工业和信息化普通高等教育“十二五”规划教材立项项目
21世纪高等教育计算机规划教材

COMPUTER

GNU/Linux 编程

GNU/Linux Programming

郑谦益 编著

- 注重基础，面向应用，知识点较多
- 剖析实现技术，展现软件体系架构
- 通过实例演示应用接口的使用方法

人民邮电出版社
POSTS & TELECOM PRESS

息化普通高等教育“十二五”规划教材立项项目
高等教育计算机规划教材

COMPUTER

GNU/Linux 编程

GNU/Linux Programming

郑谦益 编著

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

GNU/Linux编程 / 郑谦益编著. -- 北京 : 人民邮电出版社, 2012.8
21世纪高等教育计算机规划教材
ISBN 978-7-115-28260-6

Ⅰ. ①G… Ⅱ. ①郑… Ⅲ. ①Linux操作系统－程序设计－高等学校－教材 Ⅳ. ①TP316.89

中国版本图书馆CIP数据核字(2012)第123473号

内 容 提 要

Linux 作为一种自由和源码开放的类 UNIX 操作系统，虽然诞生至今只有 20 年的时间，但已经在各个领域中得到了广泛的应用，对软件行业产生了深远的影响。本书通过大量实例讲述 Linux 环境下进行软件开发所必须掌握的基础知识。全书内容由四个部分组成，第一部分介绍 Linux 相关背景知识和 GNU 命令工具的使用方法；第二部分为 Shell 程序设计，讲述 Shell 脚本语言的语法结构；第三部分介绍 Linux 开发环境，讲述基于 C 语言开发的 GNU 工具的使用方法；第四部分为 Linux 环境下的 C 语言编程，系统讲述与 Linux 内核有关的应用编程接口函数的使用方法。

本书可以作为高校计算机相关专业的高年级学生、研究生学习 Linux 编程的教材或教学参考书。

21 世纪高等教育计算机规划教材

GNU/Linux 编程

◆ 编　　著　郑谦益
责任编辑　董　楠
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress com.cn
网址　http://www.ptpress.com cn
北京艺辉印刷有限公司印刷
◆ 开本：787×1092　1/16
印张：15.5　　2012 年 8 月第 1 版
字数：407 千字　　2012 年 8 月北京第 1 次印刷

ISBN 978-7-115-28260-6

定价：32.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前 言

Linux 作为开源操作系统，有众多的发行版本，例如 Red Hat、Debian、Ubuntu 和 Mandrake 等。GNU 工具和 Linux 内核构成了 Linux 系统的基础，GNU 项目开发了大量的自由软件，其中包括 vi 编辑器、bash 命令解释器、函数库 glic、编译器 gcc 和其他各种命令工具等，这些工具也为 Linux 内核的产生与发展提供了良好的环境支持。Linux 内核是操作系统的核心，负责管理系统的软硬件资源，对处理器、内存、部设备、文件和网络等进行统一管理。Linux 内核采用了基于模块可配置的大内核结构，可移植到多种处理器，提供了大量的设备驱动程序，实现了对多种文件系统和网络协议的支持。经函数库 glibc 的封装，Linux 系统还为上层应用开发提供标准的应用编程接口服务。

目前，Linux 系统已在各领域得到了广泛应用，从高端服务器，到各种嵌入式系统。Linux 已成功安装到多种计算机硬件设备，从 PC、手机、路由器到大型计算机系统。因此，学习和掌握 Linux 操作系统有着十分重要的现实意义。

本书通过大量实例演示 Linux 系统的命令、语句和函数的使用方法，对较难理解的部分，借助软件模型从软件体系结构和行为模式的角度，对实现技术进行了描述，便于读者将理论和技术实现有机结合起来，使读者从系统框架的角度，深入理解 Linux 编程。为了适应不同层次读者的需要，本书由四部分内容构成。

第一部分，Linux 基础，包括第 1～3 章，介绍 Linux 相关概念，通过实例讲述常用命令的使用方法。

第 1 章介绍与 Linux 相关的背景知识，内容包括 UNIX 的起源和衍生版本、GNU 项目、Linux 内核的产生与发展和 Linux 商业运营模式等。

第 2 章对 Linux 的基本命令进行了分类，以实例的方式讲述常用命令的使用方法。其中对文件权限管理进行了较为详细的介绍，对进程、进程组、会话和控制终端的概念及相互关系进行了阐述。

