



北京市高等教育精品教材立项项目

高等学校计算机科学与技术专业教材

操作系统

Operating System

孟庆昌 编著



北京市高等教育精品教材立项项目

高等学校计算机科学与技术专业教材

操作系统

Operating System

孟庆昌 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本教材理论与实际相结合,全面系统地介绍现代操作系统的基本理论和最新技术。全书共分14章:第1章概述操作系统的定义、功能、特征、发展历程和结构;第2章至第7章分别讲述进程和线程管理、死锁、调度、存储管理、文件管理和输入/输出管理;第8章讲述中断和信号机制,通过示例简单介绍操作系统的协调工作;第9章介绍网络操作系统;第10章讲述分布式操作系统;第11章讲述系统的安全与保护;第12章至第14章分别介绍UNIX, Linux 和 Windows 2000 三个常用操作系统的实现技术。附录中提供习题参考答案,并为教师提供教学资源。

本书是北京市高等教育精品教材建设重点项目,可作为大学本科及专科计算机专业教材或考研参考书,也可作为计算机工作者的自学用书。

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

操作系统/孟庆昌编著. —北京:电子工业出版社, 2004.5
高等学校计算机科学与技术专业教材
ISBN 7-5053-9899-7

I. 操… II. 孟… III. 操作系统—高等学校—教材 IV. TP316
中国版本图书馆CIP数据核字(2004)第040152号

策划编辑:章海涛

责任编辑:童占梅

印 刷:北京牛山世兴印刷厂

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

经 销:各地新华书店

开 本:787×980 1/16 印张:29 字数:688千字

印 次:2004年5月第1次印刷

印 数:5000册 定价:34.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

本书是北京市教育委员会评审并确认的高等教育精品教材建设重点项目之一。本书的出版是市教委指导和支持的结果，是众多专家和同仁大力帮助的结果，是与电子工业出版社多年合作的结果，是作者辛勤劳作的结晶。

作者根据多年教学和科研的经验与体会，并汲取国内外操作系统方面优秀教材的精华，结合我国国情，本着提高学生素质、培养创新意识的精神，遵循本科教学大纲的要求，兼顾考研的需要，力求编写一本在国内有较大影响的精品教材。

目前，用于普通高校本科生的操作系统教材较多，各有特色。

本教材采用理论与实际相结合的方式，既讲述传统和现代理论，又介绍最新开发应用技术，以求达到学以致用的目的。具体来说，本教材注意了以下4个方面。

① 传统经典理论与现代最新技术相结合，便于学习上循序渐进，不割断历史，又着眼未来发展，考虑到与今后研究生课程的衔接及实际工作中的应用；既介绍作业、进程等方面的传统理论，又讲解线程、分布式操作系统、系统安全与保护等最新概念和技术。

② 常用操作系统实例介绍与Linux系统的重点分析。不仅对UNIX, Windows 2000的结构及特点进行简要介绍，而且分析介绍了Linux的结构与算法，以期学生能尝试开发操作系统。

③ 理论要系统，但不是“大全”；知识要先进，又不同于研究生教材；概念要准确，内容讲述又不能呈现简单“条文”式；突出基本点和要点，便于自学和网上教学；篇幅适中，以便于教与学。

④ 体现“以学生和学习为中心”的新理念。在讲授方法上注意由浅入深，由表及里；先引出问题，再给出概念、实现技术和典型算法，然后予以适当总结。

全书共分14章：第1章概述操作系统的定义、功能、特征、发展历程和结构；第2章至第7章分别讲述进程和线程管理、死锁、调度、存储管理、文件管理和输入/输出管理；第8章讲述中断和信号机制，通过示例简单介绍操作系统的协调工作；第9章介绍网络操作系统；第10章讲述分布式操作系统；第11章讲述系统的安全与保护；第12章至第14章分别介绍UNIX, Linux和Windows 2000三个常用操作系统的实现技术。

