

廊坊市速威网络技术有限公司培训教材

Linux 简明操作手册

Based on Ubuntu Server

Jie

2/2/2010

Contents

第 1 章 命令行操作及常用命令使用	6
1.1 命令行操作	6
1.1.1 Shell 简介	6
1.1.2 Shell 控制台使用方法	7
1.2 常用命令使用	7
1.2.1 登录和退出 Linux 系统	7
1.2.2 文件和目录操作命令	9
1.2.3 用户及用户组管理命令	11
1.2.4 进程及任务管理命令	14
1.2.5 磁盘及文件系统管理命令	16
1.2.6 软件安装命令	18
1.2.7 man 帮助命令	20
1.2.8 sudo 命令	21
1.2.9 wget 下载命令	21
第 2 章 文件查看/编辑方法介绍	23
2.1 cat 命令介绍	23
2.2 vi 命令介绍	23
2.2.1 vi 的进入与退出	24
2.2.2 vi 中显示行号	24
2.2.3 光标移动操作	24
2.2.4 文本插入操作	25
2.2.5 文件修改操作	26
2.2.6 文件搜索与替换操作	28
2.3 head 及 tail 命令介绍	29
第 3 章 网络配置及操作	30
3.1 网络配置命令	30
3.1.1 ifconfig	30

3.1.2 route	31
3.1.3 dhclient.....	33
3.2 网络操作命令	33
3.2.1 netstat 命令	33
3.2.2 nslookup 命令	34
3.2.3 host 命令	35
3.2.4 ping 命令	35
3.2.5 traceroute 命令	36
3.3 常用网络配置文件介绍	36
3.3.1 /etc/hosts IP 地址和主机名的映射	36
3.3.2 /etc/services	37
3.3.3 /etc/host.conf 配置名字解析器	37
3.3.4 /etc/nsswitch.conf 配置名字解析器	38
3.3.5 /etc/resolv.conf 配置 DNS 客户	38
3.4 常用网络操作方法	39
3.4.1 ftp 远程文件传输	39
3.4.2 telnet 远程登录	41
3.4.3 SSH 安全登录	42
第 4 章 常用命令技巧	44
4.1 Linux 下软件包的安装及卸载	44
4.1.1 apt-get 命令	44
4.1.2 apt-cache 命令	44
4.1.3 aptitude 命令	44
4.1.4 dpkg 命令	44
4.2 Linux 操作系统的时钟系统	45
4.2.1 时钟系统介绍	45
4.2.2 硬件时钟系统与系统时间的同步	45
4.3 系统信息及运行状态监控	46
4.3.1 df 命令	46
4.3.2 du 命令	46

4.3.3 top 命令	47
4.3.4 iostat 命令	47
4.3.5 dmesg 命令	47
4.4 归档/压缩工具 tar.....	48
4.5 文件及文件内容的查找	49
4.5.1 find 文件查找命令	49
4.5.2 grep 文件查找命令.....	49
4.5.3 xargs 命令	50
附录 1.....	52

第 1 章 命令行操作及常用命令使用

1.1 命令行操作

1.1.1 Shell 简介

Shell 是用户和 Linux 操作系统之间的接口。Linux 中有多种 shell，其中缺省使用的是 bash

Linux 系统的 shell 作为操作系统的外壳为用户提供使用操作系统的接口，它是一个命令语言解释器，拥有自己内建的 shell 命令集。Shell 也能被系统中其他应用程序所调用，用户在提示符下输入的命令都由 shell 先解释然后再传给 Linux 核心。

Linux 中的 shell 有多种类型，其中最常用的几种是 Bourne shell (sh)，C shell (csh) 和 Korn shell (ksh)，三种 shell 各有优缺点。

Bourne shell 是 UNIX 最初使用的 shell，并且在每种 UNIX 上都可以使用。Bourne shell 在 shell 编程方面相当优秀，但在处理与用户的交互方面做得不如其他几种 shell。Linux 操作系统缺省的 shell 是 Bourne Again Shell，它是 Bourne shell 的扩展，简称 bash。与 Bourne shell 完全向后兼容，并且在 Bourne shell 的基础上增加增强了很多特性。它还包含了很多 C shell 和 Korn shell 中的优点，有灵活和强大的编程接口同时又有友好的用户界面。

C shell 是一种比 Bourne shell 更适于编程的 shell，它的语法与 C 语言很相似。Linux 喜欢使用 C shell 的人提供了 Tcsh。Tcsh 是 C Shell 的一个扩展版本。Tcsh 包括命令行编程可编程单词补全、拼写校正、历史命令替换、作业控制和类似于 C 语言的语法。它不仅和 Bash shell 是提示符兼容，而且还提供比 Bash shell 更多的提示符参数。

Korn shell 集合了 C shell 和 Bourne shell 的优点并且和 Bourne shell 完全兼容 Linux 系统提供了 pdksh ksh 的扩展它支持任务控制可以在命令行上挂起后台执行唤醒或终止程序。

Ubuntu 系统默认的 shell 是 bash，对普通用户用 \$ 作提示符，对超级用户 root 用 # 作提示符。一旦出现了 shell 提示符就可以键入命令名称及命令所需要的参数。

我们来实习一下 shell 是如何工作的，在 shell 提示符下输入以下命令：

```
$ ls -al
```

该命令可以列出当前目录下的所有文件和子目录。

用户在登录到 Linux 时，由 /etc/passwd 文件来决定要使用哪个 shell。例如：

该文件中含有类似如下一行

```
myself:x:500:500:Linux User:/home/myself:/bin/bash
```

用户使用的 shell 被列每行的末尾/bin/bash。

1.1.2 Shell 控制台使用方法

首先介绍一个名词“控制台（console）”，它就是我们通常见到的使用字符操作界面的人机接口，我们说控制台命令就是指通过字符界面输入的可以操作系统的命令。我们现在要了解的是基于 Linux 操作系统的基本控制台命令。

Linux 是一个真正的多用户操作系统，这表示它可以同时接受多个用户登录。Linux 还允许一个用户进行多次登录，这是因为 Linux 和许多版本的 UNIX 一样提供了“虚拟控制台”的访问方式，允许用户在同一时间从控制台（系统的控制台是与系统直接相连的监视器和键盘）进行多次登录。

虚拟控制台的选择可以通过按下 Alt 键和一个功能键来实现，通常使用 F1-F6。例如用户登录后按一下 Alt-F2 键，用户又可以看到“login:”，提示符说明用户看到了第二个虚拟控制台；然后只需按 Alt-F1 键就可以回到第一个虚拟控制台。一个新安装的 Linux 系统允许用户使用 Alt-F1 到 Alt-F6 键来访问前六个虚拟控制台。

1.2 常用命令使用

1.2.1 登录和退出 Linux 系统

用户登录系统时，为了使系统能够识别自己，必须输入用户名和密码，经系统验证无误后方能进入系统。

说明

在系统安装过程中可以创建两种帐号：超级用户帐号和普通用户帐号。Linux 使用者一般为普通用户，而系统管理员一般使用超级用户帐号完成一些系统管理的工作。如果只需要完成一些由普通用户帐号就能完成的任务，建议不要使用超级用户帐号，以免无意中破坏系统。

用户登录分两步进行，第一步输入用户的登录名，系统根据该登录名来识别用户；第二步输入用户的口令，该口令是用户自己选择的一个字符串，对其他用户是保密的，是在登录时系统用来辨别真假用户的关键字。

1. 启动和登录系统

超级用户的用户名为 root，密码在安装系统时已设定，系统启动成功后屏幕显示下面的提示：

```
localhost login:
```

这时输入超级用户名"root"，然后键入回车键此时用户会在屏幕上看到输入口令的提示：

```
localhost login:root
```

```
Password:
```

这时需要输入口令。输入口令时，口令不会在屏幕上显示出来。如果用户输入了错误的口令，就会在屏幕上看到下列信息。

```
login incorrect.
```

这时需要重新输入。当用户正确地输入用户名和口令后，就能合法地进入系统。屏幕显示：

```
[root@localhost/root] #
```

此时说明用户已经登录到系统中，可以进行操作了，这里"#"是超级用户的系统提示符。普通用户在建立了普通用户帐号以后就可以进行登录了。

不论是超级用户还是普通用户需要退出系统时，在shell 提示符下键入exit 命令即可。还有其他退出系统的方法但上面一种是最安全的。

2. 重新启动系统

以 root 用户登录 Linux 操作系统后执行 reboot 命令可以重新启动 linux 系统：

```
[root@localhost/root] # reboot
```

3. 关闭系统

shutdown 命令可以安全地关闭或重启Linux 系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间,也可以是从现在开始的一个时间段.精确时间的格式是“hh:mm ”，表示小时和分钟，时间段由“+”和分钟数表示。系统执行该命令后会自动进行数据同步的工作。该命令的一般格式：

```
shutdown [选项] [时间] [警告信息]
```

命令中各选项的含义为：

- k 并不真正关机而只是发出警告信息给所有用户
- r 关机后立即重新启动

- h 关机后不重新启动
- f 快速关机重启时跳过fsck
- n 快速关机不经过init 程序
- c 取消一个已经运行的shutdown

需要特别说明的是该命令只能由超级用户使用。

例 1：系统在十分钟后关机并且马上重新启动。

```
# shutdown -r +10
```

例 2 系统马上关机并且不重新启动。

```
# shutdown -h now
```

1.2.2 文件和目录操作命令

1. ls 命令

这个命令就相当于 dos 下的dir 命令一样，也是Linux 控制台命令中最为重要几个命令之一。ls 最常用的参数有三个：-a、-l和-F。

```
ls -a
```

Linux 系统上的文件以.开头的文件被系统视为隐藏文件，仅用ls 命令是看不到他们的。而用ls -a，除了显示一般文件名外，连隐藏文件也会显示出来。

```
ls -l
```

（这个参数是字母 L 的小写不是数字1）

这个命令可以使用长格式显示文件内容。如果需要察看更详细的文件资料就要用到ls -l 这个指令。

```
ls -F
```

在列出的文件（目录）名称后加一符号，例如可执行文件加"*"，目录则加 "/"。

2. cd 命令

cd 这个命令是用来进出目录的，它的使用方法和在 dos 下没什么两样，但和 dos 不同的是 Linux 的目录对大小写是敏感的。如果大小写拼写有误，你的 cd 操作是成功不了的。另外 cd 如果直接输，cd 后面不加任何东西会回到使用者自己的 Home 目录。假设如果是 root 那就是回到/root，这个功能同 cd ~是一样的。

