



国外经典教材·计算机科学与技术

PEARSON
Prentice
Hall

LINUX: THE TEXTBOOK

LINUX教程

Syed Mansoor Sarwar
(美) Robert Koretsky 著
Syed Aqeel Sarwar
李善平 施韦 林欣 译



清华大学出版社

国外经典教材·计算机科学与技术

Linux 教程

(LINUX: THE TEXTBOOK)

(美) Syed Mansoor Sarwar, Robert Koretsky, Syed Aqeel Sarwar 著

李善平 施 韦 林 欣 译

清华大学出版社

北 京

内 容 简 介

这是一本 Linux 入门教科书, 为 Linux 初学者定制, 全面、系统地介绍了 Linux 及其环境。本书附有数百个精心设计的实战操作练习和习题, 帮助读者高频率地复习命令、原语和概念。各章节内容主要包括 Linux 历史、Linux 的 shell 命令和 shell 编程、编辑工具、电子邮件、文件操作、重定向和管道、进程、网络互连、Linux 软件开发、Linux 图形用户界面等。

本书适合作为高等院校计算机及相关专业的教材和教学参考书, 也可作为系统编程人员的自学教材和参考手册。

Simplified Chinese edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and Tsinghua University Press.

Original English language title from Proprietor's edition of the Work.

Original English language title: LINUX :THE TEXTBOOK, first edition by Syed Mansoor Sarwar, Robert Koretsky, Syed Aqeel Sarwar, Copyright © 2002

EISBN: 0-201-72595-9

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison Wesley Longman Inc..

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由培生教育出版集团授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-1795

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Linux 教程/(美)萨瓦(Sarwar, S.M.), (美)克雷兹(Koretsky, R.), (美)萨瓦(Sarwar, S.A.)著; 李善平, 施韦, 林欣译. —北京: 清华大学出版社, 2005.6

书名原文: LINUX: THE TEXTBOOK

(国外经典教材·计算机科学与技术)

ISBN 7-302-10768-8

I. L… II. ①萨… ②克… ③萨… ④李… ⑤施… ⑥林… III. Linux 操作系统-高等学校-教材
IV. TP316.89

中国版本图书馆CIP数据核字(2005)第028780号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 许存权

文稿编辑: 刘 丽

封面设计: 久久度文化

版式设计: 张红英

印 装 者: 三河市春园印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 37.5 字数: 864 千字

版 次: 2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

书 号: ISBN 7-302-10768-8/TP·7169

印 数: 1~4000

定 价: 65.00 元(附光盘 1 张)

译者的话

这是一本真正的 Linux 入门教科书。

本书为 Linux 初学者定制，不要求具备任何使用 Linux 的经验，因此对希望自学 Linux 的读者来说，不失为一种极好的选择。

本书的章节经过精心组织，每章都集中于若干主题。既有 Linux 重要概念的论述，又有从 Linux 用户界面到系统内部模块的层层深入的讲解。读者通过使用 Linux 工具，学习和掌握进程、文件、存储、I/O 管理等操作系统概念，进而加深和拓宽对操作系统原理的理解。本书全面、系统地涵盖了 Linux 软件开发工具、Bash 编程、TC shell 编程、术语表、Linux 安装、常用 shell 命令和常用工具索引等方面，一定会使 Linux 初学者受益匪浅。

必须着重推荐的是，本书的各章节联系和实战演练可谓别具一格。本书包含数百个精心设计的练习和习题，帮助读者高频率地复习命令、原语和概念。

本书的翻译由李善平、滕洁、何剑峰、王萍、彭刚、应杭、施韦、虞浩泽、李天佑、虞增辉、周强、付相君、林欣、王宇剑、杨琴、赖宗澐、郭鸣、吴震东、高庆、虞云翔、刘可嘉、王明伟等共同完成。由于文字功底、专业水准所限，译作难免有不足之处，敬请批评指正。

前 言

一个影响巨大的软件系统

“Linux 颇具颠覆性，就在 5 年前，谁能想到？这样一个全世界范围内广泛使用的操作系统是由几千个分散在全球，仅仅通过网线连接起来的开发人员，利用业余时间开发并整合出来的。”

Eric Raymond 《大教堂与小集市》

是的，这样一个软件系统的发展和形成的确令人不可思议。Linux 的前身甚至在小型机市场上都难以立足，而它却已成为这样一个颇具“颠覆性”的软件系统。它的成功对于那些企图破坏我们这个时代科技创新自由的势力无疑是个巨大的挑战。UNIX 社团所推崇的开源理念最终能够获得成功，还要归功于 Internet 技术的发展和成熟，以及在硬件平台上为个人用户提供最大限度发展空间的思想。如果 UNIX 最终是为黑客而存在，那么 Linux 本身就是他们个人自由的化身，过去曾经是，现在也是。

这本教科书是对 Internet 普及所带来的强大作用的证明，也是对数以千计自认为应该属于一个更大团体的个体黑客的证明。了解这个社团最好的方法就是上网，跟踪本书一些章节末尾列出的网上资源，了解 Linux 的缔造者的思想。我们确信，通过勤奋实践，对本书提出问题的思考和富有创意的解答，你就会享受到 Linux 社团用户所拥有的成就感。

