped:0 overruns:0 frame:0
      TX packets:957 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:3353494 (3.1 MiB)  TX bytes:105898 (103.4 KiB)
      Interrupt:67 Base address:0x2024

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:1984 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1984 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:4030020 (3.8 MiB)  TX bytes:4030020 (3.8 MiB)
```

```
[root@localhost ~]# ifconfig eth0 192.168.120.56
[root@localhost ~]# ifconfig eth0 192.168.120.56 netmask 255.255.255.0
```

## service

运行服务 - run a System V init script `service`

```
[root@localhost ~]# service --status-all
```

*查看指定服务的状态*

```
[root@localhost ~]# service network status
Configured devices:
lo eth0
Currently active devices:
lo eth0
```

*关闭指定服务*

```
[root@localhost ~]# service network stop
Shutting down interface eth0:  [  OK  ]
Shutting down loopback interface:  [  OK  ]
```

*开启指定服务*

```
[root@localhost ~]# service network start
Bringing up loopback interface:[  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.  [  OK  ]
```

*重新启动指定服务*

```
[root@localhost ~]# service network restart
Shutting down interface eth0:  [  OK  ]
Shutting down loopback interface:  [  OK  ]
Bringing up loopback interface:[  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.  [  OK  ]
```

## man

帮助文档 - format and display the on-line manual pages `man`

---

```
[root@localhost ~]# man cd
[root@localhost ~]# man ifconfig
[root@localhost ~]# man service
```

### netstat

查看网络状态 `netstat`

```
[root@localhost ~]# netstat -anp
```

## | 管道

在Linux的终端中执行命令的时候，会有大量的数据输出到控制台，打印在屏幕上。这时候如果想把这个输入作为下一条命令的输入，那么我们就需要借助管道来完成。非常形象的看，管道就是上一个命令的输出，流入了下一个命令，作为输入。

`cmd1 | cmd2 | cmd3`

先执行cmd1,把cmd1的标准输出作为cmd2的标准输入，执行cmd2，在把cmd2的标准输出，作为cmd3的标准输入，执行cmd3

```
[root@localhost ~]# netstat -anp | grep 3306
[root@localhost ~]# netstat -anp | grep 17780
unix  3  [ ] STREAM CONNECTED 17780  4395/python
```

### 标准输入输出重定向

在Linux的终端执行命令的时候，我们知道会有大量的数据输出到控制台，并且打印到屏幕上。从上面的管道知识中，我们明白可以把这些数据流入管道，当然我们还有别的方式，就是标准输入输出重定向。

*标准输出重定向* `>`

```
[root@localhost ~]# echo hello world > stdout.txt
```

*追加方式，新的数据追加在原来数据后面* `>>`

```
[root@localhost ~]# echo hello >> stdout.txt
[root@localhost ~]# echo hello >> stdout.txt
[root@localhost ~]# echo hello >> stdout.txt
[root@localhost ~]# cat stdout.txt
hello
hello
hello
hello
```

*标准错误输出重定向* `2>`

```
[root@localhost ~]# sdfkldsklkdlsfk 2> stderr.txt
[root@localhost ~]# cat stderr.txt
bash: sdfkldsklkdlsfk: command not found
```

*标准输入重定向* `<`

```
[root@localhost ~]# read ABC
hello world
[root@localhost ~]# echo $ABC
hello world
[root@localhost ~]# read ABC < stderr.txt
[root@localhost ~]# cat stderr.txt
bash: sdfkldsklkdlsfk: command not found
[root@localhost ~]# echo $ABC
bash: sdfkldsklkdlsfk: command not found
```

## 系统运行启动级别

```
[root@localhost ~]# cat /etc/inittab
#
# inittab   This file describes how the INIT process should set up
#           the system in a certain run-level.
#
# Author:   Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#           Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
#此处省略……
```

| 代号 | 描述 |
| --- | --- |
| `0` | ：关机状态 |
| `1` | ：单用户模式 |
| `2` | ：多用户模式（不能使用网络） |
| `3` | ：多用户模式（普通模式，但在命令行下） |
| `4` | ：不使用模式 |
| `5` | ：桌面系统模式，跟3模式一样，区别在3模式是命令行下 |
| `6` | ：重启模式 |

## chkconfig

更新或查询系统服务的运行启动级别 - updates and queries runlevel information for system services `chkconfig`

*选项 --list*

查看所有系统启动级别3开机运行的服务

```
[root@localhost ~]# chkconfig --list | grep '3:on'
acpid       0:off   1:off   2:on    3:on    4:on    5:on    6:off
anacron     0:off   1:off   2:on    3:on    4:on    5:on    6:off
apmd        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
autofs      0:off   1:off   2:off   3:on    4:on    5:on    6:off
avahi-daemon0:off   1:off   2:off   3:on    4:on    5:on    6:off
bluetooth   0:off   1:off   2:on    3:on    4:on    5:on    6:off
#此处省略……
```

查看蓝牙服务的启动级别