3. mkdir、rmdir 命令

mkdir 命令用来建立新的目录，如输入如下命令：

```
mkdir work
```

将在当前目录下新建一个 **work** 目录。

rmdir 用来删除已建立的目录，如输入如下命令：

```
rmdir work
```

将删除已存在的空目录 **work**。

4. **cp** 命令

cp 这个命令用于文件复制。具体用法是：

```
cp -r 源文件(source) 目的文件(target)
```

参数 **r** 是指连同源文件中的子目录一同拷贝。

5. **rm** 命令

rm 这个命令是用来删除文件的，**rm** 命令常用的参数有三个 **-i**, **-r**, **-f**

比如我现在要删除一个名字为 **test** 的一个文件，输入如下命令：

```
rm -i test
```

系统会询问我们是否要删除 **test** 文件，敲“**y/n**”确认是否要删除 **test** 文件；

rm -r 目录名：这个操作可以连同这个目录下面的子目录都删除，功能比上面讲到的 **rmdir** 更强大，不仅可能删除指定的目录，而且可以删除该目录下所有文件和子目录；

rm -f 文件名：这个操作可以不经确认强制删除文件

6. **mv** 命令

mv 这个命令的功能是移动目录或文件，引申的功能是给目录或文件重命名，比如

```
mv /tmp/xxx.tar /root
```

该命令将 **/tmp** 目录下的 **xxx.tar** 文件移动到 **/root** 目录下。

而：

```
mv aaa.tar bbb.tar
```

则是将当前目录下的文件 **aaa.tar** 更名为 **bbb.tar**。

7. **cat** 命令

`cat` 这个命令是 `linux` 中非常重要的一个命令，它的功能是显示或连结一般的 `ascii` 文本文件。它的用法如下

```
cat text
```

该命令显示 `text` 这个文件的内容。

```
cat file1 file2
```

该命令依顺序显示 `file1`, `file2` 的内容。

```
cat file1 file2>file3
```

该命令把 `file1`, `file2` 的内容结合起来再重定向到 `file3` 文件中。

8. `more` 命令

`more` 是显示一般文本文件的指令。如果一个文本文件太长了超过一个屏幕的画面，用 `cat` 来看实在是不理想，就可以试试 `more`，用法如下：

```
more file1
```

9. `pwd` 命令

`pwd` 这个命令的作用是显示用户当前的工作路径，直接输入 `pwd` 即可。

1.2.3 用户及用户组管理命令

1. `useradd`

`useradd` 命令可以创建一个新的用户帐号，其最基本用法为：

```
[root@localhost/root] # useradd 用户名
```

如输入以下命令：

```
[root@localhost/root] # useradd newuser
```

系统将创建一个新用户 `newuser`，该用户的 `Home` 目录为 `/home/newuser`。

`useradd` 命令的参数较多，常用的组合为：

```
useradd 用户名 -g 组名 -G 组名 -d Home 目录名 -p 密码
```

其中：

-g 指定该用户的首要组

-G 指定该用户的次要组

- d 指定该用户的Home 目录
- p 指定该用户的密码

如输入以下命令:

```
[root@localhost/root] # useradd oracle -g oinstall -G dba -d /home/oracle -p  
ora123
```

系统将创建一个用户 **oracle**, **oracle** 用户的首要组为**oinstall**, 次要组为**dba**, Home 目录为**/home/oracle**, 密码为**ora123**。

2. userdel

userdel 命令用于删除一个已存在的帐号, 其用法为:

```
[root@localhost/root] # userdel 用户名
```

3. groupadd

groupadd 命令可以创建一个新的用户组, 其最基本用法为:

```
[root@localhost/root] # groupadd 组名
```

如输入以下命令:

```
[root@localhost/root] # groupadd newgroup
```

系统将创建一个新的用户组 **newgroup**。

4. groupdel

groupdel 命令用于删除一个已存在的用户组, 其用法为:

```
[root@localhost/root] # groupdel 组名
```

5. password

出于系统安全考虑, **Linux** 系统中的每一个用户除了有其用户名外还有其对应的用户口令, 用户可以随时用**passwd** 命令改变自己的口令。该命令的一般格式为:

```
passwd
```

输入该命令后, 按系统提示依次输入密码和密码确认后, 即可完成用户密码的修改。

此外, 超级用户还可以修改其他用户的口令, 命令如下:

```
[root@localhost/root] # passwd 用户名
```

6. su

su 命令这个命令非常重要，它可以让一个普通用户拥有超级用户或其他用户的权限，也可以让超级用户以普通用户的身份做一些事情。普通用户使用这个命令时必须要有超级用户或其他用户的口令。如要离开当前用户的身份，可以键入 **exit** 命令。**su** 命令的一般形式为：

```
su - 用户名
```

7. chmod

chmod 命令是非常重要的，用于改变文件或目录的访问权限。该命令有两种用法：一种是包含字母和操作符表达式的文字设定法，另一种是包含数字的数字设定法。

由于数字设定法不太直观，本文不做介绍。文字设定法的用法如下：

```
chmod [who] [+ | - | =] [mode] 文件名
```

命令中各选项的含义为：

a) 操作对象**who** 可以是下述字母中的任一个或者它们的组合：

- u** 表示用户(**user**)，即文件或目录的所有者
- g** 表示同组(**group**)用户，即与文件属主有相同组ID的所有用户
- o** 表示其他(**others**)用户
- a** 表示所有(**all**)用户，它是系统默认值。

b) 操作符号可以是：

- +** 添加某个权限
- 取消某个权限
- =** 赋予给定权限,并取消其他所有权限

c) **mode** 表示权限常用的参数有：

- r** 可读
- w** 可写
- x** 可执行

举例：

1) 将文件 **script** 的权限设为可执行。命令如下：

```
chmod =rx text
```

执行成功后，用 `ls -l script` 命令查看文件属性的结果如下：

```
-r-xr-xr-x 1 user group 0 Feb 10 09:42 script
```

- 2) 将文件 `text` 的权限设为：文件属主可读、可写、可执行，与文件属主同组的用户可读，其他用户不可读。命令如下：

```
chmod u=rwx,g=r,o= text （注意，后无空格，o=后有空格）
```

执行成功后，用 `ls -l text` 命令查看文件属性的结果如下：

```
-rwxr----- 1 user group 0 Feb 10 09:42 text
```

8. chown

`chown` 用于更改某个文件或目录的属主和属组，这个命令也很常用。例如 `root` 用户把自己的一个文件拷贝给用户 `oracle`，为了让用户 `oracle` 能够存取这个文件，`root` 用户应该把这个文件的属主设为 `oracle`，否则用户 `oracle` 无法存取这个文件。`chown` 的基本用法为：

```
chown [用户:组] 文件
```

举例：

```
chown oracle:dba text
```

该命令将 `text` 文件的属主和属组分别改为 `oracle` 和 `dba`。

1.2.4 进程及任务管理命令

Linux 系统上所有运行的东西都可以称之为一个进程。每个用户任务、每个系统管理守护进程都可以称之为进程。Linux 用分时管理方法使所有的任务共同分享系统资源。以下将介绍一些常用的查看和控制进程的命令。

1. ps

`ps` 命令是最基本同时也是非常强大的进程查看命令。使用该命令可以查看有哪些进程正在运行以及运行的状态，进程是否结束，进程有没有僵死，哪些进程占用了过多的资源等等。

该命令的基本用法如下：

ps [选项]

其中常用的选项有：

- e 显示所有进程
- f 全格式
- l 长格式

举例：

- 1) 在控制台输入 **ps** 命令，得到类似如下的输出：
- 2) 在控制台输入以下命令：

```
ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 Jan14 ? 00:00:05 init
root 2 1 0 Jan14 ? 00:00:00 [keventd]
root 3 1 0 Jan14 ? 00:00:00 [kapmd]
.....
```

该命令将显示所有执行进程的信息。显示的项目依次为UID（执行进程的用户ID）、PID(进程ID)、PPID(父进程ID)、TTY(终端名称)、STIME(进程启动时间)、TIME(进程执行时间)、CMD(该进程的命令行输入)。

该命令的输出较多，可以使用**grep** 命令进行过滤，以方便查看，如：

```
ps -ef|grep oracle
```

将只显示出包含 **oracle** 的输出行，可以查询执行进程的用户为**oracle** 的所有进程。

2. top

top 命令和**ps** 命令的基本作用是相同的，显示系统当前的进程和其他状况。但是**top** 是一个动态显示过程，即可以通过用户按交互来不断刷新当前状态。**top** 的命令参数并不重要，简单的输入**top** 命令即可查看系统运行状态。

top 命令执行过程中可以使用交互命令。从使用角度来看，熟练的掌握这些命令比掌握选项还重要一些这些。命令都是单字母的，常用的命令有：

- <空格> 立即刷新显示
- h 或者? 显示帮助画面给出一些简短的命令总结说明
- m 切换显示内存信息
- t 切换显示进程和CPU状态信息
- c 切换显示命令名称和完整命令行
- M 根据驻留内存大小进行排序

p 根据CPU使用百分比大小进行排序

q 退出

3. kill

kill 命令可以用于终止后台进程。kill 命令是通过向进程发送指定的信号来结束进程的。kill 命令的语法格式很简单，大致有以下两种方式：

1) kill [-s 信号] 进程号

其中，发出的信号既可以是信号名也可以对应数字。

如，kill -9 1234 命令，将终止PID(进程号)为1234 的进程。

2) kill -l

该命令显示信号列表

4. cron

cron 命令用来实现定时任务的完成,如每日执行一次的任务（进程）。cron 命令是不应该手工启动的。一般情况下，cron 命令在系统启动时就由一个shell 脚本自动启动。启动后，cron 命令会搜索/var/spool/cron 目录，寻找以/etc/passwd 文件中的用户名命名的crontab 文件，被找到的这种文件将载入内存。例如一个用户名为user 的用户它所对应的crontab 文件就应该是/var/spool/cron/user。也就是说，以该用户命名的crontab 文件存放在/var/spool/cron 目录下面。cron 命令还将搜索/etc/crontab 文件，这个文件是用不同的格式写成的。cron 启动以后它将首先检查是否有用户设置了crontab文件，如果没有就转入休眠状态，释放系统资源。它每分钟醒过来一次查看当前是否有需要运行的命令。

可以使用crontab -l 命令查看目前已经存在的cron 任务

1.2.5 磁盘及文件系统管理命令

管理文件系统其实是相当复杂的工作，本文将只介绍日常维护需要的一些管理命令，主要内容有：挂装和卸载硬盘分区等。

1. df

df 命令可以显示目前磁盘剩余的磁盘空间，常用的参数为-k，如：

```
df -k
```

该语句将显示各分区的磁盘空间使用情况。

2. mount和umout

mount 命令有很多参数，其中大多数都不会在日常工作中用到。mount 命令最常见的用法如下所示：


```
mount [选项] [设备文件名 挂载点]
```

其中，最常用的选项是：

- a 把/etc/fstab 文件中列出的文件系统都挂装上。

挂装软驱和光驱的命令比较简单，直接输入以下命令即可：

```
mount /media/cdrom
mount /media/floppy
```

umount 命令用于卸载一个文件系统，命令的格式如下：

```
umount [ -f ] directory
```

其中，**directory** 是准备卸载的目录名。

umount 命令有一个不足之处：如果文件系统正在使用中（也就是说有人在那个分区上打开了文件），就无法把这个文件系统卸载下来。这时可以使用和**-f**参数强制执行卸载操作，当然这种操作可能会造成用户数据的丢失。如：

```
umount -f /mnt/cdrom
```

该命令将强制卸载挂装在/mnt/cdrom 目录下的文件系统。

mount与/etc/fstab

/etc目录下的fstab文件是用于配置Linux系统下的磁盘挂载。里面描述了在系统启动时应该如何挂载整个文件系统。如挂载软驱和光驱的信息在fstab文件里是这样描述的：