由于各学校课程设置、学时安排及学生程度等方面存在差异，所以在应用本教材授课时，可以对内容酌情进行取舍。如果课时较充分（如70个学时左右），可以讲授全部内容。如果课时较少（如50个学时左右），可对各章内容重点讲解，突出基本内容，对于较深入的算法和具体实现可以略讲或不讲，由学生自学。本书附录中给出了各章的习题参考答案，其出发点是为了便于自学自测，提高教学效果。请读者正确使用这部分内容，自觉地学习，避免对它的依赖性。为便于广大教师授课，本书将配套提供电子教案。可通过登录电子工业出版社教材资源网站<http://edu.phei.com.cn>或直接联系教材服务部010-68152204索取。

本书可作为大学本科及专科计算机软件和计算机应用专业学生的教科书或考研参考书，以及计算机工作者的自学用书。

本书主要由孟庆昌编写，参加编写、整理、录入工作的还有刘振英、牛欣源、孟欣、张震、孟平、王友兰等。

由于编者水平有限，时间又很紧，对广大读者的需求尚缺乏广泛深入的了解，书中难免存在不妥甚至错误之处，恳请广大读者批评指正，并及时反馈用书信息。

作 者

于北京信息工程学院

目 录

第 1 章 操作系统引论	(1)	习题 1	(32)
1.1 计算机系统概述	(1)	第 2 章 进程和线程	(35)
1.1.1 处理器	(2)	2.1 进程概念	(35)
1.1.2 存储器	(3)	2.1.1 多道程序设计	(35)
1.1.3 I/O 设备	(4)	2.1.2 进程概念	(36)
1.1.4 总线	(5)	2.2 进程的状态和组成	(38)
1.2 什么是操作系统	(6)	2.2.1 进程的状态及其转换	(38)
1.2.1 操作系统概念	(6)	2.2.2 进程描述	(41)
1.2.2 操作系统的主要功能	(7)	2.2.3 进程队列	(42)
1.2.3 操作系统的地位	(10)	2.3 进程管理	(44)
1.2.4 操作系统的服务与服务方式	(11)	2.3.1 进程图	(44)
1.3 操作系统的发展历程	(14)	2.3.2 进程创建	(45)
1.3.1 操作系统的形成	(14)	2.3.3 进程终止	(47)
1.3.2 操作系统的发展	(16)	2.3.4 进程阻塞	(48)
1.3.3 推动操作系统发展的动力	(18)	2.3.5 进程唤醒	(48)
1.4 操作系统的类型	(18)	2.4 线程	(49)
1.4.1 多道批处理系统	(18)	2.4.1 线程概念	(49)
1.4.2 分时系统	(19)	2.4.2 在用户空间实现线程	(52)
1.4.3 实时系统	(21)	2.4.3 在核心空间实现线程	(53)
1.4.4 个人机 (PC) 系统	(22)	2.4.4 组合方式	(54)
1.4.5 多处理器操作系统	(23)	2.4.5 线程池	(55)
1.4.6 嵌入式操作系统	(24)	2.5 进程的同步和通信	(56)
1.4.7 网络操作系统	(25)	2.5.1 进程的同步与互斥	(56)
1.4.8 分布式操作系统	(26)	2.5.2 临界资源和临界区	(58)
1.5 操作系统的特征	(26)	2.5.3 互斥实现方式	(59)
1.6 操作系统的结构	(27)	2.5.4 信号量	(61)
1.6.1 整体系统	(27)	2.5.5 信号量的一般应用	(66)
1.6.2 层次式系统	(28)	2.6 经典进程同步问题	(68)
1.6.3 虚拟机	(29)	2.7 管程	(74)
1.6.4 客户-服务器系统	(30)	2.8 进程通信	(77)
1.7 本章小结	(31)	2.8.1 消息传递系统	(78)

