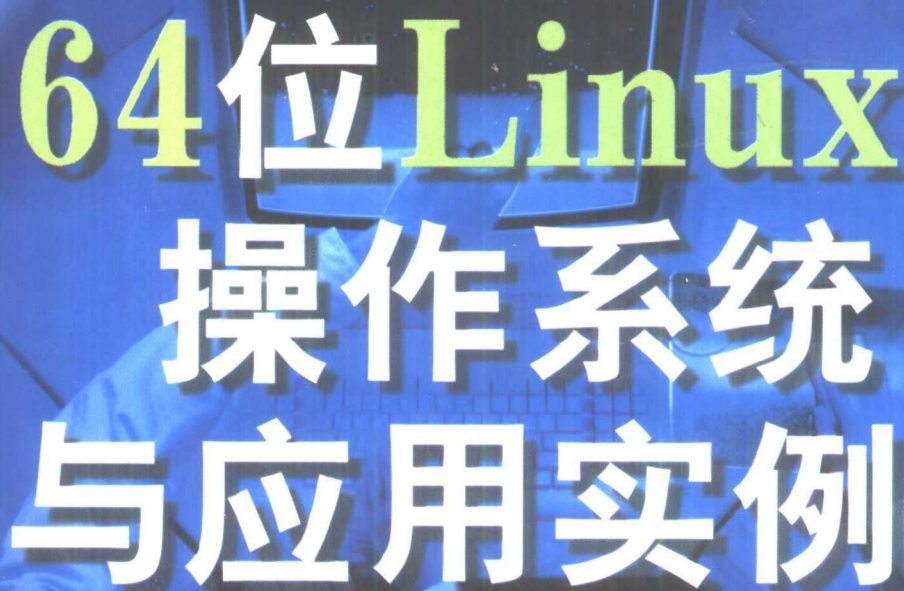


The logo for Donovan, featuring a stylized 'D' in a green circle followed by the name 'Donovan' in white text on a green background.

Donovan

The main title of the book, '64位Linux操作系统与应用实例', is displayed in large, bold characters. '64位Linux' is in green, while '操作系统与应用实例' is in white. The background is a blue-tinted image of two people working at a computer, with a globe visible behind them.

# 64位Linux 操作系统 与应用实例

赵敏哲 主编



机械工业出版社  
China Machine Press

# 64 位 Linux 操作系统与应用实例

主编 赵敏哲

参编 赵 霞 于重重 史兴国

机械工业出版社

本书由浅入深，系统讲述了现代操作系统原理的主要内容，包括进程、线程、存储器管理、文件管理、设备管理、网络系统等。与原理紧密结合，讲述 Linux 操作系统的基础知识、实现原理分析和编程方法，并设计了基于 Donovan 公司的 Penguin64Linux 操作系统平台的实例和实验，便于读者理论联系实际，掌握 Linux 的使用和基本编程技术，为从事 Linux 应用开发打下理论基础和技术基础。

本书是在参考了大量的相关书籍、论文和资料的基础上编写而成的，力求论述全面系统，内容丰富。本书既可作为全国高等学校操作系统专业系列教材，也可作为计算机工程技术人员的参考书，以及广大对 Linux 感兴趣读者的参考书。

## 图书在版编目 (CIP) 数据

64 位 Linux 操作系统与应用实例/赵敏哲主编. —北京: 机械工业出版社, 2000.12

ISBN 7-111-08449-7

I .6… II .赵… III .Linux 操作系统 IV .TP316.89

中国版本图书馆 CIP 数据核字 (2000) 第 53564 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 贡克勤 版式设计: 霍永明 责任校对: 申春香

封面设计: 李雨桥 责任印制: 路琳

北京机工印刷厂印刷·新华书店北京发行所发行

2001 年 1 月第 1 版·第 1 次印刷

787mm×1092mm<sup>1</sup>/<sub>16</sub>·23.25 印张·573 千字

0 001 - 4000 册

定价: 35.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、68326677 - 2527

## 前 言

本书讲述了现代操作系统原理的主要内容，包括进程、线程、存储器管理、文件管理、设备管理、网络系统等。

本书面向所有想学习操作系统原理或 Linux 操作系统的读者，使学习者不仅能够深入理解操作系统的精髓原理，理解 Linux 操作系统的实现原理和技术，而且在本书大量实例和实验的引导下，学习使用 Linux 以及在 Linux 环境下实用的编程技术。本书的目的是培养既懂理论，又有实际动手能力的高素质技术人才。

本书既可作为计算机专业本、专科学生的专业基础课教材，也可作为计算机工程师技术培训教材，还可作为计算机工程技术人员的参考资料。

本书的作者从事计算机及应用的科研工作和操作系统教学工作多年，并从事操作系统改研究和探索的实践。在书中既体现了一线教师在教学中积累的操作系统的理论知识和实践教学经验，也体现了 Linux 最新的技术成果，使本书不同于纯理论教科书，也不同于现有的缺乏理论体系的技术参考书，非常适用于学习操作系统原理或 Linux 操作系统核心技术及编程技术的读者，具有广泛的读者群。

本书可作为“Linux 操作系统”课的教材，也可作为技术参考书，它克服了目前操作系统教材重理论而缺少实例的不足，在讲述理论的基础上，分析 Linux 系统的实现技术，并给出了基于 Linux 操作系统的编程实例。使操作系统的学习做到理论联系实际，使读者全面掌握操作系统的原理和实际的 Linux 操作系统的使用与编程技术。

本书给出了实用的操作系统实验指导。学习操作系统的另一个难点是没有实验环境和与课程内容配套的实验内容。本书将 Linux 作为操作系统实验平台，并设计了实用的实验，在一定程度上解决了国内高校多年以来操作系统教学缺少实验的难题，相信一定会对广大工科院校的操作系统教学改革提供参考。本书中大部分实例和实验都可在 Intel 平台的 Linux 上编译运行。

本书中不仅讲述了操作系统的原理，还将现代操作系统的新知识点和最新的计算机硬件技术融入传统操作系统体系中，并以 SUN 的 UltraSPARC (64bit) 硬件及 Donovan 公司的 64bitLinux 操作系统为平台，引入了大量实例，使本书具有知识的完整性、实践性和先进性。

