

嵌入式 Linux 系统快速入门指南

目录

第 1 章命令行操作及常用命令使用.....	1
1.1 Shell 简介.....	2
1.2 Shell 控制台使用方法.....	3
1.3 登录和退出 Linux 系统.....	4
1.4 文件和目录操作命令.....	5
1.5 用户及用户组管理命令.....	6
1.6 进程及任务管理命令.....	7
1.7 磁盘及文件系统管理命令.....	8
1.8 软件安装命令.....	9
第 2 章文件查看/编辑方法介绍.....	10
2.1 cat 命令介绍.....	11
2.2 vi 命令介绍.....	12
第 3 章网络配置及操作.....	13
3.1 网络配置命令.....	14
3.2 网络操作命令.....	15
3.3 常用网络配置文件介绍.....	16
3.4 常用网络操作方法.....	17
第 4 章常用使用技巧.....	18
4.1 Linux 下软件包的安装及卸载.....	19
4.2 Linux 操作系统的时钟系统.....	20
4.3 系统信息及运行状态监控.....	21
4.4 打包/压缩工具 tar.....	22
4.5 文件及文件内容的查找.....	23
4.6 Linux 的系统服务及其配置.....	24

第 1 章命令行操作及常用命令使用

本章目录

- Shell 简介
- Shell 控制台使用方法

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 Kornshell 中的优点，有灵活和强大的编程接口同时又有很友好的用户界面。

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 为喜欢使用 C shell 的人提供了 Tcsh。Tcsh 是 C shell 的一个扩展版本。Tcsh 包括命令行编辑、可编程单词补全、拼写校正、历史命令替换、作业控制和类似 C 语言的语法。它不仅和 Bash shell 是提示符兼容而且还提供比 Bashshell 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，Redhat Linux 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)。

由于 bash 是 Redhat Linux 上缺省的 shell，本文将主要介绍 bash 及其相关知识。

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，例如 dos。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。例如 dos 命令就是控制台命令。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，用户又可以看见” 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

文章目录

- . 登录和退出 **Linux** 系统
- . 文件和目录操作命令
- . 用户及用户组管理命令
- . 进程及任务管理命令
- . 磁盘及文件系统管理命令
- . 软件安装命令

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可进入系统.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 快速关机重新启动时跳过 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

1.2.2.1 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“/”。

1.2.2.2 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 如果直接输入, 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。

1.2.2.3 mkdir rmdir

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

1.2.2.4 cp

cp 这个命令相当于 dos 下面的 copy 命令。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。

1.2.2.5 rm

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 text 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，敲了“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。

1.2.2.6 mv

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。

1.2.2.7 cat

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。

1.2.2.8 more

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

1.2.2.9 pwd

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

1.2.3.1 useradd

useradd 命令可以创建一个新的用户帐号,其最基本用法为:

useradd 用户名

如输入以下命令:

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:

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。

1.2.3.2 userdel

userdel 命令用于删除一个已存在的帐号，其用法为：

userdel 用户名

1.2.3.3 groupadd

groupadd 命令可以创建一个新的用户组，其最基本用法为：

groupadd 组名

如输入以下命令：

groupadd newgroup

系统将创建一个新的用户组 newgroup。

1.2.3.4 groupdel

groupdel 命令用于删除一个已存在的用户组，其用法为：

groupdel 组名

1.2.3.5 passwd

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：

passwd 用户名

1.2.3.6 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

1.2.3.7 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：

1、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。

2、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

3 `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
```

1.2.3.8 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。

1.2.4.1 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：

PID TTY TIME CMD

```
12039 pts/0 00:00:00 bash
```

```
18710 pts/0 00:00:00 ps
```

该命令显示当前登录用户的执行进程。显示的项目共分为四项，依次为 PID(进程 ID)、TTY(终端名称)、TIME(进程执行时间)、CMD(该进程的命令行输入)。

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。

1.2.4.2 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

1.2.4.3 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

(2)

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

1.2.4.4 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。

1.2.5.1 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。

1.2.5.2 mount 和 umount

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] 设备目录
```

