

## 第 1 章

## 初识Linux shell



本章内容

什么是Linux

Linux内核的组成部分

探索Linux桌面

了解Linux发行版

**在**深入研究如何使用Linux命令行和shell之前，最好先了解一下什么是Linux、它的历史及运作方式。本章将带你逐步了解什么是Linux，并介绍命令行和shell在Linux整体架构中的位置。

## 1.1 什么是 Linux

如果以前从未接触过Linux，你可能会对为什么会存在这么多不同的Linux发行版有些困惑。在看Linux软件包时，你肯定听过发行版、LiveCD和GNU之类的术语，也肯定被搞晕过。第一次接触Linux的人理解这些会有些困难。本章将在你了解命令和脚本之前，揭示Linux系统内部结构的一些信息。

首先，Linux可划分为以下四部分：

Linux内核；

GNU工具组件；

图形化桌面环境；

应用软件。

在Linux系统里，这四部分中的每一部分都扮演着一个特别的角色。如果将它们分开来，每一部分都没太大的作用。图1-1是Linux系统的基本结构框图，说明了各部分是如何协作起来构成整个Linux系统的。

本节将会详细介绍这四部分，然后将概述它们是如何一起协作构成一个完整的Linux系统的。

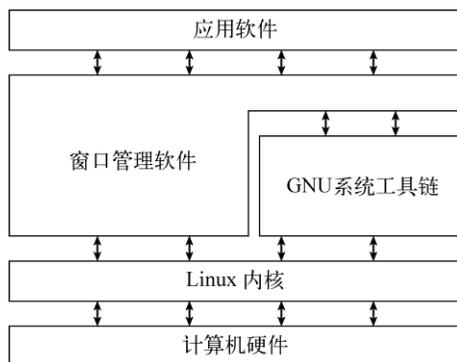


图1-1 Linux系统

### 1.1.1 深入探究 Linux 内核

Linux系统的核心是内核。内核控制着计算机系统上的所有硬件和软件：必要时分配硬件，有需要时执行软件。

## 4 第1章 初识 Linux shell

---

如果你一直都在关注Linux世界，那么毫无疑问，你肯定听说过Linus Torvalds。Linus还在赫尔辛基大学上学时就开发了第一版Linux内核。起初他只是希望Linux成为Unix操作系统的一份副本，因为当时Unix操作系统在很多大学都很流行。

当Linus完成了开发工作后，他将Linux内核发布到了互联网社区并征求改进意见。这个简单的动作引发了计算机操作系统领域内的一场革命。很快，Linus就收到了来自世界各地的学生和专业程序员的各种建议。

如果Linux内核允许任何人修改内核程序代码，那么随之而来的将是完全的混乱。简单起见，Linus担当起了所有改进建议的把关员。能否将建议代码放进内核完全取决于Linus的决定。时至今日，这种概念依然在Linux内核代码开发过程中继续使用，所不同的是，现在是由一组开发人员来做这件事，而不再是Linus一个人了。

内核基本负责以下四项主要功能：

系统内存管理；

软件程序管理；

硬件设备管理；

文件系统管理。

后面几节将会进一步探究其中的每一项功能。

### 1. 系统内存管理

操作系统内核的基本功能之一是内存管理。内核不仅管理服务器上的可用物理内存，还可以

创建和管理虚拟内存（即并不实际存在的内存）。

内核通过硬盘上的存储空间来实现虚拟内存，这块区域称为交换空间（swap space）。内核不断地在交换空间和实际的物理内存之间反复交换虚拟内存存储单元中的内容。这使得系统以为它拥有比物理内存更多的可用内存（如图1-2所示）。

内存存储单元会被按组分分成很多块，这些块称作页面（page）。内核会将每个内存页面放在物理内存或交换空间。然后，内核会维护一个内存页面表，来指明哪些页面位于物理内存内，哪些页面被换到磁盘上。

内核会记录哪些内存页面正在使用中，并自动把一段时间未访问的内存页面复制到交换空间区域（称之为换出，swapping out）——即使还有可用内存。当程序要访问一个已被换出的内存页面时，内核必须从物理内存换出另外一个内存页面来给它让出空间，然后从交换空间换入（swapping in）请求的内存页面。显然，这个过程要花费时间，并使得运行中的进程变慢。只要Linux系统在运行，为运行中的程序换出内存页面的过程就不会停歇。