2.8.2 客户-服务器系统中的通信	(82)	4.3 进程调度	(119)
2.9 本章小结	(84)	4.3.1 进程调度的功能	(119)
习题2	(85)	4.3.2 进程调度的时机	(120)
第3章 死锁	(89)	4.3.3 进程调度的基本方式	(120)
3.1 资源	(89)	4.3.4 交互式系统中常用的调度算法	(120)
3.1.1 资源使用模式	(90)	4.3.5 两级调度模型	(121)
3.1.2 可剥夺资源与不可剥夺资源	(90)	4.4 调度准则	(121)
3.2 死锁概念	(91)	4.4.1 影响调度算法选择的主要因素	(121)
3.2.1 什么是死锁	(91)	4.4.2 调度性能评价准则	(122)
3.2.2 死锁的条件	(94)	4.5 调度算法	(123)
3.2.3 资源分配图	(94)	4.5.1 先来先服务法	(123)
3.2.4 处理死锁的方法	(96)	4.5.2 短作业优先法	(124)
3.3 死锁的预防	(96)	4.5.3 最短剩余时间优先法	(126)
3.3.1 破坏互斥条件	(96)	4.5.4 优先级法	(126)
3.3.2 破坏占有且等待条件	(97)	4.5.5 轮转法	(127)
3.3.3 破坏非抢占条件	(97)	4.5.6 多级队列法	(129)
3.3.4 破坏循环等待条件	(98)	4.5.7 多级反馈队列法	(130)
3.4 死锁的避免	(99)	4.5.8 高响应比优先法	(131)
3.4.1 安全状态	(99)	4.5.9 公平共享法	(131)
3.4.2 资源分配图算法	(101)	4.5.10 几种常用调度算法的比较	(132)
3.4.3 银行家算法	(101)	4.6 线程调度	(133)
3.5 死锁的检测和恢复	(104)	4.7 多处理器调度	(134)
3.5.1 对单体资源类的死锁检测	(104)	4.7.1 多处理器系统的类型	(134)
3.5.2 对多体资源类的死锁检测	(105)	4.7.2 多处理器调度方法	(134)
3.5.3 从死锁中恢复	(107)	4.8 实时调度	(136)
3.5.4 “饥饿”状态	(108)	4.8.1 实时任务类型	(136)
3.6 处理死锁的综合方式	(108)	4.8.2 实时调度算法	(137)
3.7 本章小结	(110)	4.9 Linux 系统进程调度	(139)
习题3	(111)	4.9.1 Linux 进程调度	(139)
第4章 调度	(115)	4.9.2 shell 基本工作原理	(141)
4.1 调度类型	(115)	4.10 本章小结	(141)
4.2 作业调度	(117)	习题4	(142)
4.2.1 作业状态	(117)		
4.2.2 作业控制块和作业调度的功能	(117)		

第 5 章 存储管理	(145)	5.9.1 请求分页存储管理的基本思想	(177)
5.1 引言	(145)	5.9.2 硬件支持及缺页处理 ...	(178)
5.1.1 用户程序的主要处理阶段	(145)	5.9.3 请求分页技术的性能 ...	(179)
5.1.2 重定位	(147)	5.10 页面置换算法	(181)
5.2 分区法	(149)	5.10.1 页面置换	(181)
5.2.1 固定分区法	(149)	5.10.2 先进先出法 (FIFO) ...	(182)
5.2.2 动态分区法	(151)	5.10.3 最佳置换法 (OPT) ...	(183)
5.3 可重定位分区分配	(153)	5.10.4 最近最少使用置换法 (LRU)	(184)
5.3.1 碎片问题	(153)	5.10.5 第二次机会置换法 (SCR)	(185)
5.3.2 紧缩	(154)	5.10.6 时钟置换法 (Clock) ...	(186)
5.3.3 动态重定位	(155)	5.10.7 最少使用置换法 (LFU) ...	(187)
5.3.4 可重定位分区法的优缺点	(156)	5.10.8 页面缓冲算法 (Page Buffering)	(188)
5.4 对换技术	(156)	5.11 内存块的分配和抖动问题	(188)
5.5 分页技术	(157)	5.11.1 内存块的分配	(188)
5.5.1 分页存储管理的基本概念	(158)	5.11.2 抖动问题	(191)
5.5.2 分页系统中的地址映射 ...	(159)	5.12 请求分段技术	(194)
5.5.3 页面尺寸	(160)	5.12.1 请求分段存储管理的硬件支持	(194)
5.5.4 硬件支持	(161)	5.12.2 动态链接和链接中断处理	(195)
5.5.5 保护方式	(162)	5.13 Linux 系统的存储管理	(196)
5.5.6 页表的构造	(163)	5.13.1 Linux 的多级页表结构	(196)
5.5.7 页面共享	(167)	5.13.2 内存页的分配与释放 ...	(197)
5.6 分段技术	(168)	5.13.3 内存交换	(198)
5.6.1 分段存储管理的基本概念	(168)	5.14 本章小结	(199)
5.6.2 地址转换	(170)	习题 5	(201)
5.6.3 段的共享和保护	(171)	第 6 章 文件系统	(205)
5.7 段页式技术	(172)	6.1 概述	(205)
5.7.1 段页式存储管理的基本原理	(172)	6.1.1 文件及其分类	(205)
5.7.2 地址转换过程	(173)	6.1.2 文件命名	(208)
5.7.3 段页式技术实例: Intel Pentium 处理器	(173)	6.1.3 文件属性	(208)
5.8 虚拟存储器	(175)	6.1.4 文件存取方法	(209)
5.8.1 虚拟存储器的概念	(175)		
5.8.2 虚拟存储器的特征	(176)		
5.9 请求分页技术	(177)		

