

屠立德 王丹 金雪云 编著

操作系统基础

（第4版）

清华大学出版社



屠立德 王丹 金雪云 编著

操作系统基础

★ ★ ★ ★ ★ ★ ★
(第4版)
★ ★ ★ ★ ★ ★ ★

清华大学出版社
北京

内 容 简 介

操作系统作为核心的系统软件,负责控制和管理整个系统的资源并组织用户高效协调地使用这些资源。本书是在《操作系统基础(第3版)》的基础上修订而成的。与第3版相比,第4版在结构、内容上都做了调整、修改和增删。

本书阐述了操作系统的基本工作原理以及设计方法,力求将现代操作系统的典型特征,即多线程、微内核、分布式系统、客户/服务器模型与经典的操作系统原理紧密结合。

全书共13章,主要介绍了操作系统的基本概念和运行环境、进程和线程、处理器调度与死锁、存储管理、设备管理、文件管理、分布式系统,最后介绍了Windows和Linux操作系统的结构和实现。每章后面都有本章小结及难度适宜的习题,便于读者自学或巩固所学的知识。

本书内容丰富,结构清晰,突出基础,注重应用,强调理论与实践的结合,适合作为高等院校计算机专业或相关专业操作系统课程的教材,也可以作为从事操作系统设计与系统内核开发的技术人员的参考书籍。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

操作系统基础/屠立德等编著. —4版. —北京:清华大学出版社,2014

ISBN 978-7-302-36106-0

I. ①操… II. ①屠… III. ①操作系统—高等学校—教材 IV. ①TP316

中国版本图书馆CIP数据核字(2014)第069697号

责任编辑:梁颖 薛阳

封面设计:常雪影

责任校对:白蕾

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 21.25 字 数: 519千字

版 次: 1999年1月第1版 2014年8月第4版 印 次: 2014年8月第1次印刷

印 数: 1~2500

定 价: 34.50元

产品编号: 048885-01



前言

首先要衷心感谢广大读者的厚爱和支持,本书出版二十几年来,已印刷二十多次,发行数十万册。广大读者把他们积累的经验 and 体会无私地贡献给我们,为本书的改进和提高做出了贡献,在此向广大读者表示深深的敬意和感谢。

此次修订出版的第4版以经典操作系统基本原理和概念为框架和基线,以密切反映现代操作系统技术的新发展和新特征为重点。因为几十年来,操作系统经历着日新月异的变化,尽管现代操作系统以多线程、微内核、SMP多处理器系统、客户/服务器模式和分布式、网络系统为特征,但操作系统的基本原理和概念不但没有什么变化,而且更趋成熟,它依然是操作系统的基本骨架。在此基础上,多线程机制、微内核与客户/服务器模式、分布式系统等仍是当前发展的热点,本书通过在专门章节中加以研讨,或者将其融合在全书之中加以介绍。

本书目录中画有*号部分,授课教师可以根据教学计划进行变动,并鼓励读者通过自学(课内学时有限)来掌握它。本书内容阐述深入浅出,适合自学。

本书以现代操作系统的典型系统 Windows NT、UNIX 和 Linux 系统作为全书的范例,并将近代 UNIX 系统(如 Solaris 等)的先进技术和机制分散到各章中作为该章的典型范例来使用,以收到与全书内容紧密结合的效果,便于读者领会和理解该章所述的内容。Windows NT 和 Linux 作为一个系统范例自成一章,便于读者从系统整体角度来认识和理解一个操作系统。建议读者在阅读各章时也参考 Windows NT 和 Linux 的相关内容。

本书第9章、第10章、第12章由屠立德老师编写,第5章、第7章、第8章、第11章由金雪云老师编写,其他部分由王丹老师编写。屠立德教授负责全书的审阅、校核和定稿。由于时间仓促以及作者水平所限,错误和不妥之处在所难免,恳请读者批评指正。