写作的原因

在这本教材完成之时，Linux 已经在计算机科学和工程领域发展到一个颇为成熟的水平。这使我们觉得早该完成这样的一本 Linux 教科书了。总的来说，我们有 35 年以上的大学教学经验。完成这样一本教材的最初想法源于在教学中碰到的尴尬处境。在教授那些与 Linux 主题相关的课程时，我们发现能使用的素材，要么是非常复杂和庞大的 Linux 参考源代码，要么是“Linux 傻瓜书”之类的指导手册。显然它们用于教授那些意义非凡、体系完备的课程是不合适的。而且，用教科书的形式，通过编排一些章内指南、章内练习以及每章末尾的练习，可以更好地展示 Linux 的重要主题，使之适应教室的环境和“讲课—实验—作业”的教学模式。这样的展示显然是一种最佳的学习方式（即合理的主题编排、准备充分和适当的课时、实验内容和课后练习，同步配之教科书章节）。另外，因为本书内容兼具深度和广度，使之适合于任何想要在这一方面进一步研究和学习的读者。

写作的目的

我们的主要目的是向读者描述 Linux 应用程序的用户界面（AUI），并且在这个过程中让读者进一步深入理解系统内部的工作原理，以及一些重要的 Linux 概念、数据结构和算法。其中最重要的实例是 Linux 的文件、进程的概念，以及 I/O 重定向。其次，是为使用 C 语言和 shell 脚本的开发人员介绍一些重要的 Linux 软件工程工具。但总的来说，在本教材中我们并没有从标准 C 库

和 Linux 系统调用的角度介绍 Linux 应用程序界面(API)。编写本教材时,假设读者并不具备 Linux 或者编程方面的知识。

内容的编排格式

教材中编排了很多图表,以及大量的参考教程,交互式 shell 命令演示,练习和课后思考题。教材中对涉及到的每一个命令、工具和应用程序,都给出了相应的语法说明栏。详细介绍了命令的语法、使用目的、输出结果,以及常用的命令选项和特性。另外,每章都有一个小结。

本书的使用方法

如果将本书用作 Linux 的入门教材,那么除了第 20 章外,其他各章都应当列入教学内容。如果将本书用作操作系统概念和原理课程的辅助教材,那么应按照主要教材内容的编排顺序,对本书的内容逐一涉及。其中至少应当包含第 2 章、第 7 章以及第 13 章的内容。如果将本书作为学习 C 语言或者 shell 编程的课程的教科书,那么从第 7 章~第 20 章的全部内容,以及从第 3 章~第 6 章的相关章节对学生大有裨益。对于第 20 章的覆盖程度取决于课程的深度,导论性课程可以部分涉及,高级课程可以全部覆盖。

补充材料

通过访问 <http://lhotse.up.edu/~koretsky/linux.html> 站点可以获得章内练习题的答案、源代码,以及其他有助于更深入学习 Linux 的站点链接。也可以通过访问 Addison Wesley 的网站 www.awl.com/cseng/titles/0-201-72595-9 链接到以上站点。

有关每章习题的解决方案,可提供给教授本课程的教师。请联系当地 Addison-Wesley 销售代表。

我们为本书中所有的错误负完全责任。如果您发现本书内任何错误请发信至 sarwar@up.edu 和 koretsky@up.edu 与我们联系。我们将在下一版中加入您的反馈意见并修正您提出的错误。

致 谢

如果没有各方面的帮助,用不到一年的时间完成这本内容丰富的教材是不可能的。我们感谢 Addison-Wesley 所有参与人员。这本教材是我们第 2 次与编辑 Maite Suarez-Rivas 这位出色的专家合作。她支持我们编写这样一本教材的想法,并且提出了宝贵意见,保证了本书的及时完成。感谢 Maite! 同时也向她在 Addison-Wesley 工作的所有同事表示衷心的感谢。他们是 Lisa Hogue、Patti Mahtani、Helen Reebenacker、Regina Hagen 以及 Jarrod Gibbons。感谢他们的支持和专业精神。Lisa, 感谢你的全程帮助! 特别感谢 Pre-Press 公司项目经理 Marilyn Lloyd, 感谢她在本书出版阶段的耐心和很多有价值的建议以及出色的工作。

同时也要感谢下面这些评审人员,他们提出了有价值的反馈信息,给出了许多精确而且深刻的评论。他们的贡献毫无疑问提高了本书的质量:

Austin 社区学院的 Ronald E. Bass

Henry Ford 社区学院的 Thomas A. Burns

Ashland 大学的 Simon Gray

Carven 社区学院的 Chuck Lesko

Oregon 州立大学的 Toshimi Minoura

Minot 州立大学的 Selmer Moen

北卡罗来那大学 Chapel Hill 校区的 Gregory B. Newby

Franklin 大学的 Marianne Vakalis

田纳西州杰克逊联合大学的 G. Jan Wilms

亚里桑那州立大学的 Bill Wood

最后，还要感谢 Portland 大学工程学院计算实验室的 Dale Frakes，感谢他各方面的支持。每当我们需要他时，总能得到他的帮助。

个人致谢

Syed Mansoor Sarwar 如果没有家人给予的支持、理解、鼓励和爱，我决不可能在两年内完成两本书的写作工作。我的全家都应当得到这样特别的感谢。感激我的父母，感激他们给予我的爱，并且教会我认识自己。他们是我努力工作的榜样，也为我树立了道德的模范。至今我依然继续从他们的教育中汲取力量。感谢我的妻子 Robina，以及我的孩子 Hassaan 和 Maham。他们爱我，并且理解和支持我的工作，让我可以投入所有的时间和精力完成这本教材的写作工作。没有他们，我就写不了这本书。所以，感谢他们！感谢我的姐妹 Rizwana、Farhana 和我的兄弟 Masood、Nadeem、Aqeel 和 Nabeel，感谢他们给予的爱和友谊。尤其感谢 Nadeem 和 Nabeel，在我的工作中，他们一直不断地给我支持和鼓励。Muk hondie aiy hoon? Asif out!

感激 Tom Nelson，我的前任院长、导师和朋友。感谢在我的 Portland 大学职业生涯中所给予的鼓励。感谢我的朋友，Synplicity 公司的 Shahid Younas，以及我的同事 Aziz Inan 和 Matthew Kuhn，感谢他们的支持。

最后，特别要真诚感谢我的合作者，Robert M. Koretsky 和 Syed Aqeel Sarwar，感谢我们之间的愉快合作。

Robert M. Koretsky 感谢我的妻子 Kathe、女儿 Tara，以及我的儿子 Cody，感谢他们多年来给我的支持和爱。同时也感谢 Mansoor Sarwar，感谢他的鼓励和友谊。

Syed Aqeel Sarwar 感谢我的父母，他们耐心养育我、爱我，他们做出了很大的牺牲。也感谢我的哥哥 Syed Mansoor Sarwar 博士，感谢他不断的支持。我特别感谢我的兄弟 Masood Sarwar、Nadeem Sarwar 博士、Nabeel Sarwar 以及我亲爱的姐姐 Rizwana Sarwar 和 Farhana Sarwar 博士，感谢他们的友谊和支持。最后，感谢我的妻子 Mamoonah 的支持，还有我的儿子 Mohammad 的陪伴。

献给：

我的太太和孩子们

S.M.S.

我全家

R.M.K.

我的父母

S.A.S.

目 录

第 1 章 操作系统综述.....1	3.4 登录和退出..... 16
1.1 引言..... 1	3.5 改正命令行输入错误..... 19
1.2 操作系统是什么..... 1	3.6 重要的一些系统设置..... 19
1.3 操作系统提供的服务..... 2	3.7 对初学者有用的几个命令..... 20
1.4 文字与图形用户接口..... 2	3.7.1 修改密码..... 21
1.5 操作系统的类型..... 3	3.7.2 获取帮助..... 21
1.6 Linux 家族..... 4	3.7.3 找出用户及其 Linux 系统的名字..... 23
小结..... 4	小结..... 24
习题..... 5	习题..... 25
第 2 章 Linux 历史.....6	第 4 章 Linux shell.....26
2.1 引言..... 6	4.1 引言..... 26
2.2 开放源码..... 7	4.2 各种各样的 Linux shell..... 27
2.3 Linus Torvalds..... 7	4.2.1 各种 shell 程序..... 27
2.4 Linux 发行版..... 8	4.2.2 哪种 shell 能满足需要..... 28
2.5 内核的历史..... 8	4.2.3 更改 shell 的各种方法..... 29
2.6 Linux 的流行..... 9	4.2.4 shell 的启动文件和 环境变量..... 31
2.7 未来的发展..... 9	4.3 一些有用的多用途命令..... 32
2.8 Web 资源..... 9	4.3.1 创建和显示目录..... 32
小结..... 9	4.3.2 显示文件..... 34
习题..... 9	4.3.3 打印文件..... 34
第 3 章 开始行动..... 11	4.3.4 显示日历..... 34
3.1 引言..... 11	4.3.5 单向实时消息..... 34
3.2 计算机系统硬件..... 11	4.3.6 电子通信的通知和 允许..... 35
3.2.1 内存/主存..... 12	4.3.7 为命令创建假名..... 35
3.2.2 中央处理器 (CPU)..... 12	4.3.8 显示系统运行时间..... 37
3.2.3 磁盘..... 13	4.4 shell 元字符..... 37
3.2.4 总线..... 13	小结..... 38
3.2.5 输入/输出设备..... 13	习题..... 39
3.3 Linux 软件体系结构..... 14	第 5 章 编辑文本文件.....40
3.3.1 设备驱动程序层..... 14	5.1 引言..... 40
3.3.2 Linux 内核..... 14	5.2 如何使用 pico 编辑器进行 简短编辑..... 41
3.3.3 系统调用接口..... 15	
3.3.4 语言函数库..... 16	
3.3.5 Linux shell..... 16	
3.3.6 应用程序..... 16	

