

# Linux系统常用命令详解

## 前言

Linux提供了大量的命令，利用它可以有效地完成大量的工作，如磁盘操作、文件存取、目录操作、进程管理、文件权限设定等。所以，在Linux系统上工作离不开使用系统提供的命令。要想真正理解Linux系统，就必须从Linux命令学起，通过基础的命令学习可以进一步理解Linux系统。

## 目录

[前言](#) 2

[1.名词解释](#) 6

[1.1用户](#) 6

[1.2群组](#) 6

[1.3权限](#) 6

[1.4通配符](#) 8

[2.shell介绍](#) 8

[3.打开终端](#) 9

[4.文件管理类命令](#) 10

[cat](#) 10

[chgrp](#) 10

[chmod](#) 11

[chown](#) 11

[diff](#) 12

[find](#) 12

[ln](#) 13

[more](#) 14

[mv](#) 14

[scp](#) 14

[cp](#) 15

[rm](#) 15

[which](#) 16

[grep](#) 16

[head](#) 16

[tail](#) 17

[touch](#) 17

[file](#) 18

## [5.磁盘管理类命令](#) 18

[cd](#) 18

[df](#) 19

[du](#) 19

[mkdir](#) 20

[rmdir](#) 20

[pwd](#) 20

[ls](#) 21

[fdisk](#) 21

## [6.网络通讯类命令](#) 22

[telnet](#) 22

[ifconfig](#) 22

[netstat](#) 23

[ping](#) 23

## [7.系统管理类命令](#) 24

[kill](#) 24

[reboot](#) 24

[shutdown](#) 24

[su](#) 25

[man](#) 25

[uname](#) 25

[ps](#) 26

[date](#) 27

[exit](#) 27

[rlogin](#) 28

[ssh](#) 28

[top](#) 28

[whoami](#) 29

[uptime](#) 29

[hostname](#) 29

## [8.备份压缩类命令](#) 30

[tar](#) 30

[gzip](#) 30

[gunzip](#) 30

## 1.名词解释

### 1.1用户

用户是能够获取系统资源的权限的集合。

用户分为三类：

1. 超级用户：拥有对系统的最高管理权限，默认是root用户。
2. 普通用户：能对自己目录下的文件进行访问和修改，具有登录系统的权限，例如osa用户。
3. 系统用户：保障系统运行的用户，一般不提供密码登录系统。

### 1.2群组

群组是具有相同特征的用户逻辑集合。例如users、dba。

### 1.3权限

执行“ls -al”命令，会看到如下内容：

```

[root@scdl ~]# ls -al
总用量 324
dr-xr-x---. 27 root root    4096 4月 14 17:51 .
dr-xr-xr-x. 28 root root    4096 5月 19 14:01 ..
-rw-----.  1 root root   8429 3月 19 19:11 anaconda-ks.cfg
-rw-----.  1 root root   4379 5月 19 14:01 .bash_history
-rw-r--r--.  1 root root     18 5月 20 2009 .bash_logout
-rw-r--r--.  1 root root    176 5月 20 2009 .bash_profile
-rw-r--r--.  1 root root    176 9月 23 2004 .bashrc
drwxr-xr-x.  3 root root    4096 3月 19 20:49 .cache
drwx-----.  5 root root    4096 3月 19 20:48 .config
-rw-r--r--.  1 root root    100 9月 23 2004 .cshrc
drwx-----.  3 root root    4096 3月 19 20:48 .dbus
-rw-----.  1 root root     16 3月 19 20:48 .esd_auth
drwx-----.  4 root root    4096 3月 28 08:45 .gconf
drwx-----.  2 root root    4096 3月 28 08:45 .gconfd
drwx-----.  6 root root    4096 3月 19 20:48 .gnome2
drwx-----.  2 root root    4096 3月 19 20:48 .gnome2_private
drwxr-xr-x.  3 root root    4096 3月 19 20:48 .gnote
drwx-----.  2 root root    4096 3月 28 06:55 .gnupg
drwxr-xr-x.  2 root root    4096 3月 19 20:48 .gststreamer-0.10
-rw-r--r--.  1 root root    160 3月 28 06:55 .gtk-bookmarks
drwx-----.  2 root root    4096 3月 19 20:48 .gvfs
-rw-----.  1 root root   1252 3月 28 06:55 .ICEauthority
-rw-r--r--.  1 root root    865 3月 28 08:45 .imsettings.log
-rw-r--r--.  1 root root  102337 3月 19 19:11 install.log

