

# 第4章 Linux字符界面操作

4.1

字符界面简介

4.2

在Linux下获取帮助

4.3

Shell基础

4.4

Shell实用功能

4.5

vi编辑器基本使用

对Linux服务器进行管理时，经常需要进入字符界面进行操作，使用命令需要记住该命令的相关选项和参数。vi编辑器可以用于编辑任何ASCII文本，它功能非常强大，可以对文本进行创建、查找、替换、删除、复制和粘贴等操作。

## 4.1 字符界面简介

对Linux服务器的维护基本上都是在命令行界面下进行的，本节主要讲述如何进入命令行界面以及在命令行界面下如何关机、重启计算机。

## 4.1.1 进入Linux命令行界面

**Fedora 17**系统的命令行界面可以通过字符界面、图形界面下的终端以及虚拟控制台等多种方式进入。

# 1. Linux字符界面

安装Fedora 17系统之后，系统启动默认进入的是图形化界面，可以通过使用如下命令修改为进入字符界面：

```
[root@PC-LINUX ~]# ln -sf  
/lib/systemd/system/runlevel3.target  
/etc/systemd/system/default.target
```

所做改变在系统重新引导之后即可生效。

如果用户选择使用字符界面登录Fedora 17系统，在系统被引导后，会看到如图4-1所示的登录提示。

```
Fedora release 17 (Beefy Miracle)
Kernel 3.3.4-5.fc17.i686.PAE on an i686 (tty2)

PC-LINUX login:
```

Linux系统用户登录分两步：第一步输入用户的用户名，系统根据该用户名识别用户；第二步输入用户的口令。当用户正确地输入用户名和口令后，就能合法地进入系统，屏幕会显示如图4-2所示信息。这时就可以对系统进行各种操作了，注意超级用户登录后提示符是“#”，而其他用户登录后提示符是“\$”。

```
Fedora release 17 (Beefy Miracle)
Kernel 3.3.4-5.fc17.i686.PAE on an i686 (tty2)

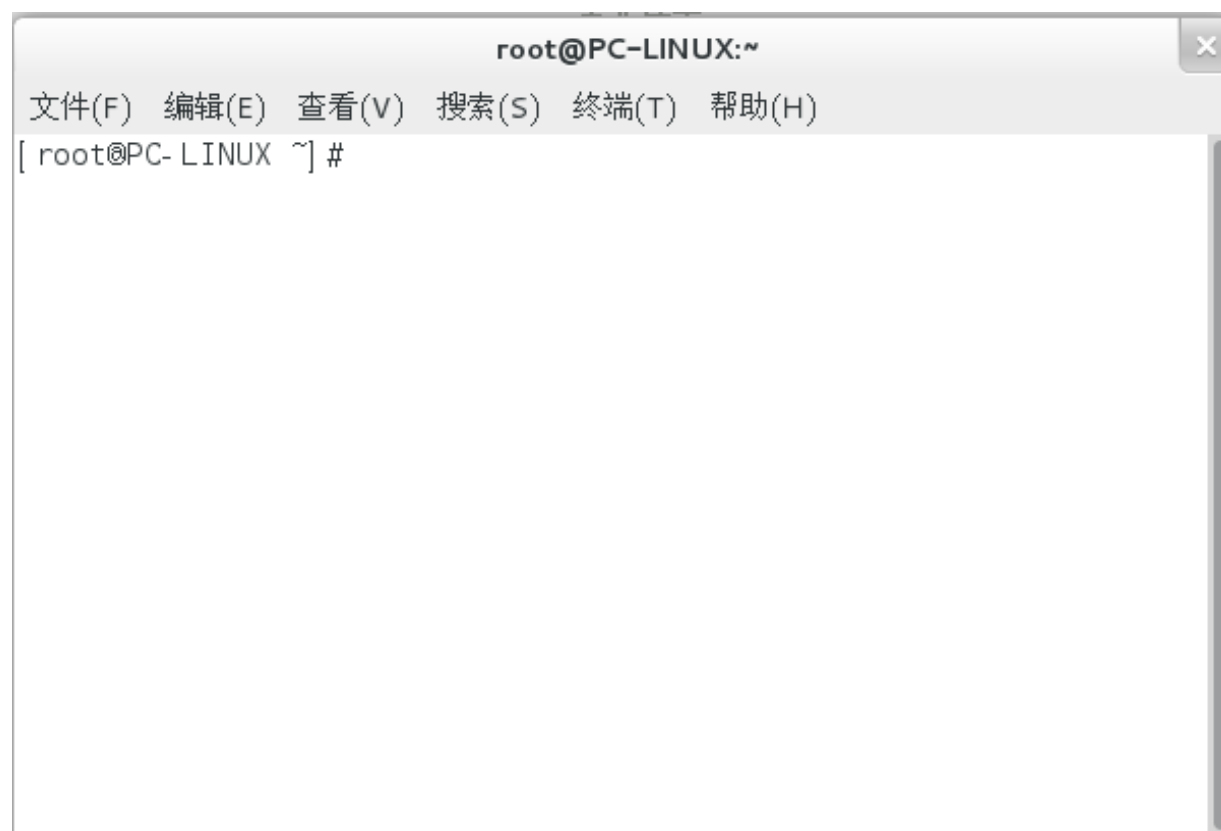
PC-LINUX login: root
Password:
Last login: Sun Jun  3 21:04:26 on tty2
[root@PC-LINUX ~]#
```



