

本章主要介绍 Linux 操作系统与 Shell 的关系、简单命令、一般命令格式、常用命令、Shell 高级操作、Linux 的 X-Window、环境变量及系统配置等内容,并详细介绍信息显示命令、查询系统命令及如何在桌面环境简单操作等问题。

本章的学习目标

- 了解 Linux 操作系统与 Shell 的关系。
- 掌握简单命令、一般命令格式和一些常用命令。
- 掌握 Shell 高级操作、环境变量及系统配置。
- 了解 Linux 的 X-Window。

3.1 Linux 系统与 Shell 的关系

Linux 操作系统主要由两部分组成:内核和系统工具。内核是 Linux 系统的核心并且驻留内存。所有直接与硬件通信的常规程序都集中在内核中,与操作系统的其他部分相比,这部分相对比较小。除了内核,其他必要的模块也驻留内存。这些模块执行一些重要功能,如输入/输出、文件管理、内存管理和处理器管理。

Linux 系统的其他部分保存在磁盘上,需要时调入内存。Linux 命令就是保存在磁盘上的程序(即系统工具)。当输入一个命令(请求程序执行)时,相应的程序就被调入内存。用户通过 Shell 与操作系统通信,而依赖于硬件的操作是由内核管理的。图 3-1 给出了 Linux 操作系统的组件,从中可见 Shell 的位置。

Shell 的原意是外壳,用来形容物体外部的架构。Linux 系统的 Shell 作为操作系统的外壳,为用户提供了使用操作系统的接口。它是命令语言、命令解释程序及程序设计语言的统称。优秀的管理员喜欢用 Shell 帮助他们自动完成各种管理工作。Linux 系统很多服务都是通过 Shell 脚本来启动,通过查看这些脚本,可以了解服务的启动过程,从而为故障诊断和系统优化做好准备。

每种操作系统都有其特定的 Shell。DOS 的标准 Shell 是 `command.com`,虽然 DOS 系统有多种,但是通常都采用 `command.com` 作为 Shell 的名称。Windows 95/Windows 98 的 Shell 是 `explorer.exe`。各主要 UNIX 类操作系统下默认的 Shell 为: AIX 默认的是 Korn Shell, Solaris 和 FreeBSD 默认的是 Bourne Shell, HP-UX 默认的是 POSIX Shell,

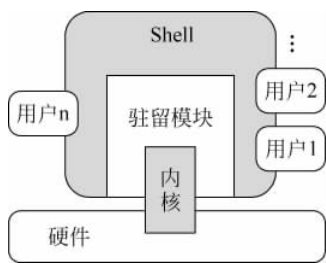


图 3-1 Linux 系统的组件

Linux 默认的是 Bash(Bourne Again Shell),也可以重新设定或切换到其他的 Shell。

Bash 是 GNU 系统的标准 Shell,正式发布于 1988 年 11 月 10 日,Brain Fox 编写了 Bash 的最初版本,1989 年年初,Chet Ramey 加入开发,负责大量的故障调试及加入许多有用的特性。现在人们可免费从 FSF(Free Software Foundation)得到 0.99 以上的版本。Red Hat Linux 9 的版本是 2.05b,Fedora Core 8 的版本是 3.1。可以用 Bash 的命令行选项-version 或打印环境变量 BASH_VERSION 的值来获得版本号。本书是以 CentOS 7 系统为例进行说明的,查看它的 Shell 解释器,Bash 的版本执行的命令为:

```
[root@localhost ~]# bash - version
GNU bash, 版本 4.2.46(2) - release (x86_64 - redhat - linux - gnu)
Copyright (C) 2011 Free Software Foundation, Inc.
```

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。

很多时候,初学者容易混淆 Shell 和命令行这两个概念,一般来说,当一个用户登录后,系统将启动一个默认的 Shell 程序,可以看到 Shell 的提示符(管理员为#,普通用户为\$),在提示符后输入一串字符后,Shell 将对这一串字符进行解释,而输入的这一串字符就称为命令行。尽管 Linux 的 GUI 功能也很强大,但控制 Linux 的最好方法是使用命令行界面,命令行操作的运行不需要占用过多的系统资源,功能也十分强大,几乎所有的 Linux 操作都可以通过命令行来完成,在计算机远程管理和服务器环境操作中 Linux 命令行的优势尤其明显。熟练掌握 Linux 命令行操作是领会 Linux 系统精髓的必然途径。

3.2 Shell 功能简介

Shell 主要有两个功能,作为命令解释器,它一端连接着 UNIX/Linux 内核,另一端连接着用户和其他应用程序,换句话说,Shell 是用户和应用程序与内核沟通的桥梁。

在 Linux 命令中,Shell 作为命令解释器的具体功能为:它接收用户输入的命令,进行分析,创建子进程,由子进程实现命令所规定的功能,等子进程终止后,发出提示符。这是 Shell 最常见的使用方式。

另外,Shell 作为一种高级程序设计语言,可以编写出代码简洁、功能强大的程序。它属于 UNIX/Linux 下的脚本编程语言,它是解释执行的,无须提前编译。

Shell 虽然没有 C/C++、Python、Java、C# 等编程语言强大,但也支持了基本的编程元素,Shell 脚本实现的功能非常强大,完全能够胜任 Linux 的日常管理工作,如文本或字符串检索、文件的查找或创建、大规模软件的自动部署、更改系统设置、监控服务器性能、发送报警邮件、压缩安装文件等。

