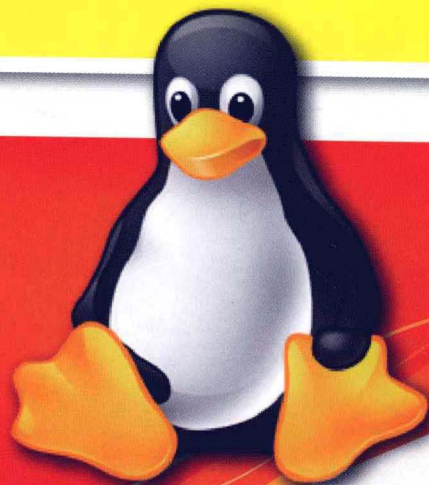


实例详解丛书

L^Cinux

编程80例详解

王友九 张艳利 郑富娥 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

实例详解丛书

Linux C 编程 80 例详解

王友九 张艳利 郑富娥 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由从事 Linux 系统应用和开发多年的工程师编写。书中以简明风趣的语言介绍 Linux 系统基础知识和 Linux C 语言开发过程中的细微知识点。本书共包括基础部分、Shell 部分、系统管理、编程基础、网络编程、编程拓展六大部分，每一部分都安排了多个章节分别展开讲解。本书不仅在讲解基础知识入门方面浅显易懂，最大的特点还在于提供了 80 个开发实例，读者可以通过这些实例理解并掌握 Linux C 语言的知识点和开发过程。

本书的读者目标主要是 Linux 系统、Linux C 编程的爱好者，同时也可供 Linux 系统初学者、大中专院校学生学习使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

Linux C 编程 80 例详解 / 王友九，张艳利，郑富娥编著. —北京：电子工业出版社，2013.2

（实例详解丛书）

ISBN 978-7-121-19286-9

I. ①L… II. ①王… ②张… ③郑… III. ①Linux 操作系统—程序设计②C 语言—程序设计

IV. ①TP316.89②TP312

中国版本图书馆 CIP 数据核字（2012）第 303996 号

策划编辑：王敬栋（wangjd@phei.com.cn）

责任编辑：徐 萍

印 刷：三河市鑫金马印装有限公司

装 订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：18.75 字数：480 千字

印 次：2013 年 2 月第 1 次印刷

印 数：4000 册 定价：48.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

关于本书

本书是 Linux C 语言开发从入门到提高的教材，通过理论与实例结合的方式，深入浅出地介绍其使用方法和技巧，目的在于让读者快速掌握 Linux C 语言编程开发方法。本书在编写过程中力求做到精益求精、浅显易懂、内容翔实，通过实用性比较强的 80 个实例细致地讲述了具体的编程方法、关键的细节技巧和工程应用实践经验。

本书从 Linux 系统的基础知识入手，介绍了 C 语言概述、Shell 之 Bash、进阶 Shell、数据流重导向与管线命令等基础内容，之后介绍了 Linux 系统管理、SecureCRT、编辑器、编译器、调试器等开发工具，从第 11 章开始讲解 make、网络编程、TCP 套接字编程、UDP 套接字编程及 Qt 图形编程等具体编程实例的开发。本书共分为 15 章，通过 80 个应用实例介绍了 Linux C 语言的开发方法。

主要内容

第 1 章 Linux 概述，包括概述、Linux 家族一览、Linux 文件系统、Linux 目录结构。

第 2 章 C 语言概述，包括数据类型、C 语言的运算符及语句、C 语言的函数、C 语言的程序。

第 3 章 Shell 之 Bash，包括 Shell、Bash、基础命令新鲜体验。

第 4 章 进阶 Shell，包括 Shell 变量及编程、Bash 的环境配置文件、Shell 脚本的应用。

第 5 章 数据流重导向与管线命令，包括数据流重导向、管线命令、多命令执行。

第 6 章 Linux 系统管理，包括基本管理操作、系统配置文件。

第 7 章 SecureCRT，包括 SSH、安装 SecureCRT、常用选项设置、主菜单介绍、主工具栏介绍、实际体验。

第 8 章 编辑器，包括编辑器 Vim、Vim 指令模式、Vim 编辑模式。

第 9 章 编译器，包括编译器 GCC、GCC 用法、GCC 延续。

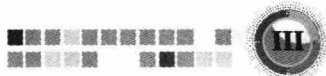
第 10 章 程序调试法宝——调试器，包括 GDB、GDB 常规接触、GDB 实战。