第 3 章通过实例讲述磁盘的结构、分区以及文件系统的构建和使用方法，对 Linux 环境下的引导加载程序 grub 的概念、原理和使用方法作了介绍，对 Linux 系统初始化过程进行了较为详细的阐述。

第二部分，Shell 程序设计，包括第 4 章，讲述 Shell 脚本的语法结构，通过实例介绍如何使用 Shell 命令进行流程控制。

第 4 章通过实例讲述 Shell 程序设计，内容包括变量的分类与使用、条件表达式、控制语句和函数。

第三部分，GNU C 语言开发环境，包括第 5 章，讲述 Linux 环境下与 C 语言开发相关的 GNU 工具的概念和使用方法。

第 5 章通过实例介绍 Linux 环境下的 C 语言开发工具，内容包括编译器 gcc、链接器 ld、项目管理工具 make，以及函数库的概念和归档工具 ar 的使用方法。

第四部分，Linux 环境下的 C 语言编程，包括第 6～10 章，通过实例讲述与 Linux 内核相关的应用编程接口函数的使用方法。

第 6 章介绍了 Linux 文件系统的应用编程接口，内容包括文件与目录的操作、标准 I/O 输入与输出和 I/O 重定向。

第 7 章介绍信号的概念，对 Linux 系统中信号的定义和相关的应用编程接口函数的使用方法进行了阐述。

第 8 章讲述 Linux 进程的相关概念，通过实例对进程的用户虚拟地址空间的结构、进程的创建与终止、可执行映像文件的加载、进程的同步控制和进程的运行环境进行了阐述。

第 9 章通过实例介绍 Linux 进程间通信的概念和方法，内容包括管道、信号量、消息队列和共享内存。

第 10 章通过实例介绍五种文件 I/O 操作模式的编程方法，对终端 I/O 行为方式的概念和编程方法进行了阐述。

本书适用对象为具有初步 C 语言程序设计经验和一定操作系统基础知识的广大读者。本书可作为高校计算机相关专业本科生、研究生学习 Linux 编程的教材或教学参考书，也可作为广大 Linux 爱好者的参考资料和培训教材。读者可根据实际情况，选读其中部分章节。