作者

2014年6月



目 录

第 1 章 引论	1
1.1 计算机系统概述	1
1.1.1 计算机的硬件组织	1
1.1.2 软件的层次与虚拟机的概念	3
1.2 操作系统的形成和发展	4
1.2.1 什么是操作系统	4
1.2.2 操作系统的形成和发展	5
1.3 多道程序设计的概念	6
1.3.1 多道程序设计的引入	6
1.3.2 多道程序设计的概念	7
1.4 操作系统的功能和特性	8
1.4.1 操作系统的功能	8
1.4.2 操作系统的特性	10
1.5 操作系统的类型	10
1.5.1 多道批处理操作系统	11
1.5.2 分时操作系统	11
1.5.3 实时系统	12
1.5.4 网络操作系统	13
1.5.5 多处理操作系统	14
1.5.6 分布式操作系统	15
1.5.7 嵌入式操作系统	15
1.5.8 多核系统	15
1.6 操作系统的设计	16
1.6.1 设计的目标和原则	16
1.6.2 操作系统设计	17
1.7 操作系统的结构	18
1.7.1 模块接口法	18
1.7.2 层次结构设计法	20

1.7.3 微内核结构	22
1.7.4 微内核的实现	24
本章小结	25
习题	26
第2章 操作系统的运行环境	28
2.1 硬件环境	28
2.1.1 中央处理器	28
2.1.2 主存储器	31
2.1.3 缓冲技术	33
2.1.4 中断技术	34
2.1.5 时钟、时钟队列	37
2.2 操作系统与其他系统软件的关系	39
2.2.1 作业、作业步、进程的关系	39
2.2.2 重定位的概念	40
2.2.3 绝对装入程序与相对装入程序	41
2.3 操作系统与用户的接口	43
2.3.1 作业控制语言	43
2.3.2 联机作业控制	45
2.4 固件——微程序设计概念*	47
2.4.1 微程序设计的概念	47
2.4.2 微程序设计与操作系统	48
本章小结	48
习题	49
第3章 进程管理	50
3.1 进程的概念	50
3.1.1 进程的引入	50
3.1.2 进程的定义	51
3.2 进程的状态	52
3.2.1 进程的状态及其变化	52
3.2.2 进程的挂起和解除挂起的状态	54
3.3 进程的描述和管理	56
3.3.1 进程的描述	56
3.3.2 进程管理	57
3.4 进程控制	58
3.4.1 进程的控制原语	58
3.4.2 操作系统与进程控制的执行	61



3.5 UNIX SVR4 的进程管理*	63
本章小结	66
习题	67
第4章 多线程	68
4.1 线程的概念	68
4.1.1 线程的引入	68
4.1.2 线程的概念	69
4.2 线程的状态和线程管理	71
4.2.1 线程的状态	71
4.2.2 线程的描述	72
4.2.3 线程的管理	74
4.3 多线程的实现	75
4.3.1 概述	75
4.3.2 用户级线程	76
4.3.3 内核级线程	78
4.3.4 KLT 和 ULT 结合的方法	79
4.3.5 线程库	79
4.4 Solaris 操作系统的线程机制*	80
4.4.1 Solaris 的多线程结构	80
4.4.2 轻质进程	81
4.4.3 内核线程	82
4.4.4 用户线程	82
4.4.5 线程的执行	82
4.4.6 内核中断线程	84
本章小结	84
习题	85
第5章 互斥与同步	86
5.1 概述	86
5.2 临界区	87
5.2.1 临界区的提出	88
5.2.2 临界区的互斥要求	89
5.3 互斥	90
5.3.1 互斥的软件解决方法	90
5.3.2 互斥的硬件解决方法	92
5.4 信号量	94
5.4.1 信号量	94

