

# Linux基础(RHEL)

2014-09-27

[Show/Hide Toc](#)



[L1](#) [L2](#) [L3](#) [L4+](#)



- [1 安装相关](#)
- [2 文件管理](#)
- [3 软件安装](#)
- [4 用户管理](#)
- [5 组管理](#)
- [6 文件权限](#)
- [7 进程控制](#)
- [8 Linux其它](#)
- [9 网络配置](#)
- [10 服务配置](#)
- [11 Linux常用简介](#)
- [12 Vim编辑器](#)
- [13 Sublime快捷键](#)
- [14 Xcode快捷键](#)



## 1 安装相关

### 1.1 安装事项

RHEL的经典版本：5.8和6.4

注意事项

从U盘安装时，GRUB引导程序默认在U盘上（/dev/sdb），需安装在/dev/sda上

新手安装时，防火墙、SELinux、kdump关闭

[Redhat网络系统安装说明](#)

[老师的ule笔记.pdf](#)

### 1.2 Linux目录结构(FHS)



- |         |                                      |                      |
|---------|--------------------------------------|----------------------|
| 1 /     | 根目录                                  |                      |
| 2 /boot | 存放程序启动所必须的文件                         |                      |
| 3       | /boot/vmlinuz-3.2.0-61vm-generic-pae | 内核文件                 |
| 4       | /boot/grub/*                         | 引导程序                 |
| 5 /var  | 系统中经常需要变化的一些文件                       |                      |
| 6       | /var/tmp                             | 程序运行时使用的临时文件，删除可能会出错 |

|    |                     |                               |
|----|---------------------|-------------------------------|
| 7  | /var/lib;/var/lib64 | 库文件                           |
| 8  | /var/log/messages   | 系统日志                          |
| 9  | /bin                | 存放系统基本用户命令，如ls,mv             |
| 10 | /sbin               | 存放系统基本管理命令 s代表super，如fdisk,ip |
| 11 | /usr                | 存放系统大量的应用程序                   |
| 12 | /usr/bin            | 用户命令                          |
| 13 | /usr/sbin           | 管理命令                          |
| 14 | /etc                | 存放系统和各种程序的配置文件                |
| 15 | /etc/init.d         | 服务启动                          |
| 16 | /dev                | 设备文件                          |
| 17 | ls /dev/sd*         | 查看所有硬盘及分区                     |
| 18 | /lib                | 库文件，驱动，由glibc提供               |
| 19 | /lib64              | 64位的库文件                       |
| 20 | /usr/lib            | 各distribute维护的库文件             |
| 21 | /usr/local/lib      | 程序或用户创建的库文件                   |
| 22 | /tmp                | 临时文件                          |
| 23 | /media              | 光盘的默认挂载点                      |
| 24 | /media/cdrom        |                               |
| 25 | /mnt                | 其它的挂载点                        |
| 26 | /proc               | 进程文件(pid)，虚拟文件系统              |
| 27 | du -sh /proc        | 统计大小为0，表示数据在内存中，不占硬盘空间        |
| 28 | /sys                | 用来代替/proc的                    |
| 29 | /opt                | 主机额外安装软件所放的目录                 |
| 30 | /srv                | 一些服务启动后，这些服务所需访问的数据目录         |

## 1.3 显示系统信息

?

```
1 luname -a -r
2 lsb_release -a      #发布版本
3 getconf LONG_BIT    #查看操作系统位数
4
5 cat /etc/*-release
6 cat /proc/version
7 cat /proc/cupinfo
8 cat /proc/meminfo
9 cat /proc/interrupts
```

## 1.4 分区设备

?

```
1 IDE硬盘
2 hda    a表示第1个      a~d
3 hda1   1表示第一个分区 1~63
4
5 SCSI硬盘
6 sda    a表示第1个      a~p
7 sda1   1表示第一个分区 1~15 最多4个主分区或3个主分区1个扩展分区
使用ls /dev/hd*, ls /dev/sd*查看
```

## 1.5 分区建议

?

```
1 /boot ext3 200M
```

```

2      swap 4096M
3/home ext3 看情况（个人资料）
4/usr  ext3 看情况（程序数据）
5/    要记得看情况 不可以与 / 分别放在不同的 partition

```

使用`sudo fdisk -l`可查看硬盘分区(partition)情况，`df -h`查看文件系统使用情况，`du -sh /path/to` 统计指定目录的大小

## 1.5.1 分区与格式化

```

?
1 parted /dev/sdb
2
3 fdisk /dev/sdb
4 1 #列出分区类型
5 p #显示
6 d #删除
7 n #创建 1扇区512字节，1柱8M
8 t #改变分区类型
9 w #保存退出
10
11mkfs -t ext3 /dev/sdb1
12mkfs.vfat -F 32 /dev/sdb1 #格式化分区，不能格式化设备
13sync #确保格式化完成

```

## 1.6 快捷键

```

?
1 ctrl alt F1~F6    图形界面切换到字符界面
2 ctrl F7           字符界面切换到图形界面
3 ctrl shift +      放大窗口
4 ctrl -            缩小窗口
5 ctrl shift t      新建窗口
6 alt <num>         切换窗口
7 快捷复制          鼠标左键选中即复制 鼠标按中键即粘贴
8
9 ctrl l            清屏 或clear命令
10ctrl c            中止任务执行
11ctrl z            任务转至后面，挂起状态
12ctrl a            shell中输入命令时，光标跳到最前
13ctrl e            shell中输入命令时，光标跳到最后
14ctrl u            shell中输入命令时，删除一行
15
16ctrl s            暂停屏幕输入
17ctrl q            恢复屏幕输入
18ctrl d            退出终端

```

### 1.6.1 history

```

?
1!16                #运行history中第16行记录
2!cd                #运行history中最近的cd命令
3!!                 #运行history中上次命令
4ctrl r             #进入history搜索
5esc .              #使用上次命令的参数

```

6alt . #引用上次命令的参数，一直按，继续向上查找参数

## 1.7 帮助

?

```

1 <command> --help
2      #查找ls命令手册中有关时间与排序的条目
3      ls --help|grep -E "time|sort"
4
5 man      #man是分章节的
6          # 1 User Commands
7          # 2 System Calls
8          # 3 C Library Functions
9          # 4 Devices and Special Files
10         # 5 File Formats and Conventions
11         # 6 Games et. Al.
12         # 7 Miscellanea
13         # 8 System Administration tools and Deamons
14         # 3p posix手册
15         #查找fopen的使用，在第3章节查找
16         man 3 fopen
17         #查询所有含有passwd的帮助文件
18         man -k passwd
19
20         #操作同less
21         #space下一页；b上一页；enter下一行；q退出；G最后；g最前
22         #?<keyword>查找光标之前；/<keyword>查找光标之后
23         #n下一个关键字；shift n上一个关键字
24
25         #将man手册转存为文本
26         man ls |col -b >ls.txt #col -b 过滤掉所有的控制字符
27
28info <command>
29      #将info手册转存为文本
30      info make -o make.txt -s

```

## 1.8 启动过程

- 加载 BIOS 的硬件资讯与进行自我测试，并依据配置取得第一个可启动的装置；
- 读取并运行第一个启动装置内 MBR 的 boot Loader (亦即是 grub, spfdisk 等程序)；
- 依据 boot loader 的配置加载 Kernel，Kernel 会开始侦测硬件与加载驱动程序；
- 在硬件驱动成功后，Kernel 会主动呼叫 init 程序，而 init 会取得 run-level 资讯；
- init 运行 /etc/rc.d/rc.sysinit 文件来准备软件运行的作业环境（如网络、时区等）；
- init 运行 run-level 的各个服务之启动（script 方式）；
- init 运行 /etc/rc.d/rc.local 文件；
- init 运行终端机模拟程序 mingetty 来启动 login 程序，最后就等待使用者登录啦；

runlevel:

- 0 - halt (系统直接关机)
- 1 - single user mode (单人维护模式，用在系统出问题时的维护)
- 2 - Multi-user, without NFS (类似底下的 runlevel 3，但无 NFS 服务)
- 3 - Full multi-user mode (完整含有网络功能的纯文字模式)
- 4 - unused (系统保留功能)
- 5 - X11 (与 runlevel 3 类似，但加载使用 X Window)

- 6 - reboot (重新启动)

## 1.9 关机

?

```
1 shutdown -h now    #关机
2 init 0             #也是关机, 不推荐
3
4 reboot             #重启
```

## 2 文件管理

### 2.1 路径

?

```
1 pwd              #查看当前路径
2 cd               #回到自己home
3 cd ~             #回到自己home
4 cd ~jack         #去 jack的home
5 cd -             #切到上次目录
```

### 2.2 查看

?

```
1 ls
2
3 -a              #显示所有文件, 包含隐藏文件
4 ls -a | sort -i    #名称排序显示
5 -l             #显示详细信息, 别名为ll
6                #显示的 -rwx rw- --- 第一个是文件类型
7                #普通- 目录d 块设备b 字符设备c 符号链接l 套接字s 管道p
8 -F             #显示后缀 目录/ 符号链接@ 套接字= 管道| 可执行*
9
10 -t            #以修改时间顺序排列查询结果
11 -ut           #以访问时间顺序排列查询结果
12 -ct           #以属性修改时间顺序排列查询结果
13 -X            #以后缀名排序
14 -S            #以文件大小排序
15 -r            #逆向排序
16
17 --time=atime|ctime
18
19 -l            #单列显示
20 -h            #人性化显示
21 -i            #显示inode号
22 -d            #只显示目录信息
23              ls -d */          #只显示目录
24              ls -ld /<dir>    #可以看到该目录的访问权限, 不加d可能直接deny
25 -R            #递归显示目录结构
26 -F            #目录以/结尾显示
27 --full-time
28 ls -l / | grep tmp #只看某项
```

```

28 cat
29     -n    #给文件所有行编号
30     -A    #控制字符都显示
31     -s    #把多个空白行合并为一个空白行
32          #使文件清空，但所有者和权限不变
33     cat /dev/null > a.txt
34          #列出两个文件的内容. 执行对输出的搜索. 统计结果行的个数
35     cat file1 file2 | grep word | wc -l
36
37 od          #查看特殊格式的文件内容。通过指定该命令的不同选项可以以十进制、八进制、十六进制
38 和ASCII码来显示文件
39     -A    #指定地址基数(最左侧一栏)，包括：
40     d     #十进制
41     o     #八进制（系统默认值）
42     x     #十六进制
43     n     #不打印位移值
44
45     -t    #指定数据的显示格式，主要的参数有(除了选项c都可以跟一个十进制数n，指定每个显示值
46 所包含的字节数)：
47     c     #ASCII字符或反斜杠序列
48     d     #有符号十进制数
49     f     #浮点数
50     o     #八进制（系统默认值为02）
51     u     #无符号十进制数
52     x     #十六进制数
53
54     #od -Ax -tcx1 file.txt
55     #echo /etc/passwd | od
56     >0000000 062457 061564 070057 071541 073563 005144
57     >0000014
58
59     #echo /etc/passwd | od -A x -x
60     >000000 652f 6374 702f 7361 7773 0a64
61     >00000c
62
63     #echo /etc/passwd | od -A x -t x1
64     >000000 2f 65 74 63 2f 70 61 73 73 77 64 0a
65     >00000c
66
67     #echo /etc/passwd | od -A x -t x1 -t c    #常用 2个十六进制为一个字节
68     >000000 2f 65 74 63 2f 70 61 73 73 77 64 0a
69     / e t c / p a s s w d \n
70     >00000c
71
72     #系统先以十六进制并以小端表示数据，如ab表示为6261，然后再依次换算成对应的十进制
73 25185或八进制061141
74
75 xxd
76     #echo /etc/passwd | xxd
77     >00000000: 2f65 7463 2f70 6173 7377 640a                /etc/passwd.
78
79     #echo /etc/passwd | xxd -gl                #常用
80     >00000000: 2f 65 74 63 2f 70 61 73 73 77 64 0a                /etc/passwd.
81