5.2.1	使用 pico 文本编辑器创建一封 E-mail	41	6.2.1	用 mail 命令发送和读取电子邮件	69
5.2.2	如何新建、保存一个文件并退出	41	6.2.2	用 mail 命令发送电子邮件 ...	70
5.2.3	常用的按键命令和光标移动命令	43	6.2.3	用 mail 命令阅读电子邮件 ...	72
5.2.4	剪切、粘贴和查找	44	6.2.4	将邮件存储到文件夹中	73
5.3	使用 vi 编辑器获取更多控制	45	6.2.5	回复、转发邮件并添加附件	73
5.3.1	shell 脚本文件	45	6.2.6	邮件别名	74
5.3.2	如何开始编辑文件, 保存文件并退出	46	6.3	elm——全屏显示电子邮件系统	75
5.3.3	vi 命令的格式和操作模式 ..	47	6.3.1	在 elm 中发送一封邮件	75
5.3.4	光标移动和编辑命令	49	6.3.2	用 elm 发送电子邮件	75
5.3.5	yank、put 命令(复制和粘贴)和 substitute 命令(搜索和替换)	50	6.3.3	用 elm 程序阅读电子邮件	77
5.3.6	设置 vi 环境	52	6.3.4	在 elm 中转发和回复邮件	79
5.3.7	从 vi 中执行 shell 命令	52	6.3.5	把邮件保存到文件夹中	80
5.4	使用 emacs 编辑器获取最大控制	53	6.3.6	elm 别名	80
5.4.1	DOS 别名	53	6.4	pine——另一个全屏电子邮件系统	82
5.4.2	如何开始编辑文件, 保存文件并退出	53	6.4.1	用 pine 发送邮件连同在 vi 中创建的附件	82
5.4.3	光标移动和编辑命令	54	6.4.2	用 pine 程序发送电子邮件 ...	82
5.4.4	键盘宏命令	56	6.4.3	用 pine 程序阅读电子邮件 ...	84
5.4.5	剪切或复制、粘贴、查找和替换	57	6.4.4	在 pine 中处理目录里的电子邮件	85
5.5	如何使用 XEmacs 进行图形化编辑	59	6.4.5	使用 pine 的地址簿	86
5.5.1	用 XEmacs 编辑数据文件 ...	59	6.4.6	pine 命令总结	88
5.5.2	XEmacs 菜单和按钮	60	6.5	图形化电子邮件系统——Kmail	90
5.5.3	如何开始编辑文件, 保存文件并退出	61	6.5.1	开始使用 Kmail	90
5.6	命令行编辑	62	6.5.2	在 Kmail 中读电子邮件	92
	小结	65	6.5.3	在 Kmail 程序发送邮件	93
	习题	65	6.5.4	将邮件简单过滤到目录中	95
第 6 章	电子邮件	67		小结	96
6.1	引言	67		习题	96
6.2	如何高效使用 Linux 中的 mail 命令	69	第 7 章	文件和文件系统结构	99
			7.1	引言	99
			7.2	Linux 文件概念	99
			7.3	文件类型	99
			7.3.1	简单/普通文件	100
			7.3.2	目录	100
			7.3.3	连接文件	101
			7.3.4	特殊文件(设备)	101

7.3.5 命名管道 (FIFO)	101	9.1 引言	138
7.4 文件系统结构	102	9.2 查看文本文件的内容	138
7.4.1 文件系统的组织	102	9.2.1 查看完整的文件	138
7.4.2 主目录和当前工作目录	102	9.2.2 每次一页地查看文件	140
7.4.3 绝对路径和相对路径	103	9.2.3 查看文件的头或尾	142
7.4.4 一些标准目录和文件	103	9.2.4 显示 NIS 数据库	145
7.5 文件结构导航	107	9.3 文件的复制、转移和删除	145
7.5.1 确定主目录的绝对路径	108	9.3.1 文件的复制	145
7.5.2 浏览文件系统	108	9.3.2 文件的转移	147
7.5.3 创建文件	111	9.3.3 文件的删除	148
7.5.4 创建和删除目录	111	9.3.4 文件大小的确定	149
7.5.5 使用目录堆栈	112	9.4 文件的追加	151
7.5.6 确定文件属性	115	9.5 文件的合并	152
7.5.7 确定文件内容的类型	117	9.6 文件的比较	153
7.6 Linux 中的文件表示和存储	118	9.7 删除重复行	157
7.7 标准文件和文件描述符	120	9.8 文件的打印与打印任务的控制	158
7.8 文件末尾 (eof) 标记	121	9.8.1 Linux 的文件打印机制	158
小 结	122	9.8.2 文件的打印	159
习 题	122	9.8.3 显示打印请求的状态	159
第 8 章 文件安全	124	9.8.4 打印任务的取消	160
8.1 引言	124	小 结	160
8.2 基于密码的保护	124	习 题	161
8.3 基于文件加密的保护	125	第 10 章 高级文件处理	163
8.4 基于访问权限的文件保护	125	10.1 引言	163
8.4.1 用户分类	125	10.2 正则表达式	163
8.4.2 文件操作/访问权限的 分类	126	10.3 文件压缩	165
8.5 读取和更改文件的访问特权	127	10.3.1 gzip 命令	165
8.5.1 读取文件的访问特权	127	10.3.2 gunzip 命令	166
8.5.2 改变文件访问特权	128	10.3.3 gzexe 命令	167
8.5.3 默认的文件访问特权	131	10.3.4 zcat 命令	167
8.6 特殊访问位	132	10.3.5 重命名文件	168
8.6.1 set-user-ID (SUID) 位	132	10.4 文件排序	168
8.6.2 set-group-ID (SGID) 位	134	10.5 搜索命令和文件	171
8.6.3 sticky 位	134	10.5.1 find 命令	171
8.7 文件访问权限和类型	135	10.5.2 whereis 命令	172
小 结	136	10.5.3 which 命令	172
习 题	136	10.6 搜索文件内容	173
第 9 章 基本文件处理	138	10.7 cutting 和 pasting	176
		10.8 编码和解码	179
		10.9 历史命令列表	180