第 11 章 make，包括 make、makefile、makefile 的编写语法及规则、隐含规则。

第 12 章 戏说网络编程，包括协议概述、套接字编程。

第 13 章 TCP 套接字编程，包括基于 TCP 的 C/S 模式、TCP 编程流程概述、TCP 编程进阶、TCP 套接字编程实战。

第 14 章 UDP 套接字编程，包括基于 UDP 的 C/S 模式、UDP 编程流程概述、UDP 编程进阶、UDP 套接字编程实战。



本书每一章均采用结构化的格式排列，主要包括以下内容：

- 知识点介绍——详细阐述基本知识点，其中画龙点睛式地加入工程人员实际使用中的观点、看法及理解。
- 实例——根据本章内容举例，详细讲解实例的每一个步骤。
- 知识点提示——回顾本章的主要内容，并加以必要的提示。
- 本章小结——对本章内容进行总结。

- 本书采用简洁、诙谐的语言，阐述编者所理解的知识点，通俗易懂，且不乏味。
- 本书涵盖了 C 语言的常用知识点，SecureCRT 的用法，Linux 系统的 Shell、Vim、GCC、GDB、make，网络编程，Qt 图形编程。
- 本书在介绍知识点的同时列举了大量实例，所有实例均经过工程技术人员的调试并成功运行。

本书主要针对 Linux 系统、Linux C 编程的爱好者编写, 可供 Linux 系统初学者、大专院校学生学习使用, 也可作为中、高级开发人员的参考用书。

本书由王友九、张艳利、郑富娥编著，同时参与编写的还有陶镇、强秀华、李式琦、张为平、严雨、李若谷、王闯、严安国、何世兰、张玉梅、韩柯华、徐慧超等，在此一并表示感谢。

由于作者水平有限，书中出现错误和疏漏在所难免，恳请广大读者批评指正，提出宝贵的建议和意见。

2012 年 11 月

目 录

第 1 章 Linux 概述	(1)
1.1 概述	(1)
1.2 Linux 家族一览	(2)
1.2.1 Ubuntu	(2)
1.2.2 CentOS	(3)
1.2.3 其他版本	(3)
1.3 Linux 文件系统	(5)
1.3.1 Linux 文件系统	(5)
1.3.2 Linux 文件类型及属性	(6)
1.4 Linux 目录结构	(7)
1.4.1 Linux 系统结构	(7)
1.4.2 目录结构	(8)
1.5 知识点提示	(9)
1.6 本章小结	(9)
第 2 章 C 语言概述	(10)
2.1 数据类型	(10)
2.1.1 基本数据类型	(10)
2.1.2 指针类型	(13)
2.1.3 构造类型	(14)
2.1.4 其他类型	(18)
2.2 C 语言的运算符及语句	(19)
2.2.1 【实例 1】——C 语言的运算符	(19)
2.2.2 【实例 2】——C 语言的语句	(20)
2.2.3 【实例 3】——C 语句全接触	(26)
2.2.4 【实例 4】——函数之 C 语句	(28)
2.3 C 语言的函数	(28)
2.3.1 函数结构	(29)
2.3.2 函数类型	(30)
2.3.3 【实例 5】——函数结构全接触	(31)
2.3.4 【实例 6】——函数深入	(32)
2.4 C 语言的程序	(34)
2.4.1 程序构成	(34)