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 /mnt/cdrom
```

```
mount /mnt/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。

1.2.6 软件安装命令

1.2.6.1 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。

1.2.6.2. rpm

rpm 命令的功能有很多，在这里我们只介绍如何利用 rpm 安装和查询已安装的软件。

1 安装

rpm 安装软件包的基本语法如下：

rpm -i rpm 包名

例如：

rpm -i tomcat4.rpm

该命令将安装 Tomcat 软件并进行相关系统配置。

2 升级

rpm 升级软件包的基本语法如下：

`rpm -U rpm 包名`

例如：

```
rpm -U tomcat4.rpm
```

该命令将升级 Tomcat 软件并进行相关系统配置。

3、查看已安装的软件

有的时候，了解系统中都已经安装了哪些软件包以及它们的用途是很有用的，RPM 的查询参数就可以做到这一点。

要想列出已经安装的全部软件包，可以通过以下命令：

```
rpm -qa
```

可以使用 `grep` 命令指定软件包的名称（或者部分名称）如下所示：

```
rpm -qa | grep 'tomcat'
```

该命令将列出包名中包含 `tomcat` 的软件包。

此外：要想找出某个特定的文件到底是属于哪个软件包的，可以输入：

```
rpm -qf filename
```

其中 `filename` 是准备要查找其归属的文件名称。

要想查出某个已经安装的软件包的功能，可以用如下命令：

```
rpm -qi packagename
```

其中，`packagename` 是要查找其用途的软件包名称。

4、删除已安装的软件

使用 RPM 删除软件包就像安装它们一样简单。大多数情况下，用户只需要输入命令：

```
rpm -e packagename
```

其中，`ackagename` 是要查删除的软件包名称。

第 2 章文件查看/编辑方法介绍

文章目录:

- . cat 命令介绍
- . vi 命令介绍
- . vi 的进入与退出
- . vi 中显示行号
- . 光标移动操作
- . 文本插入操作
- . 文本修改操作
- . 文本搜索与替换操作

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 Edit 程序在 DOS 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。

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。

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。

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）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。

`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。

`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。

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，则光标下移 n 个字符；

0（零）光标移到行首；

\$ 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；

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

2.

vi 提供了两个插入命令：**i** 和 **I**。

(1) **.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。

(2) **.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。

3. 附加（append）命令

4.

vi 提供了两个附加插入命令：**a** 和 **A**。

(1) **.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。

(2) **.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。若我们希望在某日之前或某年之后插入一些新行，则应使用 `open` 命令。

`vi` 提供了两个打开命令：`o` 和 `O`。

(1) `.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。

(2) `.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。

文本修改操作

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 (Undo)

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。

5. 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。

文本搜索与替换操作

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您需要搜索的模式。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。

第 3 章 网络配置及操作

文章目录

- Ifconfig
- Route

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。

Ifconfig

ifconfig 的语法为：

```
ifconfig [-a] [-V] [-s] []  
  
]  
[[-]broadcast [  
  
]]  
  
[netmask  
  
]  
  
[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，netmask 表示设置子网掩码，其后的 address 为子网掩码的 IP，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 子网掩码)
```

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
```

3) 查看第一块网卡的状态:

```
ifconfig eth0
```

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:

有几个状态比较重要:

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;

Route

route 命令的语法为:

```
route [-CFvnee]
```

```
route [-v] [-A family] add [-net|-host] target [netmask Nm] [gw Gw] [metric N] [mod] [reinstat
```

```
[[dev] If]
```

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

```
route [-V] [--version] [-h] [--help]
```

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) 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
```


2) 添加一条缺省路由

```
route add 0.0.0.0 gw 网关地址
```

如果没有其他匹配的路由项，就使用这个路由规则。

3) 省路由

```
route del 0.0.0.0 gw 网关地址
```

4) 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。

3.2 网络操作命令

文章目录

- netstat
- nslookup
- host 命令
- finger
- Ping 命令

Linux 系统常用的网络操作命令包括 netstat、nslookup、host finger 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; finger 命令的功能是查询用户的信息,通常会显示系统中某个用户的用户名、主目录、停滞时间、登录时间、登录 shell 等信息。ping 命令用于查看网络上的主机是否在工作。

Netstat

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

....