6.1.5	文件结构	(211)	7.2.3	设备分配算法	(254)
6.2	文件系统的功能和结构	(212)	7.2.4	SPOOLing 系统	(254)
6.2.1	文件系统的功能	(212)	7.3	I/O 软件层次	(256)
6.2.2	文件系统的结构	(213)	7.3.1	中断处理程序	(256)
6.3	目录结构和目录查询	(214)	7.3.2	设备驱动程序	(256)
6.3.1	文件控制块和文件目录	(214)	7.3.3	与设备无关的操作系统 I/O 软件	(260)
6.3.2	单级目录结构	(215)	7.3.4	用户级 I/O 软件	(263)
6.3.3	二级目录结构	(216)	7.4	磁盘调度和管理	(264)
6.3.4	树形目录结构	(217)	7.4.1	磁盘硬件	(264)
6.3.5	非循环图目录结构	(219)	7.4.2	磁盘调度	(266)
6.3.6	目录查询方法	(220)	7.4.3	磁盘管理	(271)
6.4	文件和目录操作	(221)	7.5	本章小结	(273)
6.4.1	文件操作	(221)	习题 7		(274)
6.4.2	目录操作	(223)	第 8 章 中断和信号机制		(277)
6.5	文件系统的实现	(224)	8.1	中断处理	(277)
6.5.1	文件系统的格式	(224)	8.1.1	中断概述	(277)
6.5.2	文件存储分配	(225)	8.1.2	中断的处理过程	(279)
6.5.3	空闲存储空间的管理	(231)	8.1.3	中断优先级和多重中断	(283)
6.6	管道文件	(234)	8.2	系统调用处理	(286)
6.7	文件系统的可靠性	(236)	8.2.1	陷入事件的处理方式	(286)
6.7.1	磁盘坏块管理	(236)	8.2.2	系统调用的处理方式	(287)
6.7.2	后备	(236)	8.2.3	系统调用实现过程示例	(288)
6.7.3	文件系统和一致性	(238)	8.3	信号机制	(290)
6.8	本章小结	(239)	8.3.1	信号机制概念	(290)
习题 6		(240)	8.3.2	信号的分类、产生和传送	(291)
第 7 章 输入输出管理		(243)	8.3.3	信号的处理方式	(293)
7.1	I/O 管理概述	(243)	8.3.4	信号的检测和处理	(293)
7.1.1	I/O 设备分类和标识	(243)	8.4	本章小结	(294)
7.1.2	I/O 系统结构	(245)	习题 8		(294)
7.1.3	设备控制器	(245)	第 9 章 网络操作系统		(297)
7.1.4	I/O 系统的控制方式	(247)	9.1	网络操作系统概述	(297)
7.1.5	I/O 管理的功能	(251)	9.1.1	计算机网络	(297)
7.2	设备分配	(252)	9.1.2	网络操作系统的功能	(298)
7.2.1	与设备分配相关的因素	(252)	9.1.3	网络操作系统的特性	(299)
7.2.2	设备分配技术	(253)			