2.4.2	从编辑到运行	(34)
2.4.3	【实例 7】——多文件程序	(35)
2.5	知识点提示	(37)
2.6	本章小结	(38)
第 3 章	Shell 之 Bash	(39)
3.1	Shell	(39)
3.1.1	何谓 Shell	(39)
3.1.2	Shell 种种	(40)
3.2	Bash	(40)
3.2.1	Bash 的功能概述	(41)
3.2.2	Bash 的命令	(43)
3.3	基础命令新鲜体验	(44)
3.3.1	【实例 8】——date 命令	(44)
3.3.2	【实例 9】——cal 命令	(45)
3.3.3	【实例 10】——bc 命令	(46)
3.4	知识点提示	(48)
3.5	本章小结	(48)
第 4 章	进阶 Shell	(49)
4.1	Shell 变量及编程	(49)
4.1.1	变量介绍	(49)
4.1.2	环境变量	(50)
4.1.3	变量设定规则	(51)
4.1.4	Shell Script 简介	(53)
4.2	Bash 的环境配置文件	(53)
4.2.1	login Shell 与 non-login Shell	(54)
4.2.2	系统设置	(54)
4.2.3	用户设置	(55)
4.3	Shell 脚本的应用	(55)
4.3.1	【实例 11】——Linux 优先无线上网	(55)
4.3.2	【实例 12】——安全删除和恢复文件	(56)
4.3.3	【实例 13】——定时执行指定程序	(58)
4.4	知识点提示	(60)
4.5	本章小结	(60)
第 5 章	数据流重导向与管线命令	(61)
5.1	数据流重导向	(61)
5.1.1	何谓数据流重导向	(61)
5.1.2	【实例 14】——标准输出与标准错误输出	(61)
5.1.3	【实例 15】——标准输入	(63)
5.1.4	【实例 16】——/dev/null 的用法	(64)

5.2	管线命令	(65)
5.2.1	管线命令来了	(65)
5.2.2	【实例 17】——选择/选取类命令 cut、grep	(65)
5.2.3	【实例 18】——排序类命令 sort、wc、uniq	(67)
5.2.4	【实例 19】——双向重导向命令 tee	(69)
5.2.5	【实例 20】——字符转换类命令 tr、col、join、paste、expand	(70)
5.2.6	【实例 21】——分割类命令 split	(75)
5.2.7	【实例 22】——参数代换命令 xargs	(77)
5.3	多命令执行	(78)
5.3.1	【实例 23】——多命令执行之 ;	(78)
5.3.2	【实例 24】——多命令执行之 &&	(79)
5.3.3	【实例 25】——多命令执行之 	(79)
5.4	知识点提示	(80)
5.5	本章小结	(80)
第 6 章	Linux 系统管理	(81)
6.1	基本管理操作	(81)
6.1.1	【实例 26】——系统信息管理	(81)
6.1.2	【实例 27】——磁盘管理	(84)
6.1.3	文件命令	(89)
6.1.4	【实例 28】——打包压缩命令	(94)
6.1.5	软件管理命令	(97)
6.1.6	进程管理命令	(100)
6.2	系统配置文件	(103)
6.2.1	全局配置	(103)
6.2.2	用户配置	(104)
6.3	知识点提示	(105)
6.4	本章小结	(106)
第 7 章	SecureCRT	(107)
7.1	SSH	(107)
7.2	安装 SecureCRT	(107)
7.3	常用选项设置	(109)
7.3.1	风格设置	(110)
7.3.2	字体设置	(111)
7.3.3	编码设置	(112)
7.3.4	自动断开连接	(112)
7.3.5	显示行数	(113)
7.3.6	标签设置	(113)
7.4	主菜单介绍	(114)
7.4.1	文件 (F)	(114)