/dev/scd0	/media/cdrom0	udf,iso9660	user,noauto,exec,utf8	0	0
/dev/fd0	/media/floppy0	auto	rw,user,noauto,exec,utf8	0	0

第一列是存储设备的设备名，第二列是要挂载的目录，第四列是关于在挂载时所需要的一些参数，如上面这两行配置中，都带有noauto这个参数，表示在系统启动时，Linux系统并不直接挂载光驱和软驱，而需要用户执行mount命令时才能访问光驱和软驱。

在挂载fstab文件已存在的存储设备时，可以采用简化的命令方式，如：

```
mount /media/cdrom0
```

在执行上在这个命令时，mount程序首先会去访问fstab这个文件，进一步获取这个光驱的详细信息，如光盘上的文件系统格式。

但是，fstab配置文件并不是包含所有的存储设备的信息，如U盘。在挂载U盘时就需要指定U盘上的文件系统格式的详细信息。

U盘在Ubuntu Linux下被识别为SCSI设备，那么U盘的设备名就会以sd开头。如sd0，

sd1。

```
jie@ubuntu:~$ ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5 /dev/sdb /dev/sdb1
```

在这个系统中，sda表示的是串口硬盘，而sdb就是U盘的设备名，/dev/sdb1是U盘上的第一个分区。那么挂载这个U盘时，可采用以下的命令

```
mount -t vfat /dev/sdb1 /mnt
```

表示把U盘挂载在根目录下的mnt目录，用户可以通过访问mnt目录在读写U盘上的文件。

1.2.6 软件安装命令

1. tar

tar 命令用于把多个文件合并于一个档案文件中，并提供分解的合并后的档案文件的功能。它独立于压缩工具。因此可以选择在合并前是否压缩。tar 命令的基本用法为：

```
tar [选项] 文件名
```

常用的选项包括：

- c 创建一个新的档案文件
- t 查看档案文件的内容
- x 分解档案文件的内容
- f 指定档案文件的名称
- v 显示过程信息
- z 采用压缩方式

举例：

```
1) tar -czf oracle.tar.gz /u01/app/oracle /etc/oratab /home/oracle
```

该命令将/u01/app/oracle 目录下的所有文件、/etc/oratab 文件和/home/oracle 目录下的所有文件合并到 oracle.tar 文件中、并采用压缩方式。

```
2) tar -tzf oracle.tar.gz
```

该命令列出归档文件 oracle.tar 中的所有文件列表。

```
3) tar -xzvf oracle.tar.gz
```

该命令将归档文件分解并解压缩到原有文件的路径中。

2. apt-get 命令

`apt-get` 命令用于系统中软件包的管理，Linux 下存在很多可用软件，但是很多软件都是针对于特定行业的，其它行业的用户并不需要，如果在安装时，把所有的软件都安装上，势必会占用很多的硬盘空间，而且如果把所有的软件做成光盘需要好几张，所以在 Linux 下采用了软件包管理的方式，让用户自由选择安装什么样的软件。

常用的软件包管理工具有 `apt-get`、`yast`、`rpm` 等，每款 Linux 发行版本采用的软件包管理工具都不太一样，如 RedHat 使用 RPM 软件包来管理软件，Debian 使用 Deb 软件包来管理软件，`apt-get` 是 Debian 的 Deb 软件包管理工具，它的最低底层还是调用 `dpkg` 包管理程序。Ubuntu Linux 发行版派生于 Debian，所以它也采用了 `apt-get` 的软件包管理工具。

```
apt-get update    在修改了/etc/apt/source.list 后，需要运行本命令
apt-get install package_name  安装一个新的软件包
apt-get remove package_name  卸载一个已安装的软件（保留配置文件）
apt-get --purge remove package_name  卸载一个已经安装的软件包（同时删除配置文件）
```

如何肯定安装哪个软件包，快速的方式可以采用如下方式

Ubuntu 系统在默认安装上并没有安装 `unzip` 这个程序，在命令行下输入 `unzip` 时，提示：

```
The program 'unzip' is currently not installed.  You can install it by typing:
```

```
sudo apt-get install unzip
```

```
unzip: command not found
```

提示信息里已经提示了 `sudo apt-get install unzip`，那么运行这个命令后，系统会自动安装 `unzip` 这个包。

有的包名与程序并不一样，如并没有 `ld` 这个软件包，`ld` 程序而是位于 `binutils` 软件包里，在命令行下试着执行 `ld` 时，系统会提示

```
The program 'ld' can be found in the following packages:
```

```
* binutils
```

```
* hardening-wrapper
```

```
* binutils-gold
```

```
Try: sudo apt-get install <selected package>
```

提示信息里指出了有三个软件包里面包含 `as` 这个程序，那么通过列出的软件包名大部分就可以得知需要安装哪个软件包，如要安装 `ld` 这个程序时，可采用

`sudo apt-get install bintuils` 即可

`/etc/apt/source.list` 配置文件

这个文件里面包含中在安装时，软件包下载服务器的地址，在安装时 **Ubuntu** 时，文件里面默认包含的是位于美国的服务器，为了在安装软件包时快一些，可以修改这个配置文件，修改成离自己最近的（位于中国的）服务器地址。如

```
deb http://mirrors.163.com/ubuntu/ karmic main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ karmic-security main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ karmic-updates main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ karmic-proposed main restricted universe multiverse
deb http://mirrors.163.com/ubuntu/ karmic-backports main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ karmic main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ karmic-security main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ karmic-updates main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ karmic-proposed main restricted universe multiverse
deb-src http://mirrors.163.com/ubuntu/ karmic-backports main restricted universe multiverse
```

1.2.7 man 帮助命令

`man` 就是 `manual` 的缩写，用来查看系统中自带的各种参考手册，但是手册分为好几个部分，如下所示：

- 1 Executable programs or shell commands
- 2 System calls (functions provided by the kernel)
- 3 Library calls (functions within program libraries)
- 4 Special files (usually found in `/dev`)
- 5 File formats and conventions eg `/etc/passwd`
- 6 Games
- 7 Miscellaneous (including macro packages and conventions), e.g. `man(7)`, `groff(7)`
- 8 System administration commands (usually only for root)
- 9 Kernel routines [Non standard]

最常用的就是 **section 1**, `man` 命令默认的也是在 **section 1** 里查找相应的帮助信息，**2、3** 在做 C 语言编程时可以用来查找相应库函数的使用方法及格式。

一般用法：
`man` 命令名

举例：

<code>man ls</code>	#列出 ls 命令的帮助信息
<code>man printf</code>	#列出 printf 命令的帮助信息
<code>man 3 printf</code>	#列出 C 语言函数 printf 的详细帮助信息

1.2.8 sudo 命令

`sudo` 命令用于普通用户执行需要超级权限的命令时所使用的。在使用 Linux 系统时，不推荐直接需要 `root` 用户，由于 `root` 用户拥有最高权限，在 `root` 模式下，如果误操作了系统的一些关键配置，就会导致不必要的损失。所以推荐登录 Linux 系统时采用普通用户，而需要修改系统配置时，在系统命令前加上 `sudo`，如上面提到过的

```
sudo apt-get install bintuils #在普通用户下安装 binutils 软件包
```

在普通用户下如果直接执行 `apt-get install binutils` 命令，会提示 `Permission denied` 错误信息。

在执行 `sudo` 命令时，系统首先要求需要超级权限密码如：

```
jie@ubuntu:~$ sudo apt-get install bintuils
[sudo] password for jie:
```

在输入了正确的超级权限密码后，系统会正常安装 `binutils` 软件包。

1.2.9 wget 下载命令

`wget` 是一个从网络上自动下载文件的工具，支持 HTTP，HTTPS 和 FTP 协议。在 Windows 图形界面下，用户一般是用 IE 访问网站，之后再点击需要下载文件的链接，就可以把文件下载到本地。`wget` 的作用就是这样，当然由于他不如图形界面那样直观，所以在下载文件时，需要知道文件的详细 URL。

`wget` 的基本用法：

```
wget [OPTION]... [URL]...
```

举例：

```
wget ftp://ftp.freebsdchina.org/pub/FreeBSD/README.txt
```

上面的命令就可以把 `freebsdchina` 的 ftp 服务器上的 `/pub/FreeBSD` 目录下的 `README.txt` 文件下载到当前目录

```
wget http://www.tektalk.org/wp-content/uploads/2008/03/chinasystemsoftware.pdf
```

下载了一个 pdf 文件到当前目录

有些服务器加上了用户限制，要求用户输入用户名和密码，那么就可以在下载时把

用户名和密码加在 URL 里

```
wget ftp://username:password@server-domain/download filename
```

wget 的功能不仅仅限于以上的使用，它还可以执行断点续传，递归下载等，如第一次下载一个文件时没有下载完，在第二次下载时，可以通过加参数的方式，继续前一次的下载。

断点续传的参数为-c，如

```
wget -c http://mirrors.163.com/ubuntu-releases/9.10/ubuntu-9.10-alternate-i386.iso
```

递归下载整个网站，如：

```
wget -r -p -k http://website-domain/
```

递归下载时，wget 是根据下载的首页里的链接，依次下载这些链接，再分析下载下来的文件中的 URL，再去下载新的文件。而不是整个网站的复制。

第 2 章 文件查看/编辑方法介绍

2.1 cat 命令介绍

`cat` 命令的原含义为连接（concatenate），用于连接多个文件内容并输出到标准输出流中（标准输出流默认为屏幕）。实际运用过程中，我们常使用它来显示文件内容。如果您熟悉 MS-DOS 下的 `type` 命令，相信不难掌握 `cat` 命令。该命令的常用示例如下：

```
cat file1.txt           显示 file1.txt 文件的内容；
cat file1.txt file2.txt 显示 file1.txt 和 file2.txt 文件的内容；
cat -n file1.txt        显示 file1.txt 文件的内容同时显示行号；
```

2.2 vi 命令介绍

`vi` 是 Visual Interface 的简称，它在 Linux 上的地位就仿佛记事体程序在 Windows 上一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。

`vi` 不是一个排版程序，它不象 Word 或 WPS 那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。

`vi` 没有菜单，只有命令，且命令繁多。限于篇幅，本文只介绍常用的命令。

`vi` 有三种基本工作模式：命令行模式，文本输入模式和末行模式。

1. 命令行模式

任何时候，不管用户处于何种模式，只要按一下“ESC”键，即可使 `vi` 进入命令行模式；当在 shell 环境下输入 `vi` 命令启动 `vi` 编辑器时，也是处于该模式下。

在该模式下，用户可以输入各种合法的 `vi` 命令，用于管理自己的文档。此时从键盘上输入的任何字符都被当作编辑命令来解释，若输入的字符是合法的 `vi` 命令，则 `vi` 在接受用户命令之后完成相应的动作（但需注意的是，所输入的命令并不在屏幕上显示出来）。若输入的字符不是 `vi` 的合法命令，`vi` 会响铃报警。

2. 文本输入模式

在命令模式下输入插入命令 `i`、附加命令 `a`、打开命令 `o`、修改命令 `c`、取代命令 `r` 或替换命令 `s` 都可以进入文本输入模式。在该模式下，用户输入的任何字符都被 `vi` 当做文件内容保存起来，并将其显示在屏幕上。在文本输入过程中，若想回到命令模式下，按“ESC”键即可。