9.2 网络操作系统的工作模式及体系结构	11.1.2 安全环境	(300)	(331)
.....	11.2 常见的安全性攻击	(300)	(333)
9.2.1 网络操作系统的工作模式	11.2.1 常见的攻击点	(300)	(333)
9.2.2 网络文件系统体系结构	11.2.2 网络威胁	(300)	(334)
9.3 本章小结	11.2.3 计算机病毒	(302)	(335)
习题 9	11.3 安全对策	(303)	(337)
第 10 章 分布式操作系统	11.3.1 一般性安全机制	(305)	(338)
10.1 分布式操作系统概述	11.3.2 Linux 安全问题及对策	(305)	(339)
10.1.1 分布式系统概述	11.4 保护机制	(305)	(343)
10.1.2 分布式操作系统简介	11.4.1 保护域	(306)	(343)
10.1.3 分布式系统的设计目标	11.4.2 存取控制表	(308)	(345)
10.2 分布式系统的通信	11.4.3 权力	(310)	(347)
10.2.1 网络协议	11.4.4 可信系统	(310)	(348)
10.2.2 客户-服务器模型	11.4.5 安全性能评测标准	(313)	(350)
10.2.3 远程过程调用	11.5 本章小结	(314)	(351)
10.2.4 组通信	习题 11	(314)	(352)
10.3 分布式进程管理	第 12 章 实例研究 1: UNIX	(315)	(353)
10.3.1 进程迁移	12.1 UNIX 历史简介	(315)	(353)
10.3.2 分布式系统中的同步	12.2 UNIX 核心结构	(316)	(355)
10.3.3 互斥	12.3 进程管理	(319)	(356)
10.4 分布式系统中的死锁	12.3.1 进程及其映像	(320)	(356)
10.4.1 死锁的检测	12.3.2 UNIX 进程管理	(321)	(359)
10.4.2 死锁的预防	12.3.3 线程管理中的系统调用	(322)	(361)
10.5 分布式文件系统	12.3.4 进程调度	(322)	(361)
10.5.1 文件服务接口	12.4 文件系统	(323)	(363)
10.5.2 目录服务器接口	12.4.1 UNIX 文件系统结构	(323)	(363)
10.5.3 文件共享语义	12.4.2 文件的打开与关闭	(325)	(366)
10.6 中间件	12.4.3 主要数据结构之间的联系	(326)	(368)
10.6.1 中间件概念	12.4.4 管道文件	(326)	(368)
10.6.2 中间件结构	12.5 存储管理	(327)	(369)
10.7 本章小结	12.5.1 进程的数据结构	(328)	(369)
习题 10	12.5.2 对换	(329)	(370)
第 11 章 安全性与保护机制	12.5.3 请求分页	(331)	(371)
11.1 安全性概述	12.6 I/O 管理	(331)	(374)
11.1.1 信息安全问题	12.6.1 UNIX 系统的缓冲技术	(331)	(375)

12.6.2	块设备管理	(375)	13.6.1	设备管理概述	(396)
12.6.3	字符设备管理	(376)	13.6.2	设备驱动程序和内核之间的接口	(398)
12.7	本章小结	(377)	13.7	网络系统	(400)
第 13 章	实例研究 2: Linux	(379)	13.7.1	socket	(400)
13.1	Linux 系统的历史和特点	(379)	13.7.2	网络分层结构	(401)
13.1.1	Linux 的历史	(379)	13.8	本章小结	(401)
13.1.2	Linux 的特点	(380)	第 14 章	实例研究 3: Windows 2000	(403)
13.1.3	Linux 的版本	(381)	14.1	Windows 2000 的历史和设计原则	(403)
13.2	Linux 体系结构	(381)	14.1.1	Windows 2000 的历史	(403)
13.3	进程管理	(383)	14.1.2	Windows 2000 的设计原则	(404)
13.3.1	进程状态及运行模式	(383)	14.2	Windows 2000 体系结构	(405)
13.3.2	进程的结构	(384)	14.2.1	硬件抽象层	(406)
13.3.3	进程的操作	(385)	14.2.2	内核	(406)
13.3.4	进程的调度	(386)	14.2.3	执行体	(409)
13.4	文件系统	(386)	14.3	环境子系统	(412)
13.4.1	虚拟文件系统	(387)	14.4	虚拟存储管理	(413)
13.4.2	ext2 文件系统	(391)	14.4.1	进程的虚拟地址空间	(413)
13.4.3	管道文件	(393)	14.4.2	虚拟分页地址转换机构	(413)
13.5	存储管理	(394)	14.5	本章小结	(414)
13.5.1	进程的虚存空间	(394)	附录	习题参考答案	(417)
13.5.2	Linux 的多级页表	(394)	参考文献		(454)
13.5.3	物理内存管理	(395)			
13.5.4	页面更换	(395)			
13.6	I/O 管理	(396)			