3.3 简单命令

下面介绍一些常用的简单命令,这些命令只输入命令名,可以不加选项或参数,按 Enter 键就可以正常执行。

1. pwd 当前路径命令

pwd 命令的功能是显示当前工作的全路径名。例如,在系统中登录以后马上输入命令:

```
[wdg@localhost ~]$ pwd
/home/wdg
```

表明当前所处的目录是 wdg 用户的主目录/home/wdg。Shell 提示符 \$ 前面的字符串 “[wdg@localhost ~]”中的“~”表示当前为宿主(用户主)目录,如果处于其他目录则显示目录名,“@”前面的“wdg”表示用户名,“@”后面的“localhost”表示主机名。由于在不同时间、不同环境中所用的工作目录会有很大的差异,因此由 pwd 命令显示的结果就因具体情况而异。

需要注意的是,在所有命令行字符串的结尾都要按 Enter(回车)键,系统才对该命令加以接收、分析、执行。因此这里所讲到的命令行示例都省去 Enter,作为默认方式,用户在实际上机操作时必须要在输入命令之后按下 Enter 键。

2. date 日期命令

date 命令的功能是显示系统当前的日期和时间。例如:

```
[wdg@localhost ~]$ date
2018 年 11 月 26 日  星期一 06:52:54 CST
```

日期是 2018 年 11 月 26 日,时间是 6 时 52 分 54 秒。

3. who 在线用户命令

who 命令的功能是显示当前已登录到系统的所有用户名、所有终端名和登录到系统的时间。例如:

```
[wdg@localhost ~]$ who
root  tty1    2018-11-26 06:54
root  pts/0    2018-11-26 06:49 (192.168.1.100)
wdg   pts/1    2018-11-26 06:55 (192.168.1.100)
```

这段代码表明目前有 3 个用户在系统中: root(超级用户)使用 tty1 本地终端登录,登录时间为 2018 年 11 月 26 日 6 时 54 分;另一个 root 用户使用 pts/0 异地终端登录,登录时间为 2018 年 11 月 26 日 6 时 49 分;wdg 用户使用 pts/1 异地终端登录,登录时间为 2018 年 11 月 26 日 6 时 55 分。

4. cal 日历命令

cal 命令的功能是显示日历。它可以显示公元 1~9999 年中任意一年或任意一个月的日历。可以不带任何参数直接使用该命令。例如:

```
[wdg@localhost ~]$ cal
十一月 2018
日 一 二 三 四 五 六
      1  2  3
  4  5  6  7  8  9 10
 11 12 13 14 15 16 17
 18 19 20 21 22 23 24
 25 26 27 28 29 30
```

从中可以看出,如果 cal 命令不带任何参数直接使用则显示本月的日历。在 cal 命令之后可以有一个表示年份的数字,指定显示某一年全年的日历,例如 cal 2019。注意,年份 2019 不能缩写为“19”,因为“cal 19”将显示公元 19 年的日历。可以指定显示某一年某一月的日历,如 cal 8 2019,将显示 2019 年 8 月份的日历。

注意,如果在异地终端执行以上命令,汉字可能出现乱码,需要执行如下命令之后方可正常显示汉字,该方式仅对该终端有效,若打开新的异地终端,需再次执行该命令:

```
[wdg@localhost ~]$ export LANG = zh_CN
```

5. uname 系统信息命令

uname 命令查看当前操作系统的信息,它可带多个选项。常用选项有以下几个。

- r: 显示发行版本号。
- m: 显示所用机器类型。
- i: 显示所需硬件平台。
- v: 显示操作系统版本。

例如:

```
[wdg@localhost ~]$ uname
Linux
[wdg@localhost ~]$ uname -r
3.10.0-693.el7.x86_64
[wdg@localhost ~]$ uname -mi
x86_64 x86_64
[wdg@localhost ~]$ uname -v
#1 SMP Tue Aug 22 21:09:27 UTC 2017
```

6. wc 统计命令

wc 命令用来统计给定文件的行数、字数和字符数,其格式为:

```
wc [ -lw ] [ -c ] 文件名
```

选项含义:l 为统计行数,w 为统计字数,c 为统计字节数;如果没有给出文件名,则读取标准输入。例如:

```
[wdg@localhost ~]$ wc file1
34 403 4075 file1
```

输出的列的顺序和数目固定不变,分别为行数、字数、字节数和文件名。又如:

```
[wdg@localhost ~]$ wc -w file1
403 file1
```

7. su 切换用户命令

su 命令用来切换当前用户的登录身份(前提是系统已经创建了其他用户),其常用的格

式为:

```
su [ - ] [ 用户名 ]
```

选项含义:“-”表示切换时是否带切换用户的环境变量;如果不带用户名,是指普通用户切换到 root 用户。例如:

```
[root@localhost ~]# su - wdg          # 切换到 wdg 用户,并转换其切换用户的环境
[wdg@localhost ~]$ pwd
/home/wdg
[wdg@localhost ~]$ su                # 切换到 root 用户,不转换其环境
password:
[root@localhost wdg]# pwd
/home/wdg
```

8. clear 清屏命令

clear 命令用来清除字符界面上的所有内容,只保留当前提示符,并显示在屏幕的第一行上。

3.4 Shell 命令的操作基础

Shell 命令作为使用 Linux 操作系统的最基本方式之一,用户需要熟练掌握常用的 Shell 命令,从而高效地管理 Linux 系统。

3.4.1 Shell 命令的一般格式

Linux 命令又称为 Shell 命令,当用户登录后 Shell 运行进入了内存,它遵循一定的语法将输入的命令加以解释并传给系统。

前面介绍的几个简单命令,只要在命令提示符后面输入命令名,然后按 Enter 键就可以执行了。其实许多命令的命令行还需要选项和参数。命令行中输入的第一个项目必须是一个命令的名称,第二个字是命令的选项或参数,命令行中的每个字必须由空格或 Tab 隔开,格式如下:

```
命令名称 [选项] [参数]
```

