

第3章

Linux 基本操作

Linux 是一个多用户、多任务的操作系统，允许多个用户同时登录。为了给用户提供更好的服务，方便用户操作，本章将介绍 Linux 的基本操作。本章讲解的内容主要包括 GNOME 图形界面、Bash 命令基本操作、Linux 系统常用命令、Vim 文本编辑器、Linux 系统中获取帮助等。

3.1 GNOME 图形界面基础

Linux 是一个基于命令行的操作系统，最初图形界面并不属于 Linux，但因 Linux 的广泛应用，也为了用户操作方便，进而发行了 Linux 中的图形界面，称之为 X Window 界面。图形界面有多种，比如 GNOME、KDE、Unity 界面等。因 GNOME 界面的使用和 Windows 系统的界面使用类似，故本节只做简单介绍，学习者可自行探究。

GNOME 是一个功能完善、操作简单、界面友好的桌面环境，它由面板、桌面、系统图标、图形化的文件管理等一些用于对系统进行设置和管理的实用程序组成。启动 GNOME 后，会出现如图 3.1 所示的界面。GNOME 和 Windows 系统界面的最大区别就是开始菜单和一些基本的信息设置位于界面的上方。



图 3.1 GNOME 界面

GNOME 界面包含以下几部分。

(1) 面板：桌面环境上方的区域，通过这些区域可以访问所有的系统应用程序和菜单。

(2) 桌面：位于桌面环境中的所有其他组件之后。桌面是用户界面的活动组件，将对象放在桌面上可以快速访问文件和目录，或启动常用的应用程序。若有已执行的各种程序，程序则显示在桌面上。要打开一个文件夹或启动一个程序，双击相应图标即可。

(3) 资源管理器：双击用户桌面上的“主文件夹”图标即可打开资源管理器，在它的浏览器窗口中包含文件夹和文件，可以使用鼠标拖动或放置到新位置，基本操作与 Windows 系统中类似。

综上所述，GNOME 界面与 Windows 界面的操作类似，最上面的一行叫作面板，相当于 Windows 的任务栏，面板显示打开的窗口或应用程序；中间部分是桌面，可在桌面上添加快捷方式，如图 3.2 所示的三个应用程序的图标是系统默认添加的。系统启动后在桌面上单击鼠标右键，也可以弹出快捷菜单，供用户使用。



图 3.2 GNOME 基本图形界面

Linux 安装的软件也可在左上角的“应用程序”中找到，如图 3.3 所示为系统自带的程序。

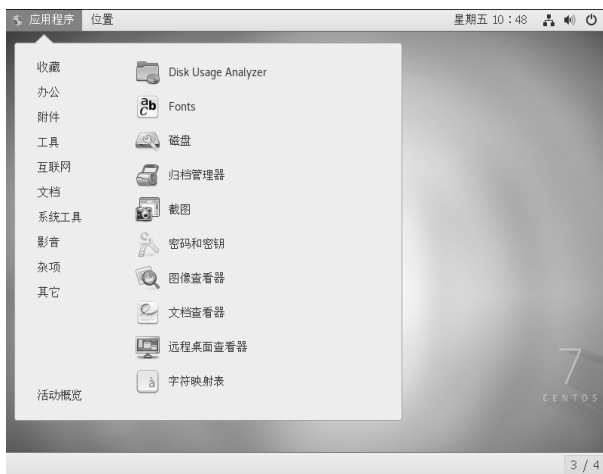


图 3.3 应用程序

虽然图形界面操作起来很直观，但 Linux 还是依靠输入命令的方式进行系统操作的。可在应用程序中打开输入命令的“终端”，或在桌面上直接单击鼠标右键，在弹出的快捷菜单中选择“打开终端”命令，如图 3.4 所示。



图 3.4 打开终端

3.2 Bash 基础及命令组成



3.2.1 Shell 简介

Bash 基础

Shell 从英文字面上看是“壳”的意思，Shell 在 Linux 中俗称“壳”，它处于 Linux Kernel 的外围，就像一个“壳”一样，如图 3.5 所示。Linux Kernel 承担的任务是接受上层的服务指令，传达给操作系统内核，并实现硬件控制等。但用户无法直接控制 Linux Kernel，即用户不允许直接与 Linux Kernel 进行通信，此时需借助一个外围的“壳”，就是 Shell。Shell 所起的是中间媒介的作用，即将用户需求转换成 Linux Kernel 能够识别的指令。但 Shell 本身其实只是一个概念，其中的操作是依靠图形或命令模式来实现的。

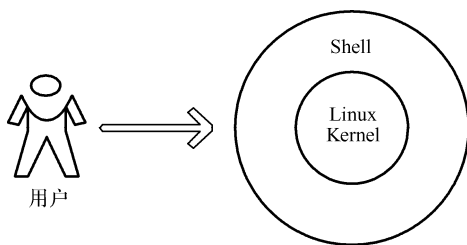


图 3.5 Shell 示意图

Shell 是操作系统最外面的一层，用来管理用户与操作系统之间的交互，等待用户输入，向操作系统解释用户的输入，并处理各种各样的操作系统输出结果。Shell 提供了用户与操作系统之间的通信方式，这种通信可以以交互方式（从键盘输入，并立即得到响应）或以非交互方式（Shell Script）执行。Shell Script 是存放在文件中的一串 Shell 和操作系统命令，可被重复使用，Shell Script 本质上是命令行，将命令简单组合到一个文件里。Shell 基本上是一个命令解释器，类似于 DOS 下的 Command，它接收用户命令（如 ls 等），然后调用相应的应用程序。较为通用的 Shell 有标准的 Bourne Shell (sh) 和 C Shell (csh)。

3.2.2 Shell 分类

Shell 可以分为两大类：图形界面 Shell 和命令行式 Shell。

(1) 图形界面 Shell (Graphical User Interface Shell, GUI Shell)，例如上一节提到的 GNOME、KDE 等。图形界面的最终使用也需转换成命令行模式去执行。

(2) 命令行式 Shell (Command Line Interface Shell, CLI Shell)，这类 Shell 有很多种，最常用的为 GNU Bourne-Again Shell，简称 Bash，也就是 Linux 中常用的命令行。

3.2.3 Bash 基础

Bash 是许多 Linux 平台的内定 Shell，它运行于大多数类 UNIX 操作系统之上。简单地说，就是用户必须通过 Shell 将输入的命令与内核进行通信，从而让内核控制硬件精准高效地完成工作。

Bash 本身也是一个命令模式下的命令，其含义是打开命令模式终端。下面详细解读一下 Bash 命令窗口中各个选项的具体含义。

1. 提示符

Linux 操作系统下的 Bash 提示符分为“#”与“\$”两种，其中“#”代表超级用户 root，“\$”代表普通用户。使用超级用户 root 登录系统，仅仅说明其权限比较大，但任何操作都可能对系统带来灾难性错误。虽然本书中有些实例是在 root 登录后实现的，但是强烈建议初学者使用普通用户登录系统。

【实例】 通过系统环境验证 Bash 提示符。

```
[linuxstudy@LinuxServer ~]$ bash
[linuxstudy@LinuxServer ~]$ su - root
密码:
上一次登录: 三 5月 15 10:56:42 CST 2019pts/0 上
```