3. 末行模式

在命令模式下用户按“:”键即可进入末行模式下，此时vi会在显示窗口的最后一行（通常也是屏幕的最后一行）显示一个“:”作为末行模式的提示符，等待用户输入命令。多数文件管理命令都是在此模式下执行的（如把编辑缓冲区的内容写到文件中等）。末行命令执行完后，vi 自动回到命令模式。

若在末行模式下输入命令过程中改变了主意，可按“ESC”键，或用退格键将输入的命令全部删除之后，再按一下退格键，即可使vi回到命令模式下。

2.2.1 vi 的进入与退出

在 shell 模式下，键入 vi 及需要编辑的文件名，即可进入 vi。例如：

```
vi example.txt
```

即可编辑 **example.txt** 文件。如果该文件存在，则编辑界面中会显示该文件的内容，并将光标定位在文件的第一行；如果文件不存在，则编辑界面中无任何内容。如果需要在进入 vi 编辑界面后，将光标置于文件的第 **n** 行，则在 vi 命令后面加上“+n”参数即可。例如需要从 **example.txt** 文件的第 5 行开始显示，则使用如下命令：

```
vi +5 example.txt
```

退出 vi 时，需要在末行模式中输入退出命令“q”。如果在文本输入模式下，首先按“ESC”键进入命令模式，然后输入“:”进入末行模式。在末行模式下，可使用如下退出命令：

:q 直接退出。如果在文本输入模式下修改了文档内容，则不能退出。

:wq 保存后退出。

:x 同 wq。

:q! 不保存内容，强制退出。

2.2.2 vi 中显示行号

在末行模式下，输入如下命令：

```
set number
```

可使 vi 在编辑界面中显示行号。此外，在末行模式下可使用如下“nu”命令（number 的简写）来显示光标所在行的行号及该行的内容。

2.2.3 光标移动操作

全屏幕文本编辑器中，光标的移动操作无疑是最经常使用的操作了。用户只有熟练

地使用移动光标的这些命令，才能迅速准确地到达所期望的位置处进行编辑。

vi 中的光标移动既可以在命令模式下，也可以在文本输入模式下，但操作的方法不尽相同。

在文本输入模式下，可直接使用键盘上的四个方向键移动光标；在命令模式下，有很多移动光标的方法。不但可以使用四个方向键来移动光标，还可以用 h、j、k、l 这四个键代替四个方向键来移动光标，这样可以避免由于不同机器上的不同键盘定义所带来的矛盾，而且使用熟练后可以手不离开字母键盘位置就能完成所有操作，从而提高工作效率。

以下命令均在命令行模式下完成光标移动：

- h 光标左移 如果在按 h 命令前输入数字 n 则光标左移 n 个字符
- l 光标右移 如果在按 l 命令前输入数字 n 则光标右移 n 个字符
- j 光标上移 如果在按 j 命令前输入数字 n 则光标上移 n 个字符
- k 光标下移 如果在按 k 命令前输入数字 n 则光标 xia 移 n 个字符
- 0 光标移到行首，（零）
- \$ 光标移到行尾
- H 光标移到屏幕上显示的第一行（并不一定是文件头）
- L 光标移到屏幕上显示的最后一行（并不一定是文件尾）
- M 光标移到屏幕的中间一行
- nG 光标移到第 n 行
- w 或 W 将光标右移至下一个单词的词首
- e 或 E 如果光标起始位置处于单词内（即非单词尾处），则该命令将把光标移到本单词词尾；如果光标起始位置处于单词尾，则该命令将把光标移动到下一个单词的词尾。
- b 或 B 如果光标处于所在单词内（即非单词首），则该命令将把光标移至本单词词首；如果光标处于所在单词的词首，则该命令将把光标移到上一个单词的词首；
- Ctrl+G 状态命令，显示当前编辑文档的状态。包括正在编辑的文件名、是否修改过、当前行号、文件的行数以及光标之前的行占整个文件的百分比。

此外，也可以通过以下末行模式下的命令完成光标在整个文件中的移动操作。

- :n 光标移到文件的第 n 行
- :\$ 光标移到文件的最后一行

2.2.4 文本插入操作

在命令模式下，用户输入的任何字符都被 vi 当作命令加以解释执行，如果用户要将输入的字符当作是文本内容时，则首先应将 vi 的工作模式从命令模式切换到文本输入模式。切换的方式是使用下面的命令：

1) 插入 Insert 命令

vi 提供了两个插入命令：i 和 I

i 命令 插入文本从光标所在位置前开始，并且插入过程中可以使用 <Backspace>键删除错误的输入。此时 vi 处于插入状态，屏幕最下行显示“--INSERT--”（插入）字样。

I 命令 该命令是将光标移到当前行的行首，然后在其前插入文本。

2) 附加 append 命令

vi 提供了两个附加插入命令：a 和 A。

a 命令 该命令用于在光标当前所在位置之后追加新文本，新输入的文本放在光标之后，在光标后的原文本将相应地向后移动，光标可在一行的任何位置。

A 命令 该命令与 a 命令不同的是，A 命令将把光标挪到所在行的行尾，从那里开始插入新文本。当输入 A 命令后，光标自动移到该行的行尾。

a 和 A 命令是把文本插入到行尾的唯一方法。

3) 打开 open 命令

不论是 Insert 命令也好，还是 append 命令也好，所插入的内容都是从当前行中的某个位置开始的。若我们希望在某行之前或某行之后插入一些新行，则应使用 open 命令。

vi 提供了两个打开命令：o 和 O。

o 命令 该命令将在光标所在行的下面新开一行，并将光标置于该行的行首，等待输入文本。要注意当使用 <Backspace>删除字符时只能删除从插入模式开始的位置以后的字符，对于以前的字符不起作用。

O 命令 和 o 命令相反，O 命令是在光标所在行的上面插入一行，并将光标置于该行的行首，等待输入文本。

2.2.5 文件修改操作

在命令模式下可以使用 vi 提供的各种有关命令对文本进行修改，包括对文本内容的删除、复制、取代和替换等。

1. 文本删除/移动

在编辑文本时，经常需要删除一些不需要的文本，我们可以用 <Backspace>

键将输错或不需要的文本删除，但此时有一个限制就是当删到行头之后，再想删上面那行的内容是不可能的。

在命令模式下，**vi** 提供了许多删除命令。这些命令大多是以 **d** 开头的。常用的有：

1) 删除单个字符

x 删除光标处的字符。若在 **x** 之前加上一个数字 **n**，则删除从光标所在位置开始向右的 **n** 个字符。

X 删除光标前面的那个字符。若在 **X** 之前加上一个数字 **n**，则删除从光标前面那个字符开始向左的 **n** 个字符。

显然这两个命令是删除少量字符的快捷方法。

2) 删除多个字符

dd 删除光标所在的整行。在 **dd** 前可加上一个数字 **n**，表示删除当前行及其后 **n-1** 行的内容。

D 或 **d\$** 两命令功能一样，都是删除从光标所在处开始到行尾的内容。

d0 删除从光标前一个字符开始到行首的内容。

dw 删除一个单词。若光标处在某个词的中间，则从光标所在位置开始删至词尾。同 **dd** 命令一样，可在 **dw** 之前加一个数字 **n**。表示删除 **n** 个指定的单词。

如果用户不小心进行了误删除操作，也不要紧，**vi** 提供了恢复误操作的命令，并且可以将恢复的内容移动，放在文本的任何地方。恢复命令用“**np**”，其中 **n** 为需要恢复的次数。例如使用 **dd** 命令删除了一行内容，然后使用“**2p**”命令，则被删除的内容会被重新插入两遍。

通过 **dd** 命令及 **p** 命令的结合使用，可以很方便的实现文本行的移动操作。

2. 文本复制命令

yy 复制光标所在的整行。在 **yy** 前可加上一个数字 **n**，表示复制当前行及其后 **n-1** 行的内容。

文本行复制后，通过使用上面介绍的 **p** 命令，可以将文本行粘贴到任何地方。

dd、**yy** 及 **p** 命令在 **vi** 中的作用，类似于 Windows 中图形编辑程序的“剪切”、“复制”及“粘贴”命令。

3. 取消上一命令

取消上一命令（Undo），也称复原命令，是非常有用的命令，它可以取消前一次的

误操作或不合适的操作对文件造成的影响,使之回复到这种误操作或不合适操作被执行之前的状态。

取消上一命令有两种形式,在命令模式下键入字符 **u** 和 **U**。它们的功能都是取消刚才输入的命令,恢复到原来的情况。小写 **u** 和大写 **U** 在具体细节上有所不同,二者的区别在于,大写 **U** 命令的功能是恢复到误操作命令前的情况,即如果插入命令后使用 **U** 命令,就删除刚刚插入的内容;如果删除命令后使用 **U** 命令,就相当于在光标处又插入刚刚删除的内容。这里把所有修改文本的命令都视为插入命令。也就是说,**U** 命令只能取消前一步操作,如果用 **U** 命令撤消了前一步操作,当再按 **U** 键时,并不是撤消再前一步的操作,而是撤消了刚才 **U** 命令执行的操作,也就是又恢复到第一次使用 **U** 命令之前的状态,结果是什么都没做。而小写 **u** 命令的功能是把当前行恢复成被编辑前的状态,而不管此行被编辑了多少次。

4. 重复命令 (Redo)

重复命令也是一个非常常用的命令。在文本编辑中经常会碰到需要机械地重复一些操作,这时就需要用到重复命令。它可以让用户方便地再执行一次前面刚完成的某个复杂的命令。

重复命令只能在命令模式下工作,在该模式下按 “.” 键既可。执行一个重复命令时,其结果是依赖于光标当前位置的。

2.2.6 文件搜索与替换操作

在进行文本编辑的时候,您可以需要搜索或定位特定的单词或单词的一部分。**vi** 编辑器有向前或向后搜索指定模式的功能。

“/” 是 **vi** 编辑器的模式搜索命令,键入 “/” 后,屏幕的底部会出现一行,同时一个斜杠会出现在底行的行首上,且光标被放置在斜杠之后。此时,您可以在光标位置键入你需要搜索的模式。输入完成并按下 **ENTER** 键后,编辑器将开始从光标所在的位置起向文件尾部搜索你键入的模式。如果输入的模式被找到,光标将停留在找到的模式上。在找到一个匹配的模式后,如果您还需要搜索下一个匹配的模式,使用命令 “n” 即可;如果需要找到上一个匹配的模式,则使用 “N” 命令即可。

此外,“?” 命令也是模式搜索命令。与 “/” 的区别在于: “/” 是从光标当前所在位置向文件尾部搜索,而 “?” 命令则是由光标当前所在位置向文件头部搜索。

文本替换操作需要使用末行模式进行操作,即所有文本替换操作命令均以 “:” 开头。

文本替换命令的格式为:

:s/text1/text2 用于将光标所在段落搜索到的第一个 “text1” 替换为 “text2”

:s/text1/text2/g 用于将光标所在段落的所有 “text1” 替换为 “text2”

:m,ns/text1/text2/g 用于将从 m 行开始至 n 行结束的所有搜索到的 “text1” 替换为 “text2”。其中可以使用 \$ 表示末行,即 “1,\$” 表示替换文档中的所有符合条