7.4.2	编辑 (E)	(115)
7.4.3	查看 (V)	(115)
7.4.4	选项 (O)	(116)
7.4.5	传输 (T)	(117)
7.4.6	脚本 (S)	(117)
7.4.7	工具 (L)	(118)
7.4.8	帮助 (H)	(118)
7.5	主工具栏介绍	(119)
7.6	实际体验	(119)
7.6.1	【实例 29】——建立连接	(119)
7.6.2	【实例 30】——断开连接	(122)
7.6.3	【实例 31】——标签移动与切换	(122)
7.6.4	【实例 32】——上传和下载文件	(123)
7.7	知识点提示	(125)
7.8	本章小结	(126)
第 8 章	编辑器	(127)
8.1	编辑器 Vim	(127)
8.1.1	Vim 概述	(127)
8.1.2	【实例 33】——Vim 的启动、退出	(128)
8.1.3	【实例 34】——Vim 模式及模式切换	(129)
8.2	Vim 指令模式	(131)
8.2.1	【实例 35】——光标移动操作	(131)
8.2.2	【实例 36】——查找操作	(132)
8.2.3	【实例 37】——复制粘贴	(132)
8.2.4	【实例 38】——删除操作	(133)
8.2.5	【实例 39】——修改操作	(134)
8.3	Vim 编辑模式	(135)
8.3.1	光标移动	(135)
8.3.2	输入	(135)
8.3.3	删除	(136)
8.4	知识点提示	(136)
8.5	本章小结	(136)
第 9 章	编译器	(137)
9.1	编译器 GCC	(137)
9.1.1	GCC 概述	(137)
9.1.2	【实例 40】——GCC 初识	(138)
9.1.3	剖析 GCC	(139)
9.2	GCC 用法	(139)
9.2.1	GCC 基本用法及其选项	(139)

9.2.2 【实例 41】——只编译子程序 (-c)	(140)
9.2.3 【实例 42】——产生目标文件 (-o)	(141)
9.2.4 【实例 43】——附加调试信息 (-g)	(141)
9.2.5 【实例 44】——多文件编译	(143)
9.2.6 【实例 45】——连接库文件	(144)
9.2.7 【实例 46】——综合示例	(144)
9.3 GCC 延续	(146)
9.3.1 【实例 47】——语法, 懂的, 却错了	(146)
9.3.2 【实例 48】——一个都不能少的头文件	(147)
9.3.3 【实例 49】——站在巨人的肩上, 却忘了巨人的存在	(147)
9.3.4 【实例 50】——变量未定义而使用	(148)
9.4 知识点提示	(149)
9.5 本章小结	(150)
第 10 章 程序调试法宝——调试器	(151)
10.1 GDB	(151)
10.1.1 GDB 初体验	(151)
10.1.2 【实例 51】——GDB 接触	(152)
10.1.3 GDB 的启动、退出	(155)
10.2 GDB 常规接触	(155)
10.2.1 程序调试法宝之运行程序	(155)
10.2.2 程序调试法宝之单步运行	(156)
10.2.3 程序调试法宝之设置断点	(157)
10.2.4 程序调试法宝之设置观察点	(157)
10.2.5 程序调试法宝之设置捕捉点	(158)
10.2.6 程序调试法宝之维护停止点	(158)
10.2.7 程序调试法宝之停止条件维护	(159)
10.3 【实例 52】——GDB 实战	(160)
10.4 知识点提示	(167)
10.5 本章小结	(167)
第 11 章 make	(168)
11.1 make	(168)
11.1.1 程序的诞生	(168)
11.1.2 make 的工作原理	(168)
11.1.3 make 的语法及参数选项	(170)
11.2 makefile	(172)
11.2.1 剖析 makefile	(172)
11.2.2 说说 makefile 的文件名	(173)
11.2.3 makefile 的包含	(173)
11.3 makefile 的编写语法及规则	(174)