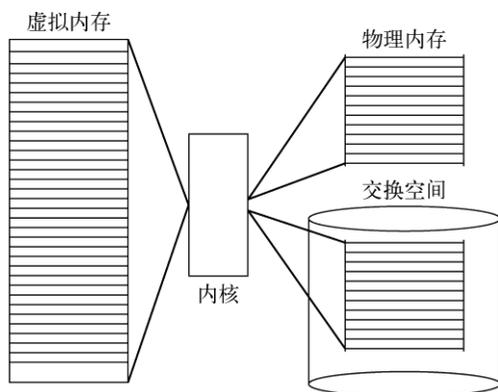


图1-2 Linux系统内存映射

## 6 第1章 初识 Linux shell

---

你可以查看专门的 `/proc/meminfo` 文件来观察Linux系统上虚拟内存的当前状态。下面是

`/proc/meminfo` 文件的一个样例。

```
rich@rich-desktop:~$ cat /proc/meminfo
MemTotal:      1026084 kB
MemFree:       666356 kB
Buffers:       49900 kB
Cached:        152272 kB
SwapCached:    0 kB
Active:        171468 kB
Inactive:      154196 kB
Active(anon):  131056 kB
Inactive(anon): 32 kB
Active(file):  40412 kB
Inactive(file): 154164 kB
Unevictable:   12 kB
Mlocked:       12 kB
HighTotal:     139208 kB
HighFree:      252 kB
LowTotal:      886876 kB
LowFree:       666104 kB
SwapTotal:     2781176 kB
SwapFree:      2781176 kB
Dirty:         588 kB
Writeback:     0 kB
AnonPages:    123500 kB
Mapped:        52232 kB
Shmem:         7600 kB
Slab:          17676 kB
SReclaimable: 9788 kB
SUnreclaim:   7888 kB
KernelStack:  2656 kB
PageTables:    5072 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp: 0 kB
CommitLimit:  3294216 kB
Committed_AS: 1234480 kB
VmallocTotal: 122880 kB
VmallocUsed:   7520 kB
VmallocChunk: 110672 kB
HardwareCorrupted: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 4096 kB
DirectMap4k:  12280 kB
```

```
DirectMap4M:      897024 kB  
rich@rich-desktop:~$
```

MemTotal:行表明这个Linux服务器有1 GB 的内存，该文件还表明大约有660 MB 的空闲空间 ( MemFree )。输出表明这个系统上大约有2.5 GB的交换空间 ( SwapTotal )。

默认情况下，运行在Linux系统上的每个进程都有各自的内存页面。进程不能访问其他进程正在使用的内存页面。内核维护着它自己的内存区域。出于安全考虑，用户进程不能访问内核进程使用的内存。

为了方便共享数据，你可以创建一些共享内存页面。多个进程可在同一块共用内存区域进行读取和写入操作。内核负责维护和管理这块共用内存区域并控制每个进程访问这块共享区域。

ipcs命令专门用来查看系统上的当前共享内存页面。以下是ipcs命令示例的输出：

```
# ipcs -m  
  
----- Shared Memory Segments -----  
key          shmid    owner    perms    bytes    nattch   status  
0x00000000  0        rich     600      52228    6        dest  
0x395ec51c  1        oracle   640      5787648  6  
  
#
```

每个共享内存段都有个所有者，也就是创建它的用户。每个段也都有标准的Linux权限设置来设定其他用户是否可以访问该段。这个键值用来限定其他用户是否可以访问共享内存段。

## 2. 软件程序管理

Linux操作系统称运行中的程序为进程。进程可以在前台运行，将输出显示在屏幕上；也可以在后台运行，隐藏到幕后。内核控制着Linux系统如何管理运行在系统上的所有进程。

内核创建了第一个进程（称为init进程）来启动系统上所有其他进程。当内核启动时，它会

## 8 第1章 初识 Linux shell

---

将init进程加载到虚拟内存中。内核在启动任何其他进程时，都会在虚拟内存中给新进程分配一块专有区域来存储该进程用到的数据和代码。

一些Linux发行版使用一个表来管理在系统开机时要自动启动的进程。在Linux系统上，这个表通常位于专门文件/etc/inittab中。

另外一些系统（比如现在流行的Ubuntu Linux发行版）则采用/etc/init.d目录，将开机时启动或停止某个应用的脚本放在这个目录下。这些脚本通过/etc/rcX.d目录下的入口<sup>①</sup>启动，这里X代表运行级（run level）。

Linux操作系统的init系统采用了运行级。运行级决定了init进程运行/etc/inittab文件或/etc/rcX.d目录中定义好的某些特定类型的进程。Linux操作系统有5个启动运行级。

运行级为1时，只有基本的系统进程会启动，同时会启动唯一一个控制台终端进程。我们称之为单用户模式。单用户模式通常用来在系统有问题时进行紧急的文件系统维护。显然，在这种模式下仅有一人（通常是系统管理员）能登录到系统上操作数据。

标准的启动运行级是3。在这个运行级上，大多数应用软件，比如网络支持程序，都会启动。另一个Linux中常见的运行级是5。在这个运行级上系统会启动图形化的X Window系统，同时允许用户通过图形化桌面窗口登录系统。