操作系统引论

本章导读

一个完整的计算机系统是由硬件和软件两大部分组成的。操作系统（Operating System）是所有软件中最基础、最核心的部分，是计算机用户和计算机硬件之间的中介程序，它为用户执行程序提供更方便、更有效的环境。从资源管理的角度来看，操作系统对整个计算机系统内的所有硬件和软件资源进行管理和调度，优化资源的利用，协调系统内的各种活动，处理可能出现的种种问题。

操作系统伴随着计算机技术的飞速发展而经历不同的阶段，从第 1 代的手工系统，到多道程序和分时系统，直到当今的掌上型系统和实时系统。回顾操作系统的发展历程和类型，有助于理解操作系统是什么、干什么和如何干。

与其他软件相比，操作系统是它们运行的基础。考察操作系统的特征可以帮助我们理解其功能和实现。

本章还介绍操作系统的用户界面和体系结构，从外部应用和内部实现两个方面加深对操作系统的理解。

1.1 计算机系统概述

如上所述，计算机系统是由硬件和软件组成的。软件裹在硬件之上。硬件是软件建立与活动的基础，而软件是对硬件功能的扩充。没有硬件，就失去了计算机系统的物理基础，软件也就无法存在了。反过来，若只有硬件而没有软件，就像最初的计算机那样，很难使用，没有活力，也就没有多大应用价值。硬件与软件有机地结合在一起，相辅相成，才使计算机技术飞速发展，且在当今信息时代占据举足轻重的地位。

下面简要考察计算机硬件结构，这些正是操作系统所管理的对象。

现代通用计算机系统是由 CPU，内存和若干 I/O 设备组成。它们经由系统总线连接在一起，实现彼此通信，如图 1-1 所示。每个设备控制器负责对特定类型的设备进行控制和管理，如硬盘控制器用来控制硬盘驱动器，视频控制器用来控制监视器，等等。CPU 和设备控制器可以并行工作，它们都要存取内存中的指令或数据。为保障对共享内存的有序存取，内存控制器对这些访问实施同步管理。

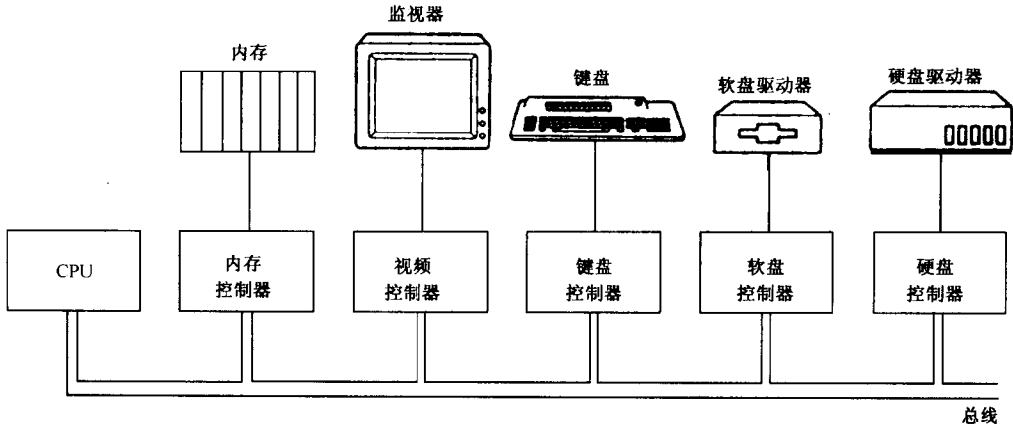


图 1-1 现代计算机硬件结构

1.1.1 处理器

CPU 是计算机的“大脑”。它从内存中提取指令并执行它们。CPU 工作的基本周期是：首先从内存中提取指令，然后对指令译码——确定其类型和操作数，最后执行它。对后面的指令按类似步骤进行处理。