本书由赵敏哲博士主编，参加编写的有北京工商大学赵霞（第 1、2、3、8 章）、于重重（第 5、6、8 章）和中科院软件所史兴国（第 4、7、8 章）。

第 1 章概论讲述了现代操作系统的基本概念、分类、功能及体系结构。在此基础上，介绍了本书的主题——Linux 操作系统的起源及发展，并以世界上第一个 64 位中文 Linux 操作系统——Penguin64 为实例结束该章。第 2 章介绍 Linux 的基本概念及如何使用 Linux，从系统启动和初始化到使用 Shell 及 C 语言编程，力图为后续章节的学习及实际操作打下坚实的基础。第 3 章是关于 Linux 的进程调度及进程间的通信，进程部分是 Linux 操作系统的最重要的组成部分，为帮助读者学习掌握该章内容，书中配备了大量的实例，特别是在 fork、

signal 以及 thread 编程方面。第 4 章介绍了 Linux 系统的存储管理，由介绍 Linux 内存管理的体系结构入手，通过虚拟内存管理和分页机制及相应的内存分配策略，使读者对内存管理建立起完整的印象。第 5 章是 Linux 文件系统，介绍用户可见的文件系统接口以及 Linux ext2 及 VFS 文件系统的特点及实现方式，为帮助读者更好地了解 Linux 文件系统的结构及管理，该章列举并分析了大量的用于文件系统的数据结构，并用大量的实例让读者熟悉 Linux 文件系统的函数调用。第 6 章讨论 Linux 的设备管理，其中包括设备驱动程序框架，设备与 I/O 系统之间的交互等。设备管理是任何操作系统最困难的部分，而且往往因硬件设备的不同而出现诸多差异。为帮助读者理解和掌握 Linux 的设备管理部分，该章配备了很多基于 Sun UltraSPARC 硬件平台的实例，第 7 章介绍了 Linux 网络系统，该章介绍了 Linux 网络的基本技术及应用，作为 Linux 操作系统的逻辑延伸及补充。有兴趣的读者可以参阅有关 Linux 网络技术的专门书籍。

本书的形成是集体智慧的结晶。特别感谢新加坡 Donovan 公司的研发中心及北京杜文 Linux 实验室，他们提供了书中大部分的实例及实验程序。另外，要特别感谢北京工商大学的韩力群副教授，她为本书的编写提供了大力帮助和协调。还要感谢参与本书出版的编辑、录排人员和美工设计人员，是他们的辛勤工作才使得本书如期付梓出版。

因出版时间紧迫，书中错误在所难免，敬请读者谅解。请将您的意见及建议发送给我们，赵敏哲的 e-mail 地址是 mzhao@donovan.com，赵霞的 e-mail 地址是 zhaoxiaxia@263.net，于重重的 e-mail 地址是 chongzhy@263.net 史兴国的 e-mail 地址是 shi@intec.iscas.ac.cn。