选项是一种标志,常用来扩展命令的特性或功能。[选项]的方括号表示语法上选项可有可无。选项往往包括一个或多个英文字母,在字母前面有一个减号(减号是必要的, Linux 用它来区别选项和参数)。例如,没有选项的 ls 命令,可列出目录中的所有文件,只列出各个文件的名称,而不显示其他更多的信息。而“ls -l”命令可以列出包含文件大小、权限、修改日期等更多信息的文件或文件夹列表。

有时也可以把几种表示不同含义的选项字母组合在一起对命令发生作用,例如:

```
[wdg@localhost ~]$ ls -la
```

大多数命令都可以接纳参数。参数是在命令行中的选项之后输入的一个或多个单词，例如，“ls -l /tmp”显示 tmp 目录下的所有文件及其信息。其结果与先进入 tmp 文件夹再执行“ls -l”的结果一致。

有一些命令可能会限制参数的数目。例如，cp 命令至少需要两个参数：

```
[wdg@localhost ~]$ cp oldfile newfile
```

在命令行中，选项要先于参数输入。在一个命令行中还可以置入多个命令，用分号“;”将各个命令隔开，例如：

```
[wdg@localhost ~]$ date;who;pwd
2018 年 11 月 26 日 星期一 07:09:45 CST
root    tty1    2018-11-26 06:54
root    pts/0    2018-11-26 06:49 (192.168.1.100)
wdg     pts/1    2018-11-26 06:55 (192.168.1.100)
/home/wdg
```

3.4.2 在线帮助命令

用户需要掌握许多命令来使用 Linux 操作系统。为了方便用户，Linux 提供了功能强大的在线帮助命令——man 命令，它可以查找到相应命令的语法结构、主要功能、主要选项说明。另外，部分命令还列举了全称以及此命令操作后所影响的系统文件等信息。格式如下：

man 命令名

例如：

```
[wdg@localhost ~]$ man who
```

通常 man 显示命令帮助的格式包含以下四部分。

- (1) NAME: 命令名称。
- (2) SYNOPSIS: 语法大纲。
- (3) DESCRIPTION: 描述说明。
- (4) OPTIONS: 选项。

除了这四部分以外，man 命令通常还会包含一些教学示例，以帮助用户进一步了解该命令的语法。man 在查询控制手册时给出了一些功能键设置，用于控制手册页滚动的主要键如下。

- 空格键：显示手册页的下一屏。
- 回车键：一次滚动手册页的一行。
- q 键：退出 man 命令。

除 man 命令外，有些命令可以用“--help”选项提供该命令的帮助信息，用户还可以用 info 和 whatis 等命令查询一些命令的帮助信息。

3.4.3 与 Shell 有关的配置文件

在 Linux 操作系统中,主要有以下几个与 Shell 有关的配置文件。

(1) /etc/profile 文件。这是系统最重要的 Shell 配置文件,也是用户登录系统最先检查的文件,文件中存放的是系统的环境变量,对所有用户都有效果,要对其更改的话,必须要在 root 用户权限下才能进行。可以使用 env 命令来查看系统当前所有的环境变量。

(2) ~/.bash_profile 文件。每个用户的 Bash 环境配置文件存在于用户的主目录中,当系统运行/etc/profile 后,将读取此文件的内容,此文件定义了 USERNAME\BASHENV 和 PATH 等环境变量,此处的 PATH 包括了用户自己定义的路径以及用户的“bin”路径。

(3) ~/.bashrc 文件。前两个文件仅在系统登录时读取,此文件将在每次运行 Bash 时读取,此文件主要定义的是一些终端设置以及 Shell 提示符等,而不定义环境变量等内容。

(4) ~/.bash_history 文件。记录了用户使用的历史命令。

3.5 Shell 命令的高级操作

Linux 系统除了提供丰富的 Shell 命令外,还提供了强大的 Shell 高级操作的扩展功能,这样不仅为用户提供方便,同时也丰富了 Shell 功能。

3.5.1 Shell 的命令补全

Linux 命令较多,有的较长,有时容易输错。其实在 Bash 中,用户在使用命令或输入文件名时不需要输入完整信息,可以让系统来补全最符合的名称,如果有多个符合,则会显示所有与之匹配的命令或文件名。例如,用户首先输入命令的前几个字母,然后按 Tab 键,如果与输入字母匹配的仅有一个命令名或文件名,系统将自动补全;如果有多个与之匹配,系统将发出报警声音;如果再按一下 Tab 键,系统将列出所有与之匹配的命令或文件名,从而方便用户操作。

例如:

```
[wdg@localhost ~]$ if          # 先输入 if,再按 Tab 键,将发出声音表示有匹配
if      ifconfig  ifenslave  ifport  ifuser
ifcfg   ifdown    ifnames    ifup
[wdg@localhost ~]$ ifconfig    # 参照对比输入所需要的 ifconfig 命令显示网卡配置
```