说明：su 命令的作用是切换用户，其后可直接加用户名，也可加参数。由 root 切换到普通用户不需要使用密码，其他用户均需知道对方密码的情况下方可切换。

2. 结构解读

```
[root@LinuxServer ~]#
```

上述 Bash 结构中，root 代表的是当前登录系统的用户名称；LinuxServer 代表当前使用的计算机主机名称；@代表使用 root 用户登录到 LinuxServer 这台主机；“~”代表的是当前所在目录。关于特殊目录的意义，将会在下一节中详细介绍。

【实例】 查看当前主机的主机名及登录的用户名称。

```
[root@LinuxServer ~]# hostname
LinuxServer
[root@LinuxServer ~]# whoami
root
```

说明：hostname 命令用于查看主机名称，whoami 命令用于查看当前登录系统的用户名。

3.2.4 Bash 命令组成

Bash 命令的一般结构:

\$ 命令名称 [命令参数] [命令对象]

- ☑ \$ 是提示符，代表使用普通用户登录系统；
- ☑ [] 中的内容都不是必需的，是可选的；
- ☑ 命令名称：对 Shell 而言正确的 Linux 命令，Bash 命令均为小写；
- ☑ 命令参数：定制命令选项的一个或多个修饰符号，是 Bash 的命令选项；
- ☑ 命令对象：受命令影响的一个或多个对象。

注意：命令名称、命令参数、命令对象之间请用空格分隔。命令参数可用长格式（完整的选项名称），也可使用短格式（单个字母的缩写），两者分别用“--”和“-”作为前缀。

【实例】 查看系统主目录“/”下各文件的详细信息。

```
[root@LinuxServer /]# ls -al /
总用量 32
dr-xr-xr-x.  18 root root  256 4月  6 15:37 .
dr-xr-xr-x.  18 root root  256 4月  6 15:37 ..
-rw-----.  1 root root  124 1月  30 16:29 .bash_history
lrwxrwxrwx.  1 root root   7 1月  26 13:48 bin -> usr/bin
dr-xr-xr-x.   5 root root 4096 5月  14 10:41 boot
.....
```

说明：ls 为命令名称，-la 为命令参数-l 和-a 合并的缩写格式，其中 l 和 a 的顺序可互换。

3.3 Linux 系统常用命令

3.3.1 查看目录

学习目录的基本操作之前，请记住以下比较特殊的目录：

- ☑ “.” 表示当前目录；
- ☑ “..” 表示当前目录的上层目录；
- ☑ “-” 表示前一个工作目录；
- ☑ “~” 表示当前用户所在的根目录；
- ☑ “~linuxstudy” 表示 linuxstudy 用户的根目录。

下面详细介绍一些系统目录操作的常用命令。

1. 切换目录

命令名称：cd（Change Directory）。

使用方式：cd [目录名称]。

说 明：切换目录。若目录名称省略，则切换至用户的根目录。

【实例】 使用 cd 命令实现目录切换。

cd ~ ：跳到当前用户的根目录；



查看目录

cd : 不加任何路径时同 cd ~功能相同;
cd ~ linuxstudy : 切换到 linuxstudy 用户的根目录;
cd .. : 切换到当前目录的上层目录;
cd - : 切换到前一个工作目录;
cd /usr/bin/ : 绝对路径, 表示切换到/usr/bin/目录;
cd ./yum : 相对路径, 表示切换到当前路径下的 yum 目录。

说明: 在使用 cd 命令进行目录切换时, cd 后一定要加空格。

根目录“/”是所有目录的顶层, 那么“/”有上层目录(..)吗? 如果有, 那它的上层目录是什么? 请学习者自己用实验证明。

2. 查看当前路径

命令名称: pwd (Print Working Directory)。

使用方式: pwd [参数]。

说明: 显示当前所在目录。

参数: -P 表示显示当前的路径, 而非使用链接路径。

【实例】 查看当前工作目录。

```
[linuxstudy@LinuxServer root]$ pwd
/root
```

【实例】 使用 pwd 命令查看实际工作目录。

```
[linuxstudy@LinuxServer root]$ cd /var/mail
[linuxstudy@LinuxServer mail]$ pwd
/var/mail
[linuxstudy@LinuxServer mail]$ pwd -P
/var/spool/mail
```

说明: 可以看到两个命令获取的路径是有区别的。

3. 查看目录列表

命令名称: ls。

使用方式: ls [参数] [目录名称或文件名称]。

说明: 列出指定目录或文件的内容。

参数:

- a: 显示所有文件与目录 (默认情况下开头为“.”的文件或目录被视为隐藏文件, 不会列出);
- d: 仅列出目录, 而不列出其中的数据;
- l: 列出文件或目录的详细信息;
- r: 将文件以相反次序显示 (默认按英文字母升序显示);
- t: 将文件按建立时间的先后次序列出;
- A: 同-a, 但不列出“.” (当前目录) 和“..” (父目录);
- F: 在列出的文件或目录后加一符号, 如为可执行文件加“*”, 为目录则加“/”;
- R: 若目录下有文件, 则将其全部依序列出。

【实例】 列出“~”目录下所有文件或目录的详细信息（包含隐藏文件）。

```
[linuxstudy@LinuxServer ~]$ ls -al
总用量 32
drwx-----. 15 linuxstudy linuxstudy 4096 1月 30 17:26 .
drwxr-xr-x. 3 root root 24 1月 26 14:02 ..
-rw-----. 1 linuxstudy linuxstudy 7 1月 30 17:26 .bash_history
-rw-r--r--. 1 linuxstudy linuxstudy 18 10月 31 2018 .bash_logout
-rw-r--r--. 1 linuxstudy linuxstudy 193 10月 31 2018 .bash_profile
-rw-r--r--. 1 linuxstudy linuxstudy 231 10月 31 2018 .bashrc
drwx-----. 14 linuxstudy linuxstudy 4096 1月 30 17:25 .cache
drwxr-xr-x. 14 linuxstudy linuxstudy 261 1月 30 17:25 .config
drwx-----. 3 linuxstudy linuxstudy 25 1月 30 17:25 .dbus
-rw-----. 1 linuxstudy linuxstudy 16 1月 30 17:25 .esd_auth
-rw-----. 1 linuxstudy linuxstudy 310 1月 30 17:25 .ICEauthority
drwx-----. 3 linuxstudy linuxstudy 19 1月 30 17:25 .local
drwxr-xr-x. 4 linuxstudy linuxstudy 39 1月 26 13:47 .mozilla
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 公共
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 模板
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 视频
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 图片
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 文档
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 下载
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 音乐
drwxr-xr-x. 2 linuxstudy linuxstudy 6 1月 30 17:25 桌面
```

说明：ls -al 命令表示以长格式显示当前目录下的所有文件（包括隐藏文件）。

【实例】 列出/tmp/sanguo 目录的详细信息，不列出其中的文件信息。

```
[linuxstudy@LinuxServer ~]$ ls -ld /tmp/sanguo
drwxrwxr-x. 2 linuxstudy linuxstudy 6 5月 16 10:01 /tmp/sanguo
```