在本书编写过程中，李养群、鲁蔚锋和肖学中老师对书稿做了大量校对工作并提出了许多宝贵意见。在此，对他们表示衷心的感谢。

由于时间和水平有限，书中一定存在许多不足之处，希望广大读者不吝赐教。

郑谦益
zhengqy@njupt.edu.cn
2012 年 3 月

目 录

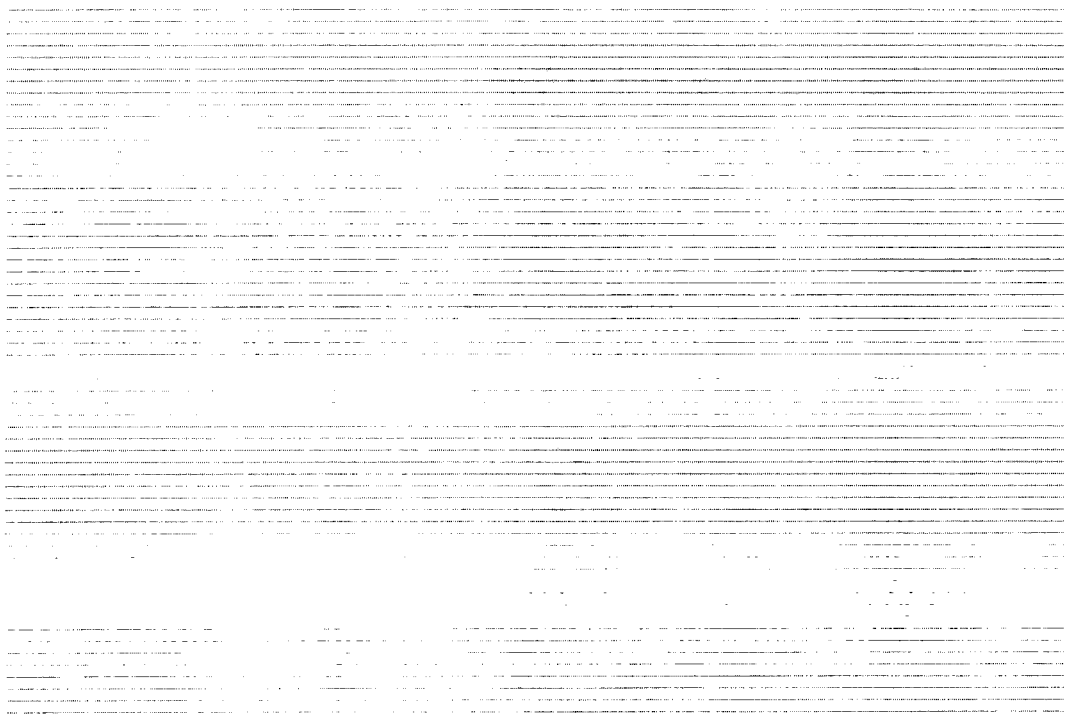
第一部分 Linux 基础

第二部分 Shell 程序设计

第三部分 CNU C 语言开发环境

第四部分 Linux 环境下的 C 语言编程

第一部分
Linux 基础



第1章 UNIX系统概述

1.1 UNIX的发展历史

1.1.1 UNIX的产生与发展

1968年，由通用电器公司、贝尔实验室和美国麻省理工学院的研究人员共同开发了一个名为Multics的操作系统，该操作系统使用户可以通过电话线接入远程终端，实现多用户访问大容量文件系统资源。Multics引入了许多现代操作系统的概念雏形，对随后的操作系统，特别是UNIX的成功有着巨大的影响。

1969～1970年，AT&T公司的贝尔实验室研究人员Ken Tompson和Dennis Ritchie在Multics操作系统的基础上用C语言开发出UNIX系统。当时，AT&T公司以低廉甚至免费的价格将UNIX源代码授权给学术机构用于研究和教学。1979年，从UNIX的V7版本开始，AT&T公司意识到UNIX的商业价值，不再将UNIX源码授权给学术机构，并对之前的UNIX及其变种声明了版权。贝尔实验室1983年发行了第一个商业版本，名为System Ⅲ，后来被拥有良好商用软件支持的System V所替代。UNIX的这一分支不断发展，直到System V第4版开始分裂，形成了System V系列。另一方面，1978年美国伯克利大学在UNIX第六版本的基础上进行了修改，增加了新的功能，发布了BSD，开创了UNIX的另一个BSD系列分支。1980年微软公司开发了名为XENIX的UNIX PC版本。各种UNIX版本的一系列变化与发展的脉络如图1-1所示，UNIX的发展主要演化为两大分支，它们分别是System V系列和BSD系列。

1. System V系列

有很多大公司在取得了UNIX的授权之后，在System V的基础上，开发出自己的UNIX产品，如表1-1所示.。

表1-1　　System V系列的衍生版本

名　称	厂　家	基于的版本
AIX	International Bussiness Machines	AT&T System V
Irix	Silicon Graphics	AT&T System V
Solaris	Sun Microsystems	AT&T System V
Unicos	Cray	AT&T System V
UNIXWare	Novell	AT&T System V
XENIX	Microsoft	AT&T System III