赵敏哲  
2000 年 10 月

# 目 录

前 言

第 1 章 概论 ..... 1

1.1 现代操作系统 ..... 1

1.1.1 操作系统的概念 ..... 1

1.1.2 操作系统的分类 ..... 2

1.1.2.1 批处理操作系统 ..... 2

1.1.2.2 分时操作系统 ..... 2

1.1.2.3 实时操作系统 ..... 3

1.1.2.4 网络操作系统和分布式操作  
系统 ..... 3

1.1.2.5 通用操作系统 ..... 3

1.1.2.6 个人操作系统 ..... 3

1.1.3 操作系统的功能 ..... 4

1.1.3.1 处理机管理 ..... 4

1.1.3.2 存储管理 ..... 4

1.1.3.3 文件管理 ..... 4

1.1.3.4 设备管理 ..... 4

1.1.3.5 用户接口 ..... 5

1.1.3.6 网络管理 ..... 5

1.1.4 操作系统的结构 ..... 5

1.1.4.1 整体式模型 ..... 5

1.1.4.2 层次式模型 ..... 5

1.1.4.3 客户/服务器模型 ..... 5

1.2 Linux 系统概论 ..... 6

1.2.1 Linux 发展史 ..... 6

1.2.1.1 Linux 的起源 ..... 6

1.2.1.2 Linux 的版本 ..... 7

1.2.2 Linux 的特点 ..... 7

1.2.3 自由软件基金会及 GNU 计划 ..... 8

1.3 Linux 的发展趋势 ..... 8

1.3.1 嵌入式 Linux ..... 9

1.3.2 桌面 Linux: 32 位 Linux ..... 9

1.3.2.1 Red Hat Linux ..... 9

1.3.2.2 XteamLinux ..... 10

1.3.3 服务器 Linux: 64 位 Linux ..... 10

1.3.3.1 64 位计算技术 ..... 10

1.3.3.2 64 位 Linux: Penguin64 中文

Linux ..... 11

1.4 小结 ..... 11

第 2 章 Linux 基础 ..... 13

2.1 Linux 系统结构及文件组织 ..... 13

2.1.1 Linux 系统结构 ..... 13

2.1.2 Linux 源文件组织 ..... 13

2.2 系统启动和初始化 ..... 14

2.2.1 内核引导 ..... 14

2.2.1.1 init 和 inittab ..... 14

2.2.1.2 系统的运行级别 ..... 16

2.2.2 关闭系统 ..... 17

2.3 文件系统操作 ..... 17

2.3.1 文件系统类型 ..... 17

2.3.2 加载文件系统 ..... 18

2.3.3 卸载文件系统 ..... 18

2.3.4 创建文件系统 ..... 19

2.4 常用命令 ..... 19

2.4.1 用户帐号管理 ..... 19

2.4.1.1 创建用户帐号 ..... 19

2.4.1.2 修改用户帐号 ..... 20

2.4.1.3 删除和查封帐号 ..... 20

2.4.1.4 几个和用户相关的命令 ..... 20

2.4.2 基本目录和文件操作 ..... 20

2.4.3 用 vi 编辑文件 ..... 21

2.4.4 进程控制 ..... 22

2.4.5 网络命令及其他 ..... 22

2.5 Shell 编程 ..... 22

2.5.1 创建和执行 Shell 脚本文件 ..... 23

2.5.1.1 执行 Shell 脚本文件 ..... 23

2.5.1.2 创建脚本文件 ..... 23

2.5.2 变量及符号 ..... 23

2.5.2.1 变量 ..... 23

2.5.2.2 符号 ..... 24

2.5.3 表达式 ..... 24

2.5.4 控制语句 ..... 25

2.5.5 实例 ..... 26

2.6 GNU C 编程 ..... 27

2.6.1 使用 gcc 编译和链接 .....	27	3.3.2 Linux 的进程调度 .....	58
2.6.2 使用 make 和 makefile 实现自动 编译 .....	28	3.3.2.1 调度的时机 .....	58
2.6.2.1 简单的 makefile 文件 .....	28	3.3.2.2 调度策略 .....	59
2.6.2.2 使用 make 命令 .....	29	3.3.2.3 调度程序源码分析 .....	60
2.6.3 gdb 调试工具 .....	29	3.4 进程通信 .....	66
2.6.3.1 gdb 简介 .....	29	3.4.1 进程的相互制约 .....	66
2.6.3.2 gdb 的命令 .....	29	3.4.1.1 进程的互斥 .....	67
2.6.4 其他调试工具 .....	32	3.4.1.2 进程的同步 .....	68
2.6.4.1 Calls .....	32	3.4.1.3 Linux 的进程通信机制 .....	69
2.6.4.2 cproto .....	32	3.4.2 信号 Signals .....	69
2.6.4.3 indent .....	32	3.4.2.1 信号机制 .....	69
2.7 小结 .....	32	3.4.2.2 信号的处理函数 .....	71
<b>第 3 章 Linux 进程管理</b> .....	<b>33</b>	3.4.2.3 实例 .....	73
3.1 现代操作系统的进程模型 .....	33	3.4.2.4 信号集及其处理 .....	74
3.1.1 程序的并发执行 .....	33	3.4.3 管道 .....	77
3.1.2 进程模型 .....	34	3.4.3.1 管道 (Pipe) 实现机制 .....	78
3.1.2.1 进程的创建及组织 .....	34	3.4.3.2 管道 (Pipe) 实例 .....	78
3.1.2.2 进程的状态 .....	34	3.4.3.3 创建管道 (Pipe) 的简单 方法 .....	82
3.1.2.3 操作系统的进程模型 .....	35	3.4.3.4 命名管道 (FIFO) .....	85
3.1.3 进程的描述 .....	35	3.4.4 System V 的进程通信机制 .....	90
3.1.3.1 进程控制块 .....	36	3.4.4.1 IPC 对象 .....	90
3.1.3.2 处理机上下文和进程 上下文 .....	36	3.4.4.2 IPC 命令 .....	91
3.1.4 Linux 的进程 .....	37	3.4.5 信号量 Semaphore .....	92
3.1.4.1 task_struct 数据结构 .....	37	3.4.5.1 信号量机制 .....	92
3.1.4.2 Linux 进程状态 .....	40	3.4.5.2 系统调用 .....	93
3.2 Linux 的进程控制 .....	41	3.4.5.3 编程实例 .....	96
3.2.1 进程的创建 fork .....	41	3.4.6 消息队列 Message Queues .....	100
3.2.1.1 fork 系统调用 .....	41	3.4.6.1 消息队列机制 .....	100
3.2.1.2 vfork 调用 .....	43	3.4.6.2 系统调用 .....	101
3.2.1.3 实例 .....	44	3.4.6.3 编程实例 .....	103
3.2.2 进程的执行 exec .....	46	3.4.7 共享内存 Shared Memory .....	113
3.2.3 进程的等待 wait .....	49	3.4.7.1 共享内存机制 .....	113
3.2.3.1 wait 函数 .....	49	3.4.7.2 编程实例 .....	115
3.2.3.2 waitpid 函数 .....	49	3.5 Linux 的线程 .....	123
3.2.3.3 实例 .....	50	3.5.1 线程的概念 .....	124
3.2.4 进程的结束 exit .....	54	3.5.1.1 与进程的关系 .....	124
3.3 进程调度 .....	55	3.5.1.2 编程初步 .....	125
3.3.1 进程调度的原理 .....	55	3.5.2 线程的互斥与同步 .....	127
3.3.1.1 调度的时机 .....	56	3.5.2.1 用信号量实现同步 .....	128
3.3.1.2 调度的处理过程 .....	56	3.5.2.2 用互斥量实现互斥 .....	130
3.3.1.3 调度的算法 .....	57	3.5.3 多线程编程 .....	132
		3.6 小结 .....	133

<b>第4章 Linux 存储管理</b> .....	134	5.2.1 文件的用户接口 .....	166
4.1 存储管理概述 .....	134	5.2.1.1 文件的类型 .....	166
4.1.1 存储管理的基本概念 .....	135	5.2.1.2 文件描述符 .....	167
4.1.2 物理内存和虚拟内存 .....	136	5.2.2 Linux 文件系统的框架 .....	168
4.1.3 分页与分段式内存管理 .....	138	5.3 ext2 文件系统 .....	168
4.2 Linux 内存管理体系结构 .....	142	5.3.1 ext2 文件系统的优点 .....	168
4.2.1 基于硬件的基本抽象概念 .....	142	5.3.2 ext2 文件的逻辑结构与物	理结构 .....
4.2.1.1 物理内存设备 .....	142	5.3.3 ext2 文件系统存储空间的	管理 .....
4.2.1.2 地址空间 .....	143	5.3.3.1 ext2 文件系统的物理存	储空间 .....
4.2.1.3 地址映射和地址	转换硬件 .....	5.3.3.2 ext2 文件系统的超级块 .....	171
4.2.2 虚拟内存管理和分页机制 .....	145	5.3.3.3 ext2 文件系统的组描	述符 .....
4.2.2.1 Linux 的虚拟存储管理 .....	145	5.3.3.4 ext2 文件系统空闲块的	分配 .....
4.2.2.2 页目录和页表 .....	147	5.3.4 ext2 文件系统目录文件的	实现 .....
4.2.2.3 页的分配和回收 .....	148	5.3.4.1 ext2 目录文件的结构 .....	176
4.2.2.4 内存映射 .....	149	5.3.4.2 文件的查询 .....	177
4.2.2.5 缺页中断 .....	149	5.3.4.3 文件的共享 .....	177
4.2.2.6 交换机制 .....	150	5.4 VFS .....	179
4.2.2.7 交换设备 .....	153	5.4.1 VFS 的超级块和索引节点 .....	180
4.2.3 系统中的缓存 .....	154	5.4.1.1 VFS 的超级块 .....	181
4.3 内存的分配策略 .....	154	5.4.1.2 VFS 的 inode .....	182
4.3.1 分配和回收 .....	154	5.4.2 对 VFS 的操作 .....	184
4.3.1.1 分配策略 .....	154	5.4.2.1 打开文件 .....	184
4.3.1.2 分配与释放操作 .....	155	5.4.2.2 注册安装文件系统 .....	185
4.3.1.3 相关函数 .....	156	5.4.3 Linux 文件系统的缓冲机制 .....	186
4.3.2 进程与内存 .....	157	5.4.3.1 VFS 索引节点缓存 .....	186
4.3.2.1 进程的创建和执行 .....	157	5.4.3.2 VFS 的目录缓存 .....	187
4.3.2.2 相关函数 .....	159	5.4.3.3 缓冲区缓存(Buffer Cache) .....	187
4.4 小结 .....	159	5.5 proc 文件系统 .....	189
<b>第5章 Linux 文件系统</b> .....	160	5.6 Linux 文件系统调用及应用实例 .....	189
5.1 文件管理概述 .....	160	5.6.1 文件的打开和关闭 .....	189
5.1.1 文件与文件系统 .....	160	5.6.1.1 open 系统调用 .....	189
5.1.1.1 文件的定义 .....	160	5.6.1.2 close 系统调用 .....	191
5.1.1.2 文件的操作 .....	160	5.6.1.3 应用实例 .....	191
5.1.1.3 文件系统的功能 .....	160	5.6.2 文件的读写 .....	192
5.1.2 文件系统的实现 .....	162	5.6.2.1 read 系统调用 .....	193
5.1.2.1 文件的逻辑结构 .....	162	5.6.2.2 write 系统调用 .....	193
5.1.2.2 文件的物理结构 .....	163	5.6.2.3 应用实例 .....	194
5.1.2.3 文件存储空间管理 .....	164		
5.1.2.4 文件目录管理 .....	165		
5.1.2.5 文件的安全设计 .....	165		
5.2 Linux 文件系统概述 .....	166		