说明：ls -ld 命令表示以长格式显示目录信息。

【实例】 列出/tmp/sanguo 目录及其子目录的详细信息。

```
[linuxstudy@LinuxServer ~]$ ls -lR /tmp/sanguo
/tmp/sanguo:
总用量 0
drwxrwxr-x. 3 linuxstudy linuxstudy 17 5月 16 10:05 liubei
/tmp/sanguo/liubei:
总用量 0
drwxrwxr-x. 2 linuxstudy linuxstudy 6 5月 16 10:04 boy
/tmp/sanguo/liubei/boy:
总用量 0
```

3.3.2 管理目录和创建文件

1. 创建目录

命令名称：mkdir（Make Directory）。

使用方式：mkdir [参数] 目录名称。

说 明：创建新目录。



管理目录和创建文件

参 数:

-m: 按照某个指定权限创建目录;

-p: 一次性递归创建多个目录。

【实例】 切换到 **tmp** 目录下, 创建一个名为 **sanguo** 的目录。

```
[linuxstudy@LinuxServer tmp]$ cd /tmp
[linuxstudy@LinuxServer tmp]$ mkdir sanguo
```

【实例】 切换到 **sanguo** 目录下, 创建如下目录: **shuhan/liubei/guanyu/zhangfei/zhaoyun**。

```
[linuxstudy@LinuxServer tmp]$ cd sanguo
[linuxstudy@LinuxServer sanguo]$ mkdir -p shuhan/liubei/guanyu/zhangfei/zhaoyun
```

说明: **mkdir** 命令使用 **-p** 参数可以递归创建多个目录。

【实例】 在 **sanguo** 目录下, 创建一个权限为 **rwxrwxrwx** 的目录, 并取名为 **caowei**。

```
[linuxstudy@LinuxServer sanguo]$ mkdir -m 777 caowei
[linuxstudy@LinuxServer sanguo]$ ls -l
总用量 0
drwxrwxrwx. 2 linuxstudy linuxstudy 65 月 16 10:38 caowei
drwxrwxr-x. 3 linuxstudy linuxstudy 20 5 月 16 10:37 shuhan
```

说明: **777** 是用数字表示权限, **r** 权限为 **4**, **w** 权限为 **2**, **x** 权限为 **1**, 三者相加为 **7**。后续章节将会详细讲解 Linux 中的权限管理。

2. 删除目录

命令名称: **rmdir** (Remove Directory)。

使用方式: **rmdir** [参数] 目录名称。

说 明: 删除空目录。

参 数: **-p** 表示当子目录被删除后, 若该目录为空, 则将该目录也一并删除, 即连同上层空目录一并删除。

【实例】 进入 **/tmp/sanguo** 目录, 并用 **ls -l** 查看。

```
[linuxstudy@LinuxServer sanguo]$ cd /tmp/sanguo
[linuxstudy@LinuxServer sanguo]$ ls -l
总用量 0
drwxrwxrwx. 2 linuxstudy linuxstudy 65 月 16 10:38 caowei
drwxrwxr-x. 3 linuxstudy linuxstudy 20 5 月 16 10:37 shuhan
```

【实例】 删除 **caowei** 目录。

```
[linuxstudy@LinuxServer sanguo]$ rmdir caowei
```

【实例】 删除 **shuhan** 目录。

```
[linuxstudy@LinuxServer sanguo]$ rmdir shuhan
rmdir: 删除 "shuhan" 失败: 目录非空
[linuxstudy@LinuxServer sanguo]$ rmdir -p shuhan/liubei/guanyu/zhangfei/zhaoyun/
```

说明: 在删除 **zhaoyun** 目录的同时, 上级目录成为空目录, 因此依次递归删除。

3. 创建文件

`touch` 的一般用法是创建一个新文件，同时 `touch` 还可以用来修改文件时间戳。该命令虽不常用，但关键时很实用。

时间戳有三个时间，即最后访问时间 `access time`，简称为 `atime`；状态更改时间 `change time`，简称 `ctime`；内容修改时间 `modification time`，简称为 `mtime`。

- ☑ **atime**：最后访问时间，是一个文件的数据最后一次被访问的时间。例如，显示一个文件的内容或运行一个 Shell 脚本都会更新文件的 `atime`。可用 `ls -lu` 命令查看 `atime`。
- ☑ **ctime**：状态更改时间，指文件或目录的属性（所有者、权限等）被更改的时间，文件内容改变时 `ctime` 也会改变。如果一个文件需要备份的话，`dump` 命令需要用到 `ctime`。可用 `ls -lc` 命令查看 `ctime`。
- ☑ **mtime**：内容修改时间，是文件的目前内容最后被修改的时间，这是在一个长目录列表中显示的时间。

文件的 `atime` 是在读取文件或者执行文件时更改的；`ctime` 是在写入文件、更改所有者、权限或链接设置时随 `inode` 的内容更改而更改的；`mtime` 是在写入文件时随文件内容的更改而更改的。例如，执行 `cat file` 命令将更新文件的 `atime`；执行 `chmod a+w file` 命令将更新文件的 `ctime`；执行 `echo "wanglaoshi shige haoren" >file` 命令将会更新文件的 `ctime` 和 `mtime`。

命令名称：`touch`。

使用方式：`touch` [参数] 文件名称。

参 数：

- a: 仅修改访问时间；
- r: 修改文件的时间为指定文件的日期时间；
- t: 指定文件的日期时间为 `yymmddhhmm`；
- c: 仅改变状态更改时间，如果文件不存在则创建空文件；
- m: 仅修改 `mtime`。

说明：如果不加参数，后面直接接文件名称，则判断该文件是否存在，若存在则改变其时间，若不存在则创建一个 0 字节的空文件；如果加上相应参数，则会按照参数的设置修改时间戳。

【实例】在 `/tmp/sanguo` 目录下创建目录 `whb`，进入 `whb` 目录，并创建空文件 `linuxtest`。

```
[linuxstudy@LinuxServer ~]$ cd /tmp/sanguo
[linuxstudy@LinuxServer sanguo]$ mkdir whb
[linuxstudy@LinuxServer sanguo]$ cd whb
[linuxstudy@LinuxServer whb]$ touch linuxtest
[linuxstudy@LinuxServer whb]$ ls -l
总用量 0
-rw-rw-r--. 1 linuxstudy linuxstudy 0 5月 16 10:20 linuxtest
```

说明：利用 `ls -l` 命令显示文件信息，其中的时间为文件的 `mtime`。`touch` 命令的作用是创建了一个空文件。

【实例】将 `~/.bashrc` 文件连同属性一起复制到 `whb` 目录并改名为 `whb_bashrc`，并查看其三个时间。

```
[linuxstudy@LinuxServer whb]$ cp -a ~/.bashrc whb_bashrc
[linuxstudy@LinuxServer whb]$ ls -lc whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 10:22 whb_bashrc
[linuxstudy@LinuxServer whb]$ ls -lu whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 09:48 whb_bashrc
[linuxstudy@LinuxServer whb]$ ll whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 10月 31 2018 whb_bashrc
[linuxstudy@LinuxServer whb]$ ll whb_bashrc ;ll --time=atime whb_bashrc ;ll --time=ctime whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 10月 31 2018 whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 09:48 whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 10:22 whb_bashrc
```

说明：本实例中使用两种方法分别查看了 whb_bashrc 文件的三个时间。ls -lc 命令查看文件的 ctime；ls -lu 命令查看文件的 atime；ll 命令查看文件的 mtime，其中 ll 是 ls -l 的简写。在本实例中还用到了同时查看多项信息的组合命令，多个命令之间用分号隔开，并同时执行。