5.4.2	信号量及同步原语	94
5.4.3	同步原语的不可分割性	96
5.4.4	用信号量实现进程间互斥	96
5.4.5	生产者和消费者问题	96
5.4.6	读者和写者问题	98
5.5	管程	99
5.5.1	管程的定义	99
5.5.2	用管程实现同步	100
5.6	进程间的通信	102
5.6.1	消息通信	102
5.6.2	间接通信模式	103
5.6.3	其他消息通信模式	104
5.7	UNIX 的进程同步和通信	105
5.7.1	管道	105
5.7.2	消息	106
5.7.3	共享主存	107
5.7.4	信号量	108
5.7.5	信号或软中断	109
	本章小结	110
	习题	110
第 6 章	死锁	112
6.1	死锁问题的提出	112
6.2	死锁的必要条件	113
6.2.1	资源的概念	113
6.2.2	死锁的必要条件	114
6.3	死锁的预防	115
6.3.1	预先静态分配法	115
6.3.2	有序资源使用法	115
6.4	死锁的避免和银行家算法	116
6.4.1	单资源的银行家算法	116
6.4.2	多资源的银行家算法	117
6.5	死锁检测与恢复	120
6.5.1	死锁的检测	120
6.5.2	死锁的恢复	121
6.6	资源分配图	121
6.6.1	资源分配图	121
6.6.2	利用资源分配图进行死锁分析	122



6.6.3 资源分配图化简法	123
本章小结	124
习题	124
第7章 实存储器管理技术	126
7.1 引言	126
7.1.1 主存储器的物理组织	126
7.1.2 主存储器的管理功能	127
7.2 固定分区	127
7.3 可变分区多道管理技术	129
7.3.1 可变分区存储管理的概念	129
7.3.2 存储分配算法	131
7.3.3 存储器的压缩和程序浮动	132
7.3.4 可变分区多道管理的地址变换	132
7.4 简单分页	134
7.5 简单分段	136
7.6 内核主存管理	138
7.6.1 内核主存管理概述	138
7.6.2 二次幂空闲表分配器	139
7.6.3 伙伴系统	139
本章小结	140
习题	141
第8章 虚拟存储管理	143
8.1 虚拟存储系统的基本概念	143
8.2 请求页式存储管理	144
8.2.1 地址转换	144
8.2.2 硬件支持	148
8.3 请求分段存储管理	150
8.3.1 请求分段概述	150
8.3.2 分段的实现	151
8.4 段页式存储管理	152
8.4.1 请求段页式存储管理的基本概念	152
8.4.2 请求段页式存储管理的地址转换	152
8.4.3 段页式存储管理算法	154
8.4.4 请求段页式存储管理的优缺点	155
8.5 页面置换算法	156
8.5.1 页面访问失效及处理	156

8.5.2	页面置换算法	156
8.5.3	交换区	159
8.6	页框分配策略	160
8.6.1	物理主存	160
8.6.2	空闲页面链表	160
8.6.3	页架分配中的有关策略	160
8.6.4	分页环境中程序的行为特性	163
8.7	UNIX SVR4 的存储管理	164
	本章小结	166
	习题	167
第9章	设备管理	169
9.1	概述	169
9.2	I/O 子系统的层次模型	170
9.2.1	I/O 子系统的设计目标	170
9.2.2	I/O 子系统的层次模型	171
9.3	I/O 硬件组成	172
9.3.1	设备和设备控制器	172
9.3.2	直接存储器访问	174
9.3.3	通道方式与输入输出处理器	175
9.4	设备驱动程序	176
9.4.1	设备和驱动程序分类	176
9.4.2	设备开关表	176
9.4.3	设备驱动程序框架	178
9.5	I/O 子系统	179
9.5.1	设备命名	180
9.5.2	输入输出缓冲区	181
9.5.3	I/O 子系统独立于设备的工作	183
9.6	流*	184
9.6.1	流的概念	184
9.6.2	消息和队列	186
9.6.3	流 I/O	188
9.7	磁盘调度	189
9.7.1	磁盘的硬件特性	189
9.7.2	磁盘调度算法	191
9.8	虚拟设备和 SPOOLing 系统	195
9.9	RAID 技术*	196
	本章小结	198