5.6.3 文件的随机存取 .....	196	6.3.1 设备驱动程序分类 .....	234
5.6.3.1 lseek 系统调用 .....	196	6.3.2 设备开关表 .....	234
5.6.3.2 应用实例 .....	196	6.3.3 驱动程序入口点 .....	236
5.6.4 文件的保护和控制 .....	197	6.4 Linux 设备的 I/O 调用 .....	236
5.6.4.1 chown 和 chmod 系统调用 .....	197	6.4.1 设备文件 .....	236
5.6.4.2 umask 系统调用 .....	199	6.4.2 Linux 文件系统对设备文 件的处理 .....	237
5.6.4.3 fcntl 系统调用 .....	199	6.4.3 字符设备 I/O .....	237
5.6.4.4 应用实例 .....	201	6.4.4 块设备 I/O .....	238
5.6.5 目录文件管理 .....	209	6.5 中文 Linux 核心技术 .....	238
5.6.5.1 mkdir, rmdir 系统调用 .....	209	6.5.1 汉字处理的基本原理 .....	238
5.6.5.2 chdir 系统调用 .....	210	6.5.1.1 基本差异 .....	239
5.6.5.3 Link, unlink .....	210	6.5.1.2 汉字处理的需求 .....	239
5.6.5.4 应用实例 .....	210	6.5.1.3 中文 Linux 对汉字的支持 .....	240
5.6.6 文件信息查询 .....	211	6.5.2 相关设备结构 .....	241
5.6.6.1 stat, fstat, lstat .....	211	6.5.2.1 显示机制 .....	241
5.6.6.2 应用实例 .....	212	6.5.2.2 Framebuffer .....	242
5.7 小结 .....	216	6.5.3 Linux 内核汉字技术 .....	243
<b>第 6 章 设备管理</b> .....	<b>217</b>	6.5.3.1 实现机制 .....	243
6.1 设备管理概述 .....	217	6.5.3.2 核心汉字支持细节介绍 .....	244
6.1.1 设备的分类和功能 .....	217	6.6 Linux 系统设备管理应用实例 .....	245
6.1.2 设备管理的实现 .....	218	6.6.1 字符设备驱动程序的实现 .....	245
6.1.2.1 总线概述 .....	218	6.6.1.1 字符设备结构 .....	245
6.1.2.2 数据传送方式 .....	219	6.6.1.2 驱动程序入口点 .....	246
6.1.2.3 缓冲技术 .....	221	6.6.1.3 字符设备驱动程序的 安装 .....	248
6.1.2.4 设备驱动程序 .....	222	6.6.1.4 实例 .....	248
6.1.2.5 SPARC 机中设备管理初始化 过程 .....	222	6.6.2 块设备驱动程序的实现 .....	254
6.2 Linux 设备管理概述 .....	223	6.6.2.1 块设备的结构 .....	254
6.2.1 Linux 中总线的使用 .....	225	6.6.2.2 驱动程序入口点 .....	254
6.2.1.1 PCI 总线概述 .....	225	6.6.2.3 相关问题 .....	257
6.2.1.2 PCI 总线的地址管理 .....	226	6.7 小结 .....	257
6.2.1.3 Linux 系统中有关 PCI 的 操作 .....	227	<b>第 7 章 Linux 网络系统</b> .....	<b>258</b>
6.2.2 中断和 DMA .....	228	7.1 概述 .....	258
6.2.2.1 中断 .....	228	7.1.1 网络协议 .....	258
6.2.2.2 计时机制的使用 .....	230	7.1.1.1 ISO/OSI 网络模型 .....	259
6.2.2.3 DMA .....	230	7.1.1.2 TCP/IP 协议 .....	261
6.2.3 内存管理 .....	231	7.1.2 BSD Socket .....	266
6.2.3.1 设备内存地址的分类 .....	232	7.1.3 网络设备 .....	267
6.2.3.2 设备内存的使用和分配 .....	232	7.2 网络设备接口 .....	268
6.2.3.3 用户空间和内核空间数据 传递 .....	233	7.2.1 核心层—网络驱动 .....	268
6.3 Linux 设备驱动程序框架 .....	234	7.2.1.1 层次结构 .....	268
		7.2.1.2 网络设备初始化 .....	273