## 2. 使用终端

Fedora 17系统桌面环境提供了打开终端命令行界面的方式，终端方式允许用户通过输入命令来管理计算机。

选择面板上的“活动”→“应用程序”→“系统工具”→“终端”，打开如图4-3所示终端界面。



在终端命令行界面中可以直接输入命令并执行，执行的结果显示在终端界面中。如果需要修改终端的设置，如常规、标题和命令、颜色、效果以及滚动等，在终端菜单栏中选择“编辑”→“配置文件首选项”，打开如图4-4所示界面进行修改。



### 3. 使用虚拟控制台

Fedora 17系统允许多个用户同时登录，还允许用户进行多次登录，这是因为Linux系统提供了虚拟控制台的访问方式。

在字符界面下，虚拟控制台的切换可以通过按下[Alt+ F1]~ [Alt+ F6]键来实现。

在图形界面下，可以使用[Ctrl+Alt+F2]~[Ctrl+Alt+F6]键切换不同的字符虚拟控制台，再使用[Ctrl+Alt+F1]可以切换回图形界面。

## 4.1.2 关闭和重启系统

在Linux系统下常用的关闭和重启系统的命令有“shutdown”，“halt”，“reboot”和“init”，但每个命令的内部工作过程是不同的。

# 1. “shutdown”命令

使用“shutdown”命令可以安全地关闭系统，有些用户会使用直接断掉电源的方式来关闭计算机，这是十分危险的。Linux系统与Windows系统不同，其后台运行着许多进程，所以强制关机可能会导致进程的数据丢失，使系统处于不稳定的状态，甚至会损坏某些系统的硬件设备。

## 命令语法:

shutdown [选项] [时间] [警告信息]

【例4.1】 立即关闭系统。

```
[root@PC-LINUX ~]# shutdown -h now
```

【例4.2】 定时45分钟后关闭系统。

```
[root@PC-LINUX ~]# shutdown -h 45
```

```
Shutdown scheduled for Sun, 03 Jun 2012  
06:29:00 +0800, use 'shutdown -c' to cancel.
```



### 【例4.3】 重新启动系统，并发出警告信息。

```
[root@PC-LINUX ~]# shutdown -r now  
"system will be reboot now."
```

Broadcast message from root@PC-LINUX on  
pts/1 (Sun, 03 Jun 2012 05:44:52 +0800):

system will be reboot now.

## 2. “halt”命令

使用“halt”命令就是调用“shutdown -h”命令执行关闭系统。

命令语法：

halt [选项]

【例4.4】 使用“halt”命令关闭系统。

```
[root@PC-LINUX ~]# halt
```

```
Broadcast message from root@PC-LINUX on pts/0 (Sun, 03  
Jun 2012 21:07:15 +0800):
```

```
The system is going down for system halt NOW!
```

### 3. “reboot”命令

“reboot”命令的工作过程与“halt”命令相似，不过“reboot”是引发主机重启，而“halt”是引发主机关闭系统。它的选项与“halt”相似。

**【例4.5】** 使用reboot命令重启计算机。

```
[root@PC-LINUX ~]# reboot
```

```
Broadcast message from root@PC-LINUX on pts/1 (Sun, 03  
Jun 2012 05:36:21 +0800):
```

```
The system is going down for reboot NOW!
```

## 4. “init”命令

“init”命令是所有进程的祖先，它的进程号始终为“1”，所以发送“TERM”信号给“init”会终止所有的用户进程和守护进程等。“shutdown”命令就是使用这种机制。“init”定义了7个运行级别，其中“init 0”为关闭系统，“init 6”为重启。

## 4.1.3 系统运行级别

Linux系统运行级别是一种状态，它由列在/etc/rc.d/rc<x>.d目录中的服务来定义，其中<x>是代表运行级别的符号。

Linux运行级别有如下7种。

0: 停止运行，所有进程中止，关闭系统。

1: 单用户模式，用于维护系统，只有少数进程运行。

2: 多用户模式，除了NFS服务没有启动外，其他和运行级别3一样。

3: 完整的多用户模式，进入Linux系统的字符界面。

- 4: 没有使用（可由用户定义）。
- 5: 完整的多用户模式（带有基于X Window的图形界面）。
- 6: 重新引导计算机。

## 【例4.6】 显示当前系统运行级别。

```
[root@PC-LINUX ~]# runlevel
```

```
N 3
```

//系统当前运行级别为3，没有上一次运行级别（用N表示）

## 【例4.7】 切换运行级别为2，并显示当前系统运行级别。

```
[root@PC-LINUX ~]# init 2
```

```
[root@PC-LINUX ~]# runlevel
```

```
3 2
```

//系统当前运行级别为2，上一次的运行级别为3



## 4.2 在Linux下获取帮助

Linux系统中的每个命令都具有众多的参数和选项，要一一记住那是不可能的，可借助Linux的帮助功能。本节主要介绍在Linux系统下如何使用帮助功能。

## 4.2.1 使用man手册页

man是一种显示Unix/Linux在线手册的命令。可以用来查看命令、函数或文件的帮助手册，另外它还可以显示一些gzip压缩格式的文件。

man命令格式化并显示在线的手册页。

命令语法:

man [选项] [命令名称]

【例4.10】 显示ls命令的man手册页。

[root@PC-LINUX ~]# man ls

LS(1) User Commands LS(1)

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILES (the current directory by default).  
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all

do not ignore entries starting with .