3.5.2 Shell 的历史命令

用户在命令行操作中输入的所有命令,系统都会将其自动记录到用户宿主目录下的一个文件中(~/.bash_history),记录的多少由用户环境变量中的 HISTSIZE 决定。在输入命令时,用户可以通过方向键的上下箭头来选择最近使用过的命令,即可完成自动输入历史命令;还可以在提示符下输入 history 命令查看所有历史命令。例如:

```
[root@localhost ~]# history | more
 1 hostname CentOS - 7
 2 bsh
 3 hostname
 4 hostname Linux - CentOS - 7
 5 bash
 6 cal
 7 vi /etc/locale.conf
 8 source /etc/locale.conf
 9 date
10 cal
11 cat /boot/grub2/grub.cfg
12 cat /etc/grub2.cfg
...
```

若执行以前历史命令列表中的某一个命令,则执行“!n”,n 为历史命令列表中的编号,如执行本例中的编号 9 的历史命令:

```
[root@localhost ~]# !9
date
2018 年 11 月 26 日 星期一 08:34:30 CST
```

3.5.3 Shell 的重定向

输入输出重定向(I/O Redirection)可以让用户从文件输入命令,或将输出结果存储在文件及设备中,从而摆脱了只有标准输入(键盘)和输出(显示器)设备的模式。其中,输出重定向符号有“>”和“>>”,“>>”称为重定向附加;而输入重定向符为“<”。另外,还有错误重定向输出“2>”,可以把命令行出错的信息保存到指定的文件中。Shell 的重定向详见 7.4.2 节。下面为一个重定向的例子。

“>”将输入的信息直接写入,“>>”将输入的信息以追加的方式写入。例如:

```
[wdg@localhost ~]$ ls
example.c m1.c m2.c m3.c
[wdg@localhost ~]$ ls > test           # 把当前目录清单信息写入 test 文件中
[wdg@localhost ~]$ cat test
example.c m1.c m2.c m3.c
[wdg@localhost ~]$ cal >> test       # 把当月日历信息附加到 test 文件后
[wdg@localhost ~]$ cat test         # 查看处理后的 test 文件内容
example.c m1.c m2.c m3.c
十一月 2018
日 一 二 三 四 五 六
      1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

3.5.4 Shell 的管道操作

管道线“|”可以将多个简单的命令集合在一起,用以完成较复杂的功能。管道线“|”前面命令的输出是管道线“|”后面命令的输入。格式为:

```
命令 1 | 命令 2 | 命令 3 | ... | 命令 n
```

例如,对当前月的日历输出进行统计。

```
[wdg@localhost ~]$ cal
 十一月 2018
日 一 二 三 四 五 六
    1  2  3
  4  5  6  7  8  9 10
 11 12 13 14 15 16 17
 18 19 20 21 22 23 24
 25 26 27 28 29 30

[wdg@localhost ~]$ cal | wc
   8   39  145
```

如果把“cal | wc”的输出再当作另一个管道线后面命令的输入,进行 wc 命令的信息统计,其示例如下:

```
[wdg@localhost ~]$ cal | wc | wc
   1    3   24
```

对照以上两个示例,比较统计的结果。

3.6 Linux 的桌面系统

Linux 系统遵循 UNIX 系统给用户提供的基于文本的命令界面,同时也为用户提供了方便、灵活、直观的图形环境——桌面系统。用户可以利用鼠标、菜单、图标等图形用户界面来使用 Linux 操作系统。

一般情况下,Linux 的桌面系统是独立于其内核之外的服务组件,和微软的 Windows 相比,Linux 的桌面系统整体功能相差不多,它们之间的区别在于:Windows 的桌面图形系统作为其内核的一部分,而 Linux 只是将桌面系统作为其系统下的一个应用程序。目前常见的 Linux 的桌面版本有 GNOME、KDE 和 Xfce 等,它们都是基于 X-Window 协议基础的桌面系统。

3.6.1 X-Window 系统简介

很多人使用计算机是从微软的 Windows 操作系统开始的,但实际上,UNIX 系统中使用窗口形式的 GUI 要早于 Windows 操作系统。在“X-Window System”这个词中不要误用

Windows, 因为 Windows 是微软公司的注册商标。尽管大多数专业 Linux 操作人员喜欢命令行界面, 但是初学者往往更喜欢图形用户界面(GUI), 尤其是某些用户使用 Linux 的目的只是办公和娱乐, 这时候 GUI 是更好的选择。Linux 提供的 GUI 解决方案是 X-Window System(也称为 X-Window 系统)。

X-Window 系统于 1984 年在麻省理工学院(MIT)开始开发, 是基于 UNIX 系统的一个图形系统的组件, 它是一个独立的程序, 且易于移植, 即使其运行时出现故障, 也不会影响操作系统的正常运行。X-Window 由服务器、客户端和通信通道三部分构成。服务器的主要作用是控制显示器、输入设备等; 客户端主要是一些功能应用程序; 通信通道主要是为服务器与客户端进行数据传输。

X-Window 系统是一套工作在 UNIX 计算机上的优良的窗口系统。严格意义上讲, X-Window 并不是一个软件, 而是一个协议, 常称为 X 协议, 现在是类 UNIX 系统中图形用户界面的工业标准。X-Window 系统最重要的特征之一是它的结构与设备无关。任何硬件只要和 X 协议兼容, 就可以执行 X 程序并显示一系列包含图文的窗口, 而不需要重新编译和链接。这种与设备无关的特征, 使得依据 X 标准开发的应用程序, 可以在不同环境下执行, 因而奠定了 X-Window 系统成为工业标准的地位。

X-Window 系统具有以下主要特征。

(1) X-Window 系统具有网络操作的透明性。应用程序的窗口可以显示在自己的计算机上, 也可以通过网络显示在其他计算机的显示器上。

(2) 支持许多不同风格的操作界面。X-Window 系统只提供建立窗口的一个标准, 至于具体的窗口形式则由窗口管理器决定。在 X-Window 系统上可以使用各种窗口管理器, 即桌面系统。

(3) X-Window 系统不是操作系统必需的构成部分。对操作系统而言, X-Window 系统只是一个可选的应用程序组件。

(4) X-Window 系统现在是开源项目, 可以通过网络或者其他途径免费获得源代码。

3.6.2 Linux 的桌面环境

X-Window 系统为 GUI 界面提供最基本的支持, 而具体的图形化工具的支持, 则需要借助于桌面环境。

几乎在所有的 Linux 发行版本中 X-Window 的图形界面都提供 GNOME 和 KDE 桌面, 它们都是完全免费的, 可以说它们是 Linux 系统的应用程序, 在实际应用中可以切换选择使用它们。一般在 Linux 系统定制安装时可以选择 GNOME 和 KDE 二者之一, 也可以二者兼选。若兼选, 则若要使用 KDE 的一些特征, 可以运行命令 `kdesktop` 调出 KDE。这两类桌面环境的大部分操作彼此互相适用。下面分别介绍这两个桌面环境。

1. GNOME

GNOME 即 GNU 网络对象模型环境(The GNU Network Object Model Environment), 是 GNU 计划的一部分, 是开放源码运动的一个重要组成部分, 是一种让使用者容易操作和设定计算机环境的工具。GNOME 的目标是基于自由软件, 为 UNIX 或者类 UNIX 操作系统构造一个功能完善、操作简单以及界面友好的桌面环境, 它是 GNU 计划的正式桌面。

GNOME 桌面是自由软件桌面系统, 支持 UNIX 和 Linux 操作系统。GNOME 设计的