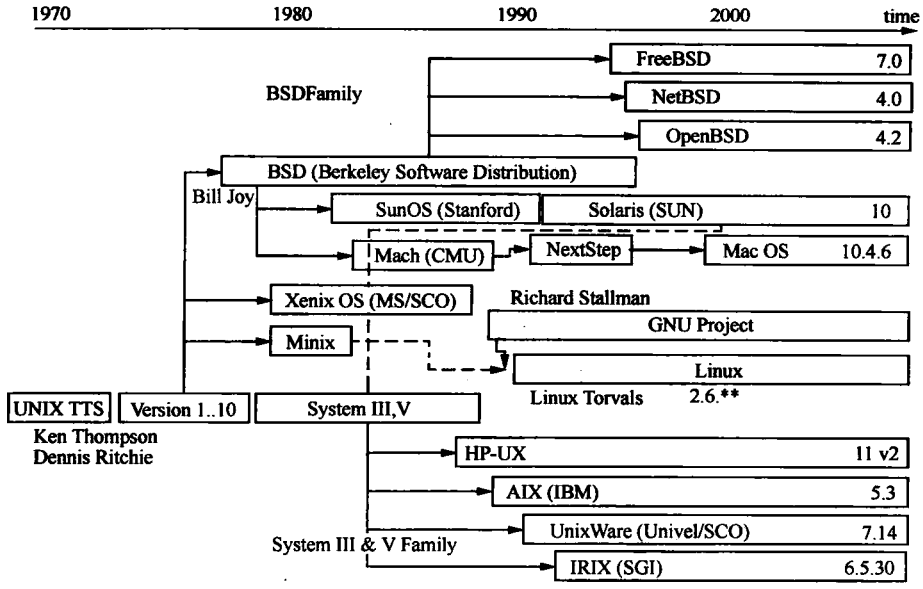


图 1-1　Unix 的发展历史

2. BSD 系列

为了不和 AT&T 公司的版权冲突，BSD 版本在版本 3 之后将代码进行了重写，BSD 版本此后不再包括有版权的 UNIX 代码。UNIX 的 BSD 版本不断发展，在 20 世纪 70 年代末期，BSD UNIX 成为美国国防部高科技研究机构科研项目的基础。为此，美国伯克利大学发行了名为 BSD Release 4.2 的有效版本。该版本结合 SVR3、BSD4.3、SunOS 和 Microsoft XENIX 的特点，包括了高级的文件管理，实现了对 TCP/IP 网络协议的支持。目前，TCP/IP 被 Internet 所使用，BSD Release 4.2 被许多厂商所采用，例如 SUN Microsystem 等。BSD 在发展过程中，衍生出不同的分支，表 1-2 列出了主要的 BSD 分支。

表 1-2　　BSD 的衍生版本

名　称	提　供　者
Dynix	Sequent
SunOS	Sun Microsystems
Ultrix	Digital Equipment Corporation
FreeBSD	支持 Intel、AMD 和 Sun UltraSPARC，基于 4.4BSD-Lite 架构
NetBSD	支持 Alpha，DraCo 和 Shark 等多种硬件，基于 4.3BSD Lite 架构
OpenBSD	衍生自 NetBSD，支持 DEC Alpha 和 Intel 等多种硬件

1.1.2　UNIX 的相关标准

UNIX 在发展过程中形成了多个分支和版本，为了使开发出来的应用程序在不加修改的情况下，能从一个系统移植到另一个系统，实现不同 UNIX 版本的相互兼容，必须制订必要的协议标准。下面给出与 UNIX 相关的几个标准。

1. ANSI C/ISO C

ANSI C 是美国国家标准局于 1989 年制定的 C 语言标准。后来被国际标准化组织接受为标准，

因此也称为 ISO C。ANSI C 的目标是为各种操作系统上的 C 程序提供可移植性保证，他不仅限于 UNIX。该标准不仅定义了 C 编程语言的语法和语义，而且还定义了一个标准库。这个库的头文件包括字符类型 <ctype.h>、错误码 <errno.h>和标准 I/O <stdio.h>等。

2. POSIX

POSIX（Portable Operating System Interface for Computing Systems）是由 IEEE 和 ISO/IEC 开发的一簇标准。该标准基于现有 UNIX 的实践经验，规范了操作系统的应用编程接口，目的是使应用程序源代码能够不加修改地移植到多种 UNIX 操作系统。1990 年，POSIX.1 与已通过的 C 语言标准联合，正式被批准为 IEEE 1003.1-1990 和 ISO/IEC 9945-1:1990 标准。POSIX.1 仅规定了系统服务应用编程接口。此后，又有多个标准相继发布。例如，命令与工具标准 POSIX.2、测试方法标准 POSIX.3 和实时编程接口 POSIX.4 等，如表 1-3 所示。

表 1-3　　POSIX 系列标准

POSIX 版本	目　　标
1003.1	库函数和系统调用标准
1003.2	命令工具标准
1003.3	测试方法标准
1003.4	实时标准
1003.5	Ada 语言相关标准
1003.6	安全标准