7.2.1.3 网络协议绑定 .....	275	8.2 Shell 编程初步 .....	301
7.2.2 核心层—灵活的网络机制 .....	277	8.2.1 目的 .....	301
7.2.2.1 网络虚拟设备 .....	277	8.2.2 要求 .....	301
7.2.2.2 IP 伪装 .....	279	8.2.3 内容 .....	301
7.2.2.3 防火墙、路由 .....	282	8.3 简单 C 语言编程实验 .....	302
7.2.2.4 其他新特性 .....	283	8.3.1 目的 .....	302
7.2.2.5 IPv6 简介 .....	284	8.3.2 要求 .....	303
7.3 系统层—网络服务系统机制 .....	284	8.3.3 内容 .....	303
7.3.1 网络守护进程 .....	285	8.4 Linux 进程管理一 .....	303
7.3.2 INETD .....	285	8.4.1 目的 .....	303
7.3.2.1 INETD 概述 .....	285	8.4.2 要求 .....	303
7.3.2.2 INETD 的机制 .....	286	8.4.3 内容 .....	303
7.3.2.3 INETD 配置说明 .....	286	8.5 Linux 进程管理二 .....	305
7.4 网络编程 .....	287	8.5.1 目的 .....	305
7.4.1 网络通信编程要点 .....	287	8.5.2 要求 .....	305
7.4.1.1 TCP 协议传输中的状态 .....	288	8.5.3 内容 .....	305
7.4.1.2 UDP 协议传输过程 .....	289	8.6 Linux 线程 .....	314
7.4.1.3 网络程序的设计 .....	290	8.6.1 目的 .....	314
7.4.1.4 网络编程中的重要函数 .....	290	8.6.2 要求 .....	314
7.4.2 客户端编程 .....	293	8.6.3 内容 .....	314
7.4.2.1 面向连接的客户程序 .....	293	8.7 Linux 存储管理 .....	322
7.4.2.2 非面向连接的客户程序 .....	294	8.7.1 目的 .....	322
7.4.3 服务器端编程 .....	294	8.7.2 要求 .....	322
7.4.3.1 面向连接的服务器端通信		8.7.3 内容 .....	322
程序 .....	295	8.8 Linux 文件管理 .....	325
7.4.3.2 非面向连接的服务器 .....	296	8.8.1 目的 .....	325
7.5 网络应用 .....	297	8.8.2 要求 .....	325
7.5.1 Internet 应用 .....	297	8.8.3 内容 .....	325
7.5.1.1 Sendmail .....	297	8.9 Linux 设备管理 .....	333
7.5.1.2 DNS .....	298	8.9.1 目的 .....	333
7.5.1.3 Web Server、FTP 等 .....	298	8.9.2 要求 .....	334
7.5.1.4 Ipchains .....	298	8.9.3 内容 .....	334
7.5.2 群组服务器应用 .....	298	8.10 Linux 网络管理 .....	345
7.5.2.1 Appletalk .....	299	8.10.1 目的 .....	345
7.5.2.2 IPX/SPX .....	299	8.10.2 要求 .....	346
7.5.2.3 SMB .....	299	8.10.3 内容 .....	346
7.6 小结 .....	300	8.11 综合实例 .....	352
<b>第 8 章 应用实例 .....</b>	<b>301</b>	8.11.1 目的 .....	352
8.1 认识 Linux 系统 .....	301	8.11.2 要求 .....	352
8.1.1 目的 .....	301	8.11.3 内容 .....	352
8.1.2 要求 .....	301	附录 .....	353
8.1.3 内容 .....	301	参考文献 .....	360

# 第 1 章 概 论

## 1.1 现代操作系统

随着计算机的发展，操作系统也有一个由简单到复杂、由低级到高级的发展过程。作为计算机与人的交互界面和一切应用系统的开发平台，操作系统成为计算机系统中变化最快、影响面最大的一个部分。现代操作系统的典型代表是 UNIX、Linux、Windows NT，它们的共同特点在于：

1. 性能更好 现代操作系统普遍采用分时策略，在调度算法、存储管理、文件管理、对外设和网络的支持等方面都采用了先进的技术，更好地适应计算机硬件的发展和应用领域对操作系统的要求。

2. 对计算机硬件支持范围更广 例如微软（Microsoft）的操作系统从 DOS、Windows 98 到 Windows NT，它们的使用范围就从个人计算机扩展到企业级的服务器和工作站上。UNIX 则从大型机向 PC 机扩展，先后出现 SCO UNIX 和 XENIX 等微机版的 UNIX。最为引人注目的是操作系统的新秀 Linux，几乎可以用在各种计算机平台上，除了个人计算机、企业级服务器和 workstation，现在又推出了 64 位平台上的 Linux 版本。

3. 分布式和网络化 现代操作系统在支持多用户多任务的基础上，对网络的支持已经成为核心的一部分，包括通信协议、网络服务、网络资源的管理和安全技术等都属于操作系统的重要组成部分。因为篇幅问题，在本书中对该部分没有过多阐述。

4. 用户界面更友好 操作系统的用户界面向着多媒体、个性化的方向发展，越来越便于使用和维护。

5. 应用范围更广 操作系统从计算机上渗透到电子信息领域的各个角落。从应用软件系统到电子信息产品。特别是随着 Internet 的普及，带动了信息家电的异军突起，嵌入式操作系统被“嵌入”到了各种电器中。因此，操作系统的原理和实现技术有了更为广泛的应用和价值。

本书将以现代操作系统理论为基础，重点研究 Linux 操作系统的实现原理和技术，并提供了大量的实例。

### 1.1.1 操作系统的概念

操作系统属于计算机系统上的系统软件，作为计算机系统的管理者，它管理和控制计算机系统中硬件资源和软件资源，合理地组织计算机的工作流程；作为计算机和用户之间的接口，它为用户使用计算机提供了高效、可靠、方便和功能强的工作环境。

计算机系统由计算机硬件和计算机软件组成，如图 1-1 所示。

没有操作系统和软件支持的计算机称为“裸机”，它是不能工作的。操作系统则位于计算机硬件和软件之间。一方面，操作系统承担了对计算机硬件的具体操作和管理维护，将计算机硬件的细节与用户隔离开来，为用户提供透明、简洁的使用硬件的方式。另一方面，操

作系统对整个计算机系统进行了组织和管理，包括处理机、文件系统、存储器、外围设备、网络等等，提供多用户多任务的工作平台，使得构建在其上的应用系统能够高效、安全地运行，合理处理多任务并发带来的资源竞争与共享等问题。