宗旨是给用户提供简单、易用、稳定的桌面系统,它使用 X-Window 服务器。

GNOME 桌面系统是使用 C 语言编程的,但也存在一些其他语言的绑定使得能够使用其他语言编写 GNOME 应用程序,如 C++、Java、Ruby、C#、Python、Perl 等。GNOME 桌面系统的 LOGO 如图 3-2 所示。

2. KDE

KDE 是 K Desktop Environment(K 桌面环境)的缩写,是一种著名的运行于 Linux、UNIX 以及 FreeBSD 等操作系统上的自由图形工作环境。KDE 和 GNOME 都是 Linux 操作系统上最流行的桌面环境系统。

KDE 现在是 UNIX 上易于使用的现代桌面环境。和一些如 GNU/Linux 这样的自由的类 UNIX 一起,UNIX/KDE 组成了一个对于任何人都可用的完全自由和开放的计算平台,而且完全免费,任何人都可以修改它的源代码。

因为 KDE 应用程序开发框架的优势,已经有大量的应用程序存在于 KDE 桌面环境了。KDE 的基本发行版中包含了这些程序的一个选择。现在 KDE 也拥有了一个基于 KDE 的 KParts 技术的,由电子表格、幻灯片制作程序、组织者、新闻客户端和更多应用组成的办公应用套件。KDE 桌面系统的 LOGO 如图 3-3 所示。



图 3-2 GNOME 的 LOGO



图 3-3 KDE 的 LOGO

CentOS 7 系统默认安装使用的 X 窗口界面就是 GNOME,KDE 可以在定制安装时选择安装。如果两个都安装了,两个桌面环境可以切换使用。本章下面的叙述都是以 GNOME 为例介绍 CentOS 7 系统的图形环境桌面系统的常用功能的使用。CentOS 7 的 GNOME 桌面如图 3-4 所示。

3.6.3 GNOME 桌面环境简介

GNOME 桌面环境的工作方式和使用微软 Windows 操作系统时所预想的一样。用户可以把文件程序的图标拖放到使用方便的地方,可以为文件和程序在桌面、面板和文件管理器中添加图标,可以改变多数工具和应用程序的外观,还可以使用系统提供的配置工具来改变系统设置。

CentOS 7 系统的 GNOME 图形化桌面环境默认提供了 5 种主要工具来使用应用程序:控制面板、桌面图标、任务条、工作区切换器以及 3.6.4 节将介绍的菜单系统。

1. 控制面板

横贯桌面顶部的长条称为控制面板。它是 GNOME 用户界面最核心的部分,如图 3-5 所示。它依次包含“应用程序”菜单、“位置”菜单,以及系统时间、网络控制器、音量控制器和



图 3-4 CentOS 7 的 GNOME 桌面

系统关机控制器 4 个功能图标。



图 3-5 GNOME 的控制面板

2. 桌面图标

桌面上的图标是用来链接到文件夹、应用程序、回收站、光盘或软盘等可移动设备(在它们被挂载后出现)的快捷途径。初始状态下,CentOS 7 系统的桌面图标如图 3-4 所示,主要有回收站、主文件夹(登录用户的宿主目录)或安装时挂载的 ISO 映像文件光盘。要打开一个文件夹或启动一个应用程序,双击相应的图标即可。

3. 任务条

任务条是 GNOME 桌面底部的一个长条,它显示任意虚拟桌面上(工作区)运行应用程序的名称。用户可以单击任务条上的名称来使相应的应用程序重现在桌面上以及工作切换区等,如图 3-6 所示。



图 3-6 GNOME 的任务条

4. 工作区切换器

在 GNOME 图形化桌面环境中提供了使用多个工作区的能力,因此不必把所有运行的应用程序都堆积在一个可视桌面区域内。CentOS 7 桌面默认有 4 个工作区,在图 3-6 所示的界面中右侧“1/4”图标的含义为当前状态显示的是 4 个工作区的第一区,而这个工作区当前有两个运行的应用程序,颜色深的为活动状态,单击右侧的“1/4”图标,则弹出 4 个工作区

的窗口菜单,如图 3-7 所示,单击可以切换相应的工作区。



图 3-7 工作区切换器

3.6.4 GNOME 桌面中的菜单系统

CentOS 7 系统的 GNOME 桌面环境中的菜单系统包括 3 个:“应用程序”菜单、“位置”菜单和“系统快捷控制器”菜单。它们各自的内容如下。

1. “应用程序”菜单

在单击最左侧的“应用程序”图标后,GNOME 会出现许多程序组,而每个组还包括其他程序内容,如图 3-8 所示。以下将对系统默认项进行说明。



图 3-8 “应用程序”菜单

在图 3-8 中显示的是“应用程序”菜单中的“系统工具”组中的程序,是系统安装时选择安装的软件。

虽然 CentOS 7 提供了功能强大的桌面环境,但是熟悉 Linux 环境的用户仍然习惯使用命令行模式。另外,部分 Linux 系统管理只能在命令行模式下完成。因此,本书在以后章节中将重点介绍 SSH 方式远程终端命令模式下的操作,当然,用户同样可以在桌面环境下的终端模拟器中使用命令行,即打开新的终端运行命令。