3. SVID

System V 接口描述（SVID）是描述 AT&T 公司 UNIX System V 操作系统的文档，是 POSIX 标准的扩展超集。

4. XPG/X/Open

X/Open 可移植性指南(由 X/Open Company,Ltd.出版),是比 POSIX 更为一般的标准。X/Open 拥有 UNIX 的版权，而 XPG 则成为 UNIX 操作系统必须满足的要求。

1.2 GNU 的诞生与发展

1.2.1 自由软件计划 GNU

自由软件的发起人 Richard Matthew Stallman（简称 RMS）认为，对所有软件知识产权的约束会妨碍技术的进步，并对社区无益。他倡导所有软件应摆脱知识产权的约束。他于 1983 年发起 GNU 计划，GNU（GNU's Not UNIX 的缩写）发音为“guh-NEW”。GNU 的目的是开发一个自由的类 UNIX 的完整操作系统。自由软件的含义是任何人可自由使用、学习、复制、修改和发布。

为了更好地开展 GNU 计划，1985 年，RMS 创立了自由软件基金会 FSF（Free Software Foundation）。由自由软件基金会负责对 GNU 计划进行组织和推广工作，依靠一些公司捐助和其他商业捐助来维持，其中也有来自个人的捐款。

经过多年努力，到 1990 年，GNU 计划已开发出大量高质量软件，包括文字编辑器 emacs、C 语言编译器 gcc、C 语言函数库 glibc、Shell 解析器 bash 以及类 UNIX 的工具软件，当时唯一未完

成的是操作系统内核 HURD。这些软件为 Linux 操作系统的开发创造了良好的环境，是 Linux 能够诞生的重要基础。

1.2.2　许可证协议

1. GPL

GNU GPL（GNU General Public License）通用公共许可证由 RMS 于 1989 年为 GNU 计划撰写，协议规定用户可以自由使用、复制、修改和发布自由软件，协议要求在对软件进行修改后，如果要再次发布，需将已修改的部分同时发布出来。GPL 协议的目的是推广自由软件的使用和学习，并且防止一些别有用心的公司在对免费软件进行修改后申请版权，阻碍软件的进一步推广。

除 GPL 外，存在多种开源许可证协议。例如，LGPL 和 BSD 等。下面对 LGPL 和 BSD 许可证协以进行简单介绍。

2. LGPL

LGPL（Lesser General Public License）是一种基于 GPL 的扩展协议，它放宽了用户使用源代码的限制，允许源代码以链接库的形式提供给商业开发。Glibc 遵守 LGPL 协议。

3. BSD

BSD（Berkley Software Distribution）是一种具有较多灵活性的开源协议，用户可以享有更多的权力，但除了下面两个限制条件。

（1）复制权必须被保留。

（2）在没有得到原作者允许的情况下，软件不能进行商业应用。

1.2.3　自由软件和开源软件

GNU 计划的创始人 RMS 倡导的自由软件（free software）是一种可以不受限制地自由使用、复制、研究、修改和分发的软件。而开源软件（open source software）是指一种公开源代码的软件，通常用户可以使用、复制、修改和分发软件的源代码。从这个意义上讲，两者没有区别，但是它们代表两种不同的哲学理念。自由软件的目的在于自由地“分享”与“协作”。开源软件则是从技术的角度，为了提高软件质量所采用的一种开发模式。

1.3　Linux 内核

1.3.1　Minix 操作系统

UNIX 从版本 7 开始，由于 AT&T 公司的商业目的，不再公布 UNIX 源代码。出于操作系统教学的需要，1987 年，荷兰籍教授 Andrew Tanenbaum 开发出基于 PC 机的类 UNIX 的操作系，统命名为 Minix。Minix 采用 C 语言和少量汇编语言，不含任何 AT&T UNIX 的代码，Minix 内核的代码量较小，采用基于模块的微内核结构，只有部分内核代码运行在内核模式下，其他则以进程的形式运行于用户模式例如，设备驱动程序和文件系统等。这样，驱动程序的故障不会导致系统的崩溃，也无需重新编译和重新启动内核，提高了系统的安全性。