```

第一列表示这个文件的属性，这个地方最需要注意。这一列共有10个属性。如下图所示：

第一个字符代表这个文件是『目录、文件或链接文件等等』：

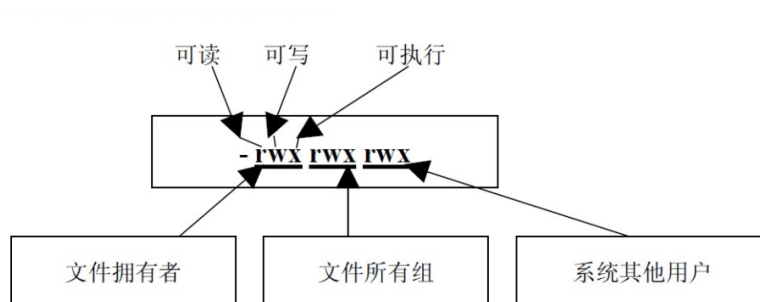
1. 当为[d]则是目录，例如上面文件名为『.cache』的那一行；
2. 当为[-]则是文件，例如上面文件名为『install.log』那一行；
3. 若是[l]则表示为连结档(link file)；
4. 若是[b]则表示为装置文件里面的可供储存的接口设备(可随机存取装置)；
5. 若是[c]则表示为装置文件里面的串行端口设备，例如键盘、

鼠标(一次性读取装置)。

接下来的字符中，以三个为一组，且均为“rwx”的三个参数的组合。

其中，[r]代表可读(read)、[w]代表可写(write)、[x]代表可执行(execute)。

要注意的是，这三个权限的位置不会改变，如果没有权限，就会出现减号[-]而已。



第一组为『文件拥有者的权限』，以『install.log』那个文件为例，该文件的拥有者可以读写，但不可执行；第二组为『同群组的权限』；第三组为『其他非本群组的权限』。

数字类型改变权限：采用八进制数组成位模式来完成权限的描述。“读取”权限r:4；“写”权限w:2；“执行”权限x:1。

每种身份文件拥有者（owner）、文件所有组（group）、系统其他用户（others）各自的三个权限（r、w、x）八进制数是需要累加的。

例如：[-rwxr-x--]:

owner:rwx=4+2+1=7

group:r-x=4+1=5

others: ---=0+0+0=0



## 1.4通配符

通配符是一种特殊语句，主要有星号(\*)和问号(?)，用来模糊搜索文件。当查找文件夹时，可以使用它来代替一个或多个真正字符；当不知道真正字符或者偷懒不想输入完整名字时，常常使用通配符代替一个或多个真正的字符。常用通配符举例：

\*：代表0个到无穷多个任意字符。

?：代表一定有一个任意字符。

## 2.shell介绍

只要有操作系统那么就离不开shell。在讨论shell之前，先了解一下计算机的运作状况。举个例子来说：当计算机传输出来音乐的时候，计算机需要什么东西呢？

1.硬件：即硬件有『声卡芯片』这个硬件配备；

2.内核管理：操作系统的核心可以支持这个芯片组，当然还需要提供芯片的驱动程序；

3.应用程序：需要用户输入发生声音的指令。

这就是一个基本的输出声音所需要的步骤！那么也就是说，必须要“输入”一个指令之后，“硬件”才会透过下达的指令来工作！那么硬件如何知道下达的指令呢？那就是kernel（内核）的控制工作了！也就是说，必须要透过shell将输入的指令与Kernel沟通，让Kernel可以控制硬件来正确无误的工作！具体步骤如下图所示：

管理整个计算机硬件的其实是操作系统的内核(kernel)，这个内核是需要被保护的，所以一般使用者(user)就只能通过shell来跟内核通信，以让内核达到用户所想要达到的工作。

shell其实就是系统的用户界面，它提供了用户和Linux（内核）之间进行交互操作的一种接口。用户在命令行中输入的每一个命令都由shell先解释，然后传给Linux内核去执行。如果把Linux内核想象成一个球体的中心，如下图，shell就是围绕内核的外层，从shell向Linux操作系统传递命令时，内核就会做出相应的反应。像man、chmod、chown、vi、fdisk等命令，都是独立的应用程序，但是用户可以通过shell（就是命令行模式）来操作这些应用程序，让这些应用程序调用内核来运行所需的工作。