```

```

82      #echo "0000000: 2f 65 74 63 2f 70 61 73 73 77 64 0a" |xxd -r
83      >/etc/passwd
84
85      #echo /etc/passwd |xxd -b
86      >0000000: 00101111 01100101 01110100 01100011 00101111 01110000  /etc/p
87      >0000006: 01100001 01110011 01110011 01110111 01100100 00001010  asswd.
88
89      #echo /etc/passwd |xxd -c4
90      >0000000: 2f65 7463  /etc
91      >0000004: 2f70 6173  /pas
92      >0000008: 7377 640a  swd.
93
94      #echo /etc/passwd |xxd -i
95      >0x2f, 0x65, 0x74, 0x63, 0x2f, 0x70, 0x61, 0x73, 0x73, 0x77, 0x64, 0x0a
96
97      #xxd -i file.txt
98      >unsigned char file_txt[] = {
99          0x2f, 0x65, 0x74, 0x63, 0x2f, 0x70, 0x61, 0x73, 0x73, 0x77, 0x64, 0x0a
100     };
101     unsigned int file_txt_len = 12;
102
103     #echo /etc/passwd |xxd -p
104     >2f6574632f7061737377640a
105
106hexdump
107     #echo /etc/passwd |hexdump
108     >0000000 652f 6374 702f 7361 7773 0a64
109     >000000c
110
111     #echo /etc/passwd |hexdump -C          #常用
112     >00000000 2f 65 74 63 2f 70 61 73 73 77 64 0a          | /etc/passwd. |
113     >0000000c
114
115     #还原文件
116     for buff in $(cat a.hex |cut -c10-58) ; do printf "\\x$buff"; done > a.txt
117     awk ' {for(i=2;i<=17;i++) printf "%c",strtonum("0X"$i)}' a.hex > a.txt
118
119echo -e "\x68\x65\x6c\x6c\x6f"    #16进制转换到字符串
120
121less      #分屏，可前翻
122      #space下一页；b上一页；enter下一行；q退出；G最后；g最前
123      #?<keyword>查找光标之前关键字；/<keyword>查找光标之后
124      #n下一个关键字；shift n上一个关键字
125more      #分屏，翻到最后直接退出，无法前翻
126
127tail
128      -f      #实时监控
129head
130
131tree      #树形显示目录
132      -a      #显示所有 文件
133      -d      #只显示 目录
134      -f      #显示全路径
135      -h      #人性化

```

```

136 -C #颜色
137 -L #层级
138 fdisk -l #查看硬盘分区情况
139 du #显示当前 目录 中 各子目录 占磁盘大小
140 -s #显示当前 目录 占磁盘大小
141 -h #人性化显示
142 -a #包括 文件
143 --max-depth=1 #设置显示深度
144 df #查看文件系统使用情况
145 -h #人性化显示
146 #查看某目录所属硬盘
147 df /<path> -h
148
149 nl #向文件添加行号
150 wc #统计行数，字符，字节
151 rev #左右相反
152 tac #cat相反
153 fmt #把文本格式化指定宽度
154 pr #给文件分页，把文本文件转换为标有页码的版本
155 fold #把输入行换行，以适应指定的宽度

```

## 2.6.3 查找

```

157 stat #文件信息查看
158 file #查看文件属性
1 getfacl #查看文件权限
1 grep <key> <path/to/file>
2 --color #高亮
3 -n #显示行号
4 -r #递归
5
6 -E, --extended-regexp PATTERN is an extended regular expression (ERE)
7 -F, --fixed-strings PATTERN is a set of newline-separated fixed strings
8 -G, --basic-regexp PATTERN is a basic regular expression (BRE)
9 -P, --perl-regexp PATTERN is a Perl regular expression
10
11 -v, --invert-match select non-matching lines(不匹配的)
12 #将配置文件中注释行和空行去掉显示
13 cat /etc/dhcpd/dhcpd.conf | grep -v "^#" | grep -v "^$"
14
15 grep "key" ./ * #查找当前目录所有文件
16
17 find <path>
18 -name "pattern"
19 -size +1000M
20 #查找指定类型的文件，然后指定按时间排序
21 find . -name *.php | xargs ls -alt
22 ls -alt $(find . -name *.php)
23 ls -alt `find . -name *.php`
24 #统计所有c文件的行数总和，不包括空行
25 find . -name "*.c" | xargs cat | grep -v ^$ | wc -l
26 find . -name "*.c" | xargs cat | sed '/^$/d' | wc -l
27
28 -i #lignore
29 -o #or
30 ! #not

```



## 2.4 创建

[?](#)

```
1 touch      #已创建的不会被清空，但会更新访问时间
2
3 mkdir
4           #创建多个目录
5           mkdir dir1 dir2
6 -p         #创建层次目录
7           mkdir -p aa/bbb/ccc
8
9 ln         #创建硬连接
10 -s        #创建符号连接
```

## 2.5 复制

[?](#)

```
1cp <src> <dest>
2 -r        #recursive, 实现同时拷贝文件和文件夹
3 -f        #force, 强制覆盖
4 -v        #verbose
5 -i        #interactive
6 -a        #权限也复制
7
8watch du <target> -h -d 0 #通过不停的监视目标目录的大小来确定复制进度
```

## 2.6 移动

[?](#)

```
1mv <src> <dest>      #需要重命名，只需在同目录下执行就行
2
3find -type f|grep 'jpg'|xargs rename 's/\.jpg$/\.png/' #批量重命名
```

## 2.7 删除

[?](#)

```
1rm          #安全的删除，建议先进入目录后再删除
2 -r         #recursive
3 -f         #force
4rm -rf /etc/yum.repos.d/* #清空目录
5rm -rf /etc/yum.repos.d/  #删除目录
```

## 2.8 归档压缩

归档压缩其实是两个步骤，常用选项说明

[?](#)

```
1-c        #create
2-v        #verbose
3-z        #gzip
4-j        #bzip2
5-f        #file指定文件
6-x        #extract
• 归档
```

[?](#)

```
ltar cvf ab.tar a b c dfile
```

- 解包

[?](#)

```
ltar xvf ab.tar [-C /path]
```

- 压缩

[?](#)

```
lgzip a b c dfile #文件是分开压缩的
```

```
2bzip2
```

- 解压

[?](#)

```
lgunzip a.gz
```

```
2bunzip2 a.bz2
```

- 归档压缩

[?](#)

```
ltar cvzf ab.tar.gz a b c dfile
```

```
2tar cvjf ab.tar.bz2 a b c dfile
```

- 解压解包

[?](#)

```
ltar xvzf ab.tar.gz
```

```
2tar xvjf ab.tar.bz2
```

## 3 软件安装

### 3.1 RPM安装

软件包名，指安装时的文件名，而软件名，指系统中识别的名称

#### 3.1.1 挂载镜像

[?](#)

```
lmount -o loop /path/to/rhel.iso /mnt
```

iso文件，即iso9660，这是光盘所使用的一种文件系统

[?](#)

```
lls /mnt/Packages
```

将看到许多.rpm的文件，通常格式如<软件名>.<版本>.<OS>.<CPU>.rpm，其中

- OS: e15, e16
- CPU: i386~i686(32位), x86\_64(64位), noarch(通用)

#### 3.1.2 安装

[?](#)

```
lrpm -ivh <软件包名>
```

```
2 -i #install
```

```
3 -vh #显示安装过程信息
```

```
4 --force #强制安装
```

不需要指定安装目录，默认安装于/usr（分区时最好单独分出来）

### 3.1.3 查询

- 查询已安装的软件包-q

?

```
1rpm -qa | grep vnc    #模糊查询
2rpm -q gcc            #明确知道软件名
```

- 查询软件安装位置-l

?

```
1rpm -ql tigervnc
```

- 查询文件所属软件，方便根据文件找到软件 再安装，或根据命令找到程序-f

?

```
1rpm -qf /path/to/file
```

- 查询已安装软件详细信息-i

?

```
1rpm -qi <软件名> (不是包)
```

### 3.1.4 删除

?

```
1rpm -e <软件名> (不是包)
2    -e          #erase
3    --nodeps    #不查看依赖性关系删除，防止缺失文件不能删除
实在卸载不了时，可强制安装--force，再删除
```

## 3.2 YUM安装

### 3.2.1 配置源

?

```
1rm -rf /etc/yum.repos.d/*    #清空目录
2vim /etc/yum.repos.d/yum.repo #以.repo结尾
键入如下信息
```

?

```
1[rhel6]
2name=rhel6
3baseurl=file:///mnt/Server
4enabled=1
5gpgcheck=0
```

一个文件中可存在多个五行，从上往下依次查找安装，enabled=0则不查找该源

### 3.2.2 安装

?

```
1yum -y install <软件名> #可带通配符
2yum -y groupinstall <软件组名> #如：通用桌面
```

### 3.2.3 查询

?

```
1yum repolist
2yum list          #第三列是yum源名字时，说明没装上
3yum grouplist
4yum search keyword>
```

### 3.2.4 卸载

?

```
1yum -y remove
2yum -y erase
```

### 3.2.5 使用163源

rhel6中使用163的centos源

?

```
wget http://mirrors.163.com/centos/6/os/x86_64/Packages/python-iniparse-0.3.1-
2.1.el6.noarch.rpm
1 wget http://mirrors.163.com/centos/6/os/x86_64/Packages/yum-metadata-parser-1.1.2-
2 16.el6.x86_64.rpm
3 wget http://mirrors.163.com/centos/6/os/x86_64/Packages/yum-3.2.29-60.el6.centos.noarch.rpm
4 wget http://mirrors.163.com/centos/6/os/x86_64/Packages/yum-plugin-fastestmirror-1.1.30-
5 30.el6.noarch.rpm
6
7 rpm -aq|grep yum|xargs rpm -e --nodeps
8
9 rpm -ivh python-iniparse-0.3.1-2.1.el6.noarch.rpm
10rpm -ivh yum-metadata-parser-1.1.2-16.el6.x86_64.rpm
11rpm -ivh yum-3.2.29-60.el6.centos.noarch.rpm yum-plugin-fastestmirror-1.1.30-
1230.el6.noarch.rpm
13
14wget http://mirrors.163.com/.help/CentOS6-Base-163.repo
15mv CentOS6-Base-163.repo /etc/yum.repos.d/
16vim /etc/yum.repos.d/CentOS6-Base-163.repo
17 :%s/\$releasever/6/g
18 :wq
19
20yum clean all
21yum makecache
```

### 3.3 RPM vs YUM

两者都安装rpm结尾的软件

- rpm不解决依赖关系，优先使用
- yum解决依赖关系，自动安装，需要配置源，在64位机器上默认不查找32位的软件，特殊情况下可手工rpm安装需要的软件

## 3.4 编译安装

### 3.4.1 安装

利用命令将源代码的包（如.tar.gz,.tar.bz2）进行解压解包

?

```
1./configure --prefix=<path/to/setup> #建议指定，删除时方便
2                                     #否则分散在usr/locals各子目录
3                                     #此步骤将产生配置文件
4make                                #编译
```

### 3.4.2 启动 #安装

手动安装的一般在usr/locals目录下，如httpd，此时应该在/usr/local/apache2，通过/usr/local/apache2/bin/apachectl start来启动

### 3.4.3 删除

直接删除由--prefix指定的目录

## 3.5 apt-get 安装

ubuntu中使用apt-get

[?](#)