-A, --almost-all

do not list implied . and ..

--author

Manual page ls(1) line 1 (press h for help or q to quit)

表4-1 man手册页组成内容

手册页内容	说 明
Header	标题
NAME	man的命令/函数的功能概述
SYNOPSIS	man的命令/函数用法的简单描述
AVAILABILITY	可用性说明
DESCRIPTION	man的命令/函数的详细描述
OPTIONS	该命令的所有可选项的详细说明
RETURN VALUE	如果是函数，则列出函数返回值
ERRORS	如果函数调用出错，则列出所有出错的值和可能引起错误的原因
FILES	该命令/函数所用到的相关系统文件
ENVIRONMENT	和该命令/函数相关的环境变量
NOTES	表示不常用的用法或实现的细节
BUGS	已知的错误和警告
HISTORY	该命令/函数的历史发展
SEE ALSO	可以参照的其他的相关命令/函数
Others	和一些具体命名/函数有关的特殊信息



表4-2                    man手册页类型

类     型	含     义
1	一般使用者的命令
2	系统调用的命令
3	C语言函数库的命令
4	驱动程序和系统设备的有关解释
5	配置文件的解释
6	游戏程序的命令
7	其他的软件或是程序的命令
8	有关系统维护的命令

## 4.2.2 使用help命令

使用help命令可以查找Shell命令的用法，只需在所查找的命令后输入“help”命令，然后就可以看到所查命令的内容了。

命令语法：

[命令] --help

【例4.11】 查看mkdir命令帮助。

```
[root@PC-LINUX ~]# mkdir --help
```

## 4.2.3 whereis命令

使用whereis命令可以查找命令所在的位置。

命令语法:

whereis [选项] [命令名]

**【例4.12】** 查找ls命令在什么目录下。

```
[root@PC-LINUX ~]# whereis ls
```

```
ls: /bin/ls /usr/bin/ls /usr/share/man/man1p/ls.1p.gz  
/usr/share/man/man1/ls.1.gz
```



## 4.3 Shell基础

在Linux系统中，Shell是最常使用的程序，其主要作用是侦听用户指令、启动指令所指定的进程并将结果返回给用户，本节主要讲述Shell的基本使用方法。

## 4.3.1 Shell起步

在AT&T工作的Dennis Ritchie和Ken Thompson两人设计UNIX的时候，他们想要为用户创建一种与他们的新系统交流的方法。那时的操作系统带有命令解释器。命令解释器接受用户的命令，然后解释它们，因而计算机可以使用这些命令。

当登入系统或打开一个终端窗口时，首先看到的是Shell提示符。

以普通用户zhangsan登入名为PC-LINUX的主机，当前工作目录是/home/zhangsan，如下所示：

```
[zhangsan@PC-LINUX ~]$
```

以root用户登录系统的提示符如下所示：

```
[root@PC-LINUX ~]#
```

要运行命令的话，只需要在提示符后敲进命令，然后再按“回车”键。

一个Shell命令可能含有一些可选项和参数，其一般格式为：

[Shell命令][可选项][参数]

Linux系统由以下4个重要的部分组成：

- (1) 内核；
- (2) Shell；
- (3) 文件系统；
- (4) 应用程序。

内核部份，操作者不易和它直接沟通，因此，必须要有一个友善的界面，使得操作时能更为方便，这个界面便是Shell。通俗地讲，Shell就是位于核心和操作者之间的一层使用者界面。

Shell的本意是“壳”的意思。在核心的外面，包覆着一层外壳，用来负责接收使用者输入的指令，然后将指令解译成核心能够了解的方式，传给核心去执行，再将结果传回至预设的输出周边设备。

## 4.3.2 Shell类型和功能

Linux系统中的Shell，除了可作为命令编译器之外，它也是一种不错的编程语言，是系统管理维护时的重要工具。

由于Linux系统对Shell的处理，采用独立自由开放的方式，因此，Shell的种类相当多，目前流行的Shell有sh，csh，ksh，tcsh和bash等。



在Linux系统中的bash具有以下功能：

- (1) 兼容Bourne Shell (sh)；
- (2) 包含C Shell以及Korn Shell中最好的功能；
- (3) 具有命令列编写修改的能力；
- (4) 具有工作控制的能力，可控制前台和后台程序；
- (5) 具有Shell编程能力。