Linux系统可以通过调整运行级来控制整个系统的功能。通过将运行级从3调整成5，系统就可以从基于控制台的系统变成更先进的图形化X Window系统。

---

<sup>①</sup> 这些入口实际上是到/etc/init.d目录中启动脚本的符号链接。——译者注

在第4章，你将会学习如何使用ps命令查看当前运行在Linux系统上的进程。下面示例演示了

1

使用ps命令时获得的输出：

```
$ ps ax
  PID TTY          STAT TIME  COMMAND
    1 ?            S     0:03   init
    2 ?            SW    0:00   [kflushd]
    3 ?            SW    0:00   [kupdate]
    4 ?            SW    0:00   [kpiod]
    5 ?            SW    0:00   [kswapd]
  243 ?            SW    0:00   [portmap]
  295 ?            S     0:00   syslogd
  305 ?            S     0:00   klogd
  320 ?            S     0:00   /usr/sbin/atd
  335 ?            S     0:00   crond
  350 ?            S     0:00   inetd
  365 ?            SW    0:00   [lpd]
  403 ttyS0        S     0:00   gpm -t ms
  418 ?            S     0:00   httpd
  423 ?            S     0:00   httpd
  424 ?            SW    0:00   [httpd]
  425 ?            SW    0:00   [httpd]
  426 ?            SW    0:00   [httpd]
  427 ?            SW    0:00   [httpd]
  428 ?            SW    0:00   [httpd]
  429 ?            SW    0:00   [httpd]
  430 ?            SW    0:00   [httpd]
  436 ?            SW    0:00   [httpd]
  437 ?            SW    0:00   [httpd]
  438 ?            SW    0:00   [httpd]
  470 ?            S     0:02   xfs -port -1
  485 ?            SW    0:00   [smbd]
  495 ?            S     0:00   nmbd -D
  533 ?            SW    0:00   [postmaster]
  538 tty1        SW    0:00   [mingetty]
  539 tty2        SW    0:00   [mingetty]
  540 tty3        SW    0:00   [mingetty]
  541 tty4        SW    0:00   [mingetty]
  542 tty5        SW    0:00   [mingetty]
  543 tty6        SW    0:00   [mingetty]
  544 ?            SW    0:00   [prefdm]
  549 ?            SW    0:00   [prefdm]
  559 ?            S     0:02   [kwm]
  585 ?            S     0:06   kikbd
  594 ?            S     0:00   kwmsound
  595 ?            S     0:03   kpanel
  596 ?            S     0:02   kfm
  597 ?            S     0:00   krootwm
```

## 10 第1章 初识 Linux shell

```
598 ?      S        0:01 kbgndwm
611 ?      S        0:00 kcmlaptop -daemon
666 ?      S        0:00 /usr/libexec/postfix/master
668 ?      S        0:00 qmgr -l -t fifo -u
787 ?      S        0:00 pickup -l -t fifo
790 ?      S        0:00 telnetd: 192.168.1.2 [vt100]
791 pts/0   S        0:00 login -- rich
792 pts/0   S        0:00 -bash
805 pts/0   R        0:00 ps ax
$
```

第一列输出显示了进程的进程号 ( Process ID , 即PID )。注意 , 第一个进程就是我们的好朋友init进程 , Linux系统分配给它的PID值是1。之后所有其他进程的PID都是按序分配的。没有两个进程拥有相同的PID , 虽然旧的PID数值在原进程结束后会被系统重新使用。

第三列显示了进程的当前状态 ( S代表在睡眠 , SW代表在睡眠和等待 , R代表在运行中 )。进程名字显示在最后一列。方括号中的进程是由于不活动而被从内存中换出到磁盘交换空间的进程。你能发现有些进程被换出了 , 但大部分运行中的进程未被换出。

### 3. 硬件设备管理

内核的另一职责是管理硬件设备。任何Linux系统需要与之通信的设备 , 都需要在内核代码中加入其驱动程序代码 ( driver code )。驱动程序代码相当于应用程序和硬件设备的中间人 , 允许内核同设备之间交换数据。在Linux内核中有两种方法用来插入设备驱动代码 :

编译进内核的设备驱动代码 ;

可插入内核的设备驱动模块。

以前 , 插入设备驱动代码的唯一途径是重新编译内核。每次给系统添加新设备时 , 你都要重新编译一遍内核代码。随着Linux内核支持越来越多的硬件设备 , 这个过程也变得越来越低效。

不过好在Linux开发人员设计出了一种更好的将驱动代码插入运行中的内核的方法。

开发人员提出了内核模块的概念。它允许将驱动代码插入到运行中的内核而无需重新编译内核。同时，当设备不再使用时也可将内核模块从内核中移走。这种方式极大地简化和推动了硬件设备在Linux上的使用。

Linux系统将硬件设备当成特殊的文件，称为设备文件。设备文件有3种不同的分类：

字符型设备文件；

块设备文件；

网络设备文件。

字符型设备文件是指处理数据时每次只能处理一个字符的设备。大多数类型的调制解调器和终端都是作为字符型设备文件创建的。块设备文件是指处理数据时每次能处理大块数据的设备，比如硬盘。

网络设备文件是指采用数据包发送和接收数据的设备，包括各种网卡和一个特殊的回环设备。这个回环设备允许Linux系统使用通用的网络编程协议同自己通信。

Linux为系统上的每个设备都创建一种特殊的文件，称为“节点”。与设备的所有通信都是通过设备节点完成的。每个节点都有一个唯一的数值对，供Linux内核标识它。数值对包括一个主设备号和一个次设备号。类似的设备被划分到同样的主设备号下。次设备号用于标识同一主设备号下的某个特殊设备。以下是在Linux服务器上的一些设备文件的例子：

```
rich@rich-desktop: ~$ cd /dev
rich@rich-desktop:/dev$ ls -al sda* ttyS*
brw-rw---- 1 root disk    8,  0 2010-09-18 17:25 sda
brw-rw---- 1 root disk    8,  1 2010-09-18 17:25 sda1
brw-rw---- 1 root disk    8,  2 2010-09-18 17:25 sda2
```