习题.....	199
第 10 章 文件系统	201
10.1 文件	202
10.1.1 文件的命名	202
10.1.2 文件的结构	202
10.1.3 文件的类型	204
10.1.4 文件的属性	205
10.1.5 文件的操作	206
10.2 目录	207
10.2.1 目录内容	207
10.2.2 文件目录的结构	208
10.2.3 路径名	211
10.2.4 符号链接	212
10.2.5 目录操作	213
10.3 文件系统的实现	213
10.3.1 文件空间的分配和管理	213
10.3.2 UNIX 系统的目录实现	217
10.3.3 磁盘空间管理	217
10.3.4 文件系统在主存的数据结构和打开操作	219
10.3.5 文件系统挂载	220
10.4 安全性和保护	221
10.4.1 用户确认技术	222
10.4.2 保护机制——数据安全性	223
10.4.3 其他	224
10.4.4 文件的转储和恢复	225
本章小结	226
习题	226
第 11 章 分布式系统*	228
11.1 概述	228
11.1.1 什么是分布式系统	228
11.1.2 分布式系统的优点	229
11.2 进程通信	230
11.2.1 进程通信的概念	230
11.2.2 TCP/IP 通信协议	231
11.2.3 分布式环境的客户/服务器模式	234
11.2.4 分布式进程通信	236

11.3	分布式文件系统	240
11.3.1	分布式文件系统概述	240
11.3.2	分布式文件系统的组成	241
11.3.3	分布式文件系统的体系结构	242
11.3.4	客户机高速缓存和一致性	244
11.4	分布式系统中的互斥与死锁	245
11.4.1	逻辑钟和逻辑时	246
11.4.2	时间戳算法(Lamport 算法)	247
11.4.3	令牌传送算法	248
11.5	进程迁移	249
11.5.1	进程迁移的原因	249
11.5.2	进程迁移机制	249
	本章小结	250
	习题	250
第 12 章	Windows NT 操作系统*	252
12.1	Windows NT 操作系统概述	252
12.2	Windows NT 的系统模型	253
12.3	Windows NT 的基元成分	256
12.3.1	对象	256
12.3.2	进程	258
12.3.3	线程	260
12.3.4	进程管理程序	262
12.4	Windows NT 的线程状态及调度	263
12.4.1	线程状态转换	263
12.4.2	内核调度程序	264
12.4.3	进程和线程的优先级	265
12.5	Windows NT 的同步对象	266
12.5.1	线程同步概述	266
12.5.2	用 Windows NT 对象同步	267
12.6	虚拟存储管理	269
12.6.1	进程的虚拟地址空间	269
12.6.2	虚拟分页	270
12.6.3	页面调度策略和工作集	273
12.6.4	页架状态和页架数据结构	274
12.6.5	主存映射文件和视图	275
12.7	输入输出系统	278
12.7.1	输入输出系统的结构	278



目 录

12.7.2	统一的驱动程序模型	279
12.7.3	异步 I/O 操作和 I/O 请求处理过程	280
12.7.4	映像文件 I/O	280
12.8	Windows NT 的内装网络	281
12.8.1	Windows NT 内装网络的特色	281
12.8.2	Windows NT 内装网络的体系结构	281
12.9	对象管理程序	282
12.10	本地过程调用	283
12.11	Windows NT 的安全性	284
12.11.1	Windows NT 的安全性	284
12.11.2	存取令牌和安全描述体	284
	本章小结	285
	习题	286
第 13 章	Linux 操作系统*	287
13.1	Linux 操作系统概述	287
13.1.1	Linux 的诞生和发展	287
13.1.2	Linux 的版本	287
13.1.3	Linux 内核的组成	288
13.1.4	Linux 的特点	290
13.2	Linux 进程管理	291
13.2.1	Linux 进程概述	291
13.2.2	Linux 进程的状态	292
13.2.3	Linux 进程的标识	293
13.2.4	Linux 进程的调度	294
13.2.5	Linux 进程的创建和撤销	295
13.3	Linux 的存储管理	296
13.3.1	Linux 的虚拟存储空间	296
13.3.2	Linux 的地址映射	297
13.3.3	Linux 物理内存的管理	298
13.3.4	页面分配算法	298
13.3.5	缺页中断	300
13.3.6	缓存和刷新机制	301
13.4	Linux 文件管理	301
13.4.1	Linux 虚拟文件系统概述	301
13.4.2	VFS 超级块	302
13.4.3	VFS 索引节点对象	303
13.4.4	目录项对象	307