## 4.3.3 开始使用bash

当用户登录到Linux系统后，一旦出现提示符号时，便可以开始输入操作命令了。命令可分为如下两大类：

- (1) bash内置的命令；
- (2) 应用程序。

# 1. 通配符

表4-3          bash中使用的通配符

符 号	含 义
?	代表任何单一字符
*	代表任何字符
[字符组合]	在中括号中的字符皆符合，如：[a~z]代表所有的小写字母
[!字符组合]	不在中括号中的字符皆符合，如：[!0~9]代表非数字的皆符合

## 2. 特殊字符及引号

表4-4 特殊字符及引号

符号	意 义	符号	意 义
~	用户主目录		管道
`	命令取代	?	通配符，代表任何单一字符
#	解释	<	输入重定向
\$	变量取值	>	输出重定向
&	后台进程工作	'	单引号（不具有变数置换的功能）
*	通配符，代表任何字符	"	双引号（具有变数置换的功能）
(	子Shell开始	/	路径分隔符
)	子Shell结束	;	命令分隔符
\	使命令持续到下一行		

## 3. 常用控制组合键

表4-5            常用控制组合键

组合键	意      义
<b>Ctrl+C</b>	终止目前的命令
<b>Ctrl+\</b>	终止目前的命令
<b>Ctrl+D</b>	输入结束，即EOF之意（如使用mail信件结束时），或注销Linux
<b>Ctrl+Z</b>	暂停目前的命令
<b>Ctrl+M</b>	相当按“Enter”键
<b>Ctrl+S</b>	暂停屏幕输出
<b>Ctrl+Q</b>	恢复屏幕输出
<b>Ctrl+U</b>	将命令列整列删除

## 4.4 Shell实用功能

Linux系统是在命令行下面诞生的，因此，Linux中的命令行有许多非常实用的功能，在本节中将详细讲解这些实用功能。

## 4.4.1 命令行自动补全

在Linux系统中有太多的命令和配置文件，可以使用命令补全功能快速地写出文件名和命令名称。

如果需要快速地从当前所在的目录跳转到 /usr/src/redhat/ 目录，可以执行以下操作：

```
[root@PC-LINUX ~]# cd  
/u<Tab>/sr<Tab>/r<Tab>
```

<Tab>是按“Tab”键的意思，使用“Tab”键也称为“命令行自动补全”，这在平常应用中是不可缺少的。

## 4.4.2 命令历史记录

通过按向上方向键，可以向后遍历最近在  
该控制台下输入的命令。用向下方向键可以向  
前遍历命令。与“shift”键连用的话，还可  
以遍历以往在该控制台中的输出。也可以编辑  
旧的命令，然后再运行。



# 1. 编辑命令行

通过光标和功能键（Home，End等键），可以浏览并编辑命令行，还可以用键盘的快捷方式来完成一般的编辑，主要功能键如下。

[Ctrl+k]：删除从光标到行尾的部分。

[Ctrl+u]：删除从光标到行首的部分。

[Alt+d]：删除从光标到当前单词结尾的部分。

[Ctrl+w]：删除从光标到当前单词开头的部分。

[Ctrl+a]：将光标移到行首。

[Ctrl+e]: 将光标移到行尾。

[Alt+a]: 将光标移到当前单词头部。

[Alt+e]: 将光标移到当前单词尾部。

[Ctrl+y]: 插入最近删除的单词。

[!\$]: 重复前一个命令最后的参数。

【例4. 13】 使用命令行记录功能键。

```
[root@PC-LINUX ~]# mkdir /root/aaa
```

//创建目录/root/aaa

```
[root@PC-LINUX ~]# cd !$
```

```
cd /root/aaa
```

//!\$是指重复前一个命令最后的参数，参数是/root/aaa

```
[root@PC-LINUX aaa]# pwd
```

```
/root/aaa
```

//显示用户当前目录是/root/aaa

## 2. 可用的Shell快捷方式

Linux系统带有不少快捷方式，其中一部分是bash原来就有的，还有一些则是预先设置的。由于主目录是每位用户的活动中心，许多Linux系统对此有特殊的快捷方式。

“~”就是用户的主目录的简写形式。

## 4.4.3 命令的排列、替换和别名

在Shell中可以使用命令排列同时执行多个命令，可以使用命令替换将一个命令的输出当作另一个命令的输入，也可以将复杂命令定义别名。

# 1. 命令的排列

如果希望一次执行多个命令，Shell允许在不同的命令之间，放上特殊的排列字符。这里将介绍最常用的两种方法。

(1) 使用 “;”