Nslookup

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 并回车即可。

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
```

finger

finger 命令的功能是查询用户的信息，通常会显示系统中某个用户的用户名、主目录、停滞时间、登录时间、登录 shell 等信息。如果要查询远程机上的用户信息，需要在用户名后面接“@主机名”采用[用户名@主机名]的格式，不过要查询的网络主机需要运行 finger 守护进程。该命令的一般格式为：

finger [选项] [使用者] [用户@主机]

finger 的常用参数如下：

-s 显示用户的注册名、实际姓名、终端名称、写状态、停滞时间、登录时间等信息。

-l 除了用-s 选项显示的信息外，还显示用户主目录、登录 shell、邮件状态等信息，以及用户主目录下的.plan 、project 和.forward 文件的内容

-p 除了不显示.plan 文件和.project 文件以外与-l 选项相同

示例：

1) 显示用户 oracle 的信息

```
finger oracle
```

其显示结果为

```
Login: oracle Name: (null)
```

Directory: /home/oracle Shell: /bin/bash

On since Fri Feb 6 16:26 (CST) on pts/1 from 10.153.21.166 33 seconds idle

On since Wed Feb 4 20:10 (CST) on pts/2 from 10.153.21.166 1 second idle

No mail.

No Plan.

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 64ICMP 数据字节

3.3 常用网络配置文件介绍

文章目录

- ./etc/sysconfig/network 网络设置
- ./etc/hosts IP 地址和主机名的映射

- ./etc/services
- ./etc/host.conf 配置名字解析器
- ./etc/nsswitch.conf 配置名字解析器
- ./etc/resolv.conf 配置 DNS 客户
- ./etc/xinetd.conf
- ./etc/modules.conf 文件
- ./etc/sysconfig/network-scripts/ifcfg-ethN 文件

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/sysconfig/network

/etc/hosts

/etc/services

/etc/host.conf

/etc/nsswitch.conf

/etc/resolv.conf

/etc/xinetd.conf

/etc/modules.conf

/etc/sysconfig/network-scripts/ifcfg-ethN

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。

/etc/sysconfig/network 网络设置

该文件用来指定服务器上的网络配置信息，包含了控制和网络有关的文件和守护程序的行为的参数。下面是一个例子文件：

NETWORKING=yes

HOSTNAME=machine1

GATEWAY=210.34.6.2

FORWARD_IPV4=yes

GATEWAYDEV=

其中， NETWORK=yes/no 表示网络是否被配置；

HOSTNAME=hostname hostname 表示服务器的主机名；

GATEWAY=gw-ip gw-ip 表示网络网关的 IP 地址；

FORWARD_IPV4=yes/no 是否开启 IP 转发功能；

GAREWAYDEV=gw-dev gw-dw 表示网关的设备名，如： eth0 等；

/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 otheralias。一旦配置完机器的网络配置文件，应该重新启动网络以使修改生效，使用下面的命令来重新启动网络：

```
service network restart
```

/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。

/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,下面是 RedHat 安装时缺省的/etc/services 中的前几行:

```
tcpmux 1/tcp # TCP port service multiplexer
```

```
echo 7/tcp
```

```
echo 7/udp
```

```
discard 9/tcp sink null
```

```
discard 9/udp sink null
```

```
systat 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。

/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。地址下面是 RedHat 安装后缺省的“/etc/host.conf”内容:

```
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。

“rccorder” 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。

/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 文件的名字。下面是可以被包含的关键字：

aliases 邮件别名；

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；

/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：