13.4.5	文件对象	308
13.5	EXT 文件系统	308
13.5.1	EXT2/EXT3/EXT4 文件系统的特性	309
13.5.2	EXT2 文件系统的磁盘结构	310
13.5.3	EXT2 超级块	311
13.5.4	组描述符	312
13.5.5	块位图	313
13.5.6	EXT2 文件系统 inode 结构	313
13.5.7	Linux 文件系统的控制	316
13.6	Linux 设备管理	319
13.6.1	Linux 设备管理概述	319
13.6.2	Linux I/O 子系统的设计	320
13.6.3	Linux 的字符设备管理	321
13.6.4	Linux 的块设备管理	322
13.6.5	缓冲区与 buffer 结构	323
	本章小结	324
	习题	325
	参考文献	326

第 1 章 引 论

1.1 计算机系统概述

一个完整的计算机系统,均由两大部分组成,即计算机的硬件部分和计算机的软件部分。通常的硬件部分是指计算机的各种处理器(如中央处理器、输入输出处理器和包含在该计算机系统中的其他处理器)、存储器、输入输出设备和通信装置。计算机软件部分是指计算机系统中安装的所有软件。包括由计算机硬件执行的、可以完成一定任务的程序及其数据和相关文档。如各种语言的编译程序、解释程序、汇编程序、连接和装入程序、用户应用程序、数据库管理系统、数据通信系统和操作系统。那么,计算机系统的各硬件部分是如何连接以构成一个完整的计算机,各软件部分的关系如何,硬件和软件之间又是怎么的关系呢?

1.1.1 计算机的硬件组织

计算机的基本组成如图 1.1 所示,通常也称为冯·诺依曼结构,它由输入设备、输出设备、运算器、控制器、存储器 5 部分组成。最初的冯·诺依曼结构的计算机是以运算器为中心的。随着计算机体系结构的不断发展,已逐渐演变到现在的以存储器为中心的结构。