- 1 apt-cache search package 搜索包
  - 2 apt-cache show package 获取包的相关信息，如说明、大小、版本等
  - 3 sudo apt-get install package 安装包
  - 4 sudo apt-get install package=version 安装指定版本
  - 5 apt-show-versions -p packagename 查看当前特定软件版本
  - 6 sudo apt-get install package -- reinstall 重新安装包
  - 7 sudo apt-get -f install 修复安装“-f = --fix-missing”
  - 8 sudo apt-get remove package 删除包
  - 9 sudo apt-get remove package -- purge 删除包，包括删除配置文件等
  - 10 sudo apt-get update 更新源
  - 11 sudo apt-get upgrade 更新已安装的包
  - 12 sudo apt-get dist-upgrade 升级系统
  - 13 sudo apt-get dselect-upgrade 使用 dselect 升级
  - 14 apt-cache depends package 了解使用依赖
  - 15 apt-cache rdepends package 是查看该包被哪些包依赖
  - 16 sudo apt-get build-dep package 安装相关的编译环境
  - 17 apt-get source package 下载该包的源代码
  - 18 sudo apt-get clean && sudo apt-get autoclean 清理无用的包
  - 19 sudo apt-get check 检查是否有损坏的依赖
  - 20
  - 21 dpkg -i file.deb 直接利用包管理器安装包
- 当出现E: Sub-process /usr/bin/dpkg returned an error code (1)

[?](#)

- 1 sudo mv /var/lib/dpkg/info /var/lib/dpkg/info.bak #现将info文件夹更名
- 2 sudo mkdir /var/lib/dpkg/info #新建一个info文件夹
- 3 sudo apt-get update

### [更新源](#)

更新后需要sudo apt-get update

不能成功时，删除/var/lib/apt/lists/partial/下的失败项

## 3.6 搜索程序

[?](#)

- 1 which #定位一个命令，搜索磁盘上命令，不能找到内置命令
- 2 whereis #定位一个命令的二进制、源文件、手册
- 3

```

4 type                #查看是否是内置命令
5
6 apropos              #查看相关名字命令的说明 同man -k
7 whatis              #查看相关名字命令的说明
8
9 locate              #搜索文件，需要配置/etc/updatedb.conf（运行updatedb）
10 slocate
11
12 ldd                 #列出程序的动态依赖关系 ldd /bin/cp

```

## 3.7 curl使用

```

?
curl http://www.linuxidc.com
1
2 #指定输出文件
3 curl -o page.html http://www.linuxidc.com
4 #指定代理
5 curl -x 123.45.67.89:1080 -o page.html http://www.linuxidc.com
6 #保存cookie
7 curl -x 123.45.67.89:1080 -o page.html -D cookie0001.txt http://www.linuxidc.com
8 #使用cookie
9 curl -x 123.45.67.89:1080 -o page1.html -D cookie0002.txt -b cookie0001.txt
10 http://www.linuxidc.com
11 #指定referer
12 curl -A "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)" -x 123.45.67.89:1080 -e
13 "mail.linuxidc.com" -o page.html -D cookie0001.txt http://www.linuxidc.com
14
15 #下载文件
16 curl -O http://cgi2.tky.3web.ne.jp/~zzh/screen1.JPG
17 #文件名匹配下载
18 curl -O http://cgi2.tky.3web.ne.jp/~zzh/screen[1-10].JPG
19 #下载多个目录下的文件
20 curl -O http://cgi2.tky.3web.ne.jp/~{zzh,nick}/[001-201].JPG
21
22 #302跟踪
23 curl -L -O http://download.savannah.gnu.org/releases/nmh/nmh-1.1-RC4.tar.gz
24
25 #断点续传，比如我们下载screen1.JPG中，突然掉线了，我们就可以这样开始续传
26 curl -c -O http://cgi2.tky.3wb.ne.jp/~zzh/screen1.JPG
27 #分块下载
28 curl -r 0-10240 -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.mp3 &\
29 curl -r 10241-20480 -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.mp3 &\
30 curl -r 20481-40960 -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.mp3 &\
31 curl -r 40961- -o "zhao.part1" http://cgi2.tky.3web.ne.jp/~zzh/zhao1.mp3
32
33 cat zhao.part* > zhao.mp3
34
35 #ftp下载
36 curl -u name:passwd ftp://ip:port/path/file
37 curl ftp://name:passwd@ip:port/path/file
38
39 #上传文件
40 curl -T localfile -u name:passwd ftp://upload_site:port/path/ #ftp

```

```

41curl -T localfile http://cgi2.tky.3web.ne.jp/~zzh/abc.cgi #http put
42
43#提交表单
44curl -d "user=nickwolfe&password=12345" http://www.linuxidc.com/login.cgi #http post
45
46#表单中上传文件
47curl -F upload=@localfile -F nick=go http://cgi2.tky.3web.ne.jp/~zzh/up_file.cgi

```

## 4.1 创建

?

```

1useradd <username>
2useradd -u <uid> <username>
3useradd -g <gid> <username> #创建时指定主属组
4useradd -G <gid>[,<gid>...] <username> #创建时指定附属组

```

## 4.2 查看

?

```

1id <username>
出现类似uid=502(jim) gid=502(jim) 组=502(jim),0(root)

```

- uid:user identify, 数字, 可能出现uid相同, 但用户名不同的情况, 以uid为准
- gid:group identify, 数字
- 组:主属组+附属组, 数组, 通过usermod -G root jim给jim设置 附属组

但可能会使得以前执行过的gpasswd -a命令失效

?

```

1cat /etc/passwd | grep <username>
出现类似jim:x:502:502::/home/jim:/bin/bash的信息, 依次为

```

- 用户名
- 密码占位, 删除后用户就没密码, 虽然不能改密码, 但能置空密码(可通过删除root的此信息, 达到进入root账户)
- uid
- gid
- 描述字段
- 家目录
- 登录shell

/etc/shadow才是真正存储密码的地方

## 4.3 修改

?

```

1usermod -u <uid> <username> #修改用户id
2usermod -g <gid> <username> #修改用户主属组
3usermod -G <gid>[,<gid>...] <username> #修改用户附属组

```

## 4.4 切换用户

?

1su - <username> #加载环境变量 如.profile

同理,在arm开发中,不加载环境变量,不修改脚本中的-,也是用于加载环境设置的

## 4.5 设置密码

?

1passwd [<username>] #不加<username>给自己重设

2 #root用户能设置简单的密码,其他用户不行

## 4.6 删除账户

?

1userdel -r <username>

2 -r #家目录删除

## 4.7 查看登录信息

实时的登录

?

1whoami

2who

3w

4ps u

5users

登录日志

?

1last <username> <tty>

2finger <username>

发信息

?

1write <username> <tty>

2wall

## 5 组管理

添加账户时,就会产生一个同名的组,这个组称为用户的 主属组

### 5.1 添加组

?

1groupadd <groupname>

### 5.2 查看组

?

1cat /etc/group

格式为组名:组密码占位(目前无用):gid:组员(数组)

### 5.3 切换组



[?](#)

```
lnewgrp <groupname> #临时切换到用户的某个附属组，exit退出
```

## 5.4 添加用户进组

[?](#)

```
lgpasswd -a <username> <groupname>
```

## 5.5 删除用户出组

[?](#)

```
lgpasswd -d <username> <groupname>
```

## 5.6 删除组

[?](#)

```
lgroupdel <groupname>
```

# 6 文件权限

## 6.1 查看

[?](#)

```
lls -l <file>
```

得到- rwx rw- ---，第一个字符表示文件类型：普通- 目录d 块设备b 字符设备c 符号链接文件l 套接字文件s 管道文件p；剩下的分成三组，简称ugo权限：第一组代表user权限；第二组代表group权限；第三组代表others权限

## 6.2 修改

[?](#)

```
1chmod u-x a.txt
```

```
2chmod g+x a.txt
```

```
3chmod o+w a.txt
```

```
4chmod u-x,g+x,o+w a.txt
```

```
5chmod u+w-x a.txt
```

以数字方式修改 read:r=4 write:w=2 execute:x=1

[?](#)

```
lchmod 744 a.txt
```

对于root用户，x权限同属主，rw的设置是无效的，依然能读能写

### 6.2.1 suid

表示设置文件在 执行阶段(权限仅对二进制程序(binary program)有效，不能够用在 shell script 上面) 具有 文件所有者 的权限

### [参考](#)

如ll /bin/ping，得到-rwsr-xr-x，其中s就是suid，普通用户也是能使用ping的(此处为root)的权限

如普通用户无法通过cat查看/etc/shadow，但修改了cat的权限chmod u+s /bin/cat后，就可以了

## 6.2.2 sgid

### [参考](#)

如ll /usr/bin/locate, 得到-rwx--s--x, 其中s就是sgid

- 对二进制程序, 运行者在运行过程中将会获得该程序群组的支持, 上面普通用户将得到slocate群组支持
- 针对目录
- 使用者若对此目录具有 r 与 x 的权限时, 该使用者能够进入此目录
- 使用者在此目录下的有效群组(effective group)将会变成该目录的群组
- 若使用者在此目录下具有 w 的权限(可以新建文件), 则使用者所创建的新文件, 该新文件的群组与此目录的群组相同。

## 6.2.3 skicky

### [参考](#)

只针对 目录 有效, 当使用者在该目录下创建文件或目录时, 仅有自己与 root 才有权力删除该文件

## 6.3 修改文件所属者/组

间接修改了权限

### [?](#)

```
1chown <username> <file>    #修改文件所属者
2chown .<groupname> <file>  #修改文件所组, 前面加点
3chown <username>:<groupname> <file> #同时修改文件所属者与组
4chgrp <groupname> <file>    #修改文件所组
```

## 7 进程控制

### 7.1 查看进程

#### [?](#)

```
ps          #当前终端上进程
1  a        #所有终端上进程, -a则为只显示非终端上进程
2  ax       #所有进程, 包括非终端上进程
3  aux      #把进程启动的用户(及起始时间)包含进来
4  e        #显示shell环境信息
5  f        #格式化父子进程的显示
6  l        #较长、较详细的将该 PID 的资讯列出
7  j        #job的格式
8  axjf     #连同部分程序树状态
9
10ps -l
11 #F: 代表这个程序旗标 (process flags), 说明这个程序的总结权限
12     #4 表示此程序的权限为 root;
13     #1 则表示此子程序仅进行复制(fork)而没有实际运行(exec)
14 #S: 代表这个程序的状态 (STAT), 主要的状态有
15     #R (Running): 该程序正在运行中;
16     #S (Sleep): 该程序目前正在睡眠状态(idle), 但可以被唤醒(signal)。
```

```

17      #D : 不可被唤醒的睡眠状态, 通常这支程序可能在等待 I/O 的情况(ex>列印)
18      #T : 停止状态(stop), 可能是在工作控制(背景暂停)或除错 (traced) 状态;
19      #Z (Zombie): 僵尸状态, 程序已经终止但却无法被移除至内存外。
20      #UID/PID/PPID: 代表『此程序被该 UID 所拥有/程序的 PID 号码/此程序的父程序 PID 号码』
21      #C: 代表 CPU 使用率, 单位为百分比;
22      #PRI/NI: Priority/Nice 的缩写, 数值越小代表该程序越快被 CPU 运行
23      #ADDR/SZ/WCHAN:
24          #ADDR 是 kernel function, 指出该程序在内存的哪个部分, 如果是个 running 的程序, 一般
25就会显示『 - 』
26          #SZ 代表此程序用掉多少内存
27          #WCHAN 表示目前程序是否运行中, 同样的, 若为 - 表示正在运行中
28      #TTY: 登陆者的终端机位置, 若为远程登陆则使用动态终端介面 (pts/n);
29      #TIME: 使用掉的 CPU 时间, 注意, 是此程序实际花费 CPU 运行的时间, 而不是系统时间;
30      #CMD: 就是 command 的缩写, 造成此程序的触发程序之命令为何
31
32ps aux
33      #USER: 该 process 属于那个使用者帐号的?
34      #PID : 该 process 的程序识别码。
35      #%CPU: 该 process 使用掉的 CPU 资源百分比;
36      #%MEM: 该 process 所占用的实体内存百分比;
37      #VSZ : 该 process 使用掉的虚拟内存量 (Kbytes)
38      #RSS : 该 process 占用的固定的内存量 (Kbytes)
39      #TTY : 该 process 是在那个终端机上面运行, 若与终端机无关则显示 ?,
40          #tty1-tty6 是本机上面的登陆者程序, 若为 pts/0 等等的, 则表示为由网络连接进主机的
41程序。
42      #STAT: 该程序目前的状态, 状态显示与 ps -l 的 S 旗标相同 (R/S/T/Z)
43      #START: 该 process 被触发启动的时间;
44      #TIME : 该 process 实际使用 CPU 运行的时间。
45      #COMMAND: 该程序的实际命令为何?
46
47pstree
48# 查看运行时系统状态
49# -p 列出每个 process 的 PID
50# -u 列出每个 process 的所属帐号名称