件的字符。

2.3 head 及 tail 命令介绍

当需要查看一个文本文件的头部或尾部时，**head** 命令及 **tail** 命令可以非常方便的完成该操作。**head** 命令用于查看一个文本文件的开头部分；而 **tail** 命令则用于显示文本文件的末尾几行。这两个命令举例如下：

```
head example.txt 显示文件 example.txt 的前十行内容；
head -n 20 example.txt 显示文件 example.txt 的前二十行内容；
tail example.txt 显示文件 example.txt 的后十行内容；
tail -n 20 example.txt 显示文件 example.txt 的后二十行内容；
tail -f example.txt 显示文件 example.txt 的后十行内容并在文件内容增加后自动显示新增的文件内容。
```

注意：

最后一条命令非常有用，尤其在监控日志文件时，可以在屏幕上一直接显示新增的日志信息。

第 3 章 网络配置及操作

3.1 网络配置命令

Linux 系统中最常用的网络配置命令包括 `ifconfig`、`route`，其中 `ifconfig` 用来查看和配置网络接口（通常是网卡）信息，包括网络接口设备的 IP 地址、掩码等；`route` 用来管理 Linux 系统内核中的路由表，它最大的用途就是用来设定静态的路由表项，通常是在系统用 `ifconfig` 配置网络接口后，用它来设定主机或者一网段的 IP 地址应该通过什么接口发送等。

3.1.1 ifconfig

`Ifconfig` 是用于查看网络接口信息，修改网络接口 IP 地址、网络掩码及其它可配置参数。

`ifconfig` 的语法为：

```
ifconfig [-a] [-V] [-s] <interface> [[<AF>] <address>]
[[-]broadcast [<address>]]
[netmask <address>]
[up|down]
```

其中 `interface` 为网卡名，`address` 为 IP 地址，`broadcast` 表示设置广播地址，其后的 `address` 为广播地址的 IP 地址，`netmask` 表示设置子网掩码，其后的 `address` 为子网掩码的 IP 地址。`up`、`down` 设置当前网卡的状态，`up` 表示启用当前网卡，`down` 表示关闭当前网卡。

`Ifconfig` 的常用参数说明如下：

- a 查看所有网卡状态
- V 查看 `ifconfig` 命令的版本信息
- s 查看统计信息

示例：

- 1) 将第一块网卡的 IP 地址设置为 192.168.0.1

```
ifconfig eth0 192.168.0.1    （格式 ifconfig 网络设备名 IP 地址）
```

- 2) 暂时关闭或启用网卡

关闭第一块网卡: `ifconfig eth0 down`

启用第一块网卡: `ifconfig eth0 up`

3) 将第一块网卡的子网掩码设置为 255.255.255.0

```
ifconfig eth0 netmask 255.255.255.0    (格式: ifconfig 网络设备名 netmask 掩码)
```

我们也可以同时设置 IP 地址和子网掩码:

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

4) 查看第一块网卡的状态

```
ifconfig eth0
```

如果要查看所有的网卡状态, 如果要查看所有的网卡状态 则直接使用不带参数的 `ifconfig` 命令即可。`ifconfig` 输出的状态信息是十分有用的, 下面, 我们就简单说明一下:

UP/DOWN: 网卡是否启动了, 如果是 DOWN 的话, 那肯定无法用的

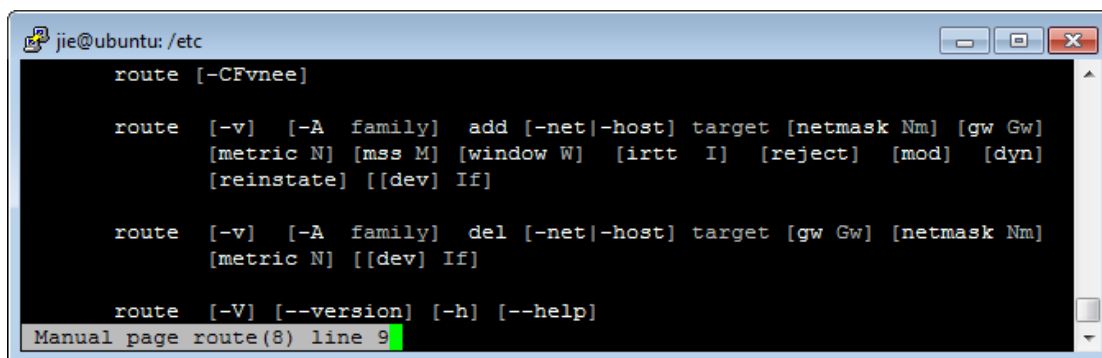
RX packets 中的 errors 包的数量如果过大说明网卡在接收时有问题

TX packets 中的 errors 包的数量如果过大说明网卡在发送时有问题

3.1.2 route

`route` 命令用于查看本机的内核路由表及设置静态路由。

`route` 命令的语法为:



```
jie@ubuntu: /etc
route [-CFvnee]

route [-v] [-A family] add [-net|-host] target [netmask Nm] [gw Gw]
[metric N] [mss M] [window W] [irtt I] [reject] [mod] [dyn]
[reinststate] [[dev] If]

route [-v] [-A family] del [-net|-host] target [gw Gw] [netmask Nm]
[metric N] [[dev] If]

route [-V] [--version] [-h] [--help]
Manual page route(8) line 9
```

`route` 的常用参数说明如下:

-v 使用冗余输出模式;

-A family 指定特定的地址族 (例如“inet”, “inet6”);

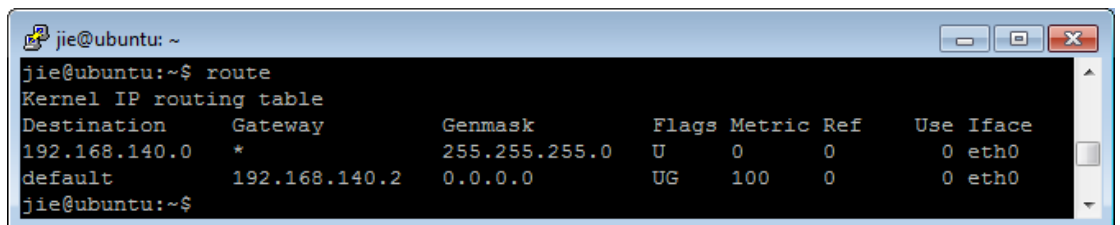
-n 使用数字显示的地址 (例如, 202.38.75.75) 而不是去解释域名;

-e 使用与 `netstat` 相同的输出格式;

- ee 参数会产生很长的输出，包括内核路由表的几乎所有信息。
- net 目标（target）是一个网段。
- host 目标（target）是一个单独的主机。
- F 显示内核 FIB 路由表。结果可能被-e 和-ee 参数改变。
- C 显示内核中路由缓存信息
- del 删除一个路由表项
- add 增加一个路由表项
- target 配置的目的网段或者主机。可以是 IP，或者是网络或主机名。
- netmask Nm 用来指明要添加的路由表项的网络掩码。
- gw Gw 任何通往目的（target）的 IP 分组都要通过这个网关。
- metric M 设置路由表中该项的尺度域（metric field）为 M。

示例：

1) 查看本机路由表



```
jie@ubuntu: ~
jie@ubuntu:~$ route
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.140.0  *              255.255.255.0   U        0      0      0 eth0
default        192.168.140.2  0.0.0.0         UG       100    0      0 eth0
jie@ubuntu:~$
```

2) 添加一条路由表项，网段 192.168.0.x 应该从接口"eth0"走

```
route add -net 192.168.0.0 netmask 255.255.255.0 dev eth0
```

3) 添加一条缺省路由

```
route add default gw 192.168.5.1
```

如果没有其他匹配的路由项，就使用这个路由规则。也可以叫做网关设置。

4) 删除缺省路由

```
route del default gw 网关地址
```

5) route 命令的输出结果示例说明

输出的格式有以下几栏：

Destination 目标网段或者主机

Gateway 网关地址，如果没有设置，则是"*"表示

Genmask 网络掩码

Flags 一些可能的标记如下：

U 路由是活动的
H 目标是一个主机
G 使用网关 (gateway)
R reinstate route 动态路由产生的表项
D dynamically installed by daemon or redirect
M modified from routing daemon or redirect
! reject route

Metric 路由距离。

Ref 路由项引用次数。(linux 内核中没有使用)

Use 查找路由项的次数。

Iface 该路由表项对应的输出接口。

3.1.3 dhclient

dhclient 的作用是为现有网络接口自动获取 IP 地址

dhclient 的一般用法:

dhclient 网卡接口名

网络接口名是 Linux 为每块网卡分配的唯一的可读名称, 一般为 eth0, eth1, 分别代表第一块网卡, 第二块网卡。

举例:

dhclient eth0

为第一块网卡自动分配 IP 地址。

3.2 网络操作命令

Linux 系统常用的网络操作命令包括 netstat、nslookup、host 和 ping。netstat 命令的功能是显示网络连接、路由表和网络接口信息, 可以让用户得知目前都有哪些网络连接正在运作。nslookup 命令的功能是查询一台机器的 IP 地址和其对应的域名。它通常需要一台域名服务器来提供域名服务。如果用户已经设置好域名服务器, 就可以用这个命令查看不同主机的 IP 地址对应的域名。host 命令可根据主机名查询主机的 IP 地址或根据主机 IP 地址查询主机名; ping 命令用于查看网络上的主机是否在工作。

3.2.1 netstat 命令

netstat 命令的语法为:

netstat [选项]

netstat 的常用参数说明如下:

- a 显示所有 **socket** 包括正在监听的
- c 每隔 1 秒就重新显示一遍直到用户中断它
- i 显示所有网络接口的信息格式同“ifconfig -e”
- n 以网络 IP 地址代替名称显示出网络连接情形
- r 显示核心路由表格式同“route -e”
- t 显示 **TCP** 协议的连接情况
- u 显示 **UDP** 协议的连接情况
- v 显示正在进行的工作

示例:

- 1) 在本机使用 **netstat** 命令, 其输出为如下

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 256 10.153.89.66:ssh 10.153.21.168:1054 ESTABLISHED
tcp 0 0 10.153.89.66:ssh 10.153.21.166:1222 ESTABLISHED
tcp 0 0 10.153.89.66:1521 10.153.21.166:2625 ESTABLISHED
tcp 0 0 10.153.89.66:1521 10.153.21.166:2623 ESTABLISHED
.....
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type State I-Node Path
unix 9 [ ] DGRAM 919 /dev/log
unix 2 [ ] STREAM CONNECTED 50246124
unix 3 [ ] STREAM CONNECTED 3914245 /tmp/.ICE-unix/dcop18694-1074144356
unix 3 [ ] STREAM CONNECTED 3914244
unix 3 [ ] STREAM CONNECTED 3914127 /tmp/.X11-unix/X0
.....
```

3.2.2 nslookup 命令

nslookup 命令的功能是查询一台机器的 IP 地址和其对应的域名。它通常需要一台域名服务器来提供域名服务。如果用户已经设置好域名服务器, 就可以用这个命令查看不同主机的 IP 地址对应的域名。命令的语法为:

```
nslookup [IP 地址/域名]
```

示例:

在本机执行 **nslookup** 命令

```
Default Server: name.tlc.com.cn
Address: 192.168.1.99
```

>

在符号 > 后面输入要查询的 IP 地址或域名并回车即可。如果要退出该命令，输入 `exit` 并回车。

3.2.3 host 命令

`host` 命令的功能基本与 `nslookup` 相同。命令的语法为：

`host [选项] [IP 地址/域名/主机名]`

`host` 的常用参数如下：

-a 显示所查域名或 IP 地址、域名或主机名的相关信息。

示例：

```
[root]# host huawei-3com.com
huawei-3com.com has address 172.19.1.228
```

3.2.4 ping 命令

`ping` 命令用于查看网络上的主机是否在工作，它向该主机发送 `ICMP_ECHO_REQUEST` 包。有时我们想从网络上的某台主机上下载文件，可是又不知道那台主机是否开着，就需要使用 `ping` 命令查看。该命令的一般格式为：

`ping [选项] [主机名/IP 地址]`

`ping` 的常用参数如下：

-c 数目 在发送指定数目的包后停止

-d 设定 `SO_DEBUG` 的选项

-f 大量且快速地送网络封包给一台机器看它的回应

-I 秒数 设定间隔几秒送一个网络封包给一台机器预设值是一秒送一次

-l 次数 在指定次数内以最快的方式送封包数据到指定机器（只有超级用户可以使用此选项）

-q 不显示任何传送封包的信息只显示最后的结果

-r 不经由网关而直接送封包到一台机器，通常是查看本机的网络接口是否有问题

-s 字节数 指定发送的数据字节数，预设值是 56，加上 8 字节的 `ICMP` 头，一共是 64 `ICMP` 数据字节。

3.2.5 traceroute 命令

traceroute 可以说是 ping 命令的加强版，它可以显示从本机到目标设备所经过的所有路由器地址及路由器响应的时间。traceroute 的原理与根据 IP 地方的 TTL(Time to Live) 的值实现的，该命令的一般格式为：

```
traceroute [选项] [目标主机名/IP 地址]
```

举例：

```
traceroute www.chinaunix.net
```

此命令会显示从本机到 www.chinaunix.net 中间所经过的所有路由器地址及路由器回应的时间。

3.3 常用网络配置文件介绍

在 Linux 系统中，TCP/IP 网络是通过若干个文本文件进行配置的，需要编辑这些文件来完成联网工作。系统中重要的有关网络配置文件为：

```
/etc/hosts  
/etc/services  
/etc/host.conf  
/etc/nsswitch.conf  
/etc/resolv.conf
```

接下来我们将对这些文件逐一讲述，这些文件都可以在系统运行时进行修改，不用启动或者停止任何守护程序，更改会立刻生效（除了 `/etc/sysconfig/network`）。另外，这些文件都支持由“#”开头的注释，每一个文件都有在 UNIX 手册页中的第 5 部分中有一项，可以用 man 命令来获取它们。

3.3.1 /etc/hosts IP 地址和主机名的映射

`/etc/hosts` 中包含了 IP 地址和主机名之间的映射，还包括主机名的别名，IP 地址的设计使计算机容易识别，但对于人却很难记住它们，为了解决这个问题，创建了 `/etc/hosts` 这个文件。下面是一个例子文件：

```
127.0.0.1 machine1 localhost.localdomain localhost  
192.168.1.100 machine7  
192.168.1.101 otherpc otheralias
```

在这个例子中，本机名是 machine1，otherpc 还有别名 otheralias，它可以指向

otheraliases。一旦配置完机器的网络配置文件，就可以通过别名访问了。

`/etc/hosts` 文件通常含有主机名、`localhost` 和系统管理员经常使用的系统别名，有时候 `telnet` 到 Linux 机器要等待很长时间，可以通过在 `/etc/hosts` 加入客户的机器的 IP 地址和主机名的匹配项，就可以减少登录等待时间。在没有域名服务器情况下，系统上的所有网络程序都通过查询该文件来解析对应于某个主机名的 IP 地址，否则，其他的主机名通常使用 DNS 来解决，DNS 客户部分的配置在文件 `/etc/resolv.conf` 中。

3.3.2 `/etc/services`

`/etc/services` 中包含了服务名和端口号之间的映射，不少的系统程序要使用这个文件。例：

```
tcpmux      1/tcp      # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp      sink null
discard     9/udp      sink null
sysstat     11/tcp     users
```

最左边一列是主机服务名，中间一列是端口号，“#”后面是端口类型，可以是 TCP 也可以是 UDP。任何后面的列都是前面服务的别名。在这个文件中也存在着别名，它们出现在端口号后面，在上述例子中 `sink` 和 `null` 都是 `discard` 服务的别名。

3.3.3 `/etc/host.conf` 配置名字解析器

有两个文件声明系统到哪里寻找名字信息来配置 UNIX 名字解析器的库。文件 `/etc/host.conf` 由版本 5 的 `libc` 库所使用，而 `/etc/nsswitch.conf` 由版本 6 使用 (`glibc`)。问题在于一些程序使用其中一个，而一些使用另一个，所以将两个文件都配置正确是必要的。

`/etc/host.conf` 文件指定如何解析主机名，Linux 通过解析器库来获得主机名对应的 IP 地址。例：

```
order hosts,bind
multi on
```

“order”指定主机名查询顺序，其参数为用逗号隔开的查找方法，支持的查找方法为 `bind`、`hosts` 和 `nis`，分别代表 DNS、`/etc/hosts` 和 NIS，这里规定先查询 `/etc/hosts` 文件然后再使用 DNS 来解析域名。

“trim”表明当通过 DNS 进行地址到主机名的转换时，域名将从主机名中被裁剪掉，

trim 可以被多个域包含多次,对/etc/hosts 和 NIS 查询方法不起作用,注意在/etc/hosts 和 NIS 表中主机名是被适当地(有或没有全域名)列出的。

"multi" 指定是否"/etc/hosts"文件中指定的主机可以有多个地址,值为 on 表示允许,拥有多个 IP 地址的主机一般称为具有多个网络界面。

"nospoof " 指是否允许对该服务器进行 IP 地址欺骗,值为 on 表示不允许,IP 欺骗是一种攻击系统安全的手段,通过把 IP 地址伪装成别的计算机,来取得其它计算机的信任。

"alert" 当 nospoof 指令为 on 时,alert 控制欺骗的企图是否用 syslog 工具进行记录,值为 on 表示使用,缺省值为 off。

"recorder" 如果被设置为 on,所有的查询将被重新排序,所以在同一子网中的主机将首选被返回,缺省值为 off。

3.3.4 /etc/nsswitch.conf 配置名字解析器

/etc/nsswitch.conf 文件是由 Sun 公司开发并用于管理系统中多个配置文件查找的顺序,它比/etc/host.conf 文件提供了更多的功能。/etc/nsswitch.conf 中的每一行或者是注释(以#号开头)或者是一个关键字后跟冒号和一系列要试用的有顺序的方法。每一个关键字是在/etc/目录可以被/etc/nsswitch.conf 控制的/etc 文件的名称。下面是可以被包含的关键字:

passwd	系统用户
group	用户组
shadow	隐蔽口令
hosts	主机名和 IP 地址
networks	网络名和号
protocols	网络协议
services	端口号和服务名称
ethers	以太网号
rpc	远程进程调用的名称和号
netgroup	网内组

3.3.5 /etc/resolv.conf 配置 DNS 客户

文件/etc/resolv.conf 配置 DNS 客户,它包含了主机的域名搜索顺序和 DNS 服务器的地址,每一行应包含一个关键字和一个或多个的由空格隔开的参数。

下面是一个例子文件:

```
domain hiwifi.net
```

```
search hiwifi.net
nameserver 192.168.140.2
```

合法的参数及其意义如下：

nameserver 表明 DNS 服务器的 IP 地址。可以有很多行的 **nameserver**，每一个带一个 IP 地址。在查询时就按 **nameserver** 在本文件中的顺序进行，且只有当第一个 **nameserver** 没有反应时才查询下面的 **nameserver**。

domain 声明主机的域名。很多程序用到它，如邮件系统；当为没有域名的主机进行 DNS 查询时，也要用到。如果没有域名，主机名将被使用，删除所有在第一个点(.)前面的内容。

search 它的多个参数指明域名查询顺序。当要查询没有域名的主机，主机将在由 **search** 声明的域中分别查找。**domain** 和 **search** 不能共存；如果同时存在，后面出现的将会被使用。

sortlist 允许将得到域名结果进行特定的排序。它的参数为网络/掩码对，允许任意的排列顺序。

3.4 常用网络操作方法

Linux 提供了一组强有力的网络命令来为用户服务，这些工具能够帮助用户登录到远程计算机上、传输文件和执行远程命令等。本节介绍下列几个常用的有关网络操作的命令：

ftp 传输文件
telnet 登录到远程计算机上

3.4.1 ftp 远程文件传输

ftp 命令是标准的文件传输协议的用户接口。**ftp** 是在 TCP/IP 网络上的计算机之间传输文件的简单有效的方法。它允许用户传输 ASCII 文件和二进制文件。在 **ftp** 会话过程中，用户可以通过使用 **ftp** 客户程序连接到另一台计算机上。从此，用户可以在目录中上下移动、列出目录内容、把文件从远程机拷贝到本地机上、把文件从本地机传输到远程系统中。需要注意的是，如果用户没有那个文件的存取权限，就不能从远程系统中获得文件或向远程系统传输文件。为了使用 **ftp** 来传输文件，用户必须知道远程计算机上的合法用户名和口令。这个用户名/口令的组合用来确认 **ftp** 会话，并用来确定用户对要传输的文件可以进行什么样的访问。

ftp 命令的一般格式如下：
\$ ftp 主机名/IP

如果没有指定主机名，**ftp** 将给出提示符，等待用户输入命令：