---

## 12 第1章 初识 Linux shell

---

```
brw-rw---- 1 root disk      8,  5 2010-09-18 17:25 sda5
crw-rw---- 1 root dialout  4, 64 2010-09-18 17:25 ttyS0
crw-rw---- 1 root dialout  4, 65 2010-09-18 17:25 ttyS1
crw-rw---- 1 root dialout  4, 66 2010-09-18 17:25 ttyS2
crw-rw---- 1 root dialout  4, 67 2010-09-18 17:25 ttyS3
rich@rich-desktop:/dev$
```

不同的Linux发行版在处理设备时采用不同的设备名。在这个发行版上，sda设备是第一个ATA硬盘，ttyS设备是标准的IBM PC COM端口。这个清单显示了示例Linux系统上创建的所有sda设备。虽然并不都在实际中使用，但它们都被创建出来了以备系统管理员不时之需。类似地，这个列表显示了已创建的所有ttyS设备。

第5列是主设备节点号。注意，所有sda设备都拥有同一主设备号8；而所有的ttyS设备都使用4。第6列是次设备节点号。同一个主设备节点号下的每个设备都拥有自己唯一的次设备节点号。

第1列显示了该设备文件的权限。权限的第一个字符表示的是设备文件的类型。注意，ATA硬盘文件都被标记为b（block device，块设备），而COM端口设备文件则被标记为c（character device，字符型设备）。

### 4. 文件系统管理

不同于其他一些操作系统，Linux内核支持多种不同类型的文件系统来从硬盘中读取或写入数据。除了自有的诸多文件系统外，Linux还支持从其他操作系统（比如Microsoft Windows）所采用的文件系统中读取或写入数据。内核必须在编译时就加入对所有可能用到的文件系统的支持。表1-1列出了Linux系统用来读写数据的标准文件系统。

表1-1 Linux文件系统

文件系统	描述
------	----

ext	Linux扩展文件系统，最早的Linux文件系统
ext2	第二扩展文件系统，在ext的基础上提供了更多的功能
ext3	第三扩展文件系统，支持日志功能
ext4	第四扩展文件系统，支持高级日志功能
hpfs	OS/2高性能文件系统
jfs	IBM日志文件系统
iso9660	ISO 9660文件系统 ( CD-ROM )
minix	MINIX文件系统
msdos	微软的FAT16
ncp	Netware文件系统
nfs	网络文件系统
ntfs	支持Microsoft NT文件系统
proc	访问系统信息
ReiserFS	高级Linux文件系统，能提供更好的性能和硬盘恢复功能
smb	支持网络访问的Samba SMB文件系统
sysv	较早期的Unix文件系统
ufs	BSD文件系统
umsdos	贮存在msdos上的类Unix文件系统
vfat	Windows 95文件系统 ( FAT32 )
XFS	高性能64位日志文件系统

任何供Linux服务器访问的硬盘都必须格式化成本表1-1所列文件系统类型中的一种。

Linux内核采用虚拟文件系统 ( Virtual File System , VFS ) 作为和每个文件系统交互的接口。这为Linux内核同任何类型文件系统通信提供了一个标准接口。当每个文件系统被挂载和使用时，VFS将信息都缓存在内存中。

### 1.1.2 GNU 工具链

除了有内核来控制硬件设备外，操作系统还需要工具链来执行一些标准功能，比如控制文件

和程序。当Linus创建Linux系统内核时，是没有系统工具链运行其上的。然而他很幸运，就在他开发Linux内核的同时，有一组人正在互联网上共同努力，模仿Unix操作系统开发一系列标准的计算机系统工具。

GNU组织（GNU代表GNU's Not Unix）开发了一套完整的Unix工具链，但没有可以运行它们的内核系统。这些工具链是在开源软件（Open Source Software，OSS）的软件开发理念下开发的。

开源软件理念允许程序员开发软件并将其免费发布。任何人都可以使用、修改该软件，或将该软件集成进自己的系统，而无需支付任何授权费用。将Linus的Linux内核和GNU操作系统工具整合起来，就可以创建一个完整的、功能丰富的免费操作系统。

尽管通常我们将Linux内核和GNU工具链的结合体称为Linux，你也会在互联网上看到一些Linux纯粹主义者将其称为GNU/Linux系统来表彰GNU组织为此所作的贡献。

### 1. 核心GNU工具链

GNU项目一开始主要是为Unix系统管理员设计的，用以提供一个类Unix环境。这个目标导致这个项目移植了很多Unix系统通用的命令行工具。为Linux系统提供的一组核心工具被称为coreutils（core utilities）软件包。