11.3.1 【实例 53】——makefile 的语法	(174)
11.3.2 【实例 54】——makefile 与命令	(175)
11.3.3 【实例 55】——makefile 与变量	(176)
11.3.4 【实例 56】——makefile 与条件语句	(178)
11.3.5 【实例 57】——makefile 与函数	(181)
11.4 隐含规则	(183)
11.4.1 【实例 58】——使用隐含规则	(183)
11.4.2 隐含规则一览表	(183)
11.5 知识点提示	(184)
11.6 本章小结	(185)
第 12 章 戏说网络编程	(186)
12.1 协议概述	(186)
12.1.1 TCP 协议简介	(187)
12.1.2 IP 协议简介	(188)
12.1.3 UDP 协议	(189)
12.1.4 端口号	(190)
12.2 套接字编程	(190)
12.2.1 套接字简介	(190)
12.2.2 套接字类型及地址结构	(190)
12.2.3 【实例 59】——C/S 模型	(192)
12.2.4 【实例 60】——常见服务器模型	(195)
12.3 知识点提示	(198)
12.4 本章小结	(199)
第 13 章 TCP 套接字编程	(200)
13.1 基于 TCP 的 C/S 模式	(200)
13.2 TCP 编程流程概述	(201)
13.2.1 【实例 61】——Socket 建立	(201)
13.2.2 【实例 62】——Socket 配置	(202)
13.2.3 【实例 63】——Socket 连接	(205)
13.2.4 【实例 64】——Socket 数据收发	(209)
13.2.5 【实例 65】——域名转换 IP	(213)
13.2.6 【实例 66】——其他相关函数	(215)
13.3 TCP 编程进阶	(217)
13.3.1 I/O 模式	(217)
13.3.2 【实例 67】——select 函数	(217)
13.3.3 【实例 68】——poll 函数	(220)
13.3.4 【实例 69】——epoll 函数	(224)
13.4 【实例 70】——TCP 套接字编程实战	(228)
13.5 知识点提示	(235)



13.6 本章小结	(236)
第 14 章 UDP 套接字编程	(237)
14.1 基于 UDP 的 C/S 模式	(237)
14.2 UDP 编程流程概述	(238)
14.2.1 UDP 编程流程	(238)
14.2.2 TCP 与 UDP 编程的区别	(238)
14.3 UDP 编程进阶	(239)
14.3.1 sendto()函数	(239)
14.3.2 recvfrom()函数	(240)
14.3.3 数据包的丢失和处理	(241)
14.4 【实例 71】——UDP 套接字编程实战	(241)
14.5 知识点提示	(246)
14.6 本章小结	(246)
第 15 章 Qt 图形编程	(247)
15.1 Qt 与 Qt Creator 接触	(247)
15.1.1 初识 Qt 与 Qt Creator	(247)
15.1.2 【实例 72】——Hello World 程序	(249)
15.1.3 【实例 73】——多窗口程序	(253)
15.1.4 【实例 74】——登录对话框	(258)
15.2 Qt Creator 简单应用	(262)
15.2.1 【实例 75】——常见的菜单和图标	(262)
15.2.2 【实例 76】——文本编辑的实现	(266)
15.2.3 【实例 77】——文本查找的实现	(273)
15.2.4 【实例 78】——状态栏制作	(275)
15.3 Qt Creator 深入	(277)
15.3.1 【实例 79】——简易 FTP 客户端	(277)
15.3.2 【实例 80】——完善 FTP 客户端	(280)
15.4 知识点提示	(287)
15.5 本章小结	(287)

第 1 章

Linux 概述

1.1 概 述

Linux 是由芬兰大学生 Linus Torvalds 于 1991 年开发的类 UNIX 系统，起初只有一批爱好者和学习 UNIX 的人在使用，后来逐渐发展成为一个成熟的操作系统。讲到 UNIX，它开发于 1969 年的贝尔实验室，是一个强大的多用户、多任务操作系统，但由于它比较庞大，而且价格昂贵，所以不便于普通用户使用。Linux 不仅继承了 UNIX 的优点，还弥补了 UNIX 庞大、昂贵等的缺点，因而得到了广泛的应用。小到手机、平板电脑、路由器和机顶盒，大到台式计算机和超级计算机，随处可见 Linux 的身影。

Linux 采用一只坐着的小企鹅图案作为其系统的 Logo，如图 1-1 所示。企鹅是南极洲的标志性动物，根据国际公约，南极洲为全人类共同所有，不属于当今世界上的任何国家，任何国家都无权将南极洲纳入其版图。Linux 选择企鹅图案作为标志，其意图也是想表明：开源的 Linux，为全人类共同所有，任何公司无权将其私有化。