目前Linux系统有以下可使用的shell：

/bin/sh（已经被/bin/bash所替代）

/bin/bash（Linux系统默认的shell）

/bin/ksh（Kornshell由AT&T Bell lab.开发，兼容于bash）

/bin/tcsh（整合C shell，提供更多的功能）

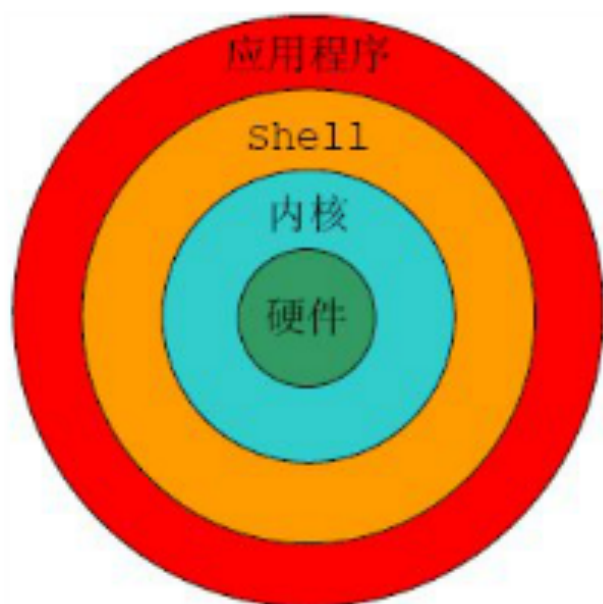
/bin/csh（已经被/bin/tcsh所替代）

/bin/zsh（基于ksh所开发出来，功能更强大的shell）

每一个shell功能相似，但在某些语法的执行方面则有所不同。委内

瑞拉项目采用的是Linux系统默认的bash shell。

了解了shell，即可进行命令操作。



### 3.打开终端

1.Linux操作系统：启动计算机，输入用户名和密码后进入系统，在桌面上点击鼠标右键，选择“打开终端”，即可进行Linux命令操作。

2.Windows操作系统：需安装远程登录软件，打开软件后，输入IP地址、登录用户名以及密码后即可远程访问终端。

### 4.文件管理类命令

**cat**

#### 1. 功能说明:

“concatenate”的缩写，直接查看文件的内容。

#### 2. 语法:

cat [options] 文件1 文件2……

### 3. 主要参数:

- n: 由第一行开始对所有输出的行数编号。
- b: 和- n相似, 只不过对于空白行不编号。

### 4. 应用举例:

```
$ cat /etc/named.conf
```

```
$ cat -b /etc/named.conf
```

## chgrp

### 1. 功能说明:

“change group”的缩写, 用于变更文件或目录的所属群组。

### 2. 语法:

```
chgrp [-R] [--help] [--version] [所属群组][文件或目录]
```

### 3. 主要参数:

- R: 递归处理, 将指定目录下的所有文件及子目录一并处理。
- help: 在线帮助。
- version: 显示版本信息。

### 4. 应用举例:

命令: 将install.log的所属组改为users。

```
$chgrp users install.log
```

## chmod

### 1. 功能说明:

改变用户的权限。

## 2. 语法:

`chmod [-R] [--help] [--version] 文件或目录。`

## 3. 主要参数:

-R:对目前目录下的所有档案与子目录进行相同的权限变更(即以递归的方式逐个变更)。

--help:显示辅助说明。

--version:显示版本。

## 4. 应用举例:

命令: 修改.bashrc文件的权限为可读可写可执行:

```
$chmod 777 .bashrc
```

# chown

## 1. 功能说明:

改变文件所有者。

## 2. 语法:

`chown [-R] [--help] [--version] 账号名称 文件或目录`

## 3. 主要参数:

-R: 对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递归的方式逐个变更)