```
nameserver 192.168.0.36
```

```
nameserver 192.168.0.37
```

search huawei-3com.com

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。**Red Hat** 中没有提供缺省的 `/etc/resolv.conf` 文件,它的内容是根据在安装时给出的选项动态创建的。

`/etc/xinetd.conf`

redhat 7.1 及以后版本计和以往的版本有一个明显的区别,就是用 `xinetd.conf` 代替原来的 `inetd.conf`,并且直接使用了 `firewall` 服务。`xinetd`(eXtended InterNET services daemon)对 `inetd` 功能进行了扩展,`xinetd` 的默认配置文件是 `/etc/xinetd.conf`,它看起来尽管和老版本的 `/etc/inetd.conf` 完全不同,其实只是以一个脚本的形式将 `inetd` 中每一行指定的服务扩展为一个 `/etc/xinetd.d/` 下的配置文件,其格式为:

service service-name

{

disabled = yes/no; //是否禁用

socket_type = xxx; //TCP/IP socket type,such as stream,dgram,raw,....

protocol = xxx; //服务使用的协议

server = xxx; //服务 daemon 的完整路径

server_args = xxx; //服务的参数

port = xxx; //指定服务的端口号

wait = xxx; //是否阻塞服务即单线程或多线程

user = xxx; //服务进程的 uid

group = xxx; //gid

REUSE = xxx; //可重用标志

.....

}

示例:

1) 配置 ftp 服务

在/etc/xinetd.d 目录下, 编辑 wu-ftpd

```
# default: on
```

```
# description: The wu-ftpd FTP server serves FTP connections. It uses
```

```
# normal, unencrypted usernames and passwords for authentication.
```

```
service ftp
```

```
{
```

```
disable = no
```

```
socket_type = stream
```

```
wait = no
```

```
user = root
```

```
server = /usr/sbin/in.ftpd
```

```
server_args = -l -a
```

```
log_on_success += DURATION
```

```
nice = 10
```

```
}
```

2) 配置 telnet 服务

在/etc/xinetd.d 目录下, 编辑 telnet

```
# default: on
```

```
# description: The telnet server serves telnet sessions; it uses
```

```
# unencrypted username/password pairs for authentication.
```

```
service telnet

{

disable = no

flags = REUSE

socket_type = stream

wait = no

user = root

server = /usr/sbin/in.telnetd

log_on_failure += USERID

}
```

配置完成后重新启动服务

```
#/etc/rc.d/init.d/xinetd restart
```

/etc/modules.conf 文件

该配置文件定义了各种需要在启动时加载的模块的参数信息,文件格式如下:

```
alias eth0 tg3
```

```
alias eth1 3c905
```

/etc/sysconfig/network-scripts/ifcfg-ethN 文件

系统网络设备的配置文件保存在” /etc/sysconfig/network-scripts” 目录下, ifcfg-eth0 包含第一块网卡的配置信息, ifcfg-eth1 包含第二块网卡的配置信息。

下面是 “/etc/sysconfig/network-scripts/ifcfg-eth0” 文件的示例:

```
DEVICE=eth0
```

```
ONBOOT=yes
```

BOOTPROTO=static

IPADDR=192.168.0.66

NETMASK=255.255.255.0

GATEWAY=192.168.0.65

若希望手工修改网络地址或在新的接口上增加新的网络界面，可以通过修改对应的文件（ifcfg-ethN）或创建新的文件来实现。

3.4 常用网络操作方法

文章目录

· ftp 远程文件传输

· telnet

· r-系列命令

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

r - 使用各种远程命令。

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

示例:

使用 ftp 传输文件过程如下