```
[root@localhost ~]# chkconfig --list bluetooth
bluetooth   0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

*选项 --level*

将蓝牙服务在345级别下关闭

```
[root@localhost ~]# chkconfig --level 345 bluetooth off
[root@localhost ~]# chkconfig --list bluetooth
bluetooth   0:off   1:off   2:on    3:off   4:off   5:off   6:off
```

将蓝牙服务在345级别下开启

```
[root@localhost ~]# chkconfig --level 345 bluetooth on
[root@localhost ~]# chkconfig --list bluetooth
bluetooth   0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

## 进程管理

程序：可执行的文件

进程：程序执行一次的过程

操作系统为系统中每一个进程分配一个id，称为pid

线程：

### ps

查看进程

*查看当前终端下的进程 processes*

```
[root@localhost ~]# ps
  PID TTY      TIME CMD
 4310 pts/100:00:00 bash
 4511 pts/100:00:00 ps
```

*查看系统所有进程信息*

```
[root@localhost ~]# ps aux
[root@localhost ~]# ps -ef
```

### pstree

以树状结构显示进程的层次关系 `pstree`

```
[root@localhost ~]# bash
[root@localhost ~]# bash
[root@localhost ~]# pstree | grep bash
 |-gnome-terminal-+-2*[bash]
 |                |-bash---bash---bash-+-grep
```

### kill

结束进程 `kill`

强制结束一个进程：

强制结束进程号为pid的进程

`kill -9 pid`

尽力结束进程号为PID的进程

`kill -15 pid`

## top

显示任务管理 - display Linux tasks `top`

```
[root@localhost ~]# top
```

## du

评估文件空间的占用 - estimate file space usage `du`

```
[root@localhost ~]# du /etc/bluetooth/
24  /etc/bluetooth/
```

## df

报告文件系统的硬盘占用 - report file system disk space usage `df`

```
[root@localhost ~]# df
Filesystem       1K-blocks      Used Available Use% Mounted on
/dev/sda2         17981340   2292852  14760336  14% /
/dev/sda1           295561     16118    264183   6% /boot
tmpfs               517552         0    517552   0% /dev/shm
```

## ln

链接文件 - make links between files `ln`

链接文件有两种：实体链接（硬链接） 和 符号链接（软链接）

*实体链接（硬链接，Hard Link）*

每个文件都会占用一个inode，文件的内容由inode指向

若要读取文件，必须要在目录中通过正确的文件名，和该文件指向的正确的inode才可以读取。

```
[root@localhost ~]# ln /etc/bluetooth/
hcid.confrfcomm.conf
[root@localhost ~]# ln /etc/bluetooth/rfcomm.conf .
[root@localhost ~]# ls
abc           bluetooth      install.log         rfcomm.conf
anaconda-ks.cfg  bluetooth.tar  install.log.syslog
[root@localhost ~]# ll -i /etc/bluetooth/rfcomm.conf ./rfcomm.conf
197400 -rw-r--r-- 2 root root 297 Jul  9  2008 /etc/bluetooth/rfcomm.conf
197400 -rw-r--r-- 2 root root 297 Jul  9  2008 ./rfcomm.conf
```

实体链接，不可以链接目录，只可以链接文件。

*符号链接（软连接，Symbolic Link）*

符号链接，类似于Windows系统的快捷方式。

选项 -s

```
[root@localhost ~]# ln -s /etc/bluetooth/rfcomm.conf rfcomm2.conf
[root@localhost ~]# ls
abc           bluetooth      install.log         rfcomm2.conf
anaconda-ks.cfg  bluetooth.tar  install.log.syslog  rfcomm.conf
[root@localhost ~]# ll -i /etc/bluetooth/rfcomm.conf ./rfcomm2.conf
 197400 -rw-r--r-- 2 root root 297 Jul  9  2008 /etc/bluetooth/rfcomm.conf
1929553 lrwxrwxrwx 1 root root  26 Dec 25 01:34 ./rfcomm2.conf -> /etc/bluetooth/rfcomm.conf
```

软连接的两个文件指向了不同的inode。

选项 -f

## setup

设置设备和文件系统 - setup devices and file systems, mount root file system `setup`

setup 设置命令（ip,防火墙等）。

1. firewall configuration （防火墙设置）
2. network configuration(ip或dns的设置)

修改IP

1. 修改IP配置信息 `vi /etc/sysconfig/network-scripts/ifcfg-eth0`
2. BOOTPROTO=static，IPADDR=172.16.112.11；
3. 重起服务使配置生效`service network restart`

## whoami

打印有效的账户名 - print effective userid `whoami`

```
[root@localhost ~]# whoami
root
```

## shutdown

关闭电源

`shutdown -h now` 关机命令

`shutdown -r now` （`reboot`）立即重启命令

## halt

关机

`halt -h`

标准情况下是关机 但是要手动关闭电源。

```
[root@localhost ~]# halt -h
root
```