每个 CPU 可以执行的指令集是专用的，所以 Pentium CPU 不能执行 SPARC 程序，反之亦然。由于访问内存、获得指令或数据字所花费的时间要比执行该指令的时间长得多，因而，所有 CPU 都包含某些寄存器，用来存放关键变量和中间结果。除了这些通用寄存器之外，多数计算机都有几个专用寄存器，其中之一是程序计数器，它保存下面要提取的指令的内存地址。当这条指令被提取之后，程序计数器就指向后面的一条指令。所以，改变程序计数器的内容就改变了程序执行的流向。

另一个寄存器存放栈指针，它指向当前内存栈的顶端。该栈中保存有关函数（过程）调用时的现场信息，包括输入参数、局部变量和未在寄存器中保存的临时变量。程序执行过程中调用的每个函数在栈中占有一个帧面。

还有一个专用寄存器保存 PSW（程序状态字），其中包括条件码位（由比较指令设置）、CPU 的优先级、程序执行模式（用户态或者核心态）和各种其他控制位。在系统调用和 I/O 中 PSW 起重要作用。

一般系统都提供核心态和用户态两种处理机执行状态。其目的是为了保护操作系统程序（特别是其内核部分），防止受到用户程序的损害。当执行操作系统程序时，处理机处于核心态（也称做系统态、管理态）。这时，它有较高特权，可以执行所有的指令，包括普通用户程序中不能使用的特权指令，从而能对所有寄存器和内存进行访问，以及启动 I/O 操作等。用户程序是在用户态下执行的，它的权限较低，只能执行指令集中非特权指令。

为了改善性能，当前 CPU 采用的体系结构主要有流水线方式和超级标量方式两种。前者将 CPU 的工作方式分为提取、译码和执行三级，各对应 CPU 的一个工作单元。这样，就有三

条指令可以同时得到 CPU 的处理，像工厂的生产线那样，连续不断地对指令进行加工和执行。

超级标量方式比流水线方式更先进，其中包括多个执行单元、提取单元和译码单元。一次可有多条指令被提取和译码，并且被放入收容缓冲区中。一旦执行单元空闲，就从收容缓冲区中取出指令予以执行。

1.1.2 存储器

在任何计算机中，存储器都是最主要的组成部分之一。按照速度、容量和成本划分，存储系统构成一个层次结构，如图 1-2 所示。

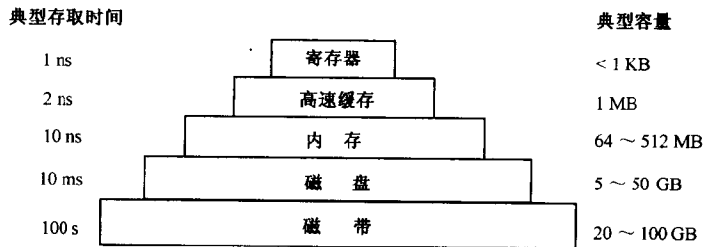


图 1-2 典型的存储器层次结构

顶层是 CPU 内部寄存器，其速度与 CPU 一样快，所以存取它们没有延迟。但它的成本高，容量小，通常都小于 1KB。软件中的程序必须管理寄存器（决定其中存放什么）。

下面一层是高速缓存（Cache），它们大多由硬件控制。Cache 的速度很快，它们放在 CPU 内部或非常靠近 CPU 的地方。当程序需要读取具体信息时，Cache 硬件先查看它是否在 Cache 中，如果在其中（称做“命中”），就直接使用它；如果不在，就从内存中获取该信息，并把它放入 Cache 中，以备今后再次使用。但 Cache 的成本很高，容量较小，一般小于 1MB。统计表明，仔细地选择 Cache 的大小和置换策略，可使命中率达到 80%~99%，从而大大提高存取性能。