```

?

```

top
1      -d      #后面可以接秒数, 就是整个程序画面升级的秒数。默认是 5 秒;
2      -n      #与 -b 搭配, 意义是, 需要进行几次 top 的输出结果
3      -p      #指定某些个 PID 来进行观察监测而已
4
5vmstat
6      -a      #使用 inactive/active(活跃与否) 取代 buffer/cache 的内存输出资讯;
7      -f      #启动到目前为止, 系统复制 (fork) 的程序数;
8      -s      #将一些事件 (启动至目前为止) 导致的内存变化情况列表说明;
9      -S      #后面可以接单位, 让显示的数据有单位。例如 K/M 取代 bytes 的容量;
10     -d      #列出磁碟的读写总量统计表
11     -p      #后面列出分割槽, 可显示该分割槽的读写总量统计表
12     #procs
13         #r : 等待运行中的程序数量
14         #b: 不可被唤醒的程序数量
15         #这两个项目越多, 代表系统越忙碌 (因为系统太忙, 所以很多程序就无法被运行或一直在等待
16而无法被唤醒之故)。
17     #memory
18         #swpd: 虚拟内存被使用的容量;

```

```

19      #free: 未被使用的内存容量;
20      #buff: 用于缓冲内存;
21      #cache: 用于高速缓存。
22      #这部份则与 free 是相同的。
23  #swap
24      #si: 由磁碟中将程序取出的量;
25      #so: 由于内存不足而将没用到的程序写入到磁碟的 swap 的容量。
26      #如果 si/so 的数值太大, 表示内存内的数据常常得在磁碟与主内存之间传来传去, 系统效能
27会很差!
28  #io
29      #bi: 由磁碟写入的区块数量;
30      #bo: 写入到磁碟去的区块数量。
31      #如果这部份的值越高, 代表系统的 I/O 非常忙碌!
32  #system
33      #in: 每秒被中断的程序次数;
34      #cs: 每秒钟进行的事件切换次数;
35      #这两个数值越大, 代表系统与周边设备的沟通非常频繁! 这些周边设备当然包括磁碟、网络
36卡、时间钟等。
37  #cup
38      #us: 非核心层的 CPU 使用状态;
39      #sy: 核心层所使用的 CPU 状态;
40      #id: 闲置的状态;
41      #wa: 等待 I/O 所耗费的 CPU 状态;
42      #st: 被虚拟机器 (virtual machine) 所盗用的 CPU 使用状态 (2.6.11 以后才支持)。
43
44vmstat -a <延迟> <总计侦测次数> #CPU/内存等资讯
45vmstat -fs                      #内存相关
46vmstat -S <单位>                #配置显示数据的单位
47vmstat -d                      #与磁碟有关
48vmstat -p <分割槽>              #与磁碟有关

```

## 7.3 杀死进程

```

49free #内存空间
50-m   #字节单位为M
?     -t   #在输出的最终结果, 显示实体内存与 swap 的总量
1ps aux | grep <keyword>          #查到进程pid
2ps aux | egrep '(cron|syslog)'    #找出与 cron 与 syslog 这两个服务有关的PID
3pidof sshd                        #查找sshd的进程pid
4
5kill -9 <pid>
kill -l 列出所有信号:

```

- -9 强制
- -15 默认
- -19 挂起
- -18 唤醒

?

```

1w          #看到所有登录系统的账户, 查看是否异常
2killall -9 sshd #杀死进程名称为sshd的进程,
3           #即把所有通过ssh登录到系统的账户登出
4
5pkill -9 httpd #杀死进程名称为httpd的进程

```

## 7.4 后台任务

- 在命令尾处键入&把作业发送到后台
- 也可以把正在运行的命令发送到后台运行，首先键入Ctrl+Z挂起作业（此时是挂起状态），然后键入bg移动后台继续执行
- bg %jobnumber 或bg %name
- fg %jobnumber 把后台作业带到前台来
- kill -18 pid 也是唤醒
- kill %jobnumber 删除后台作业
- jobs -l将PID显示 -r运行中显示 -s显示停止
- disown %jobnumber从后台列表中移除任务，并没有终止
- nohup command & 如果你正在运行一个进程，而且你觉得在退出帐户时该进程还不会结束，那么可以使用nohup命令。该命令可以在你退出帐户之后继续运行相应的进程。

## 8 Linux其它

### 8.1 重定向

?

```

1 >          #输出重定向 1> 的简写
2          ls > /dev/null  #丢掉输出
3          ls > /dev/pts/2 #输出到另一个终端
4 >>        #输出重定向追加
5 2>        #错误重定向
6 2>>       #错误重定向追加
7
8 &>        #标准与错误一起重新向
9          ls > 0.txt 2>&1  #同上
10 >> 2>>    #标准与错误一起重新向追加
11
12 <        #输入重定向
13          cat < /dev/zero -A
14
15 <<       #输入重定向，并定义结束符
16          cat << EOF  #打开文件缓冲区，输入完后，最后输入'EOF'
17
18          #下载ftp上内容，这类 交互性 的东西也可以写成脚本
19          lftp 192.168.1.245 << EOF
20          > get somefile
21          > EOF
22
23 tee      #在管道过程中产生分支
24          grep -v "a" file | tee file1 | grep -v "b" file
25
26 xargs
27          find . -name *.php |xargs ls -alt
28          #等同于
29          ls -alt $(find . -name *.php)
30
31          -n 能确定命令使用的参数个数,下面一次传递一个参数
32          ls *.php | xargs -n 1 ls -alt
except可用于写交互性的脚本

```

### 8.2 环境变量

[?](#)

lenv #查看当前用户的环境变量

一般情况下，环境变量由以下几个文件顺序进行加载

- /etc/profile
  - /etc/profile.d/\*.sh
- ~/.bash\_profile
  - ~/.bashrc
    - /etc/bashrc

非登录shell情况如 su tom, 先

- ~/.bash\_profile
  - ~/.bashrc
    - /etc/bashrc

然后/etc/profile.d/\*.sh

没有/etc/profile

[?](#)

```
1 vim /etc/profile
2 export LC_ALL=zh_CN.GB18030          #设置语言信息
3 export PS1='\u@\h:\w\$'              #修改命令行提示符
4
5 vim /etc/sysconfig/i18n
6 LANG="zh_CN.GB18030"                 #当前系统的语言环境变量设置
7 SUPPORTED="zh_CN.GB18030:zh_CN:zh:en_US.UTF-8:en_US:en" #系统支持哪些字符集
8 SYSFONT="latarcyrheb-sun16"          #系统终端字符的字体
9
10 yum install "@Chinese Support"
设置时区与同步时间
```

[?](#)

```
1 cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
2 ntpdate cn.pool.ntp.org
```

## 8.3 Shell相关

[?](#)

```
1 alias                                #查看别名，别名的优先级高
2 alias la="ls -a"                    #设置别名
3
4 unalias la                          #取消别名
5 \<command>                          #不使用别名时，直接使用原始命令
6
7 <command1>&&<command2>               #顺序执行，前面的不出错，后面的才执行
8 <command1>||<command2>              #顺序执行，前面的出错，后面的才执行
9 <command1>;<command2>               #顺序执行，不管前面结果，后面照样执行
10
11 {}                                  #命令的展开
12 mkdir dir{5..8}
13 mkdir dir{a..d}
14 mkdir dir{a,d}
15                                     #相当于括号外面 * 括号里现
```

```

16 cp /abc{0,1} #cp /abc0 /abc1
17
18#保持不变, 防止shell翻译
19 touch "a b"
20 touch a\ b
21 touch a' 'b
22
23echo -n #不换行, 默认换行
24echo -e "a\tb\tc" #翻译转义, 否则原样输出
25
26echo ${PAGER:-more} #如果变量PAGER定义则其非空, 则使用, 否则使用字符串more

```

## 8.4 Shell脚本

[?](#)

```

1 $1~$9 #位置变量
2 $* #具体参数数组, 同$@
3 $# #共几个参
4 $$ #PID
5 $? #上次命令执行的返回值, 0为正常, 非0为异常
6
7 test #if后面的条件都是 test 条件, 可简写成 [ .. ]
8 -f #是否是文件
9 -r #是否可读
10 -w #是否可写
11 -u #是否具有suid
12 -ot #是否比某文件旧
13 -gt #数字比较 great than
14 -a #and
15
16for((i=1;i<255;i++)); do
17 ping -c1 192.168.1.$i > /dev/null 2>&1
18 if [ $? -eq 0 ]; then
19 echo "192.168.1.$i is running"
20 fi
21done
22
23#双重[[ .. ]] 解决单层 [ .. ] 不能逻辑运算和正则匹配的问题
24[[ !cmd ]]
25[[ cmd && cmd ]]
26[[ cmd || cmd ]]
27[[ $z =~ "o" ]]

```

## 8.5 数学运算

[?](#)

```

1 bc
2 # echo "(6+3)*2" |bc
3 # echo "3+4;5*2;5^2;18/4" |bc
4 # echo "scale=2;15/4" |bc
5 # echo "ibase=16;A7" |bc
6 # echo "ibase=2;11111111" |bc
7 # echo "ibase=16;obase=2;B5-A4" |bc
8 # bc calc.txt
9

```

```

10expr
11          # expr 6 + 3
12          > 9
13          # expr 2 /* 3      (有转义符号)
14          > 6
15
16          # a=3
17          # expr $a+5      (无空格)
18          > 3+5
19          # expr $a + 5    (有空格)
20          > 8
21          # expr length "abc"
22          > 3
23          # expr substr "abcdefgh" 2 3 (从1开始计数)
24          > bcd
25          # expr index "abcdef" cd    (从1开始计数)
26          > 3
27
28echo
29          # echo $((3+5))
30          > 8
31          # x=2;y=3
32          # echo $((x+y))
33          > 5
34          # echo $x+$y | bc
35          > 5
36          # echo $[x+y]
37          > 5
38
39let
40          # let x=$x+1
41          # echo $x
42          > 3
43
44akw
45          # awk 'BEGIN{a=3+2;print a}'
46          > 5
47          # awk 'BEGIN{a=(3+2)*2;print a}'
48          > 10