【实例】 将文件 whb_bashrc 的三个时间均改为当天。

```
[linuxstudy@LinuxServer whb]$ touch -d "today" whb_bashrc
[linuxstudy@LinuxServer whb]$ ll whb_bashrc ;ll --time=atime whb_bashrc ;ll --time=ctime whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 10:28 whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 10:28 whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 10:28 whb_bashrc
```

说明：本实例中将时间更改为当天，命令中使用了“today”。那如果调整为三天前呢？把时间写成“3 days ago”吗？请学习者在实践中检验、证明。

【实例】 将文件 whb_bashrc 的日期改为 2018/12/12 12:12。

```
[linuxstudy@LinuxServer whb]$ touch -t "201812121212" whb_bashrc
[linuxstudy@LinuxServer whb]$ ll whb_bashrc ;ll --time=atime whb_bashrc ;ll --time=ctime whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 12月 12 12:12 whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 12月 12 12:12 whb_bashrc
-rw-r--r--. 1 linuxstudy linuxstudy 231 5月 16 10:30 whb_bashrc
```

3.3.3 复制、移动与删除

对于文件与目录最基本的管理操作是复制、移动与删除，下面详细讲解这三个操作命令。

1. 复制命令

命令名称：cp (Copy)。

使用方式：cp [参数] 源文件 目标文件。

说明：将一个文件复制至另一个文件，或复制至另一个目录。

参数：

- f: 文件在目标路径中存在时，直接覆盖；
- i: 文件在目标路径中存在时，提示询问是否覆盖；
- r: 复制指定文件中所有内容和结构；
- b: 生成覆盖文件的备份；



复制、移动与删除

- a: 保持文件原有属性;
- s: 复制为软链接, 也就是快捷方式;
- l: 复制为硬链接;
- d: 如果源文件是链接文件, 则复制链接文件属性, 而非文件本身。

【实例】 在前面创建的 `sanguo` 目录中, 创建 `caowei` 目录, 并将 `/etc/passwd` 文件复制到该目录。

```
[linuxstudy@LinuxServer sanguo]$ cd /tmp/sanguo
[linuxstudy@LinuxServer sanguo]$ mkdir caowei
[linuxstudy@LinuxServer sanguo]$ cp /etc/passwd ./caowei/
[linuxstudy@LinuxServer sanguo]$ ls -l caowei/
总用量 4
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:46 passwd
```

【实例】 将 `/etc` 目录下的 `passwd` 文件复制到 `caowei` 目录下, 并改名为 `caocao`。

```
[linuxstudy@LinuxServer sanguo]$ cp /etc/passwd /tmp/sanguo/caowei/caocao
[linuxstudy@LinuxServer sanguo]$ ls -l caowei/
总用量 8
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:49 caocao
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:46 passwd
```

说明 1: 以上两个实例中分别应用了相对路径“.”和绝对路径“/etc/passwd”“/tmp/sanguo/caowei/caocao”, 后续章节中会介绍到两者的概念与区别。

说明 2: 在这两个实例中大家可看到, `cp` 命令的功能不仅有复制, 还可在复制的同时重命名。

【实例】 将 `/var/log/wtmp` 文件复制到 `caowei` 目录中, 查看其属性, 加上 `-a` 参数后再次查看。

```
[linuxstudy@LinuxServer sanguo]$ cp /var/log/wtmp caowei/
[linuxstudy@LinuxServer sanguo]$ ls -l caowei/wtmp /var/log/wtmp
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 16 10:51 caowei/wtmp
-rw-rw-r--. 1 root          utmp          18816 5月 15 18:53 /var/log/wtmp
[linuxstudy@LinuxServer sanguo]$ cp -a /var/log/wtmp caowei/wtmp1
[linuxstudy@LinuxServer sanguo]$ ls -l caowei/wtmp1 /var/log/wtmp
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 15 18:53 caowei/wtmp1
-rw-rw-r--. 1 root          utmp          18816 5月 15 18:53 /var/log/wtmp
```

说明: 通过本实例可以发现, `-a` 参数的作用相当于 `-pdr` 参数, 即复制后保持原有文件的所有属性。

【实例】 使用 `root` 用户将 `/etc` 目录复制到 `caowei` 目录下。

```
[root@LinuxServer sanguo]# cp -r /etc caowei/
[root@LinuxServer sanguo]# ls -l caowei/
总用量 60
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:49 caocao
drwxr-xr-x. 138 root          root          8192 5月 16 10:56 etc
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:46 passwd
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 16 10:51 wtmp
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 15 18:53 wtmp1
```

说明：通过本实例可以看到，如果复制的是一个目录，则必须加上-r 参数，-r 表示递归，即连同目录中的文件或子目录一并复制过来。

【实例】为 caowei 目录下的 passwd 文件分别创建软、硬链接。

```
[root@LinuxServer caowei]# cp -s passwd ruanlianjie
[root@LinuxServer caowei]# cp -l passwd yinglianjie
[root@LinuxServer caowei]# ls -l
总用量 64
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:49 caocao
drwxr-xr-x. 138 root root 8192 5月 16 10:56 etc
-rw-r--r--. 3 linuxstudy linuxstudy 2277 5月 16 10:46 passwd
lrwxrwxrwx. 1 root root 6 5月 16 11:00 ruanlianjie -> passwd
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 16 10:51 wtmp
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 15 18:53 wtmp1
-rw-r--r--. 3 linuxstudy linuxstudy 2277 5月 16 10:46 yinglianjie
```

说明：通过本实例发现，ruanlianjie 的大小仅有 6 个字节，正好是 passwd 文件名字的长度，即软链接中只存了源文件的名称，而硬链接大小与 passwd 相同。

【实例】在刚才创建的软、硬链接的基础上删除 passwd 文件，并查看。

```
[root@LinuxServer caowei]# rm passwd
rm: 是否删除普通文件 "passwd"? y
[root@LinuxServer caowei]# ls -l
总用量 60
-rw-r--r--. 1 linuxstudy linuxstudy 2277 5月 16 10:49 caocao
drwxr-xr-x. 138 root root 8192 5月 16 10:56 etc
lrwxrwxrwx. 1 root root 6 5月 16 11:00 ruanlianjie -> passwd
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 16 10:51 wtmp
-rw-rw-r--. 1 linuxstudy linuxstudy 18816 5月 15 18:53 wtmp1
-rw-r--r--. 2 linuxstudy linuxstudy 2277 5月 16 10:46 yinglianjie
```

说明：本实例中提前应用了文件的删除命令 rm，源文件删除后 yinglianjie 文件没有变化，而软链接出现了问题，这是因为名称为 ruanlianjie 的文件本身就是个快捷方式，源文件被删除后，快捷方式也就出问题了。

2. 移动命令

命令名称：mv (Move)。

使用方式：mv [参数] 源文件 目标文件。

说明：重命名文件或将文件移动至另一目录。

参数：

- i: 如果存在文件重名，则提示是否覆盖；
- b: 建立覆盖文件的备份；
- f: 如果存在文件重名，则直接覆盖；
- u: 移动时如果存在同名文件，则比较新旧，源文件新则覆盖目标文件。