--help: 显示辅助说明。

--version: 显示版本。

## 4. 应用举例:

命令：将档案file1.txt的拥有者设为users，群体的使用者设为osa:

```
$ chown osa:users file1.txt
```

## diff

### 1. 功能说明：

diff命令用于两个文件之间的比较，并指出两者的不同，它的使用权限是所有用户。

### 2. 语法：

```
diff [options] 源文件 目标文件
```

### 3. 主要参数：

-b：忽略空格造成的不同。

-B：忽略空行造成的不同。

### 4. 应用举例：

命令：比较.bash\_profile和.bash\_profile\_bak两个文件有哪些不同：

```
$ diff .bash_profile .bash_profile_bak
```

## find

### 1. 功能说明：

在目录中搜索文件。

### 2. 语法：

```
find [path] [options] [expression]
```

path为指定目录路径，系统从这里开始沿着目录树向下查找文件。

如果不写path，那么默认为当前目录。

### 3. 主要参数：

-name: 按文件名查找, 支持通配符\*和?。

-version: 打印版本。

#### 4. 应用举例:

find命令查找文件的几种方法:

##### 1. 根据文件名查找

例如, 想要查找一个文件名是lilo.conf的文件, 可以使用如下命令:

```
$find / - name lilo.conf
```

find命令后的“/”表示搜索整个硬盘。

##### 2. 快速查找文件

根据文件名查找文件会遇到一个实际问题, 就是要花费相当长的一段时间, 特别是大型Linux文件系统和大容量硬盘文件放在很深的子目录中时。如果用户知道了这个文件存放在某个目录中, 那么只要在这个目录中往下寻找就能节省很多时间。比如smb.conf文件, 从它的文件后缀“.conf”可以判断这是一个配置文件, 那么它应该在/etc目录内, 此时可以使用下面命令:

```
$ find /etc - name smb.conf
```

##### 3. 根据部分文件名查找方法

有时用户知道只有某个文件包含有abvd这4个字, 那么要查找系统中所有包含有这4个字符的文件可以输入下面命令:

```
$ find / - name '*abvd*'
```

输入这个命令以后, Linux系统会将在/目录中查找所有的包含有abvd这4个字符的文件(其中\*是通配符), 比如abvdrmyz等符合条件的

文件都能显示出来。

## ln

### 1. 功能说明：

用来在文件之间创建链接。

### 2. 语法：

ln [options] 源文件 [链接名]

### 3. 主要参数：

-s：进行软链接(Symbolic Link)。

链接有两种，一种被称为硬链接（Hard Link），另一种被称为符号链接（Symbolic Link）。默认情况下，ln命令产生硬链接。ln后接-s参数为符号链接，符号链接文件有点类似于Windows的快捷方式。它实际上是特殊文件的一种。在符号连接中，文件实际上是一个文本文件，其中包含的有另一文件的位置信息。

### 4. 应用举例：

在主目录中创建一个符号链接gg，使之能方便地进入Web服务器(/usr/local/httpd/htdocs)文档所在的目录。在主目录下可以使用以下命令：

```
$ ln -s /usr/local/httpd/htdocs gg
```

## more

### 1. 功能说明：

可翻页查看。

### 2. 语法：



`more [文件名]`

当输入“空格”时则显示下一页内容；当输入“回车”时显示下一行内容；当输入Q或q时则退出。

### 3. 应用举例：

逐页显示 `testfile` 之档案内容，如有连续两行以上空白行则以一行空白行显示：

```
$more testfile
```

## mv

### 1. 功能说明：

`mv`命令用来为文件或目录改名，或者将文件由一个目录移入另一个目录中，它的使用权限是所有用户。该命令如同DOS命令中的`ren`和`move`的组合。

### 2. 语法：

```
mv [options] 源文件或目录 目标文件或目录
```

### 3. 应用举例：

1. 将`/usr/cbu`中的所有文件移到当前目录（用“.”表示）中：

```
$ mv /usr/cbu/* .
```

2. 将文件`cjh.txt`重命名为`wjz.txt`：

```
$ mv cjh.txt wjz.txt
```

## scp

### 1. 功能说明：

`scp`是`secure copy`的缩写,它是linux系统下基于`ssh`登陆进行安全的远程

文件拷贝命令。linux的scp命令可以在linux服务器之间复制文件和目录。

## 2. 语法:

scp [参数] [文件名] [目标路径]