小结	185	习题	222
习题	186	第 13 章 进程	224
第 11 章 文件共享	188	13.1 引言	224
11.1 引言	188	13.2 同一时刻执行的进程	224
11.2 通过副本共享	188	13.3 Linux 进程的状态	225
11.3 通过同一用户名登录共享	188	13.4 shell 命令的执行	226
11.4 为共享文件设立适当的访问 权限	189	13.5 进程属性	229
11.5 为团队成员建立一个用户组	189	13.6 进程和作业控制	236
11.6 通过文件链接共享	189	13.6.1 前后台进程以及相关 命令	236
11.6.1 硬链接	189	13.6.2 Linux 守护进程	241
11.6.2 硬链接的缺点	195	13.6.3 命令的顺序和并行执行	241
11.6.3 软链接/符号链接	196	13.6.4 命令和进程的异常终止	244
11.6.4 符号链接的类型	199	13.6.5 命令的有条件执行	247
11.6.5 符号链接的优缺点	199	13.7 Linux 进程的层次关系	248
11.7 查找符号链接	200	小结	251
小结	201	习题	252
习题	202	第 14 章 连网与网络互联	253
第 12 章 重定向和管道	203	14.1 引言	253
12.1 引言	203	14.2 计算机网络和计算机互联网	253
12.2 标准文件	203	14.3 计算机网络和网络互联的动机	254
12.3 输入重定向	204	14.4 网络模型	255
12.4 输出重定向	205	14.5 TCP/IP 协议	256
12.5 输出重定向和输入重定向的 合并	206	14.5.1 TCP 和 UDP 协议	256
12.6 带文件描述符的 I/O 重定向 操作	207	14.5.2 数据的路由——IP 协议 (The Internet Protocol)	257
12.7 标准出错的重定向	207	14.5.3 点分 10 进制的 IPv4 地址	258
12.8 用一条命令实现标准输出和 标准出错的重定向	208	14.5.4 符号名	259
12.9 用一条命令重定向标准输入、 标准输出和标准出错	211	14.5.5 将名字翻译到 IP 地址 ——域名系统	261
12.10 不覆盖文件内容的重定向 (追加)	212	14.6 互联网服务和协议	262
12.11 Linux 管道	213	14.7 client-server 软件模型	262
12.12 结合管道使用重定向	216	14.8 应用软件	264
12.13 在 TC shell 中的出错重定向	218	14.8.1 显示主机名	264
12.14 I/O 及错误重定向总结	220	14.8.2 显示当前网络中所有 登录用户的用户列表	264
小结	221	14.8.3 显示网络中所有主机 状态	266

14.8.4 测试网络连接	266	16.4 here 文件	331
14.8.5 显示用户信息	267	16.5 中断 (信号) 处理	333
14.8.6 远程登录	270	16.6 exec 命令和文件输入/输出	336
14.8.7 远程命令执行	275	16.6.1 不创建新的进程执行 命令	336
14.8.8 文件传输	276	16.6.2 通过 exec 命令的文件输 入/输出	337
14.8.9 远程复制	279	16.7 Bash 中的函数	343
14.8.10 交互式聊天	280	16.7.1 需要函数的原因	343
14.8.11 追踪从一个站点到另 一个站点的路由	281	16.7.2 函数定义	343
小结	282	16.7.3 函数调用	343
习题	282	16.7.4 函数的例子	344
第 15 章 Bash 编程	284	16.8 调试 shell 程序	345
15.1 引言	284	小结	347
15.2 运行一个 Bash 脚本	284	习题	347
15.3 shell 变量和相关的命令	285	第 17 章 TC shell 编程介绍	348
15.3.1 控制 shell 提示符	289	17.1 引言	348
15.3.2 变量声明	290	17.2 运行一个 TC shell 脚本	348
15.3.3 读写 shell 变量	292	17.3 shell 变量和相关命令	349
15.3.4 命令替换	294	17.3.1 读、写 shell 变量	352
15.3.5 输出环境变量	295	17.3.2 命令替换	353
15.3.6 重设变量	296	17.3.3 输出环境	354
15.3.7 创建只读的用户定义的 变量	297	17.3.4 重置变量	355
15.3.8 从标准输入设备读入	298	17.3.5 创建用户自定义只读 变量	356
15.4 给 shell 脚本传递参数	299	17.3.6 从标准输入中读	357
15.5 注释和程序头	302	17.4 向 shell 脚本传递参数	358
15.6 程序流程控制命令	303	17.5 注释与程序头	360
15.6.1 if-then-elif-else-fi 语句	303	17.6 流程控制命令	361
15.6.2 for 语句	310	17.6.1 if-then-else-endif 语句	361
15.6.3 while 语句	311	17.6.2 foreach 语句	367
15.6.4 until 语句	313	17.6.3 while 语句	369
15.6.5 break 和 continue 命令	315	17.6.4 break、continue 和 goto 语句	370
15.6.6 case 语句	315	17.6.5 switch 语句	371
小结	318	小结	374
习题	319	习题	374
第 16 章 高级 Bash 编程	321	第 18 章 高级 TC shell 编程	376
16.1 引言	321	18.1 引言	376
16.2 数值数据处理	321		
16.3 数组处理	327		