```
[root@localhost root]# ftp 192.168.1.66
```

```
Connected to 192.168.1.66 (192.168.1.66).
```

```
220 localhost.localdomain FTP server (Version wu-2.6.2-5) ready.
```

```
Name (192.168.1.66:root): test
```

```
331 Password required for test.
```

```
Password:
```

```
230 User oracle logged in. Access restrictions apply
```

```
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
```

```
ftp> ls
```

```
227 Entering Passive Mode (192.168.1.66,228,168)
```

```
150 Opening ASCII mode data connection for directory listing.
```

```
total 312
```

-rw-r--r-- 1 root root 1215 Feb 4 13:49 button1.jpg

-rw-r--r-- 1 root root 1133 Feb 4 13:49 button2.jpg

-rw-r--r-- 1 root root 1435 Feb 4 13:46 index.html

226 Transfer complete.

ftp> lcd

Local directory now /root

ftp> lcd tmp

Local directory now /root/tmp

ftp> bin

200 Type set to I.

ftp> mget *.jpg

mget button1.jpg? y

227 Entering Passive Mode (10,153,89,66,187,198)

150 Opening BINARY mode data connection for button1.jpg (1215 bytes).

226 Transfer complete.

1215 bytes received in 0.0119 secs (1e+02 Kbytes/sec)

mget button2.jpg? y

227 Entering Passive Mode (10,153,89,66,243,218)

150 Opening BINARY mode data connection for button2.jpg (1133 bytes).

226 Transfer complete.

1133 bytes received in 0.00223 secs (5e+02 Kbytes/sec)

ftp> quit

221-You have transferred 2348 bytes in 2 files.

221-Total traffic for this session was 3315 bytes in 3 transfers.

221-Thank you for using the FTP service on localhost.localdomain.

221 Goodbye.

telnet

用户使用 telnet 命令进行远程登录。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 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。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 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：

```
# telnet 192.168.1.66
```

```
Trying 192.168.1.66...
```

```
Connected to 192.168.1.66.
```

Escape character is '^'.

Red Hat Linux release 7.3 (Valhalla)

Kernel 2.4.18-3 on an i686

login: test

Password:

Last login: Mon Feb 9 15:03:56 from 10.153.21.166

[test@localhost test]\$

r-系列命令

除 `ftp` 和 `telnet` 以外，还可以使用 r-系列命令访问远程计算机和在网络上交换文件。使用 r-系列命令需要特别注意因为如果用户不小心，就会造成严重的安全漏洞。用户发出一个 r-系列命令后，远程系统检查名为 `/etc/hosts.equiv` 的文件，以查看用户的主机是否列在这个文件中，如果它没有找到用户的主机，就检查远程机上同名用户的主目录中名为 `rhosts` 的文件，看是否包括该用户的主机。如果该用户的主机包括在这两个文件中的任何一个之中，该用户执行 r-系列命令就不用提供口令。

rlogin 命令

rlogin 是“remote login”（远程登录）的缩写。该命令与 `telnet` 命令很相似，允许用户启动远程系统上的交互命令会话。rlogin 的一般格式是：

```
rlogin [ -8EKLdx ] [ -e char ] [-k realm ] [ -l username ] host
```

一般最常用的格式是：

```
rlogin host
```

rsh 命令

rsh 是“remote shell”（远程 shell）的缩写。该命令在指定的远程主机上启动一个 shell 并执行用户在 rsh 命令行中指定的命令。如果用户没有给出要执行的命令，rsh 就用 rlogin 命令使用户登录到远程机上。rsh 命令的一般格式是：