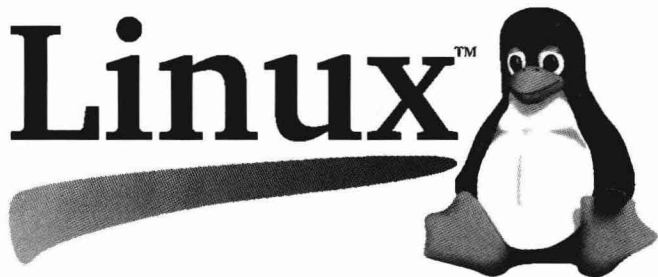


图 1-1 Linux 系统 Logo

Linux 的源码是开放的，可运用于许多硬件平台，支持多达 32 种文件系统，支持大量的外部设备。它包含人们期待的操作系统所能拥有的优良特性，包括真正的多任务、虚拟内存、目前最快的 TCP/IP 驱动程序、共享库和理想的多用户支持；它还符合 X/Open 标准，具有完全自由的 X-Window 实现方式；Linux 同 UNIX 一样，具有最先进的网络特性，且支持所有通用的 Internet 协议，既可以作为客户端也可以作为服务器。这些优异的特性使 Linux 具有广泛的用途，它被应用于以下我们比较熟悉的领域：

➤ Android 手机系统；

- Megoo 操作系统;
- 机顶盒;
- 路由器;
- 自动柜员机 (ATM);
- 大型服务器。

1.2 Linux 家族一览

Linux 系统真的算得上是一个大家族,已有的 Linux 发行版数量已经超过 300 个,最普遍被使用的发行版大约有 12 个。一个典型的 Linux 发行版包括:Linux 核心,一些 GNU 库和工具,命令行 Shell,图形界面的 X 窗口系统和相应的桌面环境,如 KDE 或 GNOME,以及数千种从办公包、编译器、文本编辑器到科学工具的应用软件。

主流的 Linux 发行版包括 Ubuntu、DebianGNU/Linux、Fedora、Gentoo、Mandriva Linux、PCLinuxOS、SlackwareLinux、openSUSE、ArchLinux、Puppylinux、Mint、CentOS、RedHat 等。其中,Ubuntu 是个人用户使用最为广泛的发行版之一,CentOS 是服务器用户使用最为广泛的发行版之一。

作为新手,要求系统安装简单、使用方便、配置容易、上手速度快且兼容性好,Ubuntu、Fedora、openSUSE 等发行版本正好满足这些需求,推荐使用 Ubuntu。当然,如果更习惯 Windows 系统的界面风格,也可以选择上手容易的红旗、雨林木风等国产的 Linux 操作系统。至于 Gentoo、Arch 等 Linux 系统,虽然运行速度快,但入门较难,且等我们成为 Linux 高手以后再尝试吧。

1.2.1 Ubuntu

Ubuntu 的 Logo 如图 1-2 所示。



图 1-2 Ubuntu 系统 Logo

Ubuntu 是一个以桌面应用为主的 Linux 操作系统,其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词(译为吾帮托或乌班图),意思是“人性”、“我的存在是因为大家的存在”,是非洲传统的一种价值观,类似华人社会的“仁爱”思想。Ubuntu 基于 Debian 发行版和 GNOME 桌面环境,与 Debian 的不同在于它每 6 个月会发布一个新版本。Ubuntu 的目标在于为一般用户提供一个最新的同时又相当稳定的、主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量,用户可以方便地从社区获得帮助。

Ubuntu 是一个相对较新的 Linux 发行版,默认采用 GNOME 桌面系统,如果用户比较

喜欢使用 KDE 桌面, 则可以使用 Kubuntu 来代替 Ubuntu。安装 Ubuntu 极其简单, 只需按照提示一步一步进行即可, 其安装和 Windows 一样简便。值得一提的是, Ubuntu 被誉为对硬件支持最好、最全面的 Linux 发行版之一, 许多在其他发行版上无法使用的硬件, 在 Ubuntu 上都可以轻松搞定。