18.2 数值型数据的处理	376	21.1 引言	459
18.3 数组操作	377	21.2 XFree86 基础	460
18.4 here document	382	21.2.1 比较与优势	460
18.5 中断 (信号) 处理	383	21.2.2 交互的关键组件——事 件和请求	461
18.6 调试 shell 程序	386	21.2.3 窗口管理器在用户界面 中的角色——FVWM	462
小结	387	21.2.4 定制 XFree86 和 FVWM	466
习题	388	21.3 GNOME 桌面管理器	471
第 19 章 文件系统备份	389	21.3.1 GNOME 桌面的登录、 注销及外观显示	471
19.1 引言	389	21.3.2 GNOME 面板	472
19.2 用 tar 命令存档和恢复文件	389	21.3.3 自定义面板	473
19.2.1 档案文件	390	21.3.4 使用 GnoRPM 安装新 软件包	476
19.2.2 恢复档案文件	393	21.3.5 Sawfish 窗口管理器	479
19.2.3 复制目录层	396	21.3.6 GNOME Control Center (GNOME 控制中心)	482
19.3 以 tar 格式发行软件	397	21.3.7 使用 GNOME 文件管 理器进行文件管理	483
小结	398	21.4 KDE 桌面管理器	484
习题	398	21.4.1 登录和注销	484
第 20 章 Linux 软件开发工具	399	21.4.2 KDE 面板	485
20.1 引言	399	21.4.3 新软件包的安装	488
20.2 计算机编程语言	399	21.4.4 K 窗口管理器	490
20.3 编译过程	401	21.4.5 KDE 控制中心	492
20.4 软件工程的生命周期	401	21.4.6 使用 KFM 进行文件 管理	493
20.5 程序生成工具	402	21.5 Internet 资源	495
20.5.1 生成 C 语言源代码	402	小结	495
20.5.2 缩进 C 语言代码	402	习题	496
20.5.3 C 程序的编译	405	附录 A Linux 的安装	498
20.5.4 处理模块化的 C 程序	407	附录 B 命令字典	507
20.5.5 创建、修改和释放库/ 归档文件	415	术语表	567
20.5.6 版本控制	420		
20.6 静态分析工具	442		
剖析 C 源程序	442		
20.7 动态分析工具	446		
20.7.1 代码调试	446		
20.7.2 运行效率分析	455		
小结	456		
习题	457		
第 21 章 XFree86 和 Linux 图形用户 界面	459		

第 1 章 操作系统综述



目标

- 解释什么是操作系统；
- 简要描述操作系统提供的服务；
- 描述字符用户界面和图形用户界面；
- 讨论各种类型的操作系统；
- 提及什么是 Linux。

1.1 引言

现今用到许多种操作系统，其中有一些可在任何计算机上（从个人计算机，或者 PC 到大型机）通用，还有一些是为某类计算机所特别设计的，例如用来控制机器人运动的实时计算机系统。本章将讨论操作系统的用途和操作系统的不同种类。不过在介绍操作系统的不同种类和 Linux 在分类中的定位之前，先看看现代计算机系统（contemporary computer system）的层次结构图，并讨论一下操作系统的基本目的。然后将介绍操作系统的不同类型及作为划分依据的参数。最后会指出 Linux 所属的类型，并简单介绍 Linux 家族的成员。

1.2 操作系统是什么

如图 1.1 所示，一个计算机系统包括多种硬件和软件资源。操作系统的主要目的就是简单、高效、公平、有序和安全地使用这些资源。它允许计算机用户使用应用软件——电子制表软件、字处理器、网页浏览器和电子邮件软件等。程序员利用编程语言函数库、系统调用和程序生成工具（例如文本编辑器、编译器和版本控制系统）来开发软件。对单用户来说，不存在公平性问题，但是如果允许多个用户同时使用一个计算机系统，公平性和安全性就成为操作系统设计者必须要考虑的两个主要问题。