```
$ ftp
ftp >
```

此时在 **ftp>**提示符后面输入 **open** 命令加主机名或 IP 地址，将试图连接指定的主机。不管使用哪一种方法，如果连接成功，需要在远程机上登录。用户如果在远程机上有帐号，就可以通过 **ftp** 使用这一帐号并需要提供口令。在远程机上的用户帐号的读写权限决定该用户在远程机上能下载什么文件和将上载文件放到哪个目录中。

一旦用户使用 **ftp** 在远程站点上登录成功，将得到 **ftp>** 提示符。现在可以自由使用 **ftp** 提供的命令，可以用 **help** 命令取得可供使用的命令清单，也可以在 **help** 命令后面指定具体的命令名称，获得这条命令的说明。

最常用的命令有：

ls	列出远程机的当前目录
cd	在远程机上改变工作目录
lcd	在本地机上改变工作目录
ascii	设置文件传输方式为 ASCII 模式，默认情况下， ftp 按 ASCII 模式传输文件，用户也可以指定其他模式。 ascii 和 binary 命令的功能是设置传输的模式。用 ASCII 模式传输文件对纯文本是非常好的，但为避免对二进制文件的破坏，用户可以以二进制模式传输文件
binary	设置文件传输方式为二进制模式
close	终止当前的 ftp 会话
hash	每次传输完数据缓冲区中的数据后就显示一个#号
get(mget)	从远程机传送指定文件到本地机， get 命令的一般格式为： get 文件名

mget 命令一次获取多个远程文件 **mget** 命令的一般格式为：

mget 文件名列表

使用用空格分隔的或带通配符的文件名列表来指定要获取的文件，对其中的每个文件都要求用户确认是否传送。

put(mput) 从本地机传送指定文件到远程机，**put** 命令的一般格式为：

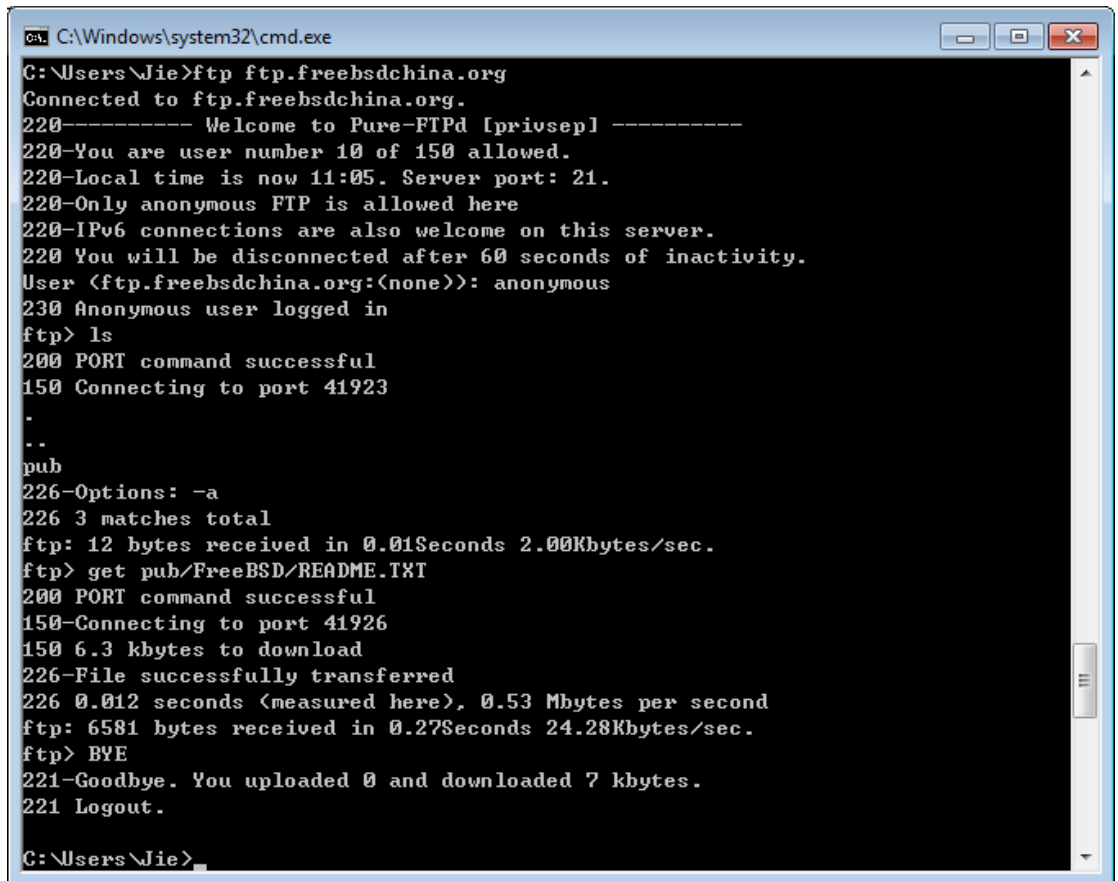
put 文件名

mput 命令一次发送多个本地文件，**mput** 命令的一般格式为：

mput 文件名列表

使用用空格分隔的或带通配符的文件名列表来指定要发送的文件，对其中的每个文件都要求用户确认是否发送。

open 连接远程 ftp 站点
quit 断开与远程机的连接并退出 ftp
? 显示本地帮助信息
! 转到 Shell 中



```
C:\Windows\system32\cmd.exe
C:\Users\Jie>ftp ftp.freebsdchina.org
Connected to ftp.freebsdchina.org.
220----- Welcome to Pure-FTPd [privsep] -----
220-You are user number 10 of 150 allowed.
220-Local time is now 11:05. Server port: 21.
220-Only anonymous FTP is allowed here
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 60 seconds of inactivity.
User <ftp.freebsdchina.org:<none>>: anonymous
230 Anonymous user logged in
ftp> ls
200 PORT command successful
150 Connecting to port 41923
.
..
pub
226-Options: -a
226 3 matches total
ftp: 12 bytes received in 0.01Seconds 2.00Kbytes/sec.
ftp> get pub/FreeBSD/README.TXT
200 PORT command successful
150-Connecting to port 41926
150 6.3 kbytes to download
226-File successfully transferred
226 0.012 seconds <measured here>, 0.53 Mbytes per second
ftp: 6581 bytes received in 0.27Seconds 24.28Kbytes/sec.
ftp> BYE
221-Goodbye. You uploaded 0 and downloaded 7 kbytes.
221 Logout.

C:\Users\Jie>
```

3.4.2 telnet 远程登录

用户使用 `telnet` 命令进行远程登录。该命令允许用户使用 `telnet` 协议在远程计算机之间进行通信，用户可以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。虽然有些系统确实为远程用户提供登录功能，但出于对安全的考虑，要限制来宾的操作权限，因此，这种情况下能使用的功能是很少的。当允许远程用户登录时，系统通常把这些用户放在一个受限制的 `shell` 中，以防系统被怀有恶意的或不小心的用户破坏。用户还可以使用 `telnet` 从远程站点登录到自己的计算机上，检查电子邮件、编辑文件和运行程序，就像在本地登录一样。但是，用户只能使用基于终端的环境而不是 `X Windows` 环境，`telnet` 只为普通终端提供终端仿真，而不支持 `X Window` 等图形环境。

`telnet` 命令的一般形式为：

`telnet` 远程主机名/IP

其中“远程主机名/IP”是要连接的远程机的主机名或 IP 地址。如果这一命令执行成功，将从远程机上得到 login: 提示符。

使用 telnet 命令登录的过程如下：

```
$ telnet 远程主机名/IP
```

一旦 telnet 成功地连接到远程系统上，就显示登录信息并提示用户输入用户名和口令。如果用户名和口令输入正确，就能成功登录并在远程系统上工作。

在 telnet 提示符后面可以输入很多命令，用来控制 telnet 会话过程，在 telnet 联机帮助手册中对这些命令有详细的说明。

示例：

3.4.3 SSH 安全登录

SSH 的英文全称是 Secure Shell，使用 SSH 时，双方所有传输的数据都是经过加密的，这样“中间人”这种攻击方式就不可能实现了，而且也能够防止 DNS 和 IP 欺骗。还有一个额外的好处就是传输的数据是经过压缩的，所以可以加快传输的速度。SSH 有很多功能，它既可以代替 telnet，又可以为 ftp、pop、甚至 ppp 提供一个安全的“通道”。

SSH 命令的一般形式为：

```
ssh 用户名@远程主机名/IP
```

其中“主机名/IP”是要连接的远程机的主机名或 IP 地址，用户名为远程主机系统已经存在的用户名，第一次登录时，会出现加密指纹信息的确认，一般情况下要选择 yes。之后会要求输入密码（默认采用当前用户名登录）。

SFTP 简介

传统 FTP 传输机制和实现原理上是没有考虑安全机制的，因为它们在网上用明文传送数据、用户帐号和用户口令，别有用心的非常容易地就可以截获这些数据、用户帐号和用户口令。而且，这些网络服务程序容易受到“中间人”（man-in-the-middle）这种攻击方式的攻击。所谓“中间人”的攻击方式，就是“中间人”冒充真正的服务器接收你的传给服务器的数据，然后再冒充你把数据传给真正的服务器。服务器和你之间的数据传送被“中间人”做了手脚之后，就会出现很严重的问题。

SFTP 是 SSH 提供的一个安全 FTP 传输功能，虽然 SFTP 提供了 FTP 的传输功能，但是在实现上 SFTP 并不依赖于 FTP，SFTP 是 SSH 下的一个子功能。

sftp 命令的使用方法基本上与 ssh 命令类似。

sftp 用户名@远程主机名/IP 地址

第 4 章 常用命令技巧

4.1 Linux 下软件包的安装及卸载

4.1.1 apt-get 命令

`apt-get autoclean apt`

会把已装或已卸的软件都备份在硬盘上，所以如果需要空间的话，可以让这个命令来删除你已经删掉的软件

`apt-get clean`

这个命令会把安装的软件的备份也删除，不过这样不会影响软件的使用的。

`apt-get upgrade`

更新所有已安装的软件包

`apt-get dist-upgrade`

将系统升级到新版本

4.1.2 apt-cache 命令

`apt-cache search string`

在软件包列表中搜索字符串

`apt-cache showpkg pkgs`

显示软件包信息。

`apt-cache dumpavail`

打印可用软件包列表。

`apt-cache show pkgs`

显示软件包记录。

`apt-cache pkgnames`

打印软件包列表中所有软件包的名称。

4.1.3 aptitude 命令

详细查看已安装或可用的软件包。与 `apt-get` 类似，`aptitude` 可以通过命令行方式调用，但仅限于某些命令——最常见的有安装和卸载命令。由于 `aptitude` 比 `apt-get` 了解更多信息，可以说它更适合用来进行安装和卸载。

4.1.4 dpkg 命令

虽然 `apt-get` 基于 `dpkg` 命令，并提供了更友好的命令接口，但是有时候也需要直接采用 `dpkg` 命令。比如某人或某组织把自己的程序做成了一个软件包，但并没有通知 Ubuntu 时，在分发时，其它用户可直接采用 `dpkg` 安装这个软件包。

如 `toolchain-mips-ls_0.1-1.deb` 这个软件包，在服务器上并没有关于它的信息，用户在安装时可以采用如下命令：

```
sudo dpkg -i toolchain-mips-ls_0.1-1.deb
```

4.2 Linux 操作系统的时钟系统

4.2.1 时钟系统介绍

对于 Linux 操作系统，主要有两种时钟系统：

1. 硬件时钟系统：该时钟系统的运行独立于任何系统控制程序，在 CPU 内部运行，即使系统断电，该时钟系统仍然保持运行。此时钟系统常常被称为实时时钟系统（RTC）、BIOS 时钟系统或 CMOS 时钟系统。
2. 系统时间：该时间为 Linux 内核的时钟系统所维护，并且被定时中断程序所驱动。该时间仅当 Linux 系统运行过程中有意义。该系统时间表示为以 UTC（通用协调时间，通用协调时间 Universal Time Coordinated）计算，自 1970 年 1 月 1 日 0 时 0 分 0 秒起至当前时刻所经过的秒数（该数字不是整数，以小数方式表示精度）。

在 Linux 系统中，真正起作用的是系统时间。硬件时钟系统的唯一作用是在 Linux 系统没有运行时，保持时间的运行。当 Linux 操作系统启动时，系统会自动将硬件时钟系统的时间同步到 Linux 系统时间，并且在之后的整个 Linux 运行过程中，不再使用该硬件时钟系统（这与 DOS 的实现方式完全不同，在 DOS 中，只有硬件时间系统的概念）。而在 Linux 操作系统关闭时，系统会自动将当前的 Linux 系统时间同步到硬件时钟系统所维护的时间。

4.2.2 硬件时钟系统与系统时间的同步

`/sbin/hwclock` 用于同步硬件时钟系统的时间与 Linux 操作系统时间。

查看当前硬件时钟系统时间的命令为：