### 1.1.2 操作系统的分类

操作系统的产生和发展与计算机硬件技术的发展密切相关。在每一个计算机硬件发展的历史阶段，就有相应类型的操作系统。

从1946年第一台计算机诞生以来，计算机和操作系统的发展经历了四个阶段：

1946~1955年：电子管时代，无操作系统。

1955~1965年：晶体管时代，简单批处理操作系统。

1965~1980年：集成电路时代，多道批处理操作系统，分时操作系统。

1980~至今：大规模和超大规模集成电路时代，分时操作系统，个人计算机操作系统，网络操作系统，分布式操作系统。

总结起来，操作系统的基本类型有：

批处理操作系统（Batch Processing Operating System）；

分时操作系统（Time Sharing Operating System）；

实时操作系统（Real Time Operating System）；

网络操作系统（Network Operating System）；

分布式操作系统（Distributed Operating System）；

通用操作系统（General Operating System）；

个人操作系统（Personal Operating System）。

#### 1.1.2.1 批处理操作系统

在批处理操作系统下，用户将自己的程序或任务提交给计算机的管理操作员，称为“作业”。提交作业之后直到获得结果之前就不再和计算机打交道。操作员把用户提交的作业分批进行处理。每批中的作业将由操作系统负责作业间的自动调度。根据作业执行的方式，分为单道批处理系统和多道批处理系统。单道批处理系统是早期的批处理系统，由监督程序转化而来。每次只向计算机提交一个作业，完成后再提交下一个，作业顺序处理，计算机系统资源利用率不高。多道批处理系统是按多道程序设计的调度原则，从一批后备作业中选取多道作业调入内存并组织它们运行。由于系统资源为多个作业所共享，作业之间自动调度执行，在运行过程中用户不干预自己的作业，从而大大提高了系统资源的利用率和作业吞吐量。其缺点是无交互性，作业周转时间长，用户使用不方便。

#### 1.1.2.2 分时操作系统

在分时操作系统中，将计算机的CPU处理时间分为时间片，按照时间片轮转的方式为多个终端用户服务，使得为每一个用户提供足够快的响应时间，并提供交互能力。

分时操作系统最大的特点是交互性，并实现了计算机系统的多用户多任务工作方式。

当今流行的操作系统，如UNIX，Linux，Windows NT都是支持多用户多任务的分时操

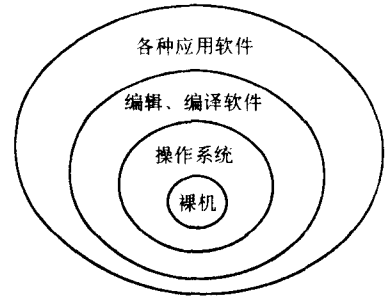


图 1-1 操作系统与计算机系统的关系

作系统。我们将在本书中详细分析操作系统在分时工作方式下的系统调度问题。

#### 1.1.2.3 实时操作系统

实时操作系统是随着计算机应用于实时控制和实时信息处理领域中发展起来的，实时性现在已经成为现代操作系统中的特性之一。

在实时操作系统的设计中主要考虑的因素有实时时钟管理、系统的中断处理策略、处理机调度策略、系统的过载保护、可靠性和安全性、系统的冗余措施等等。

Linux 操作系统在设计中，充分考虑了以上因素，因而具有较强的实时性。在实时时钟管理、系统的中断处理策略、处理机调度策略等方面都有其实时处理的特点。

#### 1.1.2.4 网络操作系统和分布式操作系统

计算机网络是通过通信设施互联起来的自主计算机的集合，它是能够实现信息交换、资源共享、可互操作和协处理的系统。网络操作系统是用于计算机网络，根据网络协议开发的，实现网络管理、通信、资源共享、网络安全、网络应用和网络服务的操作系统。

有的网络操作系统是基于原有的操作系统开发的，后来逐步将网络支持加入系统内核。也有的操作系统从底层或内核起就实现了对网络的支持。Linux 从正式版起，就包含了对 TCP/IP 协议的支持，现在已经具备了丰富的网络操作系统功能，网络性能在某些方面超过了 Windows NT。

分布式系统是建立在网络之上的软件系统，它与计算机网络有许多相似之处，如具有模块性、并行性、自制性和通用性等特点。它们的区别主要在软件（尤其是操作系统）而不是硬件。分布式操作系统的最大特征在于它的高度的整体性和透明性。

在分布式系统中，网络上的计算机对用户是透明的，用户不需了解系统中硬件和软件的分布，发给操作系统的命令和操作，由操作系统来选择合适的计算机和设备来处理。因此操作系统需要对整个系统做整体的协调和控制。而在计算机网络中，用户必须明确指定操作的源地址的目的地址，明确地递交远程任务，管理整个网络。

分布式系统的出现是基于两项重要的技术：高性能低价格微机的快速发展和普及，高速局域网的出现。这两项技术的发展使得通过高速网络把大量的 CPU 连接起来变得容易实现。因此，分布式操作系统的目的就是管理这样一个系统，它与传统的操作系统在各方面都有很大不同，特别是在系统组织结构、系统的同步、进程（线程）和处理器的管理、文件系统等方面。尽管分布式操作系统已经出现，许多问题已经取得了一些进展，但离实用和普及还有一定的距离。

#### 1.1.2.5 通用操作系统

最初，人们希望把批处理、分时、实时系统的特性集中在一起，开发出所谓的通用操作系统。实际上，在现代操作系统中，我们已不必要去区分这一点。如果愿意，可以把它们都称为通用操作系统，因为现代操作系统中，大多都具备了以上的批处理、分时、实时特性和网络等特性。这也正是现代操作系统发展的一个特色。

#### 1.1.2.6 个人操作系统

个人计算机从 IBM 个人计算机开始，迅速流行起来，它使个人拥有计算机成为可能。个人计算机上运行的操作系统主流一直是微软的操作系统，从 MS-DOS、Windows 3.x、Windows 95/98 直至 Windows NT。从 Linux 诞生起，它就成为微软个人操作系统的强有力的挑战者。

个人操作系统在 Windows 95 出现以前，和个人计算机的硬件体系结构相配套，是非常简单的操作系统，它的处理机管理、内存管理、文件管理、设备管理等都是非常简单的。比

如，在调度管理、内存管理、系统安全性等方面不支持真正的多任务。随着个人计算机硬件性能的提高，为操作系统快速发展创造了条件。目前，个人计算机性能已与小型机、微型机相当，操作系统有了施展的空间。现在个人计算机和微型机上流行的 Linux、Windows NT 已经具备了如前文所述的各种优良的特性，从个人计算机上发展起来的操作系统不再局限于个人计算机。

### 1.1.3 操作系统的功能

操作系统的功能是管理计算机系统中的所有软硬件资源，合理地组织计算机的工作流程，为用户使用计算机提供功能强、高效、可靠、方便的工作环境。一般的操作系统理论中，按照资源的类型，操作系统的功能被划分为五大部分：处理机管理、存储管理、文件管理、设备管理和用户接口。在现代操作系统中，网络功能已经成为操作系统中不可缺少的一部分。因此，研究操作系统的功能，还应该包括网络功能。

#### 1.1.3.1 处理机管理

在现代操作系统中，由于对多用户多任务的支持，处理机管理成为最重要的一个部分，它涉及处理机资源的调度策略、调度实施管理。处理机调度以进程为单位，在有的支持线程的系统中，则以线程为调度单位。操作系统不同的处理机调度策略，决定了操作系统的性能。