使用 “;” 命令时先执行命令1，不管命令1是否出错，接下来就执行命令2。

命令语法：

命令1; 命令2

**【例4.14】** 使用排列命令 “;” 同时执行两个命令。

```
[root@PC-LINUX ~]# ls -l /boot;du -hs /root
```

总用量 24131

```
-rw-r--r--. 1 root root 116892 5月 8 01:43 config-3.3.4-5.fc17.i686.PAE
drwxr-xr-x. 2 root root 1024 6月 2 23:47 grub
drwxr-xr-x. 6 root root 1024 6月 3 01:17 grub2
-rw-r--r--. 1 root root 17716383 6月 3 01:13 initramfs-3.3.4-5.fc17.i686.PAE.img
drwx-----. 2 root root 12288 6月 2 22:39 lost+found
-rw-----. 1 root root 1914110 5月 8 01:43 System.map-3.3.4-5.fc17.i686.PAE
-rw-r--r--. 1 root root 228484 1月 17 00:08 tboot.gz
-rw-r--r--. 1 root root 9220 1月 17 00:08 tboot-syms
-rwxr-xr-x. 1 root root 4696896 5月 8 01:43 vmlinuz-3.3.4-5.fc17.i686.PAE
6.6M /root
```

//先在屏幕上列出/boot目录中的所有内容，然后列出目录/root及其子目录所占磁盘大小




## (2) 使用 “&&”

使用 “&&” 命令时只有当命令1正确运行完毕后，才能执行命令2。

命令语法：

命令1&&命令2

**【例4.15】** 使用排列命令 “&&” 同时执行两个命令。



```
[root@PC-LINUX ~]# ls -a /root/bogusdir&&du -hs
```

ls: 无法访问/root/bogusdir: 没有那个文件或目录

//将返回“ls: 无法访问/root/bogusdir: 没有那个文件或目录”，  
而“du -hs”命令根本没有运行，因为没有/root/bogusdir目录

## 2. 命令替换

命令替换这种机制的语法如下。

命令1 \$(命令2)

除了使用“\$( )”之外，还可以使用后引号“`”，命令语法如下所示。

命令1 `命令2`

例如：

Pidof less //查看less的进程号，假如为4163

Kill -9 4163

或者kill -9 \$(pidof less)

或者kill -9 `pidof less`

### 3. 命令别名

在需要执行某一非常长的命令时，所有的命令以及命令的选项、参数都要一一输入，很枯燥也容易出现错误。可以为常用命令定义快捷方式，这些快捷方式可以用较简单的命令别名来定义。

命令语法：

`alias [别名]=[需要定义别名的命令]`

## 4.4.4 文件名匹配和输出重定向

### 1. 文件名匹配

文件名匹配使得用户不必一一写出文件名就可以指定多个文件。

“\*”可匹配一个或多个字符。

第二个通配符是问号“?”。在匹配时，一个问号只能代表一个字符。

## 2. 管道

Linux系统的理念是汇集许多小程序，每个程序都有特殊的专长。复杂的任务不是由大型软件完成，而是运用Shell的机制，组合许多小程序共同完成。管道就在其中发挥着重要的作用，它可以将某个命令的输出信息当作某个命令的输入，由管道符号“|”来标识。

命令语法：

[命令1]| [命令2]| [命令3]

【例4.16】 使用简单的管道。

```
[root@PC-LINUX ~]# ls /etc|more
```

```
abrt
```

```
acpi
```

```
adjtime
```

```
akonadi
```

```
alchemist
```

```
aliases
```

```
aliases.db
```

```
alsa
```

```
alternatives
```

```
anacrontab
```

```
ant.conf
```

```
ant.d
```

```
anthy-conf
```

```
asound.conf
```

```
at.deny
```

```
atmsigd.conf
```

```
at-spi2
```

```
audisp
```

```
audit
```

```
avahi
```

```
BackupPC
```

```
bash_completion.d
```

```
bashrc
```

```
--More--
```

/命令ls /etc显示/etc目录的内容，命令more是分页显示内容

### 【例4. 17】 使用复杂的管道。

```
[root@PC-LINUX ~]# rpm -qa|grep a|more
libquvi-scripts-0.4.4-1.fc17.noarch
lxappearance-0.5.1-1.fc17.i686
un-extra-vada-fonts-1.0.2-0.14.080608.fc17.noarch
ar9170-firmware-2009.05.28-4.fc17.noarch
icedax-1.1.11-10.fc17.i686
gnome-packagekit-3.4.0-1.fc17.i686
setup-2.8.48-1.fc17.noarch
java-1.7.0-openjdk-1.7.0.3-2.1.fc17.6.i686
abrt-addon-python-2.0.10-3.fc17.i686
liberation-narrow-fonts-1.07.2-3.fc17.noarch
libqalculate-0.9.7-4.fc17.i686
gstreamer-plugins-espeak-0.3.5-2.fc17.i686
yum-langpacks-0.2.4-3.fc17.noarch
perl-Carp-1.22-2.fc17.noarch
apr-util-ldap-1.4.1-2.fc17.i686
farstream-0.1.2-2.fc17.i686
gnome-search-tool-3.4.0-1.fc17.i686
gnu-free-fonts-common-20100919-5.fc17.noarch
xml-commons-resolver-1.2-8.fc17.noarch
un-extra-pilgia-fonts-1.0.2-0.14.080608.fc17.noarch
atkmm-2.22.6-1.fc17.i686
matchbox-window-manager-1.2-9.20070628svn.fc17.i686
suitesparse-3.6.1-3.fc17.i686
--More--
```