Ubuntu 采用自行加强的内核, 安全性方面更胜一筹。并且, Ubuntu 必须从第一个创建的用户通过 `su` 或 `sudo` 命令来获取 root 权限 (这似乎不太方便, 但无疑避免了用户因为粗心而损坏系统, 增强了安全性), 默认无法直接以 root 用户的身份登录。

Ubuntu 的版本每 6 个月更新一次, 弥补了 Debian 更新缓慢的不足, 而且具有人气颇高的论坛提供优秀的资源和技术支持, 便于新手遇到问题时寻求其他用户的帮助。

1.2.2

CentOS

CentOS 的 Logo 如图 1-3 所示。



图 1-3 CentOS 系统 Logo

CentOS (Community Enterprise Operating System) 是红帽 (RedHat) 操作系统的延续, 在红帽操作系统决定不再提供免费的技术支持之后, CentOS 便应运而生了。

CentOS 是通过编译 Red Hat Enterprise Linux (以下简称 RHEL) 按照开源许可规定所放出的源代码而生成的操作系统, 它完全兼容 RHEL, 与 RHEL 最大的不同之处在于 CentOS 不包含封闭的代码。当然, 所有红帽操作系统的 Logo 和标识也被替换成了 CentOS 自己的标识。

CentOS 的优势如下:

- CentOS 完全免费, 不再依赖 RHEL 所需要的序列号;
- CentOS 特有的 `yum` 命令支持在线升级, 还可以即时更新系统, 不像 RHEL 那样需要花钱购买支持服务;
- CentOS 修正了许多 RHEL 的 bug;
- CentOS 具备与 RHEL 相同的性能和感受;
- CentOS 适用于对稳定性有严格要求的场合。

1.2.3

其他版本

前面我们了解了 Linux 家族中的 Ubuntu 与 CentOS 系统, 接下来继续简要了解一下这个家族中的其他成员。

1. PCLinuxOS

PCLinuxOS 秉承“简单易用，安全无忧”的理念，无须安装便可直接从 CD 上加载并运行，集成的程序丰富。不过，由于 LiveCD 光盘的数据读取速度缓慢，导致系统性能存在瓶颈。

2. openSUSE

openSUSE 由 Novell 公司开发，是一个非常专业、优秀的发行版，以华丽的用户界面而著称。其操作简便，适合新手使用，也是当前最流行的版本之一。

3. Fedora

Fedora 属于 RedHat 的一个分支，它是由社区开放的免费操作系统。Fedora 定位于桌面应用，提供了丰富的软件包，同时，它的版本更新周期也仅有 6 个月之久。Fedora 拥有数量庞大的用户、优秀的社区技术支持及众多创新，适合新手使用，同时也是当前最流行的版本之一。

4. Debian

Debian 是迄今为止最遵循 GNU 规范的 Linux 系统。dpkg 是 Debian 系列特有的软件包管理工具，它被誉为所有 Linux 软件包管理工具中最强大的，配合 apt-get 命令在 Debian 上安装、升级、删除和管理软件变得异常容易。Debian 具有优秀的网络和社区资源，许多发行版都基于 Debian（如 Ubuntu），它是最有影响力的发行版之一。不过 Debian 的安装相对比较麻烦，不适合新手使用。

5. Mandrake

Mandrake 是基于 RedHat 的发行版，其安装过程简单，使用 KDE 作为默认桌面系统，操作方便。Mandrake 具有友好的操作界面、图形配置工具、庞大的社区技术支持，是国际上比较有影响力的版本之一。

6. Slackware

Slackware 是一个历史悠久的 Linux 发行版。其他主流发行版强调易用性的时候，它依然固执地追求最原始的效率——所有的配置均要通过配置文件来进行。Slackware 稳定、安全，所以有大批的忠实用户。由于尽量采用原版的软件包而不进行任何修改，所以制造新 bug 的概率比较小。Slackware 的版本更新周期较长，系统非常稳定、安全，并且高度坚持 UNIX 的规范，不过由于所有的配置均通过编辑文件来进行，因此只适用于对 Linux 非常熟悉的用户，而且其自动硬件检测能力较差，不适合新手使用。