#### 1.1.3.2 存储管理

存储管理的功能是对内存的组织、分配、保护与扩充。内存管理方式主要有分区管理、页式管理、段式管理和段页式管理。不同的管理方式在内存的组织、分配、保护与扩充等问题上采取了不同的策略，性能有所不同。

存储管理要保证系统及各用户进程得到所需的内存，进程之间的内存空间互不影响，保证用户程序不会破坏系统程序。当用户进程所需要的内存超过计算机所提供的物理内存容量时，通过统一管理内存和外存，为用户提供一个容量比实际内存大得多的虚拟存储器。

#### 1.1.3.3 文件管理

文件管理实现对计算机系统软件资源的管理。

文件是有名字的相关联的数据和程序的集合。文件保存在外部存储器上，如硬盘、光盘等。文件管理的主要功能之一是实现对文件的逻辑结构和物理结构组织和存取，以及外存空间的组织管理。为了实现对用户透明的按名存取功能，文件管理还包括对目录的组织和管理。文件保护、保密和共享也是文件管理的重要内容。

通过文件管理提供的接口，用户不必了解文件的物理细节，通过打开、关闭、读、写等基本操作即可完成对数据的存取。因此，在操作系统中，利用虚拟文件系统的概念将设备接口、网络通信接口组织为文件接口方式，大大地方便了用户，提高系统的安全可靠及效率，是一种非常有效的办法。在本书的文件管理、设备管理、以及网络管理中将详细介绍。

#### 1.1.3.4 设备管理

设备管理是操作系统和计算机硬件设备之间的接口，接收系统进程和用户进程的 I/O 请求，控制相应的外设完成 I/O 操作，实现设备之间、设备与处理机之间的并行工作。设备管理子系统包括与硬件相关部分（如中断实现机制、设备的分配、设备驱动程序等）以及与硬件无关部分（如缓冲技术、虚拟技术、与文件系统的接口等）。设备驱动程序处理设备的具体硬件特征，利用设备文件作为接口方式，为系统和用户使用设备提供了良好的界面和透明的接口，体现了设备独立性。

### 1.1.3.5 用户接口

以上是操作系统对资源的管理，此外操作系统还为用户提供一个友好的用户接口。一般来说，操作系统提供两种方式的接口：命令方式和系统调用。命令方式是系统提供一组控制操作命令（如 UNIX 和 Linux 中的 Shell 命令，DOS 中的 Command 命令），供用户去组织和控制自己作业的运行。系统调用接口是程序级的接口，即提供一组系统调用或函数供用户程序和其他系统程序调用，以完成某些特殊的系统功能。如打开文件、创建进程等。

### 1.1.3.6 网络管理

在现代操作系统中，在内核就实现了对通信协议和网络设备的支持。此外，网络管理还实现了网络资源的管理、网络通信、资源共享、网络安全控制和网络服务等，并向用户提供网络应用的开发接口。

## 1.1.4 操作系统的结构

操作系统由一系列软件构成。因此，操作系统作为软件系统，本身如何组织决定了操作系统的性能和可扩展性。主要有以下几种组织方式：整体式模型、层次式模型和客户/服务器模型。

### 1.1.4.1 整体式模型

早期最常用的组织方式是整体式系统。整个操作系统组织成一个过程集，每个过程都可以任意调用其他过程，扩展性差。系统中的每个过程都定义调用接口，即入口参数和返回值，相互间的调用不受约束。为了系统的安全，应用程序和操作系统本身分开，将系统的工作状态分为两种：用户态和核心态，通过处理机状态字来标志当前的运行状态。在核心态下，主要运行的是操作系统的系统程序，直接存取系统数据和硬件资源。在用户态下，用户应用程序有限使用系统资源，直接操作系统数据和硬件资源的指令被禁止执行，需要通过系统调用转入核心态来获得系统服务。早期的 DOS 操作系统采用了整体式模型，见图 1-2，它提供给用户的调用接口就是 INT 中断调用。

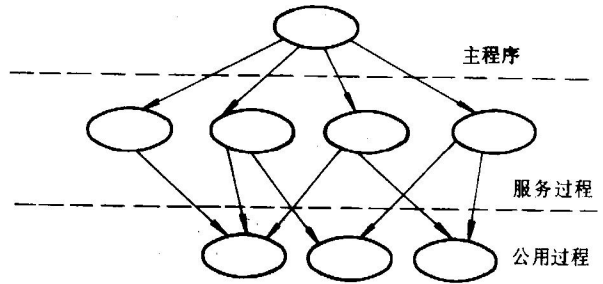


图 1-2 整体式模型

### 1.1.4.2 层次式模型

把系统根据功能模块和在系统中的地位，进一步通用化，就变成层次式系统。在系统中，上层软件都基于下一层软件之上，每一层为上层提供接口，内部的具体实现对其他层透明。Linux 属于这种层次结构，见图 1-3（详细内容见第 2 章）。但是在网络管理子系统中，Linux 采用了客户/服务器模型。

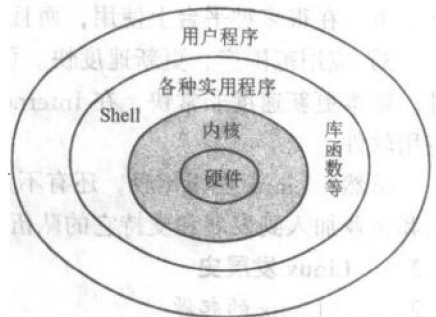


图 1-3 层次式模型

### 1.1.4.3 客户/服务器模型

现代操作系统的一个发展趋势是：操作系统中有一个很小的内核（Kernel）。在此基础上，将大多数操作系统的功能由用户进程来完成。提供系统服务的进程称为服务器进程，发出请求的进程称为用户进程。用户进程（Client）向

服务器进程 (Server) 发请求, 随后服务器进程完成操作并返回回答信息。

典型的客户/服务器模型如图 1-4 所示。操作系统内核主要负责处理客户机与服务之间的通信, 并实现和硬件的接口。和硬件实现接口的方法有两种: 一种是在内核中建立一些关键的服务进程 (如 I/O 设备驱动程序) 来专门访问硬件, 通过通常的消息机制与其他进程通信。另一种方法是把机制与策略分开, 在内核中建立最基本的硬件读写机制, 由用户空间的服务进程实现对读写之外的其他控制, 如合法性检查等。