```

## 8.6 杂项

[?](#)

```

mount -l          #查看目前所有挂载
1 mount /dev/sdb /media      #挂载U盘、光盘
2      /dev/sr0 /mnt
3 umount /dev/sdb          #取消挂载U盘
4                          #多取消几次 防止被多次挂载了
5 sudo fdisk -l | grep NTFS  #挂载ntfs的u盘
6 mount -t ntfs-3g <NTFS Partition> <Mount Point>
7
8 sed -n '5,10p' /etc/passwd      #只查看文件的第5行到第10行。
9 uniq                          # -d 只显示重复的行 -u 只显示唯一的行
10sort names | uniq -d            #显示在names文件中哪些行出现了多次

```



```

11      -r      #正常排序的反序
12      -n      #按数字次序排序
13      -f      #大小写混合在一起排序
14
15
16tr      #翻译
17      #转换大小写
18      cat /etc/passwd | tr [a-z] [A-Z]
19      #计算
20      echo "1 2 3 4 5" | tr " " "+" | bc
21      #删除文件file中出现的换行'\n'、制表'\t'字符
22      cat file | tr -d "\n\t" > new_file
23      #删除空行 -s 删除所有重复出现字符序列，只保留第一个；即将
24重复出现字符串压缩为一个字符串。
25
26      cat file | tr -s "\n" > new_file
27      #把路径变量中的冒号":", 替换成换行符"\n"
28      echo $PATH | tr -s ":" "\n"
29
30      #快速创建大文件
31      #dd if=/dev/zero of=test bs=1M count=1000
32      #创建空洞文件
33      #dd if=/dev/zero of=test bs=1M count=0 seek=100000
34      #把第一个硬盘的前 512 个字节存为一个文件:
35      dd if=/dev/hda of=disk.mbr bs=512 count=1
36      #修复硬盘(直接使用标准IO导致I/O错误时)
37      dd if=/dev/sda of=/dev/sda
38
39split      #按行数或字节数拆分文件
40csplit      #由正则来分割文件
41
42strip      #将编译链接的可执行文件进行剪切，去掉头部信息，通过nm
43将不能查看到稳中各种符号
44
45locale      #查看系统当前locale环境变量
46
47lsusb      #检查USB
48
49iwconfig      #检测无线
50iwlist
51
52stty -echo      #禁止回显
53stty echo      #打开回显
54
55stty igncr      #忽略回车符
56stty -igncr      #恢复回车符
57
58
59insmod
60rmmod
61lsmod
62dmesg
63
64taskset      #将进程分配给指定的CPU执行

```

```

65
66
67mknod <filenm> c 80 0
68
69debian kernel api has doc
70
71sudo dpkg-reconfigure gdm
72sudo /etc/init.d/lightdm start
73
74网络配置
75sudo loginctl enable-linger-session > ~/.xinitrc
76sudo /etc/init.d/gdm start #代替startx

```

## 常用网络命令

### 9.1 查看网卡

```

?
1ifconfig eth0
2ip addr show #或简写成 ip addr sh

```

### 9.2 启动网卡

```

?
1ifconfig eth0 up
2ifconfig eth0 down
3ifdown eth0
4ifup eth0

```

### 9.3 临时设置网卡

```

?
1ifconfig eth0 <ip> netmask <ip> up
2ip addr add <ip> netmask 255.255.255.0 dev eth0
3ip addr add <ip/24> dev eth0 #等同于上面，24个1（二进制）即255.255.255.0
4ifconfig eth0:0 192.168.50.50 # eth0:0 那就是在该实体网卡上，再仿真一个网络接口，一张网络
5卡上面设置多个IP

```

### 9.4 查看网关

```

?
1route -n
2ip route show

```

### 9.5 临时设置网关

```

?
1route add default gw <ip>

```

### 9.6 临时设置DNS

```

?
1vim /etc/resolv.conf
2
3nameserver 202.106.0.46 #最多三个wh
4
5#查询域名对应ip

```

```
6host baidu.com
7nslookup baidu.com
8dig baidu.com
```

## 9.7 永久的网络设置

[?](#)

```
1 vim /etc/sysconfig/network-scripts/ifcfg-eth{0,1..}
2
3 ONBOOT                                #是否启用网卡
4 NM_CONTROLLED                         #是否受networkmanager(gui)控制
5 BOOTPROTO=dhcp|none|static           #获取ip方式
6
7 NETMASK=255.255.255.0
8 PREFIX=24
9
10GETEWAY
11DNS1
12DNS2
```

## 9.8 重启网卡

[?](#)

```
lservice network restart
```

## 9.9 ubuntu下设置

[?](#)

```
1 sudo vim /etc/network/interfaces
2
3     # The primary network interface
4     auto eth0
5     iface eth0 inet static
6     address 192.168.3.90
7     gateway 192.168.3.2
8     netmask 255.255.255.0
9     network 192.168.3.0
10    broadcast 192.168.3.255
11
12sudo /etc/init.d/networking restart
13
14sudo nm-connection-editor & #图形化配置
```

## 9.10 其它

[?](#)

```
1 lsof -i:80                                #查看80端口是否启动
2 mii-tool eth0                            #eth0网卡是否有效
3 ping <ip>
4     -c1                                    #ping一个包
5
6 netstat -ntlp
7     -n                                    #以网络IP地址代替名称，显示出网络连接情形
8     -t                                    #显示TCP协议的连接情况，-u是UDP
9     -l                                    #正在listening的
```

```

10 -p #显示程序名字
11 -a #将目前系统上所有的连线、监听、Socket 数据都列出来
12
13service iptables stop #关闭防火墙
14iptables -F #关闭防火墙
15setenforce 0 #关闭SELinux
16
17nslookup <domain> #根据域名查看ip, 附带检查dns效果
18
19system-config-network #GUI方式配置, RHEL6有问题
20 #可把它关闭`/etc/init.d/NetworkManager stop`
查看指定端口

```

## ?

```

1# 查看哪些进程打开了指定端口port (对于守护进程必须以root用户执行才能查看到)
2lsof -i:<port>
3
4# 查看哪些进程打开了指定端口port, 最后一列是进程ID (此方法对于守护进程作用不大)
5netstat -nap|grep <port>
常用服务端口号, 可通过grep 3306 /etc/services查看某端口的服务, 小于1024的端口要启动时, 启动者身份必须是root才行

```

- tcp
  - ftp-data 20
  - ftp 21
  - ssh 22
  - telnet 23
  - smtp 25
  - dns 53
  - http 80
  - pop3 110
  - https 443
- udp
  - dhcp 67 68

windows下网络配置查看

```
ipconfig [/release] [/renew] [/all]
```

## 9.11 服务的防火墙

任何以 xinetd 管理的服务, 都可以透过 /etc/hosts.allow, /etc/hosts.deny来配置防火墙, 它们也是 /usr/sbin/tcpd 的配置文件, 这两个文件的判断依据是allow为优先

基本上只要一个服务受到 xinetd 管理, 或者是该服务的程序支持 TCP Wrappers 函式的功能时, 那么该服务的防火墙方面的配置就能够以 /etc/hosts. {allow, deny} 来处

## ?

```

1ldd $(which sshd httpd)
2#有无支持tcp wrappers需要看是否依赖libwrap.so
3#sshd 有支持, 但是 httpd 则没有支持

```

例: 只允许 140.116.0.0/255.255.0.0 与 203.71.39.0/255.255.255.0 这两个网域, 及

203.71.38.123 这个主机可以进入我们的 rsync 服务器；此外，其他的 IP 全部都挡掉！

?

```
1vim /etc/hosts.allow
2rsync: 140.116.0.0/255.255.0.0
3rsync: 203.71.39.0/255.255.255.0
4rsync: 203.71.38.123
5rsync: LOCAL
6
7vim /etc/hosts.deny
8rsync: ALL #利用 ALL 配置让所有其他来源不可登陆
```

## 10 服务配置

### 10.1 daemon守护进程

相关目录

- /etc/init.d/\*启动脚本放置处
- /etc/sysconfig/\*各服务的初始化环境配置文件
- /etc/xinetd.conf, /etc/xinetd.d/\*super daemon 配置文件
- /etc/\*各服务各自的配置文件
- /var/lib/\*各服务产生的数据库
- /var/run/\*各服务的程序之 PID 记录处

### 10.2 查看所有服务

?

```
1#找出目前系统开启的『网络服务』有哪些
2netstat -tulp
3
4#找出所有的有监听网络的服务（包含 socket 状态）
5netstat -lnp
6
7#观察所有的服务状态
8service --status-all
9service --status-all | grep httpd
```

### 10.3 开机启动服务

chkconfig: 管理系统服务默认启动启动与否

?

```
1#列出目前系统上面所有被 chkconfig 管理的服务
2chkconfig --list |more
3
4#显示出目前在 run level 3 为启动的服务
5chkconfig --list | grep '3:on'
6
7#让 atd 这个服务在 run level 为 3, 4, 5 时启动:
8chkconfig --level 345 atd on
```

### 10.4 SSH

代替了telnet，后者通信时使用明文，不安全，另外也不能使用root用户登录

### 10.4.1 安装

?

```
lyum install -y *openssh*
```

### 10.4.2 启动

?

```
1 service sshd start          #redhat专有
2 /etc/init.d/sshd start      #linux通用，并可借助tab得到智能感知
```

### 10.4.3 登录

?

```
1 ssh <username>@<ip>          #如果不写<username>，默认是root用户
2
3 #设置rsa以后连接时，不需要再输入密码
4 ssh-keygen                    #产生密钥对
5 ssh-copy-id -i <sshhostip>    #输入相应登录密码后，将放置对方 /.ssh/authorized_keys中
    出现ssh连接"Host key verification failed ... The authenticity of host
    can't be established"
```

?

```
1 vim /etc/ssh/ssh_config
2 StrictHostKeyChecking no
3 UserKnownHostsFile /dev/null
    有关加密
```

- 对称加密（机密性）：des 3des aes rc4/5；算法简单，适合大量加密
- 单向加密（完整性）：md5 sha1 sha2；算法简单，不占用计算资源。又称哈希加密，具有不可逆，定长输出，雪崩效应，对应的shell命令有md5sum shasum sha224sum等
- 非对称加密（身份认证，密钥传输）：rsa dsa；不适合大量加密，有公钥和私钥，公钥是私钥中的一部分
  - 身份认证：自己用私钥加密，发送给他人，他人用公钥解密，可保证发送人的身份有效，又称数字签名
  - 密钥传输：他人用公钥加密 对称加密用的密钥，传输给具有私钥的人，保证密钥不被窃取
    - 还可以使用密钥交换，又称IKE技术（dh算法）

### 10.4.4 远程拷贝

?

```
lscp <file> <self_ip>:/path
```

## 10.5 TFTP

arm板开发时会使用，平时意义不大

### 10.5.1 安装

[?](#)

```
lyum install *tftp* -y
```

## 10.5.2 配置

[?](#)

```
lvim /etc/xinetd.d/tftp
```

- 其中server\_args = -s /var/lib/tftpboot表示下载的目录，-s表示能下载，-c表示能上传
- 其中disable = yes改成no表示开启服务

为了防止权限问题，可执行

[?](#)

```
lchmod 777 /var/lib/tftpboot -R
```

## 10.5.3 启动

[?](#)

```
lchkconfig tftp on          #未修改配置的启动方式
2service xinetd restart    #修改过配置文件后启动方式
```

如果怕安全机制影响，可执行

[?](#)

```
lservice iptables stop    #关闭防火墙
2setenforce 0              #关闭SELinux
```

## 10.5.4 客户端连接

[?](#)

```
ltftp <ip>                #进入命令提示符
2 > get <file>             #指定名字，默认下载到当前目录
3 > put <file>             #默认上传当前目录中文件
```