GNU coreutils软件包由3部分构成：

用以处理文件的工具；

用以操作文本的工具；

用以管理进程的工具。

这三组主要工具中的每一组都包含一些对Linux系统管理员和程序员至关重要的工具。本书将详细介绍GNU coreutils软件包中包含的所有工具。

## 2. shell

GNU/Linux shell是个交互式工具。它为用户提供了启动程序、管理文件系统上的文件以及管理运行在Linux系统上的进程的途径。shell的核心是命令行提示符。命令行提示符是shell的交互部分。它允许你输入文本命令，之后将解释命令并在内核中执行。

shell包含了一组内置命令，你可以用这些命令来完成一些操作，例如复制文件、移动文件、重命名文件以及显示和终止系统上正运行的程序。shell也允许你在命令行提示符中输入程序的名称，它会将程序的名称传递给内核以启动它。

你也可以将shell命令放入文件中作为程序执行。这些文件被称作shell脚本。你在命令行上执行的任何命令都可放进一个shell脚本中作为一组命令执行。这为创建那种需要把几个命令放在一起工作的工具提供了便利。

在Linux系统上，通常有好几种Linux shell可用。不同的shell有不同的特性，有些更利于创建脚本，有些更利于管理进程。所有Linux发行版默认的shell基本上都是bash shell。bash shell作为标准Unix shell——Bourne shell（沿用创建者的名字）的替代，由GNU项目开发的。bash shell的名称就是针对这个Bourne shell的文字游戏，称为Bourne again shell。

除了bash shell外，我们在本书中还将介绍其他几种常见的shell。表1-2列出了Linux中常见的

## 16 第1章 初识 Linux shell

几种不同shell。

表1-2 Linux shell

shell	描 述
ash	运行在内存受限环境中简单的、轻量级 shell，但与bash shell完全兼容
korn	与Bourne shell兼容的编程shell，但支持一些高级的编程特性，比如关联数组和浮点运算
tcsh	将C语言中的一些元素引入到shell脚本中的shell
zsh	将bash、tcsh和korn的特性引入，同时提供高级编程特性、共享历史文件和主题化提示符的高级shell

大多数Linux发行版包含多个shell，虽然它们通常会采用其中一个作为默认shell。如果你的Linux发行版包含多个shell，尽情尝试不同的shell，看看哪个能满足你的需要。

### 1.1.3 Linux 桌面环境

在Linux的早期（20世纪90年代早期），系统上可用的仅是一个简单的与Linux操作系统交互的文本界面。这个文本界面允许系统管理员运行程序、控制程序的执行以及在系统中移动文件。

随着Microsoft Windows的普及，电脑用户期望的就不仅仅是对着老式的文本界面工作了。这点推动了OSS社区的更多开发活动，Linux图形化桌面环境出现了。

Linux一直都以可用多种方式来完成工作而声名在外。在图形化桌面上更是如此。Linux有各种图形化桌面可供选择。后面几节中将会介绍其中比较流行的桌面。

#### 1. X Window系统

有两项基本组件能决定你的视频环境：显卡和显示器。要在电脑上显示绚丽的画面，Linux软件就得知道如何来连接它们。X Window软件是图形显示的核心元素。

X Window软件是直接和PC上的显卡以及显示器一起工作的底层软件。它控制着Linux程序如

何在电脑上显示出绚丽的窗口和画面。

Linux并非唯一使用X Window的操作系统，X Window有针对多种不同操作系统的版本。在Linux世界里，仅有两个软件包能实现X Window。

XFree86软件包是二者中较早的那个，并且在很长一段时间里都是仅有的为Linux开发的X Window包。顾名思义，它是X Window的免费开源版本。

二者中较新的X.org在Linux世界里占据优势，现在是二者中较为普遍的。它同样提供了X Window系统的开源实现，但支持的更多是现今在用的较新显卡。

两个软件包都以同样的方式工作，控制着Linux如何使用显卡来在显示器上显示内容。为此，你需要针对你的系统专门配置它们。一般地，这个配置过程会在安装Linux时自动完成。

在首次安装Linux发行版时，它会检测显卡和显示器，然后创建一个含有必要信息的X Window配置文件。在安装过程中，你可能会注意到安装程序会检测一次显示器，以此来确定它所支持的视频模式。有时显示器会黑屏几秒。由于现在有多种不同类型的显卡和显示器，这个过程可能会需要一段时间来完成。

核心的X Window软件可以产生图形化显示环境，但仅此而已。虽然对于运行独立应用这已经足够，但在日常PC使用中却并不是那么有用。它没有桌面环境供用户操作文件或是开启程序。为此，你需要一个在X Window系统软件之上的桌面环境。

## 2. KDE桌面

KDE ( K Desktop Environment , K桌面环境 ) 最初于1996年作为开源项目发布。它会生成一

## 18 第1章 初识 Linux shell

个类似于Microsoft Windows的图形化桌面环境。如果你是Windows用户，KDE集成了所有你熟悉的。图1-3显示了运行在openSuSE Linux发行版上的KDE 4桌面。