计算机各硬件部件之间通过总线相连。数据和指令通过总线传送。图 1.2 是以存储器为中心的双总线结构。

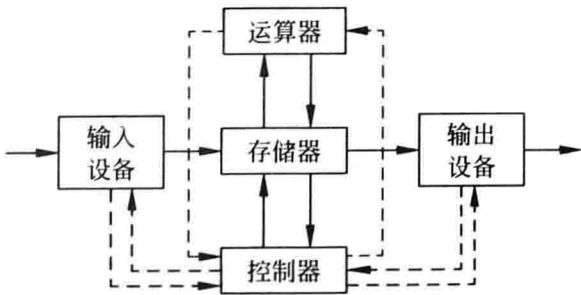


图 1.1 计算机的基本组成

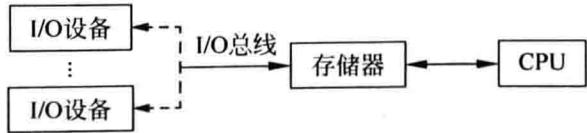


图 1.2 以存储器为中心的双总线结构

迄今为止,计算机的发展已经历了以下几代。第一代是电子管计算机,第二代是晶体管计算机,第三代是集成电路计算机,第四代是大规模集成电路计算机,第五代计算机是把信

息采集、存储、处理、通信同人工智能结合在一起的智能计算机系统。除了能进行数值计算或处理一般的信息外,主要是面向知识处理,具有形式化推理、联想、学习和解释的能力,能够帮助人们进行判断、决策、开拓未知领域和获得新的知识。在计算机发展过程中,计算机性能指标的提高,总是伴随着系统结构的发展而发展的。早期,由于硬件价格过高,设计硬件时,往往将许多工作留给软件去做,这导致了软件的复杂、价格昂贵,以致引起了软件危机。现在硬件已经非常便宜,所以计算机结构的发展,在增强功能的前提下,更关注其处理器的效率,因而并行性成为推动计算机系统结构发展的重要因素之一,即使原有部件尽可能并行运行。改进后的冯·诺依曼计算机从原来的以运算器为中心的结构演变为以存储器为中心。从系统结构上讲,主要是通过各种并行处理手段提高计算机系统性能。这方面一个重要进展就是通道的出现。通道实际上是一个专用的输入输出(I/O)处理器,由它负责和控制 I/O 设备的工作,从而较多地节省了 CPU 时间,提高了 CPU 效率。图 1.3 是一个典型的以存储器为中心的、具有通道的双总线结构。

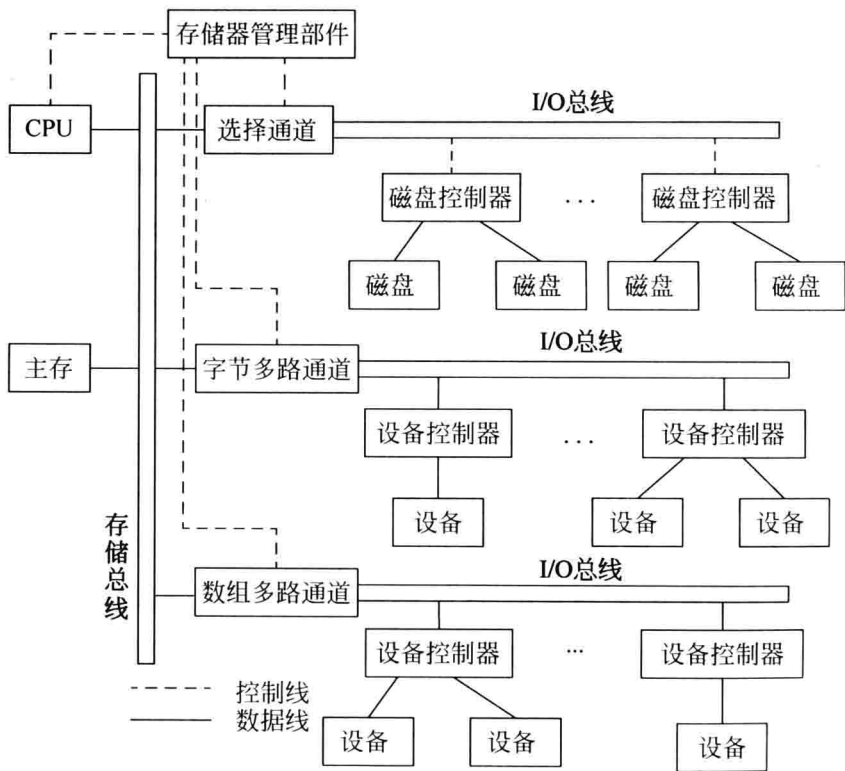


图 1.3 IBM 4300 系统结构

并行处理往往是把一件工作分为若干个相互联系的部分(如把指令的执行工作分解为取指、译码、取数、运算、存结果等),每一部分由专门的部件处理,然后按流水线原则把各部分的执行并行化。如果各部件处理进一步由处理器完成,就构成流水线多处理器系统(如果为编译过程的扫描、分析、代码生成等部分都设立专门的处理器,并与执行机器语言的通用处理器相连,就构成了高级语言处理器)。