## 3. 主要参数:

-r: 递归复制整个目录。

-p: 保留原文件的修改时间，访问时间和访问权限。

## 4. 应用举例:

命令: 将当前节点下/data/users/osa下的filetran.tar拷到scd2节点/data/users/osa下:

```
$ scp -rp /data/users/osa/filetran.tar scd2:/data/users/osa/.
```

# cp

## 1. 功能说明:

复制文件或目录。

## 2. 语法:

cp [options] 源文件 目标文件

## 3. 主要参数:

-r: 递归处理，将指定目录下的文件与子目录一并处理。

-f: 强行复制文件或目录，不论目标文件或目录是否已存在。

## 4. 应用举例:

(1) 将档案aaa复制(已存在)，并命名为bbb:

```
$ cp aaa bbb
```

(2) 将所有的C语言程序拷贝至Finished子目录中:

```
$ cp *.c Finished
```

## rm

### 1. 功能说明：

移除文件或目录。

### 2. 语法：

rm [options] 文件名

### 3. 主要参数：

-i:删除前逐一询问确认。

-f: 强制删除文件或目录。

-r:递归处理，将指定目录下的所有文件及子目录一并处理。

### 4. 应用举例：

#### 1. 删除所有C语言程序档；删除前逐一询问确认：

```
$ rm -i *.c
```

#### 2. 将 Finished 子目录及子目录中所有档案删除：

```
$ rm -r Finished
```

## which

### 1. 功能说明：

寻找“执行文件”。

### 2. 语法：

which [文件]

### 3. 应用举例：

请找出cd这个命令的完整文件名：

```
$ which cd
```

## grep

### 1. 功能说明：

查找文件里符合条件的字符串。

### 2. 语法：

grep ‘查找字符串’ 文件名

### 3. 应用举例：

```
# grep baoshi /etc/passwd
```

```
# ls -l /etc | grep inetd.conf
```

## head

### 1. 功能说明：

用来显示文件的开头至标准输出中，默认head命令打印其相应文件的开头10行。

### 4. 语法：

head [参数] [文件]