操作系统的内核之上由多个服务器组成, 分别实现某一方面功能, 如文件服务、进程服务、存储器服务等。服务器之间相对独立, 体积更小, 可以动态加/卸载, 易于管理。所有的服务都以用户进程的形式独立运行, 通过向内核或其他服务器发请求实现与硬件相关或其他功能, 不直接访问硬件, 因此如果某一个服务器中发生错误, 而不会导致整个系统崩溃。

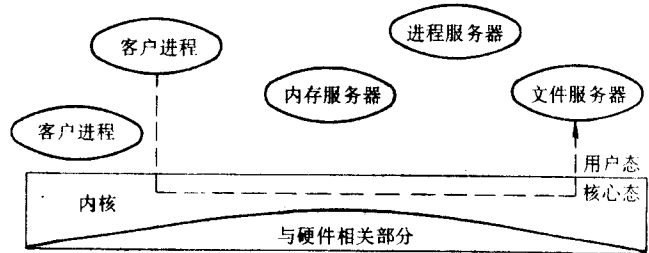


图 1-4 客户/服务器模型

客户/服务器模型的另一个优点是, 它适用于分布式系统。如果客户进程通过消息机制与服务器通信, 那么服务是由本地服务器还是通过网络的远程服务器实现, 客户所做的工作都是相同的。

## 1.2 Linux 系统概论

Linux 操作系统从它诞生起, 就成为操作系统中的一颗新星, 它伴随 Internet 日益成熟, 同时也促进了 Internet 的发展。它之所以受人瞩目, 在于以下特点:

- 1) 源代码公开、免费。任何人都可以免费得到它, 既可以将它用于应用开发, 也可以拿来作为学习操作系统的原型, 研究源代码和实现机理。
- 2) 性能优越, 功能强大, 支持平台多。Linux 系统与 UNIX 相比要小得多, 但它实现了虚拟存储器、多进程、多处理器、多文件系统、网络、线程等, 具有现代操作系统的特征, 可以在很多种平台上使用, 而且对硬件的支持越来越多。
- 3) 应用范围广, 更新速度快。可用于网络服务器、个人计算机, 有成千上万种应用软件。版本更新速度非常快, 在 Internet 上每天都会出现关于改进和扩展系统的新思想和新的应用软件。

虽然, Linux 还很年轻, 还有不足, 但它的公开免费、高性能等特性使得越来越多的个人和企业加入到发展和支持它的队伍中来, 使它日益完善和广泛应用。

### 1.2.1 Linux 发展史

#### 1.2.1.1 Linux 的起源

Linux 由芬兰赫尔辛基大学的 Linus Benedict Torvalds 创建。Linus 在学习著名计算机科学家 Andrew S. Tanenbaum 编写的用于教学的 Minix 操作系统时, 发现 Minix 的功能还不够完善, 于是决定自己在 PC 机上编写一个保护模式下的操作系统, 这就是 Linux 的原型。1991 年 10 月, Linux 第一个公开版 0.02 版发布。刚开始的 Linux 版本是被放在一个 FTP 服务器上供大家



自由下载，系统管理员为它建了 Linux 的目录来存放这些文件，于是 Linux 就成为了现在的名字。

1994年3月，Linux 1.0 正式版发布。现在 Linux 已经成为一个完整的类 UNIX 操作系统了，它的核心的最新稳定版为 2.2.16，最新测试版为 2.3.99。它有个可爱的吉祥物——一只小企鹅，企鹅取自 Linus 的家乡芬兰的吉祥物。现在，几乎各种版本的 Linux 操作系统都带着这个标志。

### 1.2.1.2 Linux 的版本

Linux 的版本号分为两种：内核 (Kernel) 版本与发行套件 (distribution) 版本。

内核版本指在 Linus 领导下的开发小组开发出的系统内核的版本号，目前最新的稳定版本为 2.2.16，测试版本为 2.3.99。一般，版本号的第二位为偶数表示是稳定版，奇数表示测试版。

发行套件版是一些组织或厂商将 Linux 内核和应用软件包装起来，并提供一些安装和系统设置工具，构成一个发行套件。最常见的如 Red Hat, Slack Ware, Debian 等，中文的如 Xteam Linux, TurboLinux, 红旗 Linux, 蓝点 Linux 等，他们也不断地推出新版本。发行套件版本由发行者决定，与内核的版本号相对独立，如 Red Hat 6.0, XteamLinux 3.0 等。

### 1.2.2 Linux 的特点

Linux 最大的特点是免费。从系统研究的角度来看，它是类 UNIX，具有 UNIX 的许多优点，同时，拥有自己独特的特点。

1. 多用户和多任务 Linux 是多用户多任务操作系统，支持多个用户同时登录，同时启动多个任务，访问多个设备。许多大学的 BBS 服务器使用运行着 Linux 的普通微机，能支持 200 人同时上线。

2. 多平台 Linux 最早运行在 x86 上，目前 Red Hat 已推出了 Alpha、Sparc 平台上的 Linux。还有其他一些平台如 MIPS, PowerPC, Mach、M68K (Amiga) 等。

3. 虚拟存储器和动态链接共享库 Linux 采用页式存储管理实现了虚拟存储器，支持多个进程共享同一个内存页面，并且采用了“Copy on Write”技术，加快了程序运行的时间，提高了管理内存的效率。

4. 优秀的磁盘缓冲调度功能 Linux 给人们留下的深刻印象之一是它的磁盘 I/O 速度。这得益于它的缓冲区管理策略。对于磁盘这一类的块设备，系统给它们定义了数据块缓冲区缓存。同时，Linux 能够将系统没有用到的物理内存全部用来做硬盘的高速缓存，当有应用程序需要更多的内存时，系统会自动释放一部分内存给应用程序。

5. 支持多种文件系统 Linux 支持的文件系统包括 minix, ext, ext2, xiafs, hpfs, fat, ms-dos, umsdos, vfat, proc, nfs, iso9660, smbfs, ncfs, affs, ufs, rornfs, sysv, xenix, coherent。Linux 缺省的文件系统是 ext2。其他的文件系统可以动态地向系统注册、安装或卸载。Linux 把安装上的文件系统挂接 (mount) 为系统目录树上的一个子目录，然后以目录的方式访问该文件系统中的目录和文件。

6. X-Window 系统 X-Window 是类 UNIX 操作系统平台上的通用图形界面，它支持许多应用程序，已经成为一个标准。近年来推出的 Linux 发行套版，大多都配备了 X-Window 及运行在其上的各种应用程序。

7. 强大的网络功能 Linux 在其核心中支持的基本协议包括 TCP, IPv4, IPX, DDP