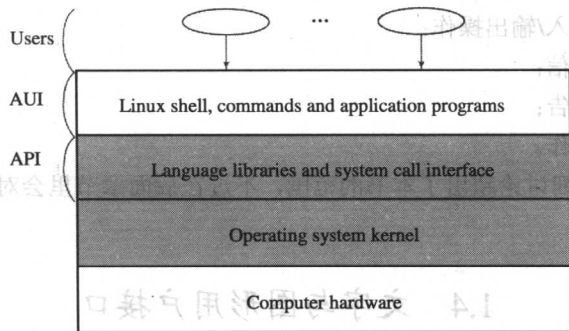


图 1.1 现代计算机系统层次结构图

硬件资源包括键盘、显示器、内存（通常是随机存取存储器 RAM）、磁盘、调制解调器和中央处理单元（CPU）。软件资源包括应用程序，例如字处理器、电子制表软件、游戏、图形和图像处理工具，以及网页浏览器之类的网相关工具。所有这些应用程序属于层次图的最上层，它们构成应用程序用户界面（AUI）。AUI 通过语言函数库和系统调用界面与操作系统内核联系。实际工作是在内核中完成的。系统调用界面由一组函数组成，这些函数可以被应用程序和函数库里的程序调用，用来启动一段提供特定服务（例如读文件）的内核代码。语言函数库和系统调用界面两者就构成了应用程序界面（API）。Linux 操作系统的层次结构将在第 3 章中作简单介绍。

有两种方式来观察操作系统的结构：自上而下和自下而上。自下而上看，操作系统可以被看成一个以高效、公平、有序和安全的方式分配和回收系统资源（硬件和软件）的软件。例如，操作系统要在装载和执行一个程序前决定分配多少内存空间给那个程序。操作系统必须保证一个时间内打印机上只有一个文件在打印。操作系统还得防止磁盘上的文件意外地被其他文件覆盖。另外，操作系统必须保证释放已经执行完的程序所占用的 CPU，以使其他程序能够执行。因此，操作系统可以被看成一个资源管理器。

本书采用的是自上而下的方式。在这种方式下，操作系统可以被看成一个作为用户的你和复杂的硬件资源隔离开来的软件。因此不需要为怎样直接和那些硬件资源打交道（大多数用户不可能做到）而烦恼。例如，只要单击鼠标就可以保存文件，指定打印机将文件打印出来，而不必费时费力地写软件。同样，不必为新购的设备（如鼠标、磁盘驱动器或者 DVD）编写程序，安装到系统上。操作系统承担了所有与硬件资源有关的任务，提供一种操作简便的机器给用户。这种机器使得用户能用几条简单命令就可以在磁盘上存储文件，在打印机上打印文件，在 DVD 影碟机上播放电影。从某种意义上来说，操作系统提供了一个比实际机器简单得多的虚拟机给用户。

举个例子来说，你可以输入 `cp memo letter` 命令来将文件 `memo` 复制为文件 `letter`，而无须考虑文件 `memo` 和 `letter` 在磁盘上的位置、磁盘的结构和大小、磁盘驱动器的品牌，以及磁盘驱动器的数量和名称等，即使系统拥有多个驱动器（软盘、CD-ROM、一个或多个硬盘）。

1.3 操作系统提供的服务

操作系统能提供多种服务（service）。其中大多数服务是为了安全有效地运行应用程序和程序开发工具。另外一些服务用于内务管理（housekeeping task），例如记录系统使用的时间。因此操作系统提供的主要服务就是提供一种机制，保证下列任务安全高效的执行：

- 程序运行；
- 程序执行的输入/输出操作；
- 进程之间的通信；
- 错误检测和报告；
- 各种文件的操作。

有关这些服务的详细讨论超出了本书的范围，不过在后面章节里会对与相应章节内容有关的服务进行简单介绍。

1.4 文字与图形用户接口

用户通过输入命令使操作系统使用计算机系统。计算机通过键盘等输入设备用于接收用户命

令。如果命令只是通过键盘传递给操作系统，那么这一操作系统采用的是字符用户界面（character user interface），即通常所说的命令行用户界面（command line user interface, CUI）。如果操作系统的最主要输入设备是鼠标等点击设备（point-and-click device），那么它采用的是图形用户界面（GUI）。有些操作系统既有字符用户界面又有图形用户界面，可以使用任意一种。有些操作系统采用 CUI 作为主要界面，但是允许用户运行提供 GUI 的软件。DOS、Linux 和 UNIX 等操作系统采用字符用户界面，而 MacOS、OS/2 和微软 Windows 等系统采用的是图形用户界面。

尽管 GUI 让计算机更容易操作，但是它所提供的自动化设置的灵活性是很低的。GUI 还在运行任务和用户之间增加一个额外的软件层，从而降低运行速度。相反，CUI 让用户享有完全的系统控制权，用户可以选择自己喜欢的方式来运行应用程序。另外，在 CUI 下运行任务和用户之间的软件层是最小的，从而加快了任务完成速度，提高了系统效率。由于大多数人已经习惯了任天堂的图形界面和网页浏览器的图形界面，字符界面给人们带来的是一种全新的与计算机系统交流的方式，同时也是一种具有挑战性的方式。尽管如此，在经过几次实习培训课之后，大多数用户都能应对这个挑战。Linux 的基本界面是字符界面，不过它可以运行基于 X Window 系统（MIT 的 Athena 项目）的软件，X Window 提供 GUI 界面。

图形用户界面是 Linux 系统的一大特色。虽然所有 Linux 命令都是在字符界面下执行，但是许多因特网工具和软件开发工具都有图形界面。因此大多数 Linux 系统都会附带一个基于 X 的桌面环境的图形界面软件包。现在使用最广泛的桌面环境是 GNU 网络对象模型环境（GNU Network Object Model Environment, GNOME）和 K 桌面环境（K Desktop Environment, KDE）。GNOME 是 Red Hat Linux 6.1 以及更高级版本的系统默认桌面环境。最近成立了一个新的联盟来推广 KDE，在第 21 章将讨论 Linux 的 GUI。