20 世纪 80 年代,微型机得到了很大发展,最简单的微型计算机的典型结构是如图 1.4 所示的单总线结构,它是一种以总线为纽带构成的计算机系统。微处理器和存储器、存储器和输入输出设备,以及微处理器和输入输出设备之间都要经过总线来交换信息。无论哪个

设备,如果需要使用总线与另一个设备交换信息,就必须先请求总线使用权,在获得总线使用权后才能进行通信。在通信双方使用总线期间,其他设备不能插入总线操作。这是其特点之一。其次,数据流的路线也有其特点,这主要表现在微处理器与输入输出设备交换数据时的两种不同的路线。当微处理器与慢速的输入输出设备(如打印机或者终端等设备)交换数据时是不经过存储器的,而是直接从(或向)输入输出设备接口(控制器)中的数据寄存器中读(或写)。当微处理器与高速的输入输出设备(如磁盘)交换数据时,这些输入输出设备在控制器的控制下首先将数据(通常是一组数据)送往存储器(或从存储器取数据),也就是说微处理器与高速输入输出设备交换数据时,必须经由存储器。这样两种不同的数据交换路线是由微型计算机的组织结构所决定的。

如图 1.4 所示的单处理器的系统通常用于比较简单的环境。目前许多微型计算机系统已经具有很复杂的、功能很强的处理能力,在这种情况下的微型计算机系统是多处理器的操作方式。

综上所述,无论是大型和小型计算机,以及功能强大的微型计算机都向着多机系统发展。这主要是由于信息爆炸时代,单 CPU 计算机系统已远远不能满足超高速、大容量、高可靠性等方面的要求。

多处理器系统指包含两个以上 CPU 的整体计算机系统,Enslow 把它定义为具有两个或者两个以上 CPU、共享存储器、I/O 通道和 I/O 设备,并由一个操作系统控制的计算机系统。多处理器系统类型很多,图 1.5 是一个对称多处理器系统的结构示意图。

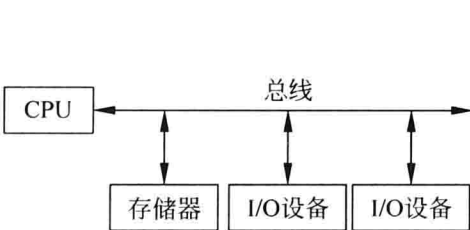


图 1.4 微型计算机的典型组织

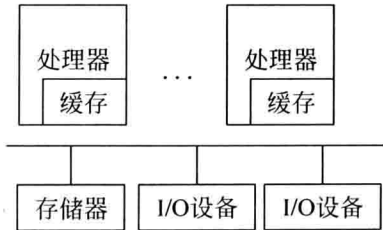


图 1.5 多机系统的结构

随着现代计算技术的发展,并行处理、超前处理、流水线技术等已经打破了单一中央处理器完成所有计算机操作的局面。处理器发展至今,时钟频率已经接近现有生产工艺的极限,通过提高频率提升处理器性能基本走到了尽头。为提升运行效率,出现了多核处理器。多核处理器技术是 CPU 设计中的一项先进技术。它把两个以上的处理器核集成在一块芯片上,以增强计算性能。通过在多个 CPU 核上分配工作负荷,并且依靠内存和输入输出的高速片上互联和高带宽管道对系统性能进行提升。多核处理器,较之单核处理器,能带来更多的性能,因而正成为一种广泛普及的计算模式。

1.1.2 软件的层次与虚拟机的概念

一个计算机系统除了硬件部分以外还有许多软件系统。一般将软件系统分为两大类,即系统软件和应用软件。系统软件用于计算机资源的管理、维护、控制和运行,以及对运行的程序进行翻译、装入、多媒体服务、网络通信等服务工作。系统软件一般包括操作系统、语