```
/sbin/hwclock -show
```

将硬件时钟系统时间同步到 Linux 操作系统时间的命令为：

```
/sbin/hwclock -hctosys
```

将 Linux 操作系统时间同步到硬件时钟系统时间的命令为：

```
/sbin/hwclock -systohc
```

如果观察系统启动/关闭时执行的初始化脚本“`/etc/rc.d/rc.sysinit`”，我们不难发现：系统启动过程中，自动调用“`/sbin/hwclock --hctosys`”将硬件时钟系统时间同步到操作系统时间；而在系统关闭过程中，自动调用“`/sbin/hwclock --systohc`”将操作系统时间同步

到硬件时钟系统时间。

4.3 系统信息及运行状态监控

Linux 的文件系统中，有一个特殊目录“/proc”，该目录下列出的文件并非保存在磁盘上，而是内存中的一个映像。在该目录下可以找到许多有意思的东西，例如：

/proc/cpuinfo 本机 CPU 的相关信息；
/proc/meminfo 本机内存及交换分区的使用信息；
/proc/modules 本机已安装的硬件模块信息；
/proc/mounts 本机已挂载 mount 上的设备信息；

此外，该目录下有一些以数字为名称的子目录，每个子目录用于维护一个正在运行的进程，而目录名即为相应的进程 ID。例如，进入名称为“1”的子目录：

```
cd /proc/1
```

该目录下，文件“cmdline”中的内容为该进程运行时执行的命令行；“environ”文件中的内容为该进程运行时的环境变量信息；“stat”文件中存放的时该进程运行的状态信息等。

通过查看“/proc”目录下的文件，可以获取几乎所有的系统信息及系统的运行状态信息。事实上，Linux 下的诸多查询系统状态信息的命令，也是通过读取该目录下的文件而获取的。

以下是一些用于查看系统状态信息的命令：

4.3.1 df 命令

用于查看 Linux 文件系统的状态信息，显示各个分区的容量、已使用量、未使用量及挂载点等信息。如：

df -k 以千字节 KB 为单位显示各分区的信息；
df -a 显示所有分区 包括大小为 0 的分区；
df -T 显示分区类型（EXT2 或 EXT3 等）；

4.3.2 du 命令

用于查看文件或文件夹的大小。如：

```
du -b /home 以字节为单位显示“/home”文件夹下各个子文件夹的大小；
```

`du -ks home` 以千字节 (KB) 为单位显示“/home”文件夹的总大小;

4.3.3 top 命令

用于实时查看系统状态信息。运行该命令后，屏幕上会显示如下信息：

CPU 状态 (CPU 状态 CPU states)：包括用户进程占用比率、系统进程占用比率、用户的 `nice` 优先级进程占用比率及空闲 CPU 资源比率等。

内存状态 (Mem)：包括内存总量、使用量、空闲量等；

交换分区状态(Swap)：包括交换分区总量、使用量、空闲量等；

每个进程的状态：包括进程 ID、用户名、优先级、CPU 和内存使用量、以及运行该进程时执行的命令行；

执行 `top` 命令后，每隔 2 秒，系统会自动刷新一次状态信息。如果要指定刷新间隔，在启动时加入“-d” 参数即可。例如：

`top -d 1` 启动 `top` 并且每隔 1 秒刷新一次。

4.3.4 iostat 命令

统计并输出 CPU 使用信息及特定设备或分区的 IO 信息。

`iostat -c` 仅显示 CPU 使用情况统计信息。

`iostat -d` 仅显示设备/分区使用情况统计信息。

`iostat -d 2` 统计设备/分区使用情况，并每隔 2 秒刷新一次。

显示的 IO 统计信息中，各部分说明如下：

`tps` 设备每秒收到的 IO 传送请求数；

`Blk_read/s` 设备每秒读入的块数量；

`Blk_wrtn/s` 设备每秒写入的块数量；

`Blk_read` 设备读入的总块数量；

`Blk_wrtn` 设备写入的总块数量；

显示的 CPU 统计信息，各部分的含义同 `top`。

4.3.5 dmesg 命令

`dmesg` 命令可用于内核打印的信息，一般地，用于查看系统启动时是否运行正常

或者是否有错误发生。如在插入 U 盘后，运行 `dmesg`，可以看到内核处理 U 盘时的一些操作信息。

```
[12069.388231] usb 1-1: new high speed USB device using ehci_hcd and address 4
[12069.778211] usb 1-1: configuration #1 chosen from 1 choice
[12069.906044] scsi5 : SCSI emulation for USB Mass Storage devices
[12069.909638] usb-storage: device found at 4
[12069.909645] usb-storage: waiting for device to settle before scanning
[12074.957989] usb-storage: device scan complete
[12074.977721] scsi 5:0:0:0: Direct-Access      SanDisk  Cruzer Micro      8.01 PQ:
0 ANSI: 0 CCS
[12074.979577] sd 5:0:0:0: Attached scsi generic sg2 type 0
[12075.067990] sd 5:0:0:0: [sdc] 7862911 512-byte logical blocks: (4.02 GB/3.74 GiB)
[12075.114100] sd 5:0:0:0: [sdc] Write Protect is off
[12075.114116] sd 5:0:0:0: [sdc] Mode Sense: 45 00 00 08
[12075.114126] sd 5:0:0:0: [sdc] Assuming drive cache: write through
[12075.332421] sd 5:0:0:0: [sdc] Assuming drive cache: write through
[12075.332793]  sdc: sdc1
[12075.628798] sd 5:0:0:0: [sdc] Assuming drive cache: write through
[12075.629196] sd 5:0:0:0: [sdc] Attached SCSI removable disk
[12076.636202] FAT: bread failed in fat_clusters_flush
[12076.866756] FAT: bread failed in fat_clusters_flush
[12076.866930] FAT: bread failed in fat_clusters_flush
```

通过上面内核输出的信息，可以看到这个 U 盘在系统中可以使用。并且内核分配给 U 盘的设备名为 `sdc`，也可以看到这个 U 盘上只是存在一个分区，分区设备名为 `sdc1`。

4.4 归档/压缩工具 tar

`tar` 是 Linux 下最常用的打包、压缩工具，它在 Linux 中扮演类似于 WinZip 或 WinRAR 在 Windows 下的角色。下面是该工具的常用命令参数及运用举例：

主要参数：

- c 创建包
- x 解包
- t 列出包中的内容
- r 增加文件到指定包中
- u 更新包中的文件

可选参数：

- j 创建或解开包时 使用 `bzip2` 进行压缩或解压

- z 创建或解开包时 使用 gzip 进行压缩或解压
- Z 创建或解开包时 使用 compress 进行压缩或解压
- f 后面跟指定的包文件名
- v 显示打包/解包过程
- C 指定解包后的路径

例如:

```
tar -czf tmp.tar.gz /tmp 将"/tmp"目录压缩打包存放为 tmp.tar.gz;  
tar -xzf tmp.tar.gz -C /home 将 tmp.tar.gz 包解到"/home" 目录下;  
tar -t tmp.tar.gz 查看 tmp.tar.gz 包中的文件信息;
```

unzip 命令可以用于解压文件名后缀为.zip 的压缩文件

例如:

unzip archive.zip 在当前目录解压 archive.zip 文件

4.5 文件及文件内容的查找

Linux 下提供完善的文件及文件内容查找命令，分别介绍如下:

4.5.1 find 文件查找命令

find 命令用于查找文件系统中的指定文件，其命令格式为:

find 要查找的路径表达式

例如:

```
find . -name 1.txt 在当前目录及其子目录下查找文件"1.txt"。  
find /tmp -name 1.txt 在 /tmp 目录及其子目录下查找文件"1.txt"。
```

4.5.2 grep 文件查找命令

grep 命令用于查找指定的模式匹配，格式为:

grep [命令选项] 要查找的匹配模式 [要查找的文件]

例如:

```
grep cams test.txt 在 "test.txt"文件中查找 cams 字符串。  
grep -r cams /root/cams 在"/root/cams" 目录及其子目录下的所有文件中，查找  
cams 字符串;
```

grep 命令除了能够查找文件外，还能够将任意输出流重定向到 grep 进行查找:

`ps -ef | grep ora` 查找进程名中包含 **ora** 的所有进程信息。

4.5.3 xargs 命令

xargs 命令的作用是将参数列表转换成小块分段传递给其他命令，以避免参数列表过长的问题。**xargs** 一般情况下是跟其它命令一起使用的，他们之间通过管道的方式来传输参数。

常用命令方式：

```
find . -name "*.foo" | xargs grep bar
```

显示出在当前目录（包括子目录）下包括有 **bar** 字符串且文件名后缀为 **foo** 的所有文件。

```
ls *.foo | xargs grep bar
```

这个命令与上面命令的区别在于，它不查找子目录。

IO 重定向

I/O 重定向简单来说就是一个过程，这个过程捕捉一个文件，或者命令，程序，脚本，甚至脚本中的代码块（**code block**）的输出，然后把捕捉到的输出，作为输入发送给另外一个文件，命令，程序，或者脚本。

如果谈到 I/O 重定向，就涉及到文件标识符(**File Descriptor**)的概念，在 Linux 系统中，系统为每一个打开的文件指定一个文件标识符以便系统对文件进行跟踪，这里有些和 C 语言编程里的文件句柄相似，文件标识符是一个数字，不同数字代表不同的含义，默认情况下，系统占用了 3 个，分别是 0 标准输入（**stdin**）,1 标准输出(**stdout**), 2 标准错误(**stderr**)，另外 3-9 是保留的标识符，可以把这些标识符指定成标准输入，输出或者错误作为临时连接。通常这样可以解决很多复杂的重定向请求。

标准输入通常指键盘的输入

标准输出通常指显示器的输出

标准错误通常也是定向到显示器

举例：

```
#ls /dev 1>filename          #注意：“1”和“>”中间没有空格
```

把命令的标准输出重新定向到一个文件 **filename**，而不是显示到屏幕上，如果不指明文件标识符，系统默认的就是 1，因此 1 可以省略。

把上面例子重的 ">"改成">>"则表示把输出追加到 **filename** 文件的末尾，如果文件不存在则创建它。

如果在执行上面 **ls** 命令的时候，出现了错误，错误信息还是打印到屏幕上，如果

想把错误信息也打印到文件里，可以采用下面的方式：

```
ls -qw /dev 1>stdout_file 2>stderr_file
```

在 C 语言编程时，打印信息时一般采用 `printf`，`printf` 输出的信息在默认情况下是打印到屏幕上的。

`printf()` 与 `sprintf(stdout,)` 是等价的，都是打印到标准输出里。

如果程序里有 `sprintf(stderr,)`，那么在执行程序时加上 `2>stderr_file` 的参数，它的错误信息将会打印到 `stderr_file` 这个文件里。

附录 1

TBD.