```
rsh [-Kdnx] [-k realm] [-l username] host [command]
```

一般常用的格式是：

rsh host [command]

command 可以是 shell 提示符下键入的任何 Linux 命令。

第 4 章常用使用技巧

- 4.1 Linux 下软件包的安装及卸载
- 4.2 Linux 操作系统的时钟系统
- 4.3 系统信息及运行状态监控
- 4.4 打包/压缩工具 tar
- 4.5 文件及文件内容的查找
- 4.6 Linux 的系统服务及其配置

4.1 Linux 下软件包的安装及卸载

文章目录

- . RPM 安装操作
- . RPM 查询操作
- . RPM 卸载操作
- . RPM 升级操作
- . RPM 验证操作
- . RPM 的其他附加命令

在 Linux 操作系统下，几乎所有的软件均通过 RPM 进行安装、卸载及管理等操作。RPM 的全称为 Redhat Package Manager，是由 Redhat 公司提出的，用于管理 Linux 下软件包的软件，Linux 安装时，除了几个核心模块以外，其余几乎所有的模块均通过 RPM 完成安装。RPM 有五种操作模式，分别为：安装、卸载、升级、查询和验证。

RPM 安装操作

命令：

rpm -i 需要安装的包文件名

举例如下：

rpm -i example.rpm 安装 example.rpm 包：

rpm -iv example.rpm 安装 example.rpm 包，并在安装过程中显示正在安装的文件信息；

rpm -ivh example.rpm 安装 example.rpm 包，并在安装过程中显示正在安装的文件信息及安装进度；

RPM 查询操作

命令:

`rpm -q ...`

附加查询命令:

a 查询所有已经安装的包

以下两个附加命令用于查询安装包的信息:

i 显示安装包的信息;

l 显示安装包中的所有文件被安装到哪些目录下;

s 显示安装版中的所有文件状态及被安装到哪些目录下;

以下两个附加命令用于指定需要查询的是安装包还是已安装后的文件;

p 查询的是安装包的信息;

f 查询的是已安装的某文件信息;

举例如下:

`rpm -qa | grep tomcat4` 查看 tomcat4 是否被安装;

`rpm -qip example.rpm` 查看 example.rpm 安装包的信息;

`rpm -qif /bin/df` 查看/bin/df 文件所在安装包的信息;

`rpm -qlf /bin/df` 查看/bin/df 文件所在安装包中的各个文件分别被安装到哪个目录下;

RPM 卸载操作

命令:

`rpm -e` 需要卸载的安装包

在卸载之前, 通常需要使用 `rpm -q ...` 命令查出需要卸载的安装包名称。

举例如下:

`rpm -e tomcat4` 卸载 tomcat4 软件包

RPM 升级操作

命令:

`rpm -U` 需要升级的包

举例如下:

`rpm -Uvh example.rpm` 升级 `example.rpm` 软件包

RPM 验证操作

命令:

`rpm -V` 需要验证的包

举例如下:

`rpm -Vf /etc/tomcat4/tomcat4.conf`

输出信息类似如下:

```
S.5....T c /etc/tomcat4/tomcat4.conf
```

其中,S 表示文件大小修改过,T 表示文件日期修改过,限于篇幅,更多的验证信息请您参考 rpm 帮助文件 `man rpm`

RPM 的其他附加命令

- `force` 强制操作(如强制安装、删除等);
- `requires` 显示该包的依赖关系;
- `nodeps` 忽略依赖关系并继续操作;

4.2 Linux 操作系统的时钟系统

文章目录

- . 时钟系统介绍
- . 硬件时钟系统与系统时间的同步

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，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。
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

文章目录

.df 命令

.du 命令

.top 命令

.iostat 命令

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：

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）。

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；

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 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 5 秒，系统会自动刷新一次状态信息，如果要指定刷新闻隔，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。

进入 **top** 后，按下如下按钮会有相应功能：

【空格】 立即刷新显示信息；

【M】 按内存使用量排序；

【N】 按进程 ID 排序；

【P】 按 CPU 使用量排序；

【c】 扩展命令行信息， 显示完整的命令行；

【?】 显示帮助信息；

【q】 退出 **top** 程序。

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。

4.4 打包/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:

主要命令:

-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

可选命令:

-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;

4.5 文件及文件内容的查找

文章目录

- . 文件查找命令 `find`
- . 文件内容查找命令 `grep`

Linux 下提供完善的文件及文件内容查找命令，分别介绍如下：

文件查找命令 `find`

`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”。

文件内容查找命令 `grep`

`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。

4.6 Linux 的系统服务及其配置

文章目录

- . 服务简介
- . 启动和停止服务
- . 服务的添加、删除、自动运行及状态

服务简介

Linux 系统服务是在 Linux 启动时自动加载，并在 Linux 退出时自动停止的系统任务。在 Linux 启动过程中，我们可以看得很多“starting ...”提示信息，该信息表示正在启动系统服务；而在 Linux 退出时，我们也能够看到相应的“Stopping ...”信息，表示系统服务正在停止。Linux 自身有许多系统任务，如“network”服务用于支持网络连接等。CAMS 在安装过程中，会自动加入三个系统任务，以保证在 Linux 启动时，CAMS 服务自动启动；而在 Linux 关闭时，CAMS 服务也自动停止。这三个服务分别是：

oracled Oracle 数据库服务

camsd CAMS 协议处理模块服务

tomcat4 Tomcat Web Server 服务，用于支持 CAMS 配置管理台、自助服务平台、Portal 服务平台及内容计费认证平台等；

Linux 系统的所有服务程序脚本均可以在“/etc/rc.d/init.d”文件夹中找到，每一项服务对应一个脚本文件。如果 CAMS 已经安装，我们可以在该文件夹下找到“oracled”、“camsd”和“tomcat4”这三个文件。

启动和停止服务

service 命令用于启动及停止某个服务，例如：

service camsd stop 停止 camsd 服务

service oracled start 启动 oracled 服务

service tomcat4 start 启动 tomcat4 服务

CAMS 引入的三个服务间有依赖关系，即 camsd 服务和 tomcat4 服务依赖与 oracled 服务，在 oracled 服务启动前不能启动 camsd 服务和 tomcat4 服务；

而在 oracled 服务停止前必须先停止 camsd 服务和 tomcat4 服务。

服务的添加、删除、自动运行及状态

CAMS 在安装过程中会自动添加相关的服务，在安装的最后过程中，会提示用户是否启动服务（使服务生效）。需要注意的是：一个服务被添加后，并不表示该服务在系统启动过程中会自动运行，仅表示可以使用 service 命令来启动、停止服务。

1. 添加服务

2.

添加服务的步骤为：

(1) 将该服务的脚本文件拷入 “/etc/rc.d/init.d” 文件夹下，保证其状态为 “可执行”；

(2) 使用 `chkconfig` 命令将服务添加到服务列表中：

```
chkconfig - add camsd （注意有两个“-”）
```

将服务添加以后， 该服务即可以使用 `service` 来启动/停止了。

2. 删除服务

如果要删除某项服务将上述命令的 “add” 改为 “del” 即可：

```
chkconfig - del camsd （注意有两个“-”）
```

3. 设置服务自动运行

Linux 运行有 7 种运行级别，分别使用 0-6 来表示，控制台界面运行级别为 3，图形界面运行级别为 5，要使某服务在系统启动时自动运行，使用 `chkconfig` 将该服务添加到相应级别的自启动服务列表中即可。

```
chkconfig - level 35 camsd on （注意有两个“-”）
```

上面的命令将 `camsd` 服务添加到级别为 3、5 的自启动服务列表中；而下面的命令则将服务 `camsd` 从级别为 3、5 的自启动服务列表中删除：

```
chkconfig - level 35 camsd off （注意有两个“-”）
```

如果需要查看某项服务的自启动状态信息， 使用如下命令即可：

```
chkconfig - list camsd （注意有两个“-”）
```

凌阳教育嵌入式培训网 (<http://www.sunplusedu.com>) 提供更多[嵌入式学习](#)教程！