图1-3 openSuSE Linux系统上的KDE 4桌面

KDE桌面允许你把应用程序图标和文件图标放置在桌面的特定位置上。单击应用程序图标，Linux系统就会运行该应用程序。单击文件图标，KDE桌面就会确定使用哪种应用程序来处理该文件。

桌面底部的横条称为面板，它由以下四部分构成。

**KDE菜单：**和Windows的开始菜单非常类似，KDE菜单包含了启动已安装程序的链接。

**程序快捷方式：**在面板上有直接从面板启动程序的快速链接。

任务栏：任务栏显示着当前桌面正运行的程序的图标。

小应用程序：面板上还有一些特殊小应用程序的图标，这些图标常常会根据小应用程序的状态发生变化。

所有的面板功能都和你在Windows上看到的类似。除了桌面功能，KDE项目还开发了大量的可运行在KDE环境中的应用程序。表1-3列出了这些程序。（注意，作为惯例，KDE应用命名时经常有个大写的K。）

表1-3 KDE应用程序

应用程序	描述
amaroK	音频播放器
digiKam	数码相机软件
dolphin	文件管理器
K3b	CD烧录软件
Kaffeine	视频播放器
Kmail	E-mail客户端
Koffice	Office应用套件
Konqueror	文件和Web浏览器
Kontact	个人信息管理器
Kopete	即时消息客户端

这里仅列出了KDE项目开发的部分应用。在KDE桌面中还包含着更多的应用。

### 3. GNOME桌面

GNOME ( The GNU Network Object Model Environment , GNU网络对象模型环境 ) 是另一个流行的Linux桌面环境。GNOME于1999年首次发布，现已成为许多Linux发行版 ( 最普遍的是Red

## 20 第1章 初识 Linux shell

Hat Linux ) 默认的桌面环境。

尽管GNOME决定不再沿用Microsoft Windows的标准外观，但它还是集成了许多Windows用户习惯的功能：

一块放置图标的桌面区域；

两个面板区域；

拖放功能。

图1-4显示了Ubuntu Linux发行版采用的标准GNOME桌面。

不甘示弱于KDE，GNOME开发人员也开发了一组集成进GNOME桌面的图形化程序，如表1-4所示。

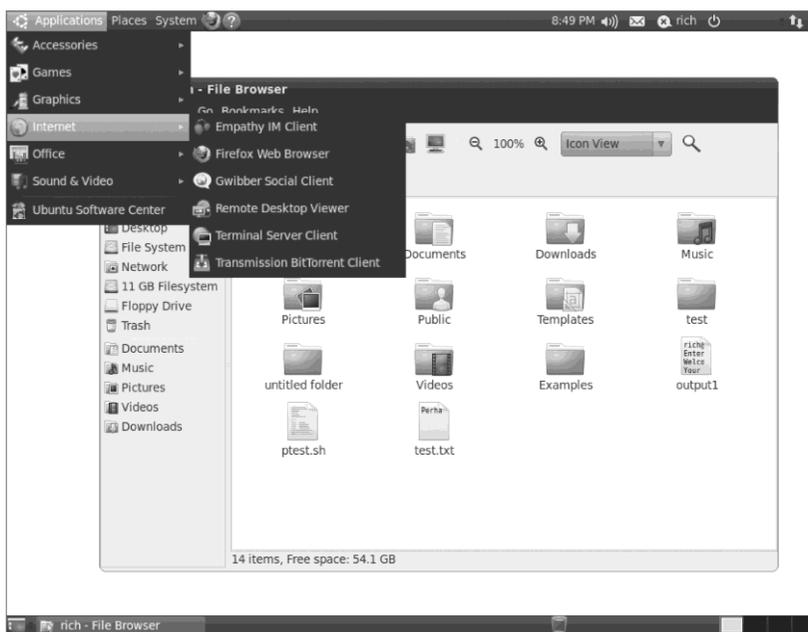


图1-4 Ubuntu Linux系统上的GNOME桌面

表1-4 GNOME应用程序

应用程序	描 述
epiphany	Web浏览器
evince	文档查看器
gcalc-tool	计算器
gedit	GNOME文本编辑器
gnome-panel	桌面面板，用以启动程序
gnome-nettool	网络诊断工具
gnome-terminal	终端模拟器
nautilus	图形化文件管理器
nautilus-cd-burner	CD烧录工具
sound juicer	音频CD抓轨工具
tomboy	笔记程序
totem	多媒体播放器

如你所见，对于GNOME桌面，同样有不少可用的应用程序。除了这些程序，大多数采用GNOME桌面的Linux发行版还集成了KDE库，允许用户在GNOME桌面上运行KDE应用程序。

#### 4. 其他桌面

图形化桌面环境的弊端在于它们要占用相当一部分系统资源。在Linux发展之初，Linux的特征和卖点之一就是，它可以运行在功能较弱的早期PC上运行，而这些PC无法运行较新的微软桌面。然而随着KDE和GNOME桌面环境的普及，情况发生了变化。运行KDE或GNOME桌面要占用和微软的最新桌面环境一样多的内存资源。

如果你的PC比较老旧，也不要泄气。Linux开发人员已经联手让Linux返璞归真。他们开发了一些低内存开销的图形化桌面环境，仅提供能在早期PC上完美运行的基本功能。