### 5. 主要参数：

-n<行数> 显示的行数

### 6. 应用举例：

显示文件的前5行命令：

```
$ head -n 5 log2014.log
```

## tail

### 1. 功能说明：

用于显示指定文件末尾内容，不指定文件时，作为输入信息进行处理。常用查看日志文件。

#### 7. 语法：

`tail [主要参数] [文件]`。

#### 8. 主要参数：

`-f`：循环读取

`-n`：显示行数

#### 9. 应用举例：

(1) 显示文件末尾5行内容命令：

```
$ tail -n 5 log2014.log
```

(2) 循环查看文件内容命令：

```
$ tail -f test.log
```

## touch

#### 1. 功能说明：

改变文件或目录时间。

#### 2. 语法：

`touch [-acdm] 文件`

#### 3. 主要参数：

`-a`：只更改存取时间。

`-c`：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。

`-m`：只更改变动时间。

-t <日期时间>: 使用指定的日期时间, 而非现在的时间。

#### 4. 应用举例:

1. 修改bashrc文件, 将日期调整为两天前:

```
$ touch -d "2 days ago" bashrc
```

2. 将bashrc日期改为2007/09/15 02:02

```
$ touch -t 0709150202 bashrc
```

## file

#### 1. 功能说明:

辨识文件类型。

#### 5. 语法:

```
file [文件或目录]
```

#### 6. 应用举例:

命令: 查看bin的文件类型。

```
osa(scd1)>file bin
```

```
bin: directory
```

## 5.磁盘管理类命令

## cd

#### 1. 功能说明:

切换目录, change directory的缩写。

#### 10. 语法:

```
cd [文件名]
```

## 11. 应用举例：

(1) 跳到/usr/bin/：

```
$ cd /usr/bin
```

(2) 跳到目前目录的上两层：

```
$ cd ../../
```

## df

### 1. 功能说明：

df命令用来检查文件系统的磁盘空间占用情况，使用权限是所有用户。

### 12. 语法：

```
df [options]
```

### 13. 主要参数：

- k：以1024字节为单位列出磁盘空间使用情况。
- h：以容易理解的格式印出文件系统大小，例如136KB、254MB、21GB。

## 14. 应用举例：

将系统内所有的文件系统列出来：

```
$ df
```

## du

### 1. 功能说明：

显示目录或文件的大小。

### 15. 语法：

du [-ahkm] 文件或目录名称。

#### 16. 主要参数:

-a: 列出所有的文件与目录容量。

-h: 以K, M, G为单位, 提高信息的可读性。

-k: 以1024 bytes为单位。

-m: 以1MB为单位。

--help: 显示帮助。

--version: 显示版本信息。

#### 17. 应用举例:

列出目前目录下的所有文件容量:

```
osa(scd1)>du
```

## mkdir

#### 1. 功能说明:

新建新目录

#### 18. 语法:

mkdir 目录名称

#### 19. 应用举例:

创建一个名为“tsk”的目录且具有rwx(即读、写、执行的权限):

```
$ mkdir - m 777 tsk
```

## rmdir

#### 1. 功能说明:

删除“空”的目录



## 20. 语法:

`rmdir` 目录名称

## 21. 应用举例:

```
$ rmdir dis
```

# pwd

## 1. 功能说明:

显示目前所在的目录。

## 22. 语法:

```
pwd [--help][--version]
```

## 23. 主要参数:

`--help`: 在线帮助。

`--version`: 显示版本信息。

## 24. 应用举例:

显示当前所在目录:

```
$ pwd
```

```
/data/users/osa
```

# ls

## 1. 功能说明:

查看文件与目录。

## 25. 语法:

```
ls [options] 目录名称
```

## 26. 主要参数:

- a: 不隐藏任何以“.”字符开始的项目。
- l: 使用较长格式列出信息。
- r: 依相反次序排列。
- t: 依时间排序，而不是用文件名。
- d: 仅列出目录本身，而不是列出目录内的文件数据。

## 27. 应用举例:

列出当前目录下所有文件和目录，并显示具体信息:

```
osa(scd1)>ls -alrt
```

总用量 11960

```
-rw-r--r--. 1 osa users 37121 8月 20 2013 bay_struct.object
```

```
drwxr-xr-x. 14 osa users 4096 3月 23 19:55 adtime
```

```
-rw-r--r--. 1 osa users 1 5月 6 14:34 sqlnet.log
```

## fdisk

### 1. 功能说明:

磁盘分区。

### 28. 语法:

```
fdisk [主要参数]
```

### 29. 主要参数:

- b<分区大小>: 指定每个分区的大小。
- l: 列出指定的外围设备的分区表状况。
- s<分区编号>: 将指定的分区大小输出到标准输出上，单位为区块。

-u: 搭配"-l"参数列表, 会用分区数目取代柱面数目, 来表示每个分区的起始地址。

### 30. 应用举例:

查看机器所挂硬盘个数及分区情况

```
osa(scd1)>fdisk -l
```

## 6.网络通讯类命令

### telnet

#### 1. 功能说明:

telnet表示开启终端机阶段作业, 并登入远端主机。telnet是一个Linux命令, 同时也是一个协议(远程登陆协议)。用户使用telnet命令可以进行远程登录, 并在远程计算机之间进行通信。用户通过网络在远程计算机上登录, 就像登录到本地机上执行命令一样。为了通过telnet登录到远程计算机上, 必须知道远程机上的合法用户名和口令。

#### 31. 语法:

```
telnet [主机名称IP地址]
```

#### 32. 应用举例:

```
$ telnet 10.15.100.1
```

### ifconfig

#### 1. 功能说明:

ifconfig用于查看和更改网络接口的地址和参数, 包括IP地址、网络掩码、广播地址。

### 33. 语法:

`ifconfig - interface [options] address`

### 34. 主要参数:

`-interface`: 指定的网络接口名, 如eth0和eth1。

`up`: 激活指定的网络接口卡。

`down`: 关闭指定的网络接口。

`address`: 设置指定接口设备的IP地址。

`netmask address`: 设置接口的子网掩码。

### 35. 应用举例:

要赋给eth0接口IP地址207.164.186.2, 并且马上激活它, 使用下面命令:

```
$ ifconfig eth0 210.34.6.89 netmask 255.255.255.128 broadcast  
210.34.6.127
```