//命令rpm -qa显示已经安装在系统上的RPM包，命令grep a是过滤软件包，命令more是分页显示这些信息



### 3. 重定向文件

有时，希望将命令的输出结果保存到文件中，或以文件内容作为命令的参数，这时就需要用到重定向。重定向不使用系统的标准输入端口、标准输出端口或标准错误端口，而进行重新的指定。重定向有5种方式，分别是：输出重定向、输入重定向、错误重定向、追加重定向以及同时实现输出和错误的重定向。

## (1) 输出重定向

输出重定向，即将command命令的输出保存到文件file中，如果存在相同的文件，则覆盖file文件中的内容。

命令语法：

```
command > file
```

【例4.18】 使用输出重定向将目录/boot的内容保存到文件/root/abc。

```
[root@PC-LINUX /]# ls /boot > /root/abc
```

```
[root@PC-LINUX /]# cat /root/abc
```

```
config-3.3.4-5.fc17.i686.PAE
```

```
grub
```

```
grub2
```

```
initramfs-3.3.4-5.fc17.i686.PAE.img
```

```
lost+found
```

```
System.map-3.3.4-5.fc17.i686.PAE
```

```
tboot.gz
```

```
tboot-syms
```

```
vmlinuz-3.3.4-5.fc17.i686.PAE
```

```
//查看文件/root/abc的内容
```

【例4.19】 使用echo命令和输出重定向创建文本文件/root/mm，内容是hello。

```
[root@PC-LINUX /]# echo hello > /root/mm
```

```
[root@PC-LINUX /]# cat /root/mm
```

```
hello
```

//显示文件/root/mm，可以看到文件的内容是hello

(2) 输入重定向  
输入重定向，即将文件file的内容作为  
command命令的输入。

命令语法；

`command < file`

【例4.20】 使用输入重定向将文件/root/mm的内容作为输入让cat命令执行。

```
[root@PC-LINUX ~]# cat < /root/mm
```

```
hello
```

//可以看到文件/root/mm的内容是hello

### （3）错误重定向

错误重定向，即将command命令执行的出错信息输出到文件file中。

命令语法：

`command 2> file`

【例4.21】 查看根本不存在的/root/kk文件，出现报错信息，将其保存到文件/root/b中。

```
[root@PC-LINUX ~]# cat /root/kk 2> /root/b
```

```
[root@PC-LINUX ~]# cat /root/b
```

```
cat: /root/kk: 没有那个文件或目录
```

//使用cat命令查看/root/b文件，可以看到其内容就是执行命令cat /root/kk的报错信息

## （4）追加重定向

追加重定向，即将command命令执行的输出添加到已存在的文件file中。

命令语法：

```
command >> file
```

【例4.22】 使用追加重定向将数据写入文件/root/a。

```
[root@PC-LINUX ~]# echo aaa>/root/a
```

//先创建文件/root/a，文件内容是aaa

```
[root@PC-LINUX ~]# echo bbb>>/root/a
```

//向文件/root/a中追加数据bbb

```
[root@PC-LINUX ~]# cat /root/a
```

aaa

bbb



( 5 ) 同时实现输出和错误的重定向  
同时实现输出和错误的重定向，即可以同时实现输出重定向和错误重定向的功能。

命令语法：

`command &>file`

【例4. 23】 同时使用输出和错误重定向。

```
[root@PC-LINUX ~]# ls /boot &> /root/kk
```

```
[root@PC-LINUX ~]# cat /root/kk
```

```
config-3.3.4-5.fc17.i686.PAE
```

```
grub
```

```
grub2
```

```
initramfs-3.3.4-5.fc17.i686.PAE.img
```

```
lost+found
```

```
System.map-3.3.4-5.fc17.i686.PAE
```

```
tboot.gz
```

```
tboot-syms
```

```
vmlinuz-3.3.4-5.fc17.i686.PAE
```

//因为/boot目录下有文件，所以最终使用了输出重定向

【例4.24】 同时使用输出和错误重定向。

```
[root@PC-LINUX ~]# ls /nn &> /root/oo
```

```
[root@PC-LINUX ~]# cat /root/oo
```