尽管这些图形化桌面环境并没有专为其设计的大量应用，但它们仍然能运行许多基本的图形

## 22 第1章 初识 Linux shell

化程序，支持如文字处理、电子表格、数据库、绘图以及多媒体等功能。

表1-5列出了一些可在配置较低的PC和笔记本电脑上运行的轻量级Linux图形化桌面环境。

表1-5 其他Linux图形化桌面

桌 面	描 述
fluxbox	一个没有面板的轻型桌面，仅有一个弹出式菜单来启动程序
xfce	和KDE很像的一个桌面，但少了很多图形以适应低内存环境
JWM	Joe的窗口管理器（Joe's Window Manager），适用于低内存低硬盘空间环境的理想超微型桌面
fvwm	支持诸如虚拟桌面和面板等高级桌面功能，但在低内存环境中运行
fvwm95	从fvwm衍生而来，但看起来更像是Windows 95桌面

这些图形化桌面环境并不如KDE或GNOME桌面一样绚丽，但它们提供了基本的图形化功能。图1-5显示了Puppy Linux antiX发行版所采用的JWM桌面的外观。



图1-5 Puppy Linux发行版所采用的JWM桌面

如果你用的是早期PC，尝试一下基于上述某个桌面环境的Linux发行版，看看会怎么样。你可能会大吃一惊。

## 1.2 Linux 发行版

到此为止，你已经了解了构成完整Linux系统所需要的4个关键部件，你可能在考虑要怎样才能把它们放在一起构建一个Linux系统。幸运的是，已经有人为你做好这些了。

我们将完整的Linux系统包称为发行版。有各种不同的Linux发行版来满足可能存在的各种运算需求。大多数发行版是为某个特定用户群定制的，比如商业用户、多媒体爱好者、软件开发人员或者普通家庭用户。每个定制的发行版都支持特定功能所需的各种软件包，比如为多媒体爱好者准备的音频和视频编辑软件，为软件开发人员准备的编译器和集成开发环境。

不同的Linux发行版通常归类为三种：

完整的核心Linux发行版；

专业发行版；

LiveCD测试发行版。

后面几节将会探讨这些不同类型的Linux发行版，然后会展示每种类型中的一些Linux发行版示例。

### 1.2.1 核心 Linux 发行版

核心Linux发行版含有内核、一个或多个图形化桌面环境以及预编译好的几乎所有能见到的

Linux应用。它提供了一站式的完整Linux安装。表1-6列出了较流行的核心Linux发行版。

表1-6 核心Linux发行版

发 行 版	描 述
Slackware	最早的Linux发行版中的一员，在Linux极客中比较流行
Red Hat	一个主要用于Internet服务器的商业发行版
Fedora	从Red Hat分离出的家用发行版
Gentoo	为高级Linux用户设计的发行版，仅包含Linux源代码
Mandriva	主要是家用（之前叫Mandrake）
openSuSE	用于商用和家用的发行版
Debian	在Linux专家和商用Linux产品中流行的发行版

在Linux的早期，发行版是作为一叠软盘发布的。你必须下载多组文件然后将其复制到软盘上。通常要用20张或更多的软盘来创建一个完整的发行版。毋庸多言，这是个痛苦的过程。

现今，家用电脑基本都有内置的CD和DVD光驱，Linux发行版也就用几张CD光盘或单张DVD光盘来发布。这大大简化了Linux的安装过程。

然而当新手在安装核心Linux发行版时，他们仍经常遇到各种各样的问题。为了照顾到想用Linux的人可能碰到的各种情形，单个发行版也必须包含很多应用软件。它们包含了从高端Internet数据库服务器到通用游戏的所有软件。鉴于Linux上可用的大量应用程序，一个完整的发行版通常至少要4张CD。

发行版中的大量可选配置对于Linux极客来说是好事，但对于新手来说就是一场噩梦。多数发行版会在安装过程中问一系列问题以决定哪些应用要默认加载、哪些硬件是连接到PC的以及怎样配置硬件设备。新手经常会被这些问题困扰，因此他们经常在机器上加载了过多的程序或没

有加载足够的程序以至于最后发现计算机没有如他们期望那样的工作。

对新手来说幸运的是，还有更简便的安装Linux的方法。

### 1.2.2 专业 Linux 发行版

Linux发行版的一个子群开始出现了。它们通常基于某个主流发行版，但仅包含主流发行版中一小部分用于某种特定用途的程序。

除了提供专业软件外（比如仅为商业用户提供的办公应用），专业发行版还尝试通过自动检测和自动配置来帮助新手安装Linux。这让安装Linux变得更容易。

表1-7列出了一些专业Linux发行版以及它们的专长。

表1-7 专业Linux发行版

发 行 版	描 述
Xandros	一个为新手配置的商业Linux发行版
SimplyMEPIS	一个免费的家用Linux发行版
Ubuntu	一个免费的学校和家庭用的Linux发行版
PCLinuxOS	一个免费的家庭和办公用的Linux发行版
Mint	一个免费的家庭娱乐用的Linux发行版
dyne:bolic	一个免费的包含音频和MIDI应用程序的Linux发行版
Puppy Linux	一个免费的适用于早期PC的小型Linux发行版