该命令的作用是设置网卡eth0的IP地址、网络掩码和网络的本地广播地址。注意, 用ifconfig命令配置的网络设备参数, 机器重新启动以后将会丢失。

如果要暂停某个网络接口的工作, 可以使用down参数:

```
$ ifconfig eth0 down
```

## netstat

### 1. 功能说明:

检查整个Linux网络状态。如开启的端口、在为哪些用户服务, 以及服务的状态等。在默认情况下, netstat只显示已建立连接的端口。

### 36. 语法:

netstat [主要参数]

### 37. 主要参数:

-a: 显示所有选项，默认不显示LISTEN相关

-t: 仅显示tcp相关选项

-n: 拒绝显示别名，能显示数字的全部转化成数字。

-l: 仅列出有在Listen(监听)的服务状态

### 38. 应用举例:

如果要显示处于监听状态的所有端口，使用-a参数:

```
$ netstat -a
```

## ping

### 1. 功能说明:

检测主机网络接口状态。通常使用它检测网络是否连通。

### 39. 语法:

ping [参数] [主机名或IP地址]

### 40. 主要参数:

-s: 设置数据包的大小。

### 41. 应用举例:

(1) 检测10.15.100.1这个地址的网络状态:

```
$ ping 10.15.100.1
```

(2) 检测10.15.100.1这个地址的网络状态，设置数量包大小为

4096:

```
$ ping -s 4096 10.15.100.1
```

## 7. 系统管理类命令

### kill

#### 1. 功能说明：

kill命令用来中止一个进程。

#### 42. 语法：

```
kill [进程id]
```

#### 43. 主要参数：

-9：强制终止。

#### 44. 应用举例：

强行中止一个进程标识号为324的进程：

```
$ kill - 9 324
```

### reboot

#### 1. 功能说明：

重新启动计算机。

#### 45. 语法：

直接执行reboot来重启计算机。

### shutdown

#### 1. 功能说明：

关闭计算机。

#### 46. 语法：

`shutdown [- h][- r]`

#### 47. 主要参数:

`-h`: 关机后关闭电源。

`-r`: 关机后重新开机

## **su**

#### 1. 功能说明:

身份切换命令。

#### 48. 语法:

`su - [用户名]`

#### 49. 应用举例:

在osa用户下切换到root用户:

```
osa(scd1)>su - root
```

## **man**

#### 1. 功能说明:

用来查看系统中自带的各种参考手册。

#### 50. 语法:

`man [命令]`

#### 51. 应用举例:

查看ls命令的参考手册:

```
$ man ls
```

## **uname**

### 1. 功能说明：

显示系统信息。

### 52. 语法：

uname [-主要参数][--help][--version]

### 53. 主要参数：

-a：显示全部的信息。

--help：显示帮助。

--version：显示版本信息。

### 54. 应用举例：

```
osa(scd1)>uname -a
```

```
Linux scd1 2.6.32-220.el6.x86_64 #1 SMP Wed Nov 9 08:03:13 EST 2011
```

```
x86_64 x86_64 x86_64 GNU/Linux
```

## ps

### 1. 功能说明：

显示瞬间行程的动态。

### 55. 语法：

ps [options] [--help]

### 56. 主要参数：

-ef：显示所有进程信息，连同命令行。

a：显示所有进程

u：指定用户的所有进程

aux：显示所有包含其他使用者的行程。



## 57. 应用举例:

### 1. 显示所有进程信息，连同命令行:

```
osa(scd1)>ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	May13 ?		00:02:39	/sbin/init
root	2	0	0	May13 ?		00:00:00	[kthreadd]
root	3	2	0	May13 ?		00:00:00	[migration/0]
root	4	2	0	May13 ?		00:00:12	[ksoftirqd/0]
root	5	2	0	May13 ?		00:00:00	[migration/0]

### 2. ps与grep常用组合用法，查找特定进程:

```
osa(scd1)>ps -ef|grep ssh
```

root	6952	1	0	May13 ?		00:00:00	/usr/sbin/sshd
root	41702	6952	0	May18 ?		00:00:00	sshd: osa [priv]
osa	41718	41702	0	May18 ?		00:00:00	sshd: osa@pts/3
root	43581	6952	0	08:37 ?		00:00:00	sshd: osa [priv]
osa	43660	43581	0	08:37 ?		00:00:00	sshd: osa@pts/0
osa	52733	43661	0	08:41 pts/0		00:00:00	grep ssh

### 3. 列出目前所有的正在内存当中的程序:

```
[root@localhost test6]# ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	10368	676	?	Ss	Nov02	0:00	init [3]
root	2	0.0	0.0	0	0	?	S<	Nov02	0:01	[migration/0]
root	3	0.0	0.0	0	0	?	SN	Nov02	0:00	[ksoftirqd/0]
root	4	0.0	0.0	0	0	?	S<	Nov02	0:01	[migration/1]
root	5	0.0	0.0	0	0	?	SN	Nov02	0:00	[ksoftirqd/1]
root	6	0.0	0.0	0	0	?	S<	Nov02	29:57	[events/0]
root	7	0.0	0.0	0	0	?	S<	Nov02	0:00	[events/1]
root	8	0.0	0.0	0	0	?	S<	Nov02	0:00	[khelper]
root	49	0.0	0.0	0	0	?	S<	Nov02	0:00	[kthread]
root	54	0.0	0.0	0	0	?	S<	Nov02	0:00	[kblockd/0]
root	55	0.0	0.0	0	0	?	S<	Nov02	0:00	[kblockd/1]
root	56	0.0	0.0	0	0	?	S<	Nov02	0:00	[kacpid]

.....省略部分结果

## date

### 1. 功能说明：

显示或设置系统时间与日期。

### 2. 语法：

date

### 3. 应用举例：

osa(scd1)>date

2015年 05月 19日 星期二 10:13:00 CST

## exit

### 1. 功能说明：

退出目前的shell。

### 2. 语法：

exit

### 3. 应用举例：

```
osa(scd1)>exit
```

```
logout
```

## rlogin

### 1. 功能说明：

远端登入。

### 2. 语法：

```
rlogin [主机名称或IP地址]
```

### 3. 应用举例：

```
$ rlogin scd1
```

## ssh

### 1. 功能说明：

远端登入。与rlogin的区别是更安全，采用ssh登录时与主机通讯的数据是加密传输的。

### 4. 语法：

```
ssh [主机名称或IP地址]
```

### 5. 应用举例：

```
$ ssh scd1
```

## top

### 1. 功能说明：

显示、管理执行中的程序。

## 58. 语法:

`top [options]`

## 59. 主要参数:

`-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。

`-n<执行次数>`: 设置监控信息的更新次数。

## 60. 应用举例:

`osa(scd1)>top d 5`

## whoami

### 1. 功能说明:

显示当前登录的用户名。

## 61. 语法:

`whoami`

## 62. 应用举例:

`osa(scd1)>whoami`

`osa`

## uptime

### 1. 功能说明:

查询linux系统当前状态。包括当前时间、系统运行时间、当前在线用户、以及最近1分钟、5分钟、15分钟系统的平均负载。

## 63. 语法:

`uptime`

#### 64. 应用举例:

```
osa(scd1)>uptime
```

```
11:11:55 up 5 days, 20:22, 5 users, load average: 0.89, 0.83, 0.80
```

### hostname

#### 1. 功能说明:

查看当前用户节点名。

#### 65. 语法:

```
hostname
```

#### 66. 应用举例:

```
osa(scd1)>hostname
```

```
scd1
```

## 8.备份压缩类命令

### tar

#### 1. 功能说明:

备份文件。

#### 67. 语法:

```
tar [-czvf][文件或目录...]
```

#### 68. 主要参数:

-c: 建立新的备份文件。

-f: 指定备份文件。

-v: 显示指令执行过程。

-x: 从备份文件中还原文件。

-z: 通过gzip指令处理备份文件。

#### 69. 应用举例:

```
$ tar -cvf temp.tar temp
```

```
$ tar -xvf temp.tar temp
```

### gzip

#### 1. 功能说明:

压缩文件。

#### 70. 语法:

```
gzip [文件名]
```

#### 71. 应用举例:

将文件temp以gzip压缩

```
$ gzip temp
```

### gunzip

#### 1. 功能说明:

解压文件。

#### 72. 语法:

```
gunzip [文件名]
```

#### 73. 应用举例:

将压缩文件解压成文件格式。

```
$ gunzip temp.gz
```