【实例】在 sanguo 目录下创建名为 shuhan 的目录，将 caocao 文件移动至该目录。

```
[root@LinuxServer sanguo]# mkdir shuhan
[root@LinuxServer sanguo]# mv caowei/caocao shuhan
[root@LinuxServer sanguo]# ls shuhan
caocao
```

【实例】 在 `sanguo` 目录下创建名为 `sunquan` 的目录，并将其改名为 `sunwu`。

```
[root@LinuxServer sanguo]# mkdir sunquan
[root@LinuxServer sanguo]# ls
caowei shuhan sunquan
[root@LinuxServer sanguo]# mv sunquan sunwu
[root@LinuxServer sanguo]# ls
caowei shuhan sunwu
```

3. 删除命令

命令名称： `rm` (Remove)。

使用方式： `rm` [参数] 目录名称或文件名称。

说 明： 删除文件或目录。

参 数：

- i: 删除前逐一询问确认；
- f: 即使原文件属性设为只读也直接删除，无须逐一确认，可强制删除；
- r: 将目录及其目录内文件逐一删除。

【实例】 删除 `shuhan` 目录下的 `caocao` 文件。

```
[root@LinuxServer sanguo]# rm shuhan/caocao
rm: 是否删除普通文件 "shuhan/caocao"? y
[root@LinuxServer sanguo]# ls shuhan
[root@LinuxServer sanguo]#
```

【实例】 删除 `caowei` 目录下的 `etc` 目录。

```
[root@LinuxServer sanguo]# rm -r caowei/etc/
rm: 是否进入目录"caowei/etc/"? y
rm: 是否删除普通文件 "caowei/etc/fstab"? y
rm: 是否删除普通空文件 "caowei/etc/crypttab"? y
rm: 是否删除符号链接 "caowei/etc/mtab"? ^C
[root@LinuxServer sanguo]# rm -rf caowei/etc/
[root@LinuxServer sanguo]# ls caowei/
ruanlianjie wtmp wtmp1 yinglianjie
```

说明： 用 `rm -r` 命令删除目录时会逐一询问是否删除，加上了 `-f` 参数后则不会再逐一询问，直接删除。但是切记，`-f` 参数不要随便使用，否则可能会造成无法挽回的损失。



3.3.4 查看文件内容

在不借助任何查看工具的前提下，想利用 Linux 命令查看文件的内容，查看文件内容应该怎么做呢？接下来将详细介绍查看命令 `cat`、`tac`、`more`、`less`、`head`、`tail`。其中，`cat` 由第一行开始显示文件内容；`tac` 从最后一行开始显示文件内容，与 `cat` 正好相反；`more` 分页显示文件内容；`less` 与 `more` 类似，但支持往上翻页；`head` 查看文件前几行内容；`tail` 查看文件后几行内容。下面详细解读一下每个命令的用法。

1. `cat` 命令

命令名称： `cat` (Concatenate)。

使用方式：cat [参数] [--help] [--version] 文件名称。

说 明：查看纯文本文件内容（适用于内容较少的文件）。

参 数：

-n 或--number：由 1 开始对所有输出的行数编号；

-b 或--number-nonblank：和-n 相似，只不过对于空白行不编号；

-s 或--squeeze-blank：当遇到有连续两行以上的空白行时，就代换为一行；

-v 或--show-nonprinting：显示具体格式。

【实例】 将/etc 目录下的 passwd 文件复制到 sanguo 目录下，并用 cat 命令查看其内容。

```
[root@LinuxServer sanguo]# cp /etc/passwd /tmp/sanguo/
[root@LinuxServer sanguo]# cat /tmp/sanguo/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
.....
```

【实例】 将上面查看的文件加上行号。

```
[root@LinuxServer sanguo]# cat -n /tmp/sanguo/passwd
 1      root:x:0:0:root:/root:/bin/bash
 2      bin:x:1:1:bin:/bin:/sbin/nologin
 3      daemon:x:2:2:daemon:/sbin:/sbin/nologin
 4      adm:x:3:4:adm:/var/adm:/sbin/nologin
 5      lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
 6      sync:x:5:0:sync:/sbin:/bin/sync
.....
```

2. tac 命令

命令名称： tac。

使用方式： tac 文件名称。

说 明：将文件从最后一行到第一行反向输出。

【实例】 用 tac 命令查看 sanguo/passwd 的内容。

```
[root@LinuxServer sanguo]# tac /tmp/sanguo/passwd
linuxstudy:x:1000:1000:LinuxStudy:/home/linuxstudy:/bin/bash
tcpdump:x:72:72:::/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
.....
```

3. more 命令

命令名称： more。

使用方式： more [参数] 文件名称。

说 明：分页显示文件的内容，空格键【Space】代表显示下一页，【B】键代表显示

上一页，还有搜寻字串的功能（与 Vim 相似），按【H】键可显示说明文件。

参 数：

- f: 统计逻辑行数而不是屏幕行数；
- p: 不滚屏，清屏并显示文本；
- s: 将多个空行压缩为一行；
- NUM: 指定每屏显示的行数为 NUM；
- +NUM: 从文件第 NUM 行开始显示；
- +/STRING: 从匹配搜索字符串 STRING 的文件位置开始显示；
- V: 输出版本信息并退出。

【实例】使用 more 命令查看 passwd 文件的信息。

```
[root@LinuxServer sanguo]# more passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
--More--(52%)
```

4. less 命令

命令名称: less。

使用方式: less [参数] 文件名称。

说 明: less 与 more 的作用十分相似，都可用于浏览文件的内容。不同的是，less 允许往上翻页，浏览已经看过的部分。less 并非一次性读入整个文件，因此针对大型文件速度会比一般的文本编辑器（如 Vim）快。

参 数：

- c: 从上到下刷新屏幕，并显示文件内容，而不是通过底部滚动完成刷新；
- f: 强制打开文件，显示二进制文件时不提示警告；
- i: 搜索时忽略大小写，除非搜索串中包含大写字母；
- I: 搜索时忽略大小写，除非搜索串中包含小写字母；
- m: 显示读取文件的百分比；
- M: 显示读取文件的百分比、行号及总行数；
- N: 在每行前输出行号。

【实例】使用 less 命令查看 passwd 的信息。

```
[root@LinuxServer sanguo]# less passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
passwd
```

5. head 命令

命令名称: head。

使用方式: head [参数] 文件名称。

说 明: 以行为单位读取文件前 N 行数据。

参 数: -n 后面接数字 N, 代表显示 N 行。

【实例】 使用 head 命令读取 passwd 文件前 3 行数据。

```
[root@LinuxServer sanguo]# head -n 3 passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

6. tail 命令

命令名称: tail。

使用方式: tail [参数] 文件名称。

说 明: 以行为单位读取文件最后 N 行数据。

参 数: -n 后面接数字 N, 代表显示 N 行。

【实例】 使用 tail 命令读取 passwd 文件后 3 行数据。

```
[root@LinuxServer sanguo]# tail -n 3 passwd
ntp:x:38:38::/etc/ntp:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
linuxstudy:x:1000:1000:LinuxStudy:/home/linuxstudy:/bin/bash
```

【实例】 使用 tail 命令读取 passwd 文件 40 行以后的数据。