## 10.6 VSFTPD

### 10.6.1 安装

[?](#)

```
lyum install -y lftp      #客户端
2yum install -y vsftpd    #服务器端
```

### 10.6.2 配置

修改文件夹权限

[?](#)

```
lchmod 777 -R /var/ftp
```

修改配置文件

[?](#)

```
lvim /etc/vsftpd/vsftpd.conf
```

找到anon\_upload\_enable=yes和anon\_mkdir\_write\_enable=yes，前者表示可上传文件，后者表示可上传目录，需要时，可以去掉注释

### 10.6.3 启动

[?](#)

## 10.6.4 vsftpd 客户端连接

[?](#)

```
1lftp <ip> #进入命令提示符
2> ls #可浏览
3> get <file> -o /path/to/file #下载到指定目录
4> put /path/to/file #上传指定目录中文件
5> mirror <dir> /path/to/dir #下载目录
6> mirror -R /path/to/dir #上传目录
```

## 10.7 NFS

全称为network file system

ext3(linux) fat32(windows) ntfs(windows) 均是本地文件系统；cifs (samba协议) 共享文件系统，用于windows与linux共享文件；nfs是linux间使用的共享文件系统

### 10.7.1 安装

[?](#)

```
1yum install -y nfs-utils
2yum install -y rpcbind
```

### 10.7.2 配置

[?](#)

```
1vim /etc/exports
可以文件中指定共享的描述，如
```

/tmp \*(rw, async, no\_root\_squash)，分别代表对于该目录，可读写，异步传输，取消root的权限压制

也可以描述为/tmp <ip>[</网段>]表示只对某ip网段才能访问

### 10.7.3 启动

[?](#)

```
1service nfs restart
2service rpcbind restart
```

### 10.7.4 客户端挂载

客户端也需要启动上述的服务，然后使用mount进行挂载

[?](#)

```
1mount -t nfs <ip>:/tmp /opt
```

## 10.8 DHCP

### 10.8.1 安装

[?](#)

```
1yum install -y dhcp
```

### 10.8.2 配置



[?](#)

```

1vim /etc/dhcp/dhcpd.conf
2
3subnet 172.24.40.0 netmask 255.255.255.0{
4    range 172.24.40.100 172.24.40.200;
5    option routers 172.24.40.254;          #gateway
6    option domain-name "uplooking.com";
7    option domain-name-servers 202.106.0.46; #dns
8}

```

### 10.8.3 启动

[?](#)

```
1/etc/init.d/dhcpd start
```

### 10.8.4 客户端续约

[?](#)

```
1dhclient eth0
```

## 10.9 DNS

### 10.9.1 概述

FQDN是完全限定域名，准备的描述出主机所在的位置。DNS是倒置的树状结构的 分布式 数据库系统，存储着FQDN名和ip地址的对应关系，负责它们之间的解析

一个局域网内最好只有一个DNS

客户端存放DNS主机IP的配置在/etc/resolv.conf

### 10.9.2 安装

[?](#)

```
1yum install -y bind
```

### 10.9.3 配置

[?](#)

```

1vim /etc/named.conf      #负责domain <-> zonefile
2vim /var/named/xxx.zone  #负责subdomain <-> ip

```

### 10.9.4 启动

[?](#)

```
1/etc/init.d/bind start
```

## 10.10 SAMBA

samba的用户名必须与Linux系统的用户名一致，但密码可以不同，即必须先有linux的用户，然后使用smbpasswd -a <username>增加同名用户

### 10.10.1 配置

/etc/samba/smb.conf

[?](#)

```
1 [global]
2  workgroup = WORKGROUP
3  display charset = UTF-8
4  unix charset = UTF-8
5  dos charset = cp936
6 [Public]
7  path = /home/zyx/Public
8  available = yes
9  browseable = yes
10 public = yes
11 writable = yes
12 valid users = zyx
13 create mask = 0664
14 directory mask = 0775
15 force user = zyx
16 force group = zyx
```

## 11 Linux常用简介

### 11.1 Linux文件管理命令

- ls 显示文件名
- cat 显示文本文件内容
- rm 删除文件
- less 分屏显示文件
- cp 复制文件
- mv 更改文件名
- grep 查找字符串
- head 显示文件头部
- tail 显示文件尾部
- sort 按顺序显示文件内容
- uniq 忽略文件中的重复行
- diff 比较两个文件
- diffstat diff结果的统计信息
- file 测试文件内容
- echo 显示文本
- date 显示日期和时间
- script 记录Linux会话信息
- apropos 搜索关键字
- locate 搜索文件
- rmdir 删除目录
- chattr 改变文件的属性
- cksum 文件的CRC校验
- cmp 比较文件差异
- split 分割文件
- dirname 显示文件除名字外的路径
- find 查找目录或者文件
- findfs 通过列表或用户ID查找文件系统
- ln 链接文件或目录
- lndir 链接目录内容
- lsattr 显示文件属性
- od 输出文件内容
- paste 合并文件的列
- stat 显示inode内容

- tee 读取标准输入到标准输出并可保存为文件
- tmpwatch 删除临时文件
- touch 更新文件目录时间
- tree 以树状图显示目录内容
- umask 指定在建立文件时预设的权限掩码
- chmod 设置文件或目录的访问权限
- chgrp 改变文件或目录所属的群组
- chown 改变文件的拥有者或群组
- more 查看文件的内容
- md5sum: MD5函数值计算和检查
- awk 模式匹配语言
- wc 输出文件中的行数、单词数、字节数
- comm 比较排序文件
- join 将两个文件中指定栏位内容相同的行连接起来
- fmt 编排文本文件
- tr 转换字符
- col 过滤控制字符
- colrm 删除指定的行
- fold 限制文件列宽
- iconv 转换给定文件的编码
- dc 任意精度的计算器
- expr 求表达式变量的值
- strings 显示文件中的可打印字符
- xargs 从标准输入读入参数
- sum 计算文件的校验和，以及文件占用的块数
- setfacl 设定文件访问控制列表
- getfacl 获取文件访问控制列表
- chacl 更改文件或目录的访问控制列表

## 11.2 Linux磁盘管理命令

- df 显示报告文件系统磁盘使用信息
- du 显示目录或者文件所占的磁盘空间
- dd 磁盘操作
- fdisk 磁盘分区
- mount 挂载文件系统
- umount 卸载文件系统
- mkfs 建立各种文件系统
- mkfs ext2: 建立一个Ext2/Ext3文件系统
- mkbootdisk 建立启动盘
- fsck 检查文件系统
- blockdev 从命令行调用区块设备控制程序
- hdparm 设置磁盘参数
- mkswap 建立交换分区
- swapon 使用交换空间
- swapoff 关闭系统交换分区
- sync 写入磁盘
- e2fsck: 设置卷标
- badblocks 检查磁盘
- quota 显示磁盘已使用的空间与限制
- quotacheck 检查磁盘的使用空间与限制
- quotaoff 关闭磁盘空间限制
- quotaon 开启磁盘空间限制

- quotastats 显示磁盘空间限制
- repquota 检查磁盘空间限制的状态
- mdadm RAID设置工具
- tune fs: 文件系统调整
- mkisofs 建立ISO 9660映象文件
- cfdisk 磁盘分区
- sfdisk 硬盘分区工具程序
- parted 磁盘分区工具
- mkinitrd 建立要载入ramdisk的映象文件
- ssm 命令行集中存储管理工具
- 使用xfs管理命令
- LVM命令列表

## 11.3 Linux进程管理命令

- accton 打开或关闭进程统计
- lastcomm 显示以前使用过的命令的信息
- sa 报告、清理并维护进程统计文件
- at 定时运行命令
- atq 显示目前使用at命令后待执行的命令队列
- atrm 删除at命令中待执行的命令队列
- batch 在系统负载水平允许的时候执行命令
- bg 后台运行命令
- fg 挂起程序
- jobs 显示后台程序
- kill 杀掉进程
- crontab 设置计时器
- ps 查看权限
- pstree 显示进程状态树
- top 显示进程
- nice 改变优先权等级
- renice 修改优先权等级
- sleep 暂停进程
- nohup 用户退出系统之后继续工作
- pgrep 查找匹配条件的进程
- fuser 用文件或者套接口表示进程
- chkconfig 设置系统的各种服务
- strace 跟踪一个进程的系统调用或信号产生的情况
- ltrace 跟踪进程调用库函数的情况
- vmstat 报告虚拟内存统计信息
- mpstat 监测CPU（包括多CPU）性能
- iostat 监测I/O性能
- sar 系统活动情况报告
- pidof 查找正在运行进程的进程ID（PID）
- ntsysv 设置系统服务

## 11.4 Linux网络管理命令

- arp 管理系统中的ARP高速缓存
- arpwatch 监听ARP记录
- arping 发送ARP请求到一个相邻主机
- arpd 收集免费arp信息的一个守护进程
- finger 查找并显示用户信息

- ifconfig 设置网络接口
- iwconfig 设置无线网卡
- iw 新一代无线网络配置工具
- hostname 显示主机名
- ifup 激活网络设备
- ifdown 禁用网络设备
- mii tool: 调整网卡模式
- route 设置路由表
- netstat 查看网络连接
- ping 检测主机的连通性
- minicom 设置调制解调器
- pppd 建立PPP连接
- pppstats 显示PPP连接状态
- chat 拨号命令
- traceroute 检查数据包所经过的路由
- tracepath 追踪连接到目标地址所经过的路由
- rcp 远程复制
- tcpdump 网络数据分析器
- ipcalc IP地址计算器
- netreport 监视网络状态
- ip 网络集成命令工具
- pppoe setup: 设置ADSL
- pppoe start: 激活ADSL连接
- pppoe stop: 断开ADSL连接
- pppoe status: 检测ADSL连接状态
- wget 下载文件
- ngrep 监控网络接口
- lsof 查看打开的文件
- ethtool 查询及设置网卡参数
- netconf 设置各项网络功能
- tc 显示和维护流量控制设置
- telnet 远程登录
- rlogin 远程登录命令
- rsh 远程登录的Shell
- usernetctl 让普通用户控制网络接口
- nmcli NetworkManager命令行网络接口配置工具
- nmtui 基于Curses的用户界面nmtui
- nc netcat, Linux下用于调试和检查网络的工具包
- linstat 显示Linux系统的网络状态
- ss 显示网络状态信息
- rexec 在指定的远程Linux系统主机上执行命令

## 11.5 Linux用户管理命令

- useradd 建立用户
- userdel 删除用户
- usermod 修改已有用户的信息
- passwd 设置密码
- chage 密码老化
- groupadd 添加组
- groupdel 删除组账户
- groupmod 修改组
- vipw 编辑/etc/passwd文件

- vigr 编辑/etc/group文件
- newgrp 转换组
- groups 显示组
- gpasswd 添加组
- whoami 显示当前用户名称
- who 显示登录用户
- id 显示用户信息
- su 切换身份
- pwck 检测账户
- grpck 检测用户组账号信息的完整性
- chsh 设置Shell
- chfn 修改用户信息
- ac 显示用户在线时间的统计信息
- grpconv 开启群组的投影密码
- grpunconv 关闭群组的投影密码
- lastlog 显示最近登录用户的用户名、登录端口和登录时间
- logname 显示当前用户的名称
- users 显示当前登录到系统的用户
- lastb 显示登录系统失败用户的相关信息

## 11.6 Linux的备份和压缩命令

- tar 备份文件
- dump 备份文件系统
- cpio copy in/out) : 建立、还原备份文件
- restore 还原备份下来的文件或整个文件系统（一个分区）
- bunzip : 解压缩.bz2文件
- bzip : 解压缩.bz2文件
- bzgrep 使用正则表达式搜索.bz2压缩包中的文件
- unzip 解压缩.zip文件
- bzip recover: 修复损坏的.bz2文件
- gzip 压缩文件
- compress 压缩、解压文件
- gzexe 压缩执行文件
- lha 压缩或解压缩文件
- unarj 解压缩文件
- zip 压缩文件
- zipinfo 显示压缩文件的信息