这只是专业Linux发行版中的一小部分而已。大约有数百个专业Linux发行版，而在互联网上还不断有新的出现。不管你的专长是什么，你都能找到一个为你量身定做的Linux发行版。

许多专业Linux发行版都是基于Debian Linux的。它们使用和Debian一样的安装文件，但仅打包了完整Debian系统中的一部分软件。

### 1.2.3 Linux LiveCD

Linux世界中一个相对比较新的现象是可引导启动的Linux CD发行版的出现。它允许不安装Linux就可以看看Linux系统是什么样的。多数现代PC都能从CD启动，而不是必须从标准硬盘启动。基于这点，一些Linux发行版创建了包含有示例Linux系统（称为Linux LiveCD）的可引导启动CD。由于单张CD容量的限制，示例系统无法包含一个完整的Linux系统，但你会为它们能加入的各种程序感到欣喜。最终你可以通过CD来启动PC并且无需在硬盘安装任何东西就能运行Linux发行版了。

这是一个不弄乱PC而体验各种Linux发行版的绝妙方法。只需插入CD就能引导了！所有的Linux软件都将直接从CD上运行。有很多Linux LiveCD可供你从互联网上下载、刻录到CD上体验。

表1-8列出了可用的一些流行Linux LiveCD。

表1-8 Linux LiveCD发行版

发 行 版	描 述
Knoppix	一个德语Linux发行版，最早的LiveCD Linux
SimplyMEPIS	为家庭用户设计的Linux发行版
PCLinuxOS	成熟的LiveCD上的Linux发行版
Ubuntu	为多种语言设计的世界级项目
Slax	基于Slackware Linux的LiveCD Linux
Puppy Linux	为早期PC设计的全功能Linux

你能在这张表中看到熟悉的面孔。许多专业Linux发行版都有Linux LiveCD版本。一些Linux LiveCD发行版，比如Ubuntu，允许直接从LiveCD安装整个发行版。这使你能从CD引导启动Linux体验此Linux发行版；如果喜欢，就安装到硬盘上。这个功能极其方便易用。

就像所有美好的事物一样，Linux LiveCD也有一些美中不足的地方。由于要从CD上访问所有东西，应用程序会运行得更慢，尤其当你把缓慢的早期机器和CD光驱放在一起用时。还有，由于无法向CD写入数据，对Linux系统作的任何修改在重启后都会失效。

不过，有一些Linux LiveCD的改进帮助解决了上述一些问题。这些改进包括：

能将CD上的Linux系统文件复制到内存中；

能将系统文件复制到硬盘上；

能在U盘上存储系统设置；

能在U盘上存储用户设置。

一些Linux LiveCD，如Puppy Linux，被设计成有极少量的Linux系统文件。当CD引导启动时，LiveCD的启动脚本直接把它们复制到内存中。这允许在Linux启动后立即把CD从光驱中取走。这不仅使得程序运行得更快（因为程序从内存中运行时更快），而且空出CD光驱供你用Puppy Linux自带的软件转录音频CD或播放视频DVD。

其他Linux LiveCD用另外的方法，同样允许你在启动后将CD从光驱中拿走。其中一种就是把核心Linux文件作为一个文件复制到Windows硬盘上。待CD启动后，系统会寻找那个文件，并从那个文件中读取系统文件。dyne:bolic Linux LiveCD采用的就是这种技术，我们称之为对接（docking）。当然，你必须在从CD引导启动之前把系统文件复制到硬盘里。

使用通用U盘（也称为闪存或闪盘）是一项非常流行的存储Linux LiveCD会话数据的技术。几乎每个Linux LiveCD都能识别插入的U盘（即使是在Windows下格式化的）并从U盘上读取或写

入文件。这允许你启动Linux LiveCD，使用Linux应用来创建文件，将这些文件存储在U盘上，然后用Windows应用（或者在另外一台电脑上）访问这些文件。这该有多酷！

### 1.3 小结

本章探讨了Linux系统以及它是如何工作的这个基本概念。Linux内核是系统的核心，控制着内存、程序和硬件是如何与对方交互的。GNU工具链也是Linux系统中的一个重要部分。本书关注的焦点Linux shell是GNU核心工具集中的一部分。本章还讨论了Linux系统中的最后一个组件Linux桌面环境。随着时间推移，一切都发生了改变。现今Linux支持几个图形化桌面环境。

本章还探讨了各种Linux发行版。Linux发行版就是把Linux系统的各个不同部分汇集起来组成的单个易于安装的包。Linux发行版有囊括各种软件的成熟的核心Linux发行版，也有只包含针对某种特定功能软件包的专业发行版。Linux LiveCD则是一种无需将Linux安装到硬盘就能体验Linux的发行版。

下一章你将开始了解开启命令行和shell脚本编程体验所需的基本知识。你将了解如何从绚丽的图形化桌面环境获得Linux shell工具。那绝非易事。