```
[root@LinuxServer sanguo]# tail -n +40 passwd
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
linuxstudy:x:1000:1000:LinuxStudy:/home/linuxstudy:/bin/bash
```

3.4 Vim 文本编辑器



Vim 编辑器

Vim 是一个类似于 Vi 的文本编辑器, 但在 Vi 的基础上增加了很多新特性, Vim 被普遍认为是类 Vi 编辑器中最好的一个。Vi 和 Vim 都是多模式编辑器, 不同的是 Vim 是 Vi 的升级版, 它不仅兼容 Vi 的所有指令, 而且还有一些新特性, 比如代码补全、编译及错误跳转、代码高亮显示等方便编程的功能, 在程序员中被广泛使用。另外, Vim 还可以很好地支持中文, 尤其是简体中文。

学习 Vim 首先要过两关。第一关是理解 Vim 的设计思路, Vim 是整个文本编辑都用键盘而非鼠标来完成的, 键盘上几乎每个键都有固定用法, 且 Vim 希望用户在普通模式(也就是命令行模式, 只可输入命令)下完成大部分的编辑工作, 将此模式设置为默认模式。初学者打开 Vim 后, 若直接输入单词, 则会“滴滴”乱响, 这是因为 Vim 把用户输入的单词理解为命令了。第二关是命令, Vim 有超过一百条命令, 若能熟练使用这些命令, 编辑速度会比使用鼠标快, 但要全都记住也是一件难事, 因此最好的方法就是多操作, 把 Vim 用在日常文本编辑中, 遇到难题及时解决, 这样 Vim 技能便会提升。

Vim 是一个多模式编辑器, 其模式大致分为以下六种。

(1) 命令模式 (n): 在其他任何一种模式下, 按【Esc】键或者【Ctrl+C】组合键可以退到命令模式。

(2) 输入模式 (i): 也叫插入模式, 在该模式下, Vim 像一个常见的编辑器。在命令模式下, 按【I】或【A】键可进入该模式。当然, 还有一些其他命令也可实现。

(3) 可视模式 (v): 在该模式下, 可以按【H】键进行选择, 然后进行复制、粘贴或其他操作。在命令模式下, 按【V】键进入该模式。

(4) 块操作模式 (V): 在命令模式下, 按【Ctrl+V】组合键进入该模式。

(5) 修改模式 (R): 即改写模式, 很多软件中用【Insert】键来完成这个切换。在 Vim 中, 在命令模式下, 按【R】键进入该模式。

(6) 末行模式 (ex): 命令执行模式, 在命令模式下, 按冒号【:】键切换到该模式。

在以上六种模式中, 最常用的是命令模式、输入模式及末行模式三种。Linux 中一般直接进入的是命令模式, 在命令模式下按【I】【A】【O】等键可进入输入模式, 输入模式下按【Esc】键可以回到命令模式; 在命令模式下按【:】键可进入末行模式, 末行模式下按【Esc】键可以回到命令模式。模式互相转换的示意图如图 3.6 所示。

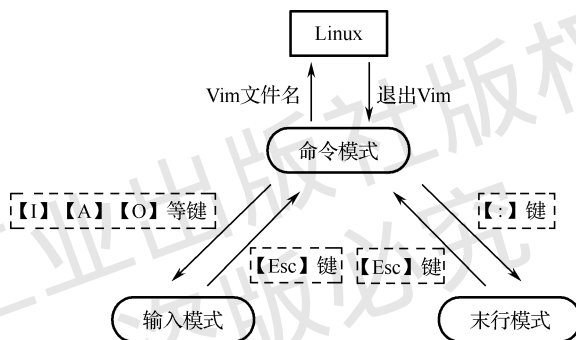


图 3.6 模式转换图

Vim 具有颜色显示的功能, 并且还支持许多程序语法, 因此当用户使用 Vim 编辑器时, 可以高亮显示, 并在程序出现错误时智能提示。在 Linux 系统中输入 Vim 命令得到如图 3.7 所示界面。

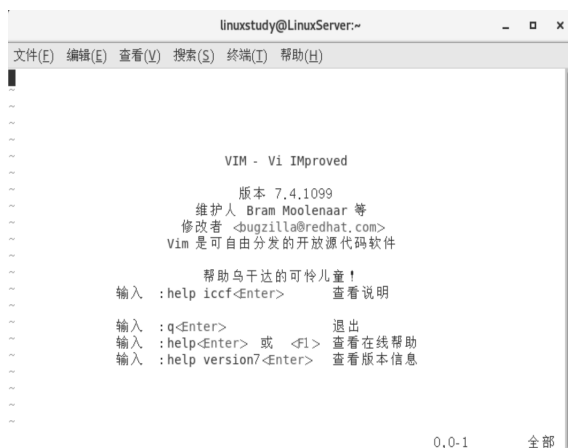


图 3.7 Vim 编辑器界面

Vim 的主要命令有进入 Vim、光标控制、添加数据、删除数据、修改数据、查找替换、复制文本、撤销重复及保存退出等，每部分都有命令符。下面就通过简单的命令符加上实例来学习 Vim。

1. 进入 Vim

- ☑ Vim 文件名称：打开或新建文件，并将光标置于第一行行首；
- ☑ Vim +n 文件名称：打开文件，并将光标置于第 n 行行首；
- ☑ Vim +文件名称：打开文件，并将光标置于最后一行行首；
- ☑ Vim 文件名称 1……文件名称 n：打开多个文件，依次进行编辑。

【实例】 使用 Vim 打开/etc/passwd 文件，结果如图 3.8 所示。

```
[linuxstudy@LinuxServer sanguo]$ vim /etc/passwd
```



```
linuxstudy@LinuxServer:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:99:99:Nobody:/:/sbin/nologin  
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin  
dbus:x:81:81:System message bus:/:/sbin/nologin  
polkitd:x:999:998:User for polkitd:/:/sbin/nologin  
libstoragemgmt:x:998:996:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin  
colord:x:997:995:User for colord:/var/lib/colord:/sbin/nologin  
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin  
saslauth:x:996:76:Saslauthd user:/run/saslauthd:/sbin/nologin  
abrt:x:173:173:./etc/abrt:/sbin/nologin  
*/etc/passwd" [只读] 43L, 2277C 1,1 顶端
```

图 3.8 /etc/passwd 文件

2. 光标控制

- ☑ h：光标左移一个字符，例如，8h 左移 8 个字符，以下可类推；
- ☑ l：光标右移一个字符，到本行右端则不再移动；
- ☑ k 或 Ctrl+p：光标移至上一行的同列位置；
- ☑ j 或 Ctrl+n：光标移至下一行的同列位置；
- ☑ Space：光标右（后）移一个字符，到本行右端将移到下一行；
- ☑ Backspace：光标左（前）移一个字符，到本行左端将移到上一行。

3. 翻页操作

- ☑ Ctrl+u 或 Ctrl+d：向文件首（Ctrl+u）或文件尾（Ctrl+d）翻半屏；
- ☑ Ctrl+f 或 Ctrl+b：向文件首（Ctrl+f）或文件尾（Ctrl+b）翻一屏。

4. 插入操作

- ☑ i 或 I：在光标前（i）或当前行首（I）插入；
- ☑ a 或 A：在光标后（a）或当前行尾（A）插入；

- ☑ o 或 O: 在当前行之下 (o) 或之上 (O) 新开一行;
- ☑ r: 替换当前字符;
- ☑ R: 替换当前字符及其后的字符, 直至按【Esc】键。

5. 删除操作

- ☑ ndw 或 ndW: 删除光标处开始及其后的 n-1 个字符;
- ☑ d0 或 d\$: 删至行首 (d0) 或行尾 (d\$);
- ☑ dL: 删至当前屏幕最后一行;
- ☑ ndd: 删除当前行及其后 n-1 行。

【实例】 将/etc/passwd 文件复制到/tmp/sanguo 目录, 用 Vim 查看, 并删除当前光标下的 5 行。