同时，Andrew Tanenbaum 出版了一本名为《操作系统教程：Minix 设计与实现》的教材，对 Minix 的实现机制进行了详细描述。目前，Minix 的最新版本为 Minix3，相关源代码可以从

www.minix.org 下载，源代码遵守 BSD 版权协议。

1.3.2 Linux 的产生与发展

1991 年，芬兰赫尔辛基大学学生 Linus Torvalds 在 Minix 设计思想的基础上，在 Internet 上发布了 Linux0.01 内核。当时正逢 GNU 的操作系统内核未完成之际，从此，Linux 内核与 GNU 相结合，奠定了此后 Linux 系统的基础。

当时，Linux0.01 版只能运行于 386 处理器上，只提供有限的设备驱动，只支持 Minix 文件系统，对网络不提供支持。此后，在全世界 Linux 开发人员的共同努力下，Linux 内核不断增加新功能和优化结构，目前仍处在发展之中。

1994 年，Linux1.0 发布，和 Linux0.01 相比，增加了新的文件系统、内存文件映射和对 TCP/IP 的支持。

1996 年，Linux2.0 内核版本发布，增加了对多种硬件体系结构和多处理器结构的支持，内存管理代码进行了改进，提升了 TCP/IP 性能，提供了对内核线程的支持。

2006 年，Linux2.6 内核版本发布，将以往的非抢占式内核升级为抢占式内核，改进了进程调度策略，以适应实时应用环境的需要。

1.3.3 Linux 内核版本

Linux 内核版本的命名格式为 Linux-X.Y.Z，其中，X 为主版本号，X 的变化表示 Linux 内核在设计上有较大变化；Y 为次版本号，Y 的变化表示 Linux 内核有了一定的改变，当 Y 是偶数时，表示该版本为发行版，代码运行稳定，当 Y 为奇数时，表示该版本为开发版，技术最新，代码处于测试阶段；Z 为末尾版本号，Z 表示仅对版本有微小改变。

1.3.4 Linux 内核的分类

Linux 内核从发行组织和应用领域的角度，可分为不同的类型，分别由不同组织进行维护，常见的 Linux 内核如下。

1. 标准 Linux 内核

标准 Linux 内核为通常意义上的 Linux 内核，由网站 www.kernel.org 维护。这些 Linux 内核并不总适用于 Linux 所支持的体系结构，该站点上的内核仅确保在 x86 体系结构上可正常运行，它是基于 x86 处理器的内核。

2. 嵌入式 Linux 内核

为了在嵌入式系统中应用 Linux 内核，在标准内核的基础上，开发出了适用于不同体系结构的 Linux 内核版本。例如，在无内存管理单元的嵌入式系统上使用的 μClinux 和为 ARM 处理器开发的 ARM Linux 等。