1) 字符终端模拟器

在“应用程序”菜单中的“系统工具”选项中选择“终端”选项,或者在桌面空白处右击,在弹出的菜单中选择“打开终端”选项,即可打开终端命令窗口 Shell 提示。要退出 Shell 提示,单击 Shell 提示窗口右上角的关闭按钮,或在提示行中输入“exit”,以及按 Ctrl+D 组合键,这 3 种方法均可退出。如图 3-9 所示为 X 图形界面下的字符终端模拟器软件。

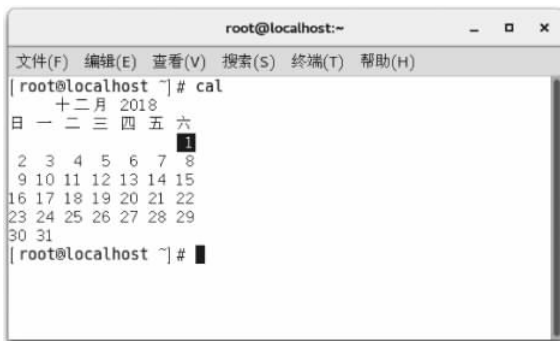


图 3-9 X 图形界面下的字符终端模拟器软件

2) 系统管理工具

在“应用程序”菜单中的“系统工具”选项中选择“设置”选项,将打开系统管理工具,如图 3-10 所示。系统管理中包括系统的全部设置,这是桌面环境中的系统管理主要窗口,该窗口的功能由个人、硬件和系统三部分组成,每个部分都由多个功能模块选项组成,通过这些功能模块可以完成系统的基本设置。当然,桌面环境的系统设置对于初学者来说相对容易,但是若想快捷、高效地对系统进行设置,还是用字符命令方式最好。

3) 设置系统 IP 地址

在图 3-10 的系统管理工具窗口中,单击“硬件”选项区域中的“网络”图标,则打开如图 3-11 所示的网络设置窗口。要设置系统的静态 IP 地址,则单击右下角的设置图形按钮,在弹出的新窗口中选择“IPv4”选项,如图 3-12 所示,单击右上角的“打开”按钮,并选择其下的“手动”下拉菜单选项,然后再填写“地址”“子网掩码”“网关”等参数信息,最后单击“应用”按钮,使参数设置生效。

2. “位置”菜单

在如图 3-13 所示的桌面环境中,单击“位置”按钮后会弹出多种选项的“位置”菜单。“位置”菜单的主要功能是显示登录的用户宿主目录文件列表及其权限之内的经常使用的工具软件列表,登录系统不同的用户,其“位置”菜单的内容有所不同。图中为 root 用户的“位置”菜单文件列表,其中,“主文件夹”为“/root”目录,“视频”“图片”“文档”等为系统安装时在 root 用户目录下自动生成的子文件夹。因为安装时选择的是“中文界面”安装系统,则生成中文的子文件夹;若用“英文界面”安装系统,则生成英文的子文件夹。“位置”菜单中的“计算机”为系统根目录。选择“位置”菜单中的“主文件夹”选项,则弹出如图 3-14 所示的窗口,为当前 root 用户的宿主目录文件窗口。

3. “系统快捷控制器”菜单

在图 3-13 所示的界面的右上角有一组系统快捷控制器图标,单击该图标则弹出一组菜



图 3-10 系统管理工具窗口



图 3-11 网络设置窗口

单,如图 3-15 所示。菜单分三部分:系统音量控制器;有线 已连接、未在使用、root; 3 个系统控制图标。

单击“有线 已连接”则展开其子菜单,包括“关闭”网络连接、“有线控制”网络配置;“未在使用”的子菜单是系统蓝牙设置内容;“root”表示当前登录的账户名,其子菜单包括“注销”用户、“账号设置”功能。

3 个系统控制图标包括“系统设置”“屏幕锁定”以及“关闭系统”,其中,单击“关闭系



图 3-12 设置静态 IP 地址窗口



图 3-13 “位置”菜单

统”，则弹出如图 3-16 所示的窗口，若系统中有其他用户登录，则关闭系统窗口将列出当前在线用户列表，以及系统关机倒计时 60 秒，此时若单击“关机”按钮，则系统进入关机状态。

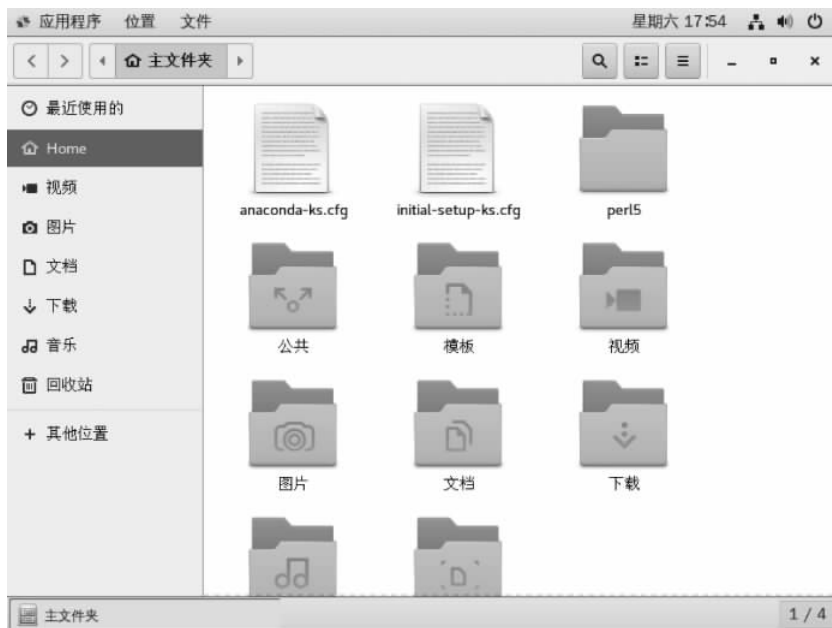


图 3-14 root 用户“位置”菜单下的“主文件夹”窗口



图 3-15 “系统快捷控制器”菜单



图 3-16 “关闭系统选项”窗口

3.6.5 GNOME 桌面的中英文版切换

CentOS 7 系统安装时可以选择语言类型,也可以安装后再选择语言类型,前提条件是安装时必须选择相关语言字库包。如果系统安装选择的是简体中文,在远程终端窗口下的中文提示经常会出现乱码,必须进行编码设置才能正常显示,而且原中文系统下的文件及文件夹的中文命名,在字符终端下操作也很不方便,所以根据个人的习惯要求,需进行必要的系统语言格式类型的重新设置。

下面以安装系统时选择的是中文简体版的系统安装为例,在系统安装完后进入 GNOME 桌面环境中,默认简体中文版切换成英文版的操作步骤如下。

(1) 打开“区域和语言”设置窗口。

打开“应用程序”菜单,选择“系统工具”选项,然后在其子菜单中选择“设置”选项,弹出

如图 3-17 所示的“全部设置”界面，在图中选择“区域和语言”选项，出现如图 3-18 所示的窗口。