```
[linuxstudy@LinuxServer tmp]$ cp /etc/passwd /tmp/sanguo/
[linuxstudy@LinuxServer tmp]$ vim /tmp/sanguo/passwd
```

说明: 光标处于 sync:x:5:0:sync:/sbin:/bin/sync 这一行的行首, 因为要一次删除 5 行数据, 所以在命令模式下执行“5dd”命令, 删除前如下:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

删除后如下:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

可见, 已经达到了删除效果。

6. 复制、粘贴操作

- ☑ yy: 将当前行的内容放入临时缓冲区;
- ☑ nyy: 将 n 行的内容放入临时缓冲区;
- ☑ p: 将临时缓冲区中的文本粘贴到光标后;
- ☑ P: 将临时缓冲区中的文本粘贴到光标前。

Vim 操作命令有很多, 详细使用方法请参看 man 手册。下面通过几个实例来完成相对

复杂一些的操作。

【实例】 使用 `root` 用户新建文件 `libai`，并在其中输入 4 行数据。

```
[root@LinuxServer ~]# vim libai
```

在打开的文件中输入李白的《静夜思》并保存，如图 3.9 所示。



图 3.9 打开文件并输入内容

说明：“:wq”用于数据的保存退出。

【实例】 光标定位到 `libai` 文件的第 3 行，将 `ls -l /root` 命令的执行结果附加到光标之下的位置。

将光标定位到第 3 行开头，并在末行模式下输入“:r!ls -l /root”，如图 3.10 和图 3.11 所示。



图 3.10 附加结果

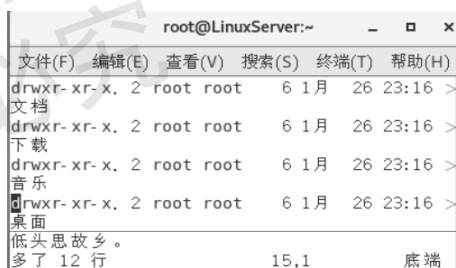


图 3.11 附加了执行结果

【实例】 用 Vim 查看刚刚创建的 `libai` 文件并显示行号，如图 3.12 所示。

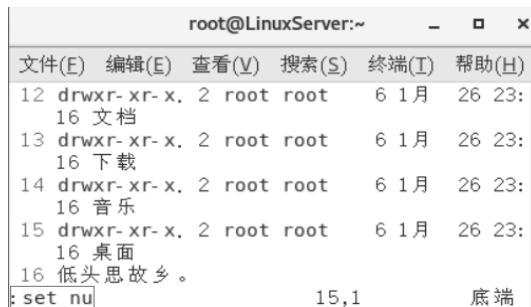


图 3.12 查看 `libai` 文件并显示行号

【实例】 将 `libai` 文件中的 `root` 替换成 `whb`，如图 3.13 所示。

说明：%s 代表整个文件，g 代表全部替换。



图 3.13 替换文字

3.5 Linux 系统中获取帮助

Linux 系统中有很多命令供用户使用，其参数也十分复杂，而不同情况下使用何种命令呢？很多初学者试图记住所有命令，勇气可嘉但没有必要，因为选择命令时可以使用在线帮助文件。下面详细讲解获取帮助的三个命令。

3.5.1 man 命令

在 Linux 系统中，如果对某个命令的意思或使用方法不熟悉，找到该命令的“使用说明书”（man）即可。man 是 manual（操作说明）的简写，所有的 man page 文件都放在目录/usr/share/man 中。

命令名称：man。

使用方式：man [参数] [命令名称]。

说明：利用 man 文件查看命令的用法。

参数：

-M: 指定搜索 man 手册页的路径，通常路径由环境变量 MANPATH 预设，若在命令行中指定另外的路径，则覆盖 MANPATH 的设定；

-P: 指定所使用的分页程序，默认使用 /usr/bin/less-is，在环境变量 MANPAGER 中预设；

-a: 显示所有的手册页内容，而不是只显示第一页；

-d: 该参数主要用于调试排错，如果用户加入了新文件，可用该参数调试排错，但不列出文件内容；

-f: 只显示出命令功能而不显示其中详细的说明文件；

-p: string 设定运行的预先处理程序的顺序；

-w: 不显示手册页，只显示文件所在位置。

【实例】 使用 man 查询 cp 命令的作用和用法。

```
[linuxstudy@LinuxServer sanguo]$ man cp
CP(1)                General Commands Manual                CP(1)
```

```

NAME
    cp - 复制文件和目录

总览
    cp [参数] 文件路径
    cp [参数] 文件...目录

POSIX 参数: [-fipRr]

GNU 参数(最短形式): [-abdfilprsvxPR] [-S
SUFFIX] [-V {numbered,existing,simple}]
    [--sparse=WHEN] [--help] [--version] [--]

描述
    cp                                复制文件(或者目录等) .
    可以使用这个命令复制一个文件到一个指定的目的地,
    或者复制任意多个文件到一个目的目录.

```

下面介绍一下 `man page` 文件中的基本标记的含义:

- Header: 标题;
- NAME: 命令的名称及作用;
- 总览: 命令的具体格式;
- 描述: 命令参数的详细说明;
- 参数: 该命令所有可选参数的详细说明。

在 `man page` 中的各按键的作用如下:

- 空格键: 向下翻一页;
- PageDown: 向下翻一页;
- PageUp: 向上翻一页;
- Home: 到第一页;
- End: 到最后一页;
- /word: 向下搜索 word 字符串, 如果要搜索 date 的话, 就输入 “/date”;
- ?word: 向上搜索 word 字符串;
- n,N: 使用/或?来搜索字符串时, 可以用 n 来继续下一个搜索;
- q: 结束并退出 `man page`。

3.5.2 info 命令

`info page` 也是在线求助的方法, 与 `man` 用法类似。`info page` 是将文件数据拆成一个一个的段落, 每个段落用独立的页面来撰写, 并在各个页面中还有类似网页的“超链接”, 可跳到不同的页面中。每个独立的页面被称为一个节点。值得注意的是, 查询的目标数据的说明文件必须以 `info` 的格式实现, 才能使用 `info` 的特殊功能。`info page` 的文件都放在 `/usr/share/info` 目录中。

命令名称: `info`。

使用方式: `info [参数] [命令名称]`。

说明：利用 info 查看命令用法。

【实例】查看命令 info 的用法。

```
[linuxstudy@LinuxServer sanguo]$ info info
File: info.info, Node: Top, Next: Getting Started, U\
p: (dir)
```

```
Info: An Introduction
```

```
*****
```

```
The GNU Project distributes most of its on-line manuals\
in the "Info
format", which you read using an "Info reader". You ar\
e probably using
an Info reader to read this now.
```