ls: 无法访问/nn: 没有那个文件或目录

//因为没有/nn目录，所以最终使用了错误重定向

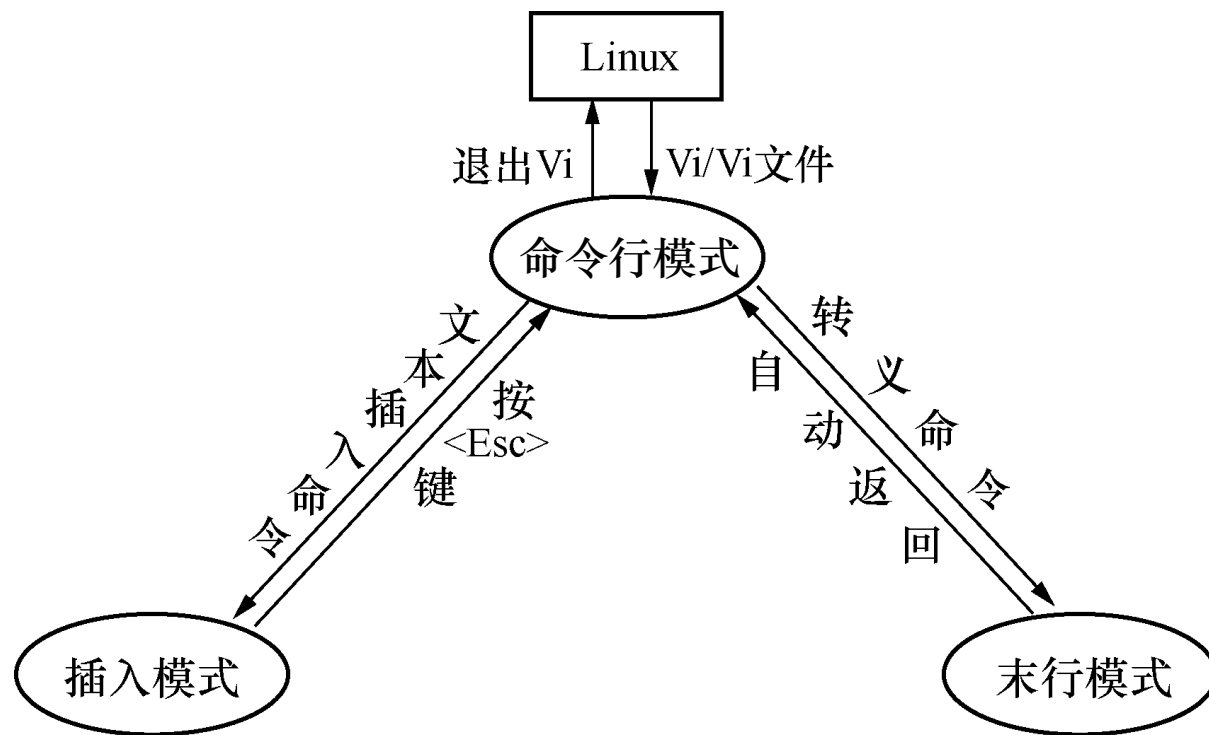
## 4.5 vi编辑器基本使用

文本编辑器有很多，图形模式下有gedit, kwrite等编辑器，文本模式下的编辑器有vi, vim（vi的增强版本）和nano。vi和vim是Linux系统中最常用的编辑器，本节主要讲述vi编辑器的使用。

## 4.5.1 vi的基本概念

vi编辑器是所有Linux系统的标准编辑器，用于编辑任何ASCII文本，对于编辑源程序尤其有用。它功能非常强大，通过使用vi编辑器，可以对文本进行创建、查找、替换、删除、复制和粘贴等操作。