1.4 Linux 系统

1.4.1 Linux 系统的概念

Linux 系统是指包含 Linux 内核、工具软件和应用程序等在内的一系列软件的集合，从软件

层次结构的角度，将 Linux 系统分为 Linux 内核、Shell 命令解释器、管理工具和图形用户界面 4 个部分，如图 1-2 所示。

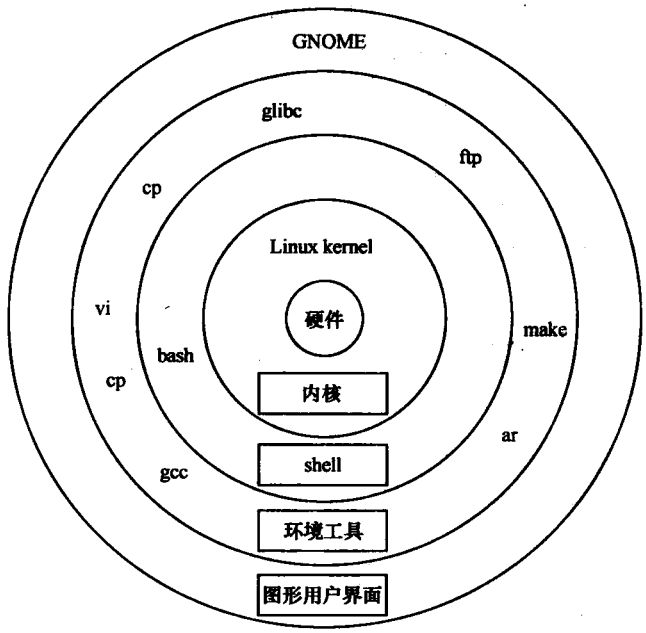


图 1-2　Linux 系统的层次结构

图 1-2 中，内核是整个操作系统的核心，为上层应用提供基本的软硬件访问服务，例如进程创建、文件存取、进程通信和信号处理等。Shell 是用户和内核交互的命令解析器。用户在 Shell 环境中可执行各种命令，实现对系统资源的访问。环境工具则是各种工具软件的集合，例如 vi 编辑器、磁盘分区和格式化工具、gcc 编译器等。用户可按实际需求定制所需的工具。图形用户界面是为了方便用户使用系统软硬件资源而开发的一种基于窗口的应用软件，用户可通过鼠标代替输入命令操作计算机，当然，用户可以不安装图形用户界面，而是通过 Shell 与计算机进行交互。

1.4.2　GNU/Linux

通常称 Linux 系统为 Linux，但从严格意义上说，Linux 指的只是 Linux 内核，Linux 内核不能构成一个完整的 Linux 系统。1991 年 Linux 的第一个版本公开发行，而此时 GNU 计划已完成除操作系统内核之外的大部分软件，因此，后来人们将 Linux 内核与 GNU 工具相结合，形成了 GNU/Linux。但并非所有的 Linux 系统都使用该名称，因为人们在 Linux 系统中也增加了其他元素，出于各种原因，多数 Linux 系统发行机构使用各自不同的称谓。

1.4.3　Linux 发行版

由企业、组织和个人开发的基于 GNU/Linux 的操作系统，称为 Linux 发行版。Linux 发行版形式多样，从功能齐全的桌面系统和服务器系统到面向特定应用的嵌入式系统，都有各自不同的特点。这些发行版可分为商业发行版与社区发行版，例如，Fedora（Red Hat）、OpenSUSE（Novell）、Ubuntu （Canonical 公司）和 Mandriva Linux 属于商业发行版，社区发行版由自由软件社区提供支持，例如 Debian 和 Gentoo 等。下面给出常用的 Linux 发行版，如表 1-4 所示。

表 1-4 Linux 发行版

Linux 发行版	特　点	网　址
RedHat	易用，易维护，专业，应用广泛	www.redhat.com
Debian	非商业组织维护，功能强大	www.debian.org
Mandrake	容易安装与使用	www.mandrakesoft.com
Novell/SuSE	欧洲大陆的 Linux	www.suse.com
Ubuntu	易于使用，版本更新快	http://www.ubuntu.com/
Gentoo Linux	使用了由 RHEL 提供的源码资源	http://www.gentoo.org/

1.5 Linux 系统的商业运营模式

自由软件的源代码是开放的，任何人可以自由复制、修改和发布，但必须遵循自由软件的许可证协议，同时，开源不等于免费。开源软件与商业本身并不冲突，开源软件可进行商业运营，事实上，它也是一种新兴的商业模式。开源软件本身的确是免费的，但开发者最初的意图是为了通过后续服务或出售赢利。下面给出开源软件的几种商业运营模式。

1. 多种产品线

利用开源软件带动商业软件的销售。例如，开源客户端软件带动了商业服务器软件的销售，借用开源版本带动商业许可版本的产品销售。这种模式应用得比较普遍。如 MySQL 产品就同时推出面向个人和企业的两种版本，即开源版本和专业版本，分别采用不同的授权方式。开源版本完全免费以便更好地推广，而从专业版的许可销售和支持服务中获得收入。又如 Red Hat 将原桌面操作系统转为 Fedora 项目，借 Fedora Core Linux 在开源社区的影响带动 Red Hat 企业版的销售。

2. 技术服务型

为开源软件提供技术服务。例如：JBoss 应用服务器完全免费，而通过提供技术文档、培训和二次开发等技术服务盈利。

3. 软、硬件一体化

在硬件产品中植入开源软件，开源软件不是利润的中心。这种模式被很多大型公司广泛采纳。例如 IBM 和 HP 等服务器厂商，通过捆绑免费的 Linux 操作系统销售硬件服务器。而 Sun 公司近期将其 Solaris 操作系统开放源代码，以确保服务器硬件的销售收入，也是这种模式的体现。

4. 附属品

出售开放源代码的附加产品。例如专业出版的文档和书籍等。O'Reilly 集团是销售开源软件附加产品公司的典型案例，它出版了很多优秀的开源软件的参考资料。