```
There are two primary Info readers: 'info', a stand-\
alone program
designed just to read Info files (*note What is Info?: \
(info-std)Top.),
and the 'info' package in GNU Emacs, a general-purpose \
editor. At
```

```
--zz-Info: (info.info.gz)Top, 80 lines --Top-----
Welcome to Info version 5.1. Type h for help, m for men
```

在 info page 文件中，部分数据的意义如下：

- ☑ File: 代表 info page 中的数据由 info.info 文件所提供；
- ☑ Node: 代表当前页面属于 top 节点；
- ☑ Next: 下一节点的名称；
- ☑ Up: 回到上一层节点的总览界面；
- ☑ Prev: 前一节点；

在 info page 中各按键的用法如下：

- ☑ PageDown: 向下翻页；
- ☑ PageUp: 向上翻页；
- ☑ Tab: 在节点之间移动；
- ☑ Enter: 当光标在节点上面时，按【Enter】键可以进入该节点；
- ☑ B: 移动光标至该 info 界面的第一个节点处；
- ☑ E: 移动光标至该 info 界面的最后一个节点处；
- ☑ N: 到下一个节点处；
- ☑ P: 到上一个节点处；
- ☑ U: 向上移动一层；
- ☑ S (/): 在 info page 中进行查询；
- ☑ H: 显示求助菜单；
- ☑ Q: 结束并退出 info page。

3.5.3 help 命令

命令名称: help。

使用方式: [命令名称] --help。

说明: 通过该命令可以查找 Shell 命令的用法，只需在所查找的命令后输入“--help”，即可看到所查命令的内容。

【实例】 查看 cal 命令的用法。

```
[linuxstudy@LinuxServer sanguo]$ cal --help
```

用法:

```
cal [参数] [[[日] 月] 年]
```

参数:

-1, --one	只显示当前月份（默认）
-3, --three	显示上个月、当月和下个月
-s, --sunday	周日作为一周第一天
-m, --monday	周一作为一周第一天
-j, --julian	输出儒略日
-y, --year	输出整年
-V, --version	显示版本信息并退出
-h, --help	显示此帮助并退出

3.6 小结

本章学习了 Linux 基本操作和相关的命令，主要包括 GNOME 图形界面的使用；Bash 基础及命令组成；Shell 简介及其分类；Linux 系统常用命令：查看目录命令（pwd、cd、ls）、创建目录和文件命令（mkdir、rmdir、touch）、对文件和目录操作命令（cp、mv、rm），以及查看文件内容命令（cat、tac、more、less、head、tail）；Vim 文本编辑器的基本操作，以及 Linux 系统中三种帮助命令的使用方法。

实训 3 Linux 基本操作

一、实训目的

熟练掌握 Linux 系统中的常用命令和 Vim 编辑器的使用，学会在 Linux 中寻求在线帮助。

二、实训内容

(1) 常用命令：pwd、cd、ls、mkdir、rmdir、touch、cp、mv、rm、cat、tac、more、less、head、tail。

- (2) 文本编辑命令: Vim。
- (3) 帮助命令: man、info、help。

三、项目背景

小 A 了解了 Linux 系统的历史背景, 安装了虚拟机和 Linux 系统, 现在小 A 要真正开始学习 Linux 系统了。小 A 知道 Linux 是基于命令的操作系统, 所以必须掌握一些常用命令的基本操作, 并学会如何寻求帮助。

四、实训步骤

任务 1: 掌握 Linux 中图形界面 GNOME 的基本操作。

任务 2: 完成下面的题目。

1. 常用命令实践题

- (1) 进入/var/mail 目录, 利用命令查看实际工作目录与链接文件本身的详细信息。
- (2) 在根目录下创建 test 目录, 进入该目录并分别创建 test1 与 test2/test3/test4/test5 目录。在 test1 目录中创建权限为 rwx--x--x 的目录 test11 和具有默认权限的目录 test12, 并通过 ls -al 命令查看其区别。
- (3) 将第(2)题中创建的 test2 中的文件及目录全部删除。
- (4) 用 root 身份查看自身的环境变量, 切换到 linuxstudy 下查看其环境变量是否有区别。
- (5) 切换到 root 用户, 将 ls 由/bin/ls 移动到/root/ls (可用 mv /bin/ls /root 命令)。在/root 目录下, 请问: ①能不能直接输入 ls? ②若不能, 该如何执行 ls 命令? ③若要直接输入 ls 即可执行, 又该如何修改环境变量 \$PATH? ④将 ls 移动回原目录。
- (6) 列出根目录下的所有文档 (含隐藏), 并显示其详细内容。
- (7) 将 root 用户根目录中的 .bashrc 文件复制到第(2)题中创建的 test11 目录中, 并改名为 bashrc; 进入 test11 文件夹, 将 /var/log/wtmp 复制到当前所在的目录下并用 ls 查看其属性。
- (8) 复制 etc 目录下的所有内容到 test12 目录中; 为第(7)题中复制的 bashrc 创建软、硬链接并分别命名为 bashrc_s 与 bashrc_h, 查看其详细信息; 将 bashrc_s 复制成 bashrc_s1 与 bashrc_s2, 查看其详细信息, 体会何时复制的是真实文档, 何时复制的是链接本身。
- (9) 将第(7)题中建立的 bashrc 文件删除, 将 test11 中所有 bashrc 开头的文件全部删除, 在 test 目录中创建 test3/test31, 将 test3/test31 全部删除。
- (10) 使用 touch 新建文件 testtouch, 并查看其时间和大小; 分别查看该文件的 atime、ctime 和 mtime 三个时间。
- (11) 将 ~/.bashrc 文件复制到普通用户 linuxstudy 的根目录下并改名为 bashrc, 复制时使用 -a 参数; 查看该文件的 atime、ctime 和 mtime 三个时间; 将 bashrc 的日期向前调整 2 天; 将 bashrc 的时间更改为当前时间。

2. Vim 编辑器实践题

- (1) 使用 Vim 编辑一个文件, 取名为 jin.c。

- (2) 打开编写的文件，执行 `ls -la /etc/` 命令，并将执行结果附加到 `jin.c` 文件中。
- (3) 将 `jin.c` 文件中的“a”字符全部替换成“linux”。

3. 帮助命令实践题

- (1) 查看 `etc` 目录下的 `issue` 文件内容，如何显示行号？如何将空白行的行号去掉？
- (2) 利用 `tac` 查看 `etc` 目录下的 `issue` 文件内容，与 `cat` 的结果进行比较，有什么不同？
- (3) 利用 `more` 查看 `/root/initial-setup-ks.cfg` 文件，熟悉 `more` 中的翻页、查找说明文件与退出操作。
- (4) 利用 `less` 查看 `/root/initial-setup-ks.cfg` 文件，熟悉 `less` 中的上、下翻页与退出操作。
- (5) 创建 `teacher.c` 文件，输入 30 行数据，利用 `head` 与 `tail` 命令进行查看，体会其区别。
- (6) 若要显示 `/etc/passwd` 文件的第 20 行到第 30 行内容，该如何实现？
- (7) 将 `/usr/bin/passwd` 文件内容以 ASCII 码方式输出，将 `/etc/issue` 文件内容以八进制方式输出。

电子工业出版社版权所有
盗版必究