1.5 操作系统的类型

根据能同时使用系统的用户数和系统能同时运行的进程（执行中的程序）数的不同，可以把操作系统分成 3 类。

1. 单用户、单进程系统

这种操作系统在一个时间内只允许一个用户使用计算机系统，并且用户每次只能运行一个进程。这种操作系统被广泛用于 PC 机上。其代表者有 MacDOS、DOS 和 Windows 3.1。

2. 单用户、多进程系统

就如名字所描述的，这种操作系统在一个时间内只允许一个用户使用，但是能同时运行多个进程。这种操作系统也被应用于 PC 机上。其代表者有 OS/2 和 Windows NT 工作站。

3. 多用户、多进程系统

这种操作系统允许多个用户同时使用计算机系统，并且每个用户能同时运行多个进程。它被广泛应用于像大学和大公司这种需要支持多用户的组织机构的计算机上。Linux、UNIX 和 Windows NT Server 都属于此类。

多用户、多进程系统通过共享（分时）诸如 CPU 之类的昂贵资源来提高计算机系统资源利用率，从而增加系统吞吐量（单位时间完成的进程数）。资源利用率之所以得以提高，是由于在多进程系统里，当一个进程在执行输入/输出操作时（例如读入键盘输入，捕获鼠标单击或者写硬盘文件），CPU 可以从这个进程转移分配给另外一个进程——相当于这两个进程同时运行。这种把当前执行 I/O 操作的进程所占用的 CPU 分配给其他进程的机制叫做多道程序设计。多道程序

设计是当前所有多用户、多进程操作系统的关键。在单进程系统中，如果当前占有 CPU 的进程执行 I/O 操作，CPU 将会空闲，因为系统中不存在第二个进程同时使用 CPU。

允许用户与执行程序交互的操作系统称作交互式操作系统，不允许交互的称作批处理操作系统。在程序运行过程中不需要用户参与的情况下，例如运行工资单处理程序，批处理系统是很有用的。VMS 操作系统拥有交互式 and 批处理两种界面。几乎当前所有著名的操作系统 (Linux、UNIX、DOS、Windows 等) 都是交互式的。Linux 和 UNIX 也允许程序在后台以批处理方式运行 (参见 13 章的 Linux 下的后台程序运行)。多用户、多进程、交互的操作系统被称为分时操作系统 (time-sharing operating system)。在分时操作系统中，CPU 快速地在不同的进程之间切换。这种操作方式让系统中所有进程得以同时运行，并且使每个用户都有独自使用系统的感觉。分时系统的代表有 Linux、UNIX 和 Windows NT Server。

1.6 Linux 家族

1991 年 10 月芬兰赫尔辛基大学 21 岁的学生 Linus Torvalds 在 comp.os.minix 新闻组上发布了如下消息：

I'm doing a (free) operating system (just a hobby, won't be big and professional like GNU) for 386 (486) AT clones.

我正在设计一个 (免费的) 支持诸如 386 (486) AT 计算机的操作系统 (只是业余爱好而已，不会太大也没有 GNU 那么专业)。

Torvalds 的“业余兴趣”最终变成了现在的 Linux 操作系统。虽然 Linux 只是一种发展还不到 10 年的类 UNIX 操作系统，但它却引起了 PC 和因特网世界的革命。世界上数百万计算机用户在他们的家用 PC 和办公室工作站上使用 Linux。Linux 最初是为使用 Intel CPU 芯片的 PC 机设计和开发的，不过现在 Linux 可以在多种其他平台上运行，包括了 Alpha、Amiga、Atari、Macintosh 和 Sun SPARC。

数年之前，Linux 这个名字只是指代一种操作系统，不过现在它已经成为由最初的系统衍生出来的一个操作系统家族的代名词。这个家族的系统提供不同的用户界面和应用程序组。其成员包括 Caldera、Corel、Debian、Mandrake、Red Hat、Slackware 和 SuSE。可以从因特网上免费下载到大多数 Linux 系统，或者花费不到 30 美元去当地计算机用品商店购买一套。第 2 章将简略介绍 Linux 系统的发展历史。

小 结

操作系统是运行在计算机系统硬件上的，用于管理系统软硬件资源的一种软件。它还给计算机系统用户提供一台更易于交互的简单虚拟机。操作系统所提供的基本服务是保证程序执行的安全性和高效性。这些服务包括程序执行、I/O 操作、进程间通信、错误检测和报告，以及文件操作。

操作系统根据能同时使用系统的用户数和能同时在系统上运行的进程数，可以分成以下 3 大类：单用户、单进程系统，单用户、多进程系统和多用户、多进程系统。此外，允许用户与其执行的程序交互的操作系统被称作交互式操作系统，不允许交互的称作批处理操作系统。多用户、多进程、交互的操作系统被称为分时操作系统，Linux 是其主要代表。多用户、多进程系统通过把系统资源在同时运行的进程之间进行切换以达到提高资源利用率 (resource utilization) 的目的，