图 3-17 “全部设置”界面



图 3-18 “区域和语言”窗口

(2) 设置语言类型。

在图 3-18 所示的界面中，上部设置系统的语言类型有两个，分别是“语言”和“格式”。单击“语言”选项则弹出如图 3-19 所示的窗口，选择所要的语言类型，如“English”，单击“完

成”按钮,则提示重新启动系统后生效。以上“语言”设置只是系统的界面菜单的语言类型。“格式”选项可以选择页面输出的格式语言类型,如日期的输出格式的语言类型等,该选项设置也非常重要,如果用户终端方式的输出含有中文,且经常出现乱码,则此项可以设置成英文格式。



图 3-19 选择“语言”类型的窗口

(3) 格式类型的转换。

完成语言及格式的设置后,重新启动系统进入桌面后,则转换成设置的语言格式类型,但出现提示转换窗口,如图 3-20 所示,提示原来的语言及格式类型所对应的文件夹的语言类型是否进行转换。这里为了终端操作方便,建议文件夹等语言类型转换成英文状态。

3.6.6 GNOME 桌面下的软件安装

CentOS 7 系统的 GNOME 桌面环境下的软件安装可以通过多种方法进行。一种常用的方法是选择桌面的“应用程序”→“系统工具”→“软件”,则出现“软件”窗口,该软件窗口是在系统连接外网的前提下,通过系统设置好的“软件仓库”进行搜索查看安装软件,但是没有连接外网的系统该功能不可用。

另一种常用的方法是通过系统安装光盘选择所需要的安装程序软件,在 2.1.1 节中已经介绍了 CentOS 7 系统安装程序的各种映像版本,一般安装的映像文件中已经集成了大多数系统所需要的软件,且通过映像文件安装系统后进入桌面环境会自动默认识别原安装的映像文件光盘,如图 3-4 所示。其通过光盘安装指定软件的步骤如下。

(1) 打开映像光盘中的软件包文件夹。

在图 3-4 所示的界面中双击桌面上的光盘图标,则打开如图 3-21 所示的界面,在光盘内的根目录下的“Packages”文件夹就是集成了系统安装的 RPM 软件包,打开该软件包文



图 3-20 语言格式类型的转换提示窗口

件夹,则出现如图 3-22 所示的 RPM 软件目录列表窗口。

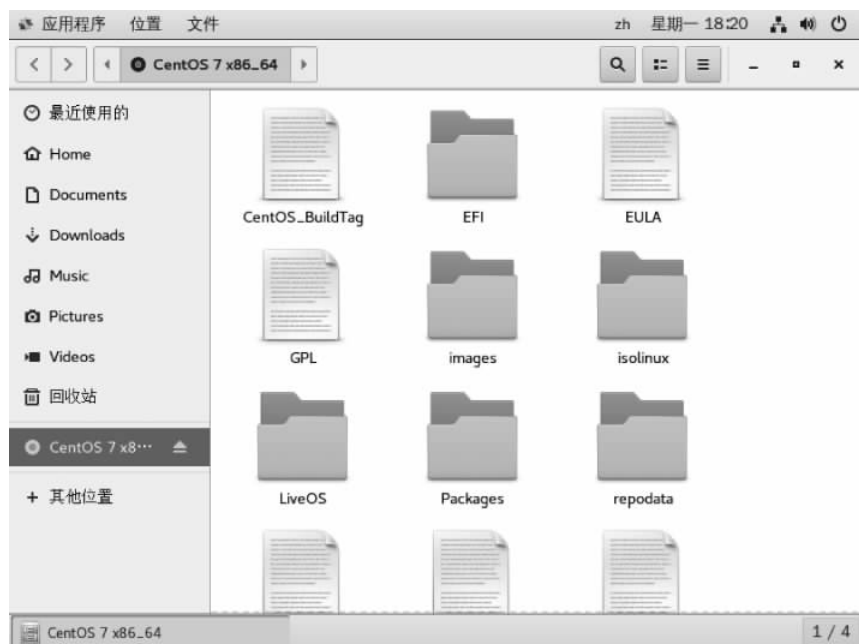


图 3-21 系统安装映像文件光盘界面

(2) 搜索指定的软件包。

在 RPM 软件包文件列表窗口中单击“搜索”图标,在其下出现的提示栏中输入所要指定软件包的关键字,则下面自动匹配该关键字的软件包,如图 3-22 所示。

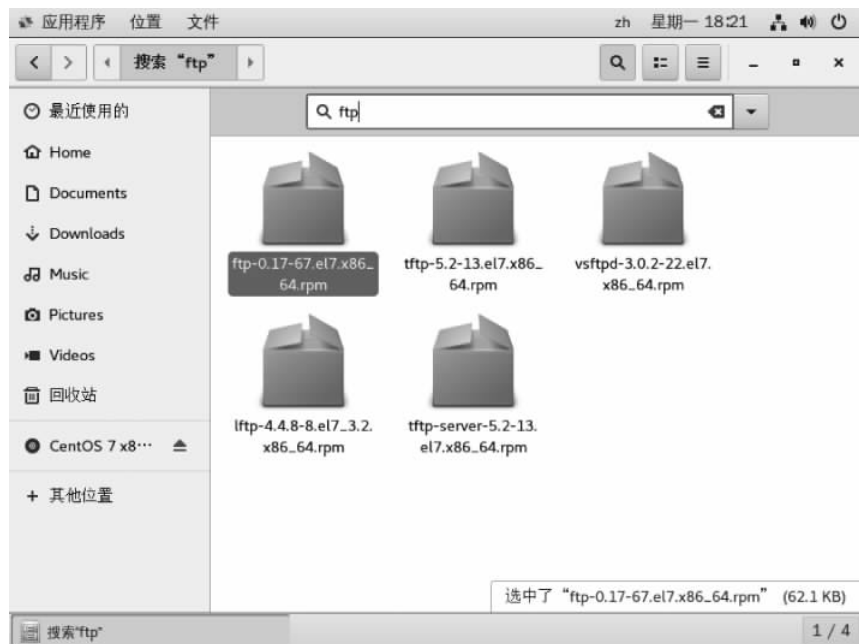


图 3-22 搜索指定的软件包

(3) 安装软件包。

在图 3-22 所示的界面中搜索到所要的软件包后,双击该软件包即可安装该程序,若提示不支持该软件包的安装时,可以把该软件包复制到本地指定目录,如/root 目录下,通过终端 RPM 命令方式进行安装,如图 3-23 所示。RPM 命令方式安装软件包的方法详见 5.2.2 节中的内容。



图 3-23 终端命令方式安装软件包

3.7 本章小结

本章主要介绍了 Linux 操作系统与 Shell 的关系,强调了命令行操作的重要性,然后叙述了常用的简单命令、一般命令格式、Shell 高级操作、Linux 的 X-Window、GNOME 桌面环境及系统菜单等内容。通过本章的学习,读者可以了解到 Linux 操作系统与 Shell 的关系,学会一些简单命令、常用命令以及 X 窗口下的图形操作。

因为读者都熟悉 Windows 操作系统,所以本书的图形界面只在本章做了简单介绍,以后的章节中主要以远程终端形式使用 Linux 操作系统,所以要求读者重点掌握 Shell 命令相关操作等内容。

3.8 思考与实践

1. Shell 有哪些高级操作?
2. 对一个命令不熟悉,有哪些方法可以获得帮助?
3. 在 CentOS 7 系统桌面环境下进行如下操作:设置系统时间、更换桌面背景、工作区切换、目录的切换、文件的查看等,比较该系统和 Windows 系统的基本操作区别。
4. 在 CentOS 7 系统下,分别在远程 SSH 终端以及本地桌面环境下以多用户方式登录(登录的用户分为普通用户及 root 用户),再在不同终端关闭系统,观察此时多用户在线状态下普通用户及 root 用户系统关机的提示及状态变化情况。
5. 在 CentOS 7 系统的桌面环境下,进行系统的语言格式由中文版转换成英文版的重新设置操作。设置成功后观察设置前后的远程终端的输出变化,即把原来终端的中文输出转换成英文输出的格式。
6. 在 CentOS 7 系统的桌面环境下,打开安装系统时的映像文件光盘,查找有关 ftp 文件的 RPM 软件包,选择其一并进行安装。