## 11.7 Linux系统管理命令

- apmd 高级电源管理
- apmsleep APM进入休眠状态
- apropos 查找使用手册的名字和相关描述
- arch 输出主机的体系结构
- alias 设置别名
- cd 切换目录
- clear 清空终端屏幕
- clock 系统RTC时间设置
- cal 显示日历
- chroot 改变根目录
- date 显示或设置系统时间
- dmesg 显示开机信息

- `dircolors` 设置ls命令在显示目录或文件时所用的色彩
- `depmod` 分析模块
- `echo` 显示文本行
- `exec` 执行完命令后交出控制权
- `exit` 退出Shell
- `eject` 弹出介质
- `enable` 启动或关闭Shell命令
- `fc` 修改或执行命令
- `fgconsole` 显示虚拟终端的数目
- `free` 显示内存信息
- `fwhois` 显示用户的信息
- `getty` 设置终端模式
- `gitps` 显示程序情况
- `logwatch` 可定制和可插入式的日志监视系统
- `logsave` 把一个命令的输出输出到一个指定的日志文件中
- `GRUB` 引导加载程序
- `halt` 关闭系统
- `history` 显示历史命令
- `hwclock` 显示与设定硬件时钟
- `init` 进程处理初始化
- `last` 显示登录用户信息
- `lilo` 引导管理器
- `login` 登录系统
- `local` 显示本地支持的语言系统信息
- `logout` 退出系统
- `logrotate` 处理Log文件
- `lsmod` 显示Linux内核的模块信息
- `man` 格式化和显示在线手册
- `manpath` 设置man手册的查询路径
- `modinfo` 显示内核信息
- `modprobe` 自动处理可载入模块
- `pmap` 显示程序的内存信息
- `procinfo` 显示系统状态
- `pwd` 显示工作目录
- `reboot` 重新启动系统
- `rlogin` 远程登录
- `rmmod` 删除模块
- `rpm` 软件包管理
- `shutdown` 令：系统关机命令
- `suspend` 暂停执行Shell
- `nproc` 打印当前进程可用的处理器数
- `tload` 显示系统负载
- `uname` 显示系统信息
- `authconfig` 配置系统的认证信息
- `declare` 显示或者设定Shell变量
- `export` 设置或者显示环境变量
- `hostid` 打印出当前主机的标识
- `insmod` 载入模块
- `rdate` 显示其他主机的日期与时间
- `runlevel` 显示执行等级
- `set` 设置Shell
- `setenv` 查询或显示环境变量
- `setserial` 设置或显示串口的相关信息
- `setup` 设置公用程序

- symlinks 维护符号链接的工具程序
- swatch 系统监控程序
- sync 将内存缓冲区内的数据写入磁盘
- startx 启动X Window
- sysctl 设置系统核心参数
- timeconfig 设置时区
- ulimit 控制Shell程序的资源
- unalias 删除别名
- unset 删除变量或函数
- up date: 软件包升级
- uptime 告知系统运行了多长时间
- mouseconfig 设置鼠标相关参数
- bind 显示或设置键盘按键及其相关的功能
- kbdconfig 设置键盘类型
- snapscreenshot 命令行截图
- mt 磁带机控制
- cdrecord CD刻录工具
- dvdrecord DVD刻录工具
- lspci 查看硬件插槽
- sane find-scanner: 扫描仪搜索
- scanimage 检测扫描仪型号
- mtools 命令集
- whereis 查找文件
- ytalk 与其他用户交谈
- apt 软件包在线管理
- yum 在线管理软件包
- vlock 锁定终端
- wait 等待程序返回状态
- watch 将结果输出到标准输出设备
- rsync 远程数据同步工具
- & 将任务放在后台执行
- screen 多重视窗管理程序
- lsb release: 显示LSB和特定版本的相关信息
- lscpu 查看 CPU 信息
- blkid 查看块设备
- journalctl 日志管理
- lsblk 列出块设备
- systemd 服务管理命令组
- GRUB2

## 11.8 Linux服务器管理命令

### 11.8.1 DNS服务器管理命令

- named 域名服务器管理命令
- rndc DNS服务器控制
- named checkconf: 检查DNS配置
- named checkzone: 检查区域文件的合法性
- dig 发送域名查询信息包到域名服务器
- nslookup 交互式查询名称服务器
- host 使用域名服务器查询主机名字
- dnssec keygen: DNSSEC密钥生成工具
- dnssec signkey: DNSSEC密钥集签名工具



- `dnssec makekeyset`: DNSSEC区域签名工具
- `dnssec signzone`: DNSSEC区域签名工具
- `dlint` Bind DNS服务器辅助工具
- `dnstop` Bind DNS服务器辅助工具

### 11.8.2 NFS服务器管理命令

- `nfsd` 启动停止NFS服务器
- `portmap` 将RPC程序号转换为因特网端口号
- `rpcinfo` 报告远程过程调用(RPC)服务器的状态
- `showmount` 显示远程已安装文件系统的所有客户机的列表
- `umount` 删除当前已挂载的远程文件系统
- `mount` 将已命名的文件系统连接到指定的挂载点
- `automount` 安装自动安装点
- `mountall` 挂载一组文件系统
- `exportfs` 重新分享/etc/exports变更的目录资源

### 11.8.3 Samba服务器管理命令

- `testparm` 检查smb.conf配置文件的内部正确性
- `smbd` samba daemon): Samba服务器程序
- `smbclient` 类似FTP操作方式的访问SMB/CIFS服务器资源的客户端
- `smbstatus` 报告当前Samba的连接状态
- `smbmount` 装载一个smbfs文件系统
- `smbpasswd` 设置用户的SMB密码

### 11.8.4 SSH服务器管理命令

- `sshd` OpenSSH守护进程
- `ssh keygen`: 生成、管理和转换认证密钥
- `ssh` SSH命令行登录工具
- `sftp` 安全互动FTP
- `scp` 将文件复制到远程主机或本地主机
- `squid` 代理服务器squid守护进程

### 11.8.5 DHCP服务器管理命令

- `dhcpd` DHCP服务器守护进程
- `dhclient` DHCPv6客户端守护进程
- `dhcp c:` DHCPv6客户端守护进程

### 11.8.6 E-mail服务器管理命令

- `sendmail` 为本地或网络交付传送邮件
- `mail` E-mail管理程序
- `mailq` 显示待寄邮件的清单
- `mailstats` 显示关于邮件流量的统计信息
- `mutt` 电子邮件管理程序

### 11.8.7 Linux防火墙管理工具iptables和firewalld

- `iptables` Linux防火墙管理工具

- ip tables: IPv6版本的iptables工具
- iptables save: iptables列表存储
- iptables restore: 装载由iptables-save保存的规则集
- firewall cmd: 下一代防火墙管理工具
- arptables 管理ARP包过滤的软件

### 11.8.8 SELinux管理命令

- setenforce 设置SELinux模式
- getenforce 查看SELinux模式
- setsebool 设置SELinux布尔值
- getsebool 查看SELinux布尔值
- togglesebool 翻转SELinux布尔值
- sestatus SELinux状态查看工具
- avcstat 显示AVC统计信息
- audit why: 转换审计消息
- audit allow: 生成策略允许规则
- load policy: 装载策略
- semanage 管理SELinux策略
- semodule 管理策略模块
- chcat 改变语境类别
- restorecon 恢复文件安全语境
- chcon 改变文件安全语境
- setfiles 设置文件安全语境
- seinfo 提取策略的规则数量统计信息
- sesearch 搜索policy.conf或二进制策略中特别的类型
- checkmodule 编译策略模块
- sealert SELinux信息诊断客户端工具
- selinuxenabled 查询系统的SELinux是否启用

### 11.8.9 虚拟化管理命令

- xen 拟化管理命令
- KVM/Qemu虚拟机管理命令

## 11.9 Linux打印管理命令

- cupsd 通用打印程序守护进程
- cupsaccept 指示打印系统接受发往指定目标打印机的打印任务
- lpadmin 配置LP打印服务
- lp 打印文件
- lpstat 显示行式打印机的状态信息
- lpr 排队打印作业
- lprm 从打印队列中删除任务
- lpc 控制打印机
- lpq 检查假脱机队列
- lpinfo 显示驱动和设备
- lpmove 将作业从一个队列移动到另一个队列
- cancel 取消已存在的打印任务
- cupsdisable 禁用指定的打印机或类
- cupsreject 指示打印系统拒绝发往指定目标打印机的打印任务
- cupsenable 启动指定的打印机

## 11.10 Linux库应用命令

- ldconfig 配置查找共享库
- nm 列举目标文件中的符号名
- ar 建立修改文件或从文件中抽取成员
- strip 去除目标文件中的无用信息
- objdump 展开目标文件、静态库和共享库中的信息
- ldd 显示共享库的依赖情况
- patch 令：修补文件
- ld 连接器

## 11.11 Linux开发应用命令

- as 标准GNU汇编程序
- nasm 汇编器
- gcc C/C++编译器
- make 维护和编译软件或软件包
- gdb GUN调试器
- gdbserver 远端GNU服务器
- autoconf 产生配置脚本
- autoheader 为configure产生模板头文件
- autoreconf 更新已经生成的配置文件
- autoscan 产生初步的configure.in文件
- autoupdate 更新configure.in文件
- automake 自动生成Makefile.in的工具
- aclocal 生成aclocal.m4文件
- configure 生成Makefile文件

## 12 Vim编辑器

### 12.1 模式切换

任何模式，多按几次ESC即进入命令模式：

- v进入可视模式
- Shift v进入可视行模式
- Ctrl v进入可视块模式
- : /进入提示符模式
- i o a等进入插入模式

### 12.2 命令模式

多按几次ESC即进入命令模式，以下操作均在命令模式下进行

#### 12.2.1 光标移动

?

- |       |             |
|-------|-------------|
| 1 h   | 光标向左移动一个字符  |
| 2 20h | 光标向左移动20个字符 |
| 3 l   | 光标向右移动一个字符  |

4  
 5 j 光标向下移动一行  
 6 20j 光标向下移动20行  
 7 k 光标向上移动一行  
 8  
 9 Ctrl + f 屏幕『向下』移动一页，相当于 [Page Down] 按键 #(常用)  
 10 Ctrl + b 屏幕『向上』移动一页，相当于 [Page Up] 按键 #(常用)  
 11  
 12 n[Space] 光标会向后面移动 n 个字符距离  
 13 n[Enter] 光标向下移动 n 行 #(常用)  
 14  
 15 G 移动到这个档案的最后一行 #(常用)  
 16 nG 移动到这个档案的第 n 行 #(常用)  
 17 1G 移动到这个档案的第一行，相当于 1G 啊! #(常用)

## 2.2 定位单词

?  
 1 w 到下一个单词的开头 #(常用)  
 2 e 到下一个单词的结尾 #(常用)  
 3 b 到前一个单词的开头 #(常用)  
 4 % 匹配括号移动，包括 (, {, [ 需要把光标先移到括号上 #(常用)  
 5 \* 和 # 匹配光标当前所在的单词，移动光标到下一个（或上一个）匹配单词 #(常用，可用查找配合使用)  
 6  
 7  
 8  
 9 0 到行头  
 10 ^ 到本行的第一个非blank字符  
 11 \$ 到行尾  
 12 fa 到下一个为a的字符处，你也可以fs到下一个为s的字符。  
 13 t, 到逗号前的第一个字符。逗号可以变成其它字符。  
 14 3fa 在当前行查找第三个出现的a。  
 15 F 和 T 和 f 和 t 一样，只不过是相反方向。