7. Puppy

Puppy 是一个十分小巧的 Linux 发行版，其 ISO 映像文件容量约为 100MB。但是其功能比较全面，在 Puppy 中包含了必要的系统工具、网络软件、办公应用、影音娱乐工具等。Puppy 不仅整个系统都可在内存中运行，而且还支持从各种存储介质启动运行，如 U 盘、光盘、硬盘等。它适合用在一些较老的、对空间要求比较苛刻的设备上。

8. Gentoo

Gentoo 是一个基于源代码的 Linux 发行版，具有高度的可定制性。尽管安装时可以选择预先编译好的软件包，但是大部分使用 Gentoo 的用户都选择自己手动编译，所以 Gentoo 适合比较有 Linux 使用经验的老手使用。需要注意的是，由于编译软件需要消耗大量的时间，因此如果所有的软件都自己编译，则安装桌面系统需要较长的时间。



1.3 Linux 文件系统

文件系统是操作系统用于表示磁盘或分区上文件的方法和数据结构。不同的操作系统需要使用不同类型的文件系统，为了与其他操作系统兼容，相互交换数据，通常操作系统都支持多种类型的文件系统。Linux 内核支持十多种不同类型的文件系统，下面对 Linux 常用的文件系统及基本文件类型作一个简要介绍。

1.3.1 Linux 文件系统

1. ext2、ext3 和 ext4 文件系统

ext 是第一个专门为 Linux 设计的文件系统类型，称为扩展文件系统，在 Linux 发展的早期发挥过重要的作用。由于稳定性、速度和兼容性方面存在许多缺陷，ext 现在已经很少使用。

ext2 是为解决 ext 文件系统存在的缺陷而设计的可扩展、高性能的文件系统，称为二级扩展文件系统。ext2 于 1993 年发布，在速度和 CPU 利用率上具有更加突出的优势，是 GNU/Linux 系统中标准的文件系统，支持 256 字节的长文件名，文件存取性能极好。

ext3 是 ext2 的升级版，兼容 ext2，在 ext2 的基础上增加了文件系统日志记录功能，称为日志式文件系统，是目前 Linux 默认采用的文件系统。日志式文件系统在因断电或其他异常事件而停机重启后，操作系统会根据文件系统的日志，快速检测并恢复文件系统到正常的状态，并可缩短系统的恢复时间，提高数据的安全性。若对数据有较高的安全性要求，建议使用 ext3 文件系统。

ext4 是 ext3 的改进版，修改了 ext3 中部分重要的数据结构，而不仅仅像 ext3 对 ext2 那样，只是增加了一个日志功能而已。ext4 可以提供更优的性能和可靠性，同时具备更为丰富的功能。Linux kernel 自 2.6.28 开始正式支持新的 ext4 文件系统。

2. swap 文件系统

众所周知，现代操作系统都实现了“虚拟内存”这一技术，不但在功能上突破了物理内存的限制，使程序可以操纵大于实际物理内存的空间，而且更重要的是，“虚拟内存”是隔离每个进程的安全保护网，使每个进程都不受其他程序的干扰。

swap 空间的作用可简单描述为：当系统的物理内存不够用的时候，就需要将物理内存中的一部分空间释放出来，以供当前运行的程序使用。这些被释放的空间可能来自一些很长时间没有什么操作的程序，可以将其临时保存到 swap 空间中，等到那些程序要运行时，再从 swap 中恢复保存的数据到内存中。需要注意的是，系统总是在物理内存不够时，才进行 swap 交换。

3. vfat 文件系统

Linux 把 DOS、Windows 中采用的 FAT 文件系统（包括 FAT12、FAT16 和 FAT32）统称为 vfat 文件系统。

4. NFS 文件系统

NFS 即网络文件系统，用于在 UNIX 系统之间通过网络进行文件共享，用户可将网络