vi编辑器有3种基本工作模式，分别是命令行模式、插入模式和末行模式。



vi的3种模式

# 1. 命令行模式

控制屏幕光标的移动，字符、字或行的删除，移动、复制某区域及进入插入模式，或者到末行模式。

## 2. 插入模式

只有在插入模式下，才可以做文字输入，按“Esc”键可回到命令行模式。



### 3. 末行模式

将文件保存或退出vi编辑器，也可以设置编辑环境，如寻找字符串、列出行号等。

## 4.5.2 vi的基本操作

本小节主要讲述vi编辑器的基本使用，如使用vi编辑器创建文件、输入数据以及保存文件并退出。

## 1. 进入vi编辑器

在系统Shell提示符下输入vi及文件名称后，就进入vi编辑画面。

## 2. 切换至插入模式编辑文件

在命令行模式下按字母“I”就可以进入插入模式，这时候就可以开始输入文字了。

### 3. 退出vi及保存文件

在命令行模式下，按冒号键“:”可以进入末行模式。

输入“wq”，存盘并退出vi。

输入“q!”，不存盘强制退出vi。

## 4.5.3 命令行模式操作

当在Shell提示符下输入“vi 文件名”之后就进入了命令行模式，在命令行模式下是不能输入任何数据的。

# 1. 进入插入模式

进入插入模式的操作如下。

按“i”：从光标当前位置开始输入文件。

按“a”：从目前光标所在位置的下一个位置开始输入文字。

按“o”：在光标所在的行的下面插入一行，从行首开始输入文字。

按“I”：在光标所在行的行首插入。

按“A”：在光标所在行的行末插入。

按“O”：在光标所在的行的上面插入一行。

按“s”：删除光标位置的一个字符，然后进入插入模式。

按“S”：删除光标所在的行，然后进入插入模式。



## 2. 从插入模式切换为命令行模式

从插入模式切换为命令行模式只需按“Esc”键。

### 3. 移动光标

vi可以直接用键盘上的光标来上下左右移动，但正规的vi是用小写英文字母“h”，“j”，“k”及“l”，分别控制光标左、下、上、右移一格。

按[Ctrl+b]：屏幕往前移动一页。

按[Ctrl+f]：屏幕往后移动一页。

按[Ctrl+u]：屏幕往前移动半页。

按[Ctrl+d]：屏幕往后移动半页。

按数字“0”：光标移到所在行的行首。

按“G”：移动到文本的最后一行。

按“\$”：移动到光标所在行的行尾。

按“^”：移动到光标所在行的行首。

按“w”：光标跳到下个字的开头。

按“e”：光标跳到下个字的字尾。

按“b”：光标回到上个字的开头。

## 4. 删除文字

删除文字操作如下。

“x”：每按一次，删除光标所在位置的一个字符。

“nx”：例如，“6x”表示删除光标所在位置开始的6个字符。

“X”：大写的X，每按一次，删除光标所在位置的前面一个字符。

“nX”：例如，“20X”表示删除光标所在位置前面20个字符。

“dd”：删除光标所在行。

“ndd”：从光标所在行开始删除n行。例如，“4dd”表示删除从光标所在行开始的4行字符。

## 5. 复制

复制操作如下。

“yw”：将光标所在之处到字尾的字符复制到缓冲区中。

“nyw”：复制n个字到缓冲区。

“yy”：复制光标所在行到缓冲区。

“nyy”：例如，“6yy”表示复制从光标所在行开始的6行字符。

“p”：将缓冲区内的内容写到光标所在位置。

## 6. 替换

替换操作如下。

“r”：替换光标所在处的字符。

“R”：替换光标所到之处的字符，直到按下“Esc”键为止。

## 7. 撤销上一次操作

“u”：如果误执行一个命令，可以马上按下“u”，回到上一个操作。按多次“u”可以执行多次撤销。



## 8. 跳至指定的行

跳至指定行的操作如下。

[Ctrl+g]：列出光标所在行的行号。

“nG”：例如，“15G”，表示移动光标至该文本的第15行行首。

## 9. 存盘退出

“ZZ”：存盘退出。

## 10. 不存盘退出

“ZQ”：不存盘退出。

## 4.5.4 末行模式操作

在使用末行模式之前，请记住先按“Esc”键确定已经处于命令行模式后，再按冒号“:”即可进入末行模式。

# 1. 列出行号

“set nu”：输入“set nu”后，会在文件中的每一行前面列出行号。

## 2. 取消列出行号

“set nonu”：输入“set nonu”后，会取消在文件中的每一行前面列出行号。

### 3. 跳到文件中的某一行

“n”：“n”号表示一个数字，在冒号后输入一个数字，再按“回车”键就会跳到该行了，如输入数字15，再按“回车”键就会跳到文本的第15行。

## 4. 查找字符

查找字符操作如下。

“/关键字”：先按“/”键，再输入想查找的字符，如果第一次查找的关键字不是想要的，可以一直按“n”会往后查找下一个关键字。

“?关键字”：先按“?”键，再输入想查找的字符，如果第一次查找的关键字不是想要的，可以一直按“n”往前查找下一个关键字。



## 5. 运行Shell命令

“!命令”：运行Shell命令。

## 6. 替换字符

替换字符操作如下。

“s /a/b”：将光标所在行第一个a替换为b。

“s /a/b/g”：将光标所在行所有的a替换为b。

## 7. 保存文件

“w”：在冒号后输入字母“w”就可以将文件保存起来。

## 8. 退出vi编辑器

离开vi编辑器操作如下。

“q”：按“q”即退出vi，如果无法离开vi，可以在“q”后跟一个“!”强制符离开vi。

“wq”：一般建议离开时，搭配“w”一起使用，这样在退出的时候还可以保存文件。

## 小 结

要进入**Fedora 17**系统的命令行界面可以通过字符界面、图形界面下的终端以及虚拟控制台等多种方式进入。在**Linux**系统下常用的关机/重启命令有**shutdown**，**halt**，**reboot**和**init**，但每个命令的内部工作过程是不同的。

# 小 结

Linux系统中的每个命令都具有众多的参数和选项，要一一记住那是不可能的。使用man手册页是一种不错的方法，man是一种显示Unix/Linux在线手册的命令。可以用来查看命令、函数或文件的帮助手册，另外它还可以显示一些gzip压缩格式的文件。

## 小 结

使用help命令可以查找Shell命令的用法，只需在所查找的命令后输入“help”命令，然后就可以看到所查命令的内容了。使用whereis命令可以查找命令所在的位置。例如，我们最常用的ls命令，是存放在/bin这个目录下的。

## 小 结

在Linux系统中，Shell是最常使用的程序，其主要作用是侦听用户指令、启动指令所指定的进程并将结果返回给用户，一个Shell命令可能含有一些可选项和参数。Shell的种类相当多，目前流行的Shell有sh，csh，ksh，tcsh和bash等。



# 小 结

Linux系统是在命令行下面诞生的，因此，Linux系统中的命令行有许多非常实用的功能，主要有命令行自动补全、命令历史记录、命令的排列、命令替换、命令别名、文件名匹配、管道、重定向文件等。

# 小 结

vi编辑器是所有Linux系统的标准编辑器，用于编辑任何ASCII文本，对于编辑源程序尤其有用。它功能非常强大，通过使用vi编辑器，可以对文本进行创建、查找、替换、删除、复制和粘贴等操作。vi编辑器有3种基本工作模式，分别是命令模式、插入模式和末行模式。