## 2.3 编辑操作

?  
 1 x 在一行字当中，x 为向后删除一个字符（相当于 [del] 按键） #(常用)  
 2 X 为向前删除一个字符（相当于 [backspace] ） #(常用)  
 3  
 4 nx n为数字，连续向后删除 n 个字符  
 5 nX n为数字，连续删除光标前面的 n 个字符  
 6  
 7 dd 删除光标所在的那一整行 #(常用)  
 8 n dd n为数字。删除光标所在行向下n行，例如 20dd 则是删除 20 行  
 9 d1G 删除光标所在行到第一行的所有数据(组合用法)  
 10 dG 删除光标所在行到最后一行的所有数据  
 11 d\$ 删除光标所在处，到该行的最后一个字符  
 12 d0 那个是数字的 0，删除光标所在处，到该行的最前面一个字符  
 13 :n1,n2 d 将 n1 行到 n2 行之间的内容删除 #(注意有冒号)  
 14  
 15 yy 复制光标所在的那一行 #(常用)  
 16 nyy n为数字。复制光标所在行向下n行，例如 20yy 则是复制 20 行  
 17 y1G 复制光标所在行到第一行的所有数据(组合用法)  
 18 yG 复制光标所在行到最后一行的所有数据

19y\$ 复制光标所在的那个字符到该行行尾的所有数据  
 20y0 复制光标所在的那个字符到该行行首的所有数据  
 21:n1,n2 d 将 n1 行到 n2 行之间的内容复制 # (注意有冒号)  
 22  
 23p 将复制的数据, 粘贴在光标的下一行 # (常用)  
 24P 将复制的数据, 粘贴到光标的上一行  
 25  
 26ddp 上下两行的位置交换 (组合用法)  
 27  
 28J 将光标所在行与下一行的数据结合成同一行 # (常用)  
 29  
 30u 撤销 # (常用)  
 31Ctrl + r 撤销的撤销 # (常用)

## 12.3 提示符模式

### 12.3.1 查找替换

?  
 /string 向光标之下寻找一个名称为string字符串 # (常用)  
 ?string 向光标之上寻找一个名称为string字符串 # (常用)  
 1  
 2 n 正向查找, 搜索出的string, 可以理解成next # (常用)  
 3 N 反向查找, 搜索出的string, 可以理解成Not next # (常用)  
 4  
 5 :g/str/s/str1/str2/g 第一个g表示对每一个包括s1的行都进行替换, 第二个g表示对每一行的  
 6 所有进行替换  
 7 包括str的行所有的str1都用str2替换  
 8  
 9 :n1,n2s/string1/string2/g n1是查找的开始行数, n2是查找结束的行数, string1是要查找的字符  
 10 串, string2是替换的字符串 # (常用)  
 11  
 12 :2,7s/ddd/fff/g 在第2行, 第7行之间, 将ddd替换成fff  
 13 :1,\$s/string1/string2/gc 从第一行到最后一行寻找并替换, 在替换前给用户确认 (confirm)  
 14  
 15 :%s/f \$/for/g 将每一行尾部的“f” (f键和空格键) 替换为for, 命令之前的“%”指  
 16 定该命令将作用于所有行上. 不指定一个范围的话, “:s”将只作用于当前行.  
 17

### 12.3.2 文件操作

:%s/^[^#]/#& 将不是#开头的加上#

?  
 1vi +n FileName 打开文件 FileName, 并将光标置于第 n 行首。  
 2vi + FileName 打开文件 FileName, 并将光标置于最后一行  
 3vi -r FileName 在上次正用 vi 编辑 FileName 发生系统崩溃后, 恢复FileName。  
 4  
 5:%!xxd 按十六进制查看当前文件  
 6:%!xxd -r 从十六进制返回正常模式

### 12.3.3 保存退出

?  
 1 :w 将编辑的数据写入硬盘档案中 # (常用)  
 2 :w! 文件为『只读』时, 强制写入, 到底能否写入跟用户对档案的权限有关  
 3  
 4 :q 离开 vi # (常用)

5 :q! 若曾修改过档案，又不想储存，使用 ! 为强制离开不储存档案。  
 6  
 7 :wq 储存后离开 #(常用)  
 8 :wq! 强制储存后离开  
 9  
 10:ZZ 若档案没有更动，则不储存离开，若档案已经被更动过，则储存后离开!  
 11  
 12:e! 重新编辑当前文件，忽略所有的修改  
 13  
 14:w [filename] 另存为  
 15:r [filename] 在编辑的数据中，读入另一个档案的数据。亦即将这个档案内容加到光标所在行后面  
 12.3.4 运行shell

?

:! command 暂时离开 vi 到指令列模式下执行 command 的显示结果! 例如 [ :! cat  
 1./test] 即可在 vi 当中察看当前文件夹中的test文件中的内容  
 2  
 3:n1,n2 w! Command 将文件中n1行到n2行的内容作为 Command的输入并执行之，若不指定 n1、n2，则将  
 4整个文件内容作为 Command 的输入。  
 5  
 12.3.5 设置环境 将命令 Command 的输出结果放到当前行。

?

1:set nu 显示行号  
 2:set nonu  
 3:syntax on  
 4:set tabstop=4  
 也可在当前用户home目录下，新建.vimrc文件，将配置写入其中（无须set前面的冒号）

## 12.4 插入模式

在命令模式下，键入以下将会进入插入模式

?

1i 在光标前插入 #(常用)  
 2I 在光标行首字符前插入 #(常用)  
 3a 在光标后插入  
 4A 在光标行尾字符后插入 #(常用)  
 5o 在当前行后插入一个新行 #(常用)  
 6O 在当前行前插入一个新行  
 7r 替换当前字符, 接着键字的字符作为替换字符  
 8R 替换当前字符及其后的字符，直至按 ESC 键

## 12.5 代码相关

- 格式化，可视模式，选择后，按=
- 代码自动补全，编辑模式，ctrl p, ctrl n|p上下选择，ctrl y确认选择
- 查找变量或函数的声明，命令模式，gd
- 代码折叠，提示符模式:set fdm=indent|syntax，以下是命令模式下
  - zc 折叠
  - zC 对所在范围内所有嵌套的折叠点进行折叠
  - zo 展开折叠
  - zO 对所在范围内所有嵌套的折叠点展开

- 调试语法错误
  - 做好Makefile
  - 在项目当前目录进入vim
  - :cope
  - :set mouse=a
  - :make
- 导航代码，在项目当前目录执行ctags -R .
  - 方法1 vim -t somefun
  - 方法2 vim中在光标置于方法处ctrl ]进入方法实现，ctrl o返回
    - :ts命令就能列出一个列表供用户选择
    - :tp为上一个tag标记文件
    - :tn为下一个tag标记文件。
    - 若当前tags文件中用户所查找的变量或函数名只有一个，:tp,:tn命令不可用
- 导航系统头文件，#include <stdio.h>光标停在stdio.h位置，按gf进入头文件，ctrl o返回

## 13 Sublime快捷键

### 13.1 编辑器

[?](#)

|                  |            |
|------------------|------------|
| 1ctrl k b        | #切换显示左侧文件夹 |
| 2ctrl p          | #搜索项目中的文件  |
| 3ctrl shift p    | #控制面板      |
| 4alt shift <num> | #分屏显示      |
| 5f11             | #全屏        |

### 13.2 查找

[?](#)

|               |           |
|---------------|-----------|
| 1ctrl f       | #当前文件查找   |
| 2ctrl shift f | #项目或文件夹查找 |
| 3ctrl h       | #替换       |

### 13.3 编辑

[?](#)

|                        |                |
|------------------------|----------------|
| 1 ctrl shift <up down> | #选中行上下移动       |
| 2 ctrl ]               | #当前行缩进         |
| 3 ctrl [               | #去除当前行缩进       |
| 4 shift tab            | #去除缩进          |
| 5                      |                |
| 6 ctrl enter           | #在当前行后 插入一行    |
| 7 ctrl shift enter     | #在当前行前 插入一行    |
| 8 ctrl j               | #合并选择的多行       |
| 9 ctrl t               | #词互换(首先选择好两个词) |
| 10                     |                |
| 11ctrl k k             | #从光标处删除至行尾     |
| 12ctrl k backspace     | #从光标处删除至行首     |
| 13                     |                |

14ctrl k u #大写  
 15ctrl k l #小写  
 16  
 17f9 #行排序（按a-z）  
 18  
 19ctrl y #恢复撤销

## 13.4 选择复制

?  
 1 ctrl d #选择单词，重复可增加选择下一个相同的单词  
 2 #ctrl f调出查找，把'Aa'选中，则本快捷大小写敏感  
 3 ctrl k d #选择单词，过滤当前，移到下一个相同单词  
 4 ctrl shift d #复制上一行整行  
 5 ctrl x #剪切当前行 可当删除使用  
 6 alt f3 #选择所有相同的词  
 7 ctrl l #选择行，重复可依次增加选择下一行  
 8 #按住ctrl 左键选择代码，可选不连续代码  
 9 #鼠标中键框选代码，可选矩形区代码  
 10#直接在当前行任意位置 ctrl c，复制的是整行

## 13.5 代码相关

?  
 1 ctrl r #列出所有method  
 2 ctrl shift v #粘贴并自动格式化  
 3  
 4 ctrl / #注释/取消注释 选中的行  
 5 ctrl shift / #注释/取消注释 当前块  
 6  
 7 ctrl m #光标移动至括号内开始或结束的位置  
 8 ctrl shift m #选择当前括号内内容（继续按则选择父括号）  
 9 ctrl shift j #选择当前括号内内容，与上有差异（继续按则选择父括号）  
 10  
 11ctrl shift [ #当前括号 折叠代码  
 12ctrl shift ] #当前括号 展开代码  
 13ctrl k 0 #展开所有代码  
 14ctrl k 1 #代码折叠至一层  
 15ctrl k 2 #代码折叠至二层  
 16  
 17alt . #闭合当前标签  
 18alt f2 #增删标记 按f2可来回切  
 19  
 20ctrl b #编译  
 21f7 #运行

## 14 Xcode快捷键

### 14.1 编辑

?  
 1 cmd [ ] #左右缩进  
 2 cmd opt [ ] #当前行上下移动  
 3  
 4 ctrl f #前移光标



|    |                   |                       |
|----|-------------------|-----------------------|
| 5  | ctrl b            | #后移光标                 |
| 6  | ctrl p            | #光标到上一行               |
| 7  | ctrl n            | #光标到下一行               |
| 8  | ctrl a            | #光标到行首                |
| 9  | ctrl e            | #光标到行尾                |
| 10 | opt left right    | #光标前进 后退一个单词          |
| 11 | #以上加上shift就会选择    |                       |
| 12 |                   |                       |
| 13 | mouse three click | #选中当前行                |
| 14 |                   |                       |
| 15 | ctrl d            | #删除光标右侧字符 同 fn delete |
| 16 | ctrl k            | #删除光标处到行尾             |
| 17 | cmd delete        | #删除光标处到行首             |
| 18 |                   |                       |
| 19 | ctrl cmd space    | #标准表情                 |

## 14.2 代码相关

|                   |                    |                 |
|-------------------|--------------------|-----------------|
| <a href="#">?</a> |                    |                 |
| 1                 | esc                | #当前单词重新补全       |
| 2                 | cmd r              | #运行             |
| 3                 | cmd /              | #注释             |
| 4                 | cmd shift j        | #转到定义           |
| 5                 | cmd opt left right | #折叠             |
| 6                 |                    |                 |
| 7                 | ctrl i             | #当前格式化          |
| 8                 | cmd shift o        | #显示open quickly |
| 9                 | opt mouse dclick   | #文档             |

---