再下面一层是内存（或称主存），它是存储器系统的主力，也称做 RAM（随机存取存储器）。CPU 可以直接存取内存及寄存器和 Cache 中的信息，但不能直接存取磁盘上的数据。因此，机器执行的指令及所用的数据必须预先存放在内存及 Cache 和寄存器中。然而，内存中存放的信息是易变的，当机器电源被关闭后，内存中的信息就全部丢失了。

下一层是磁盘（即硬盘），称做辅助存储器（简称辅存或外存），它是对内存的扩展。磁盘驱动器的构造如图 1-3 所示。磁盘上可以永久保留数据，而且容量特别大，现在常用的磁盘容量为 5~60 GB。磁头是可以移动的，由于是机械装置，所以，磁盘上数据的存取速度低于内存存取速度。

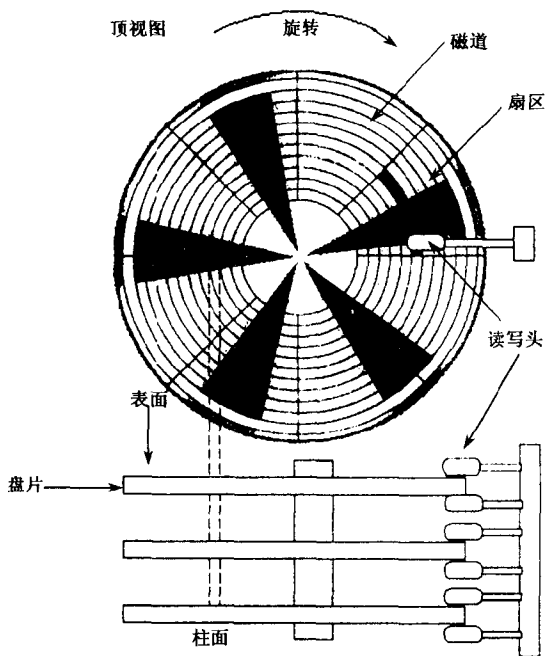


图 1-3 磁盘驱动器的构造

最下层是**磁带**。虽然它上面保存的数据更持久，容量更大，但它的存取速度很慢，而且不适宜进行随机存取。所以，磁带设备一般不能用做辅存。它的主要用途是作为文件系统的后备，存放不常用的信息或用做系统间传送信息的介质。

除了上面介绍的存储器以外，在实际应用中还有其他存储器，如软盘、光盘（CD-ROM）、闪存等。

1.1.3 I/O 设备

I/O 设备是人机交互的工具。它通常由控制器和设备本身两部分组成。

控制器是 I/O 设备的电子部分，它协调和控制一台或多台 I/O 设备的操作，实现设备操作与整个系统操作的同步。在小型机和微型机上，往往以印刷电路卡的形式插入计算机中。很多控制器可以管理 2 台、4 台甚至 8 台同样的设备。设备控制器本身有一些缓冲区和一组专用寄存器，负责在外部设备和本地缓冲区之间移动数据。

设备本身的对外接口相当简单，实际上它们隐藏在控制器的后面。因而，操作系统总是与控制器打交道，而不是与设备直接作用。

设备的种类很多，因而设备控制器的类别就很多，这就需要不同的软件来控制它们。这些向控制器发布命令并接收其回答信息的软件就是**设备驱动程序**。不同操作系统上的不同控制器分别对应不同的设备驱动程序。理论上讲，驱动程序可以在核心之外运行，但当前的系统都把

它放在操作系统中，使之在核心方式下运行。

输入和输出的工作方式主要有程序控制方式，程序中断方式和直接存储器存取方式（DMA 方式）三种。

① **程序控制方式**最简单，输入和输出过程完全由 CPU 控制。其一般过程是：CPU 选定设备，发出读/写命令，启动该设备；然后等待 I/O 完成。在此期间，CPU 不能进行其他工作。这种方式也称“忙式等待”。其缺点是：CPU 没有充分发挥高速处理的能力。

② **程序中断方式**现在得到普遍应用。当启动 I/O 操作时，CPU 加载设备控制器内的某些寄存器，之后 CPU 继续执行其他指令。同时设备控制器检查这些寄存器的内容，以确定执行什么动作。例如，发现读请求，则控制器启动设备，把数据从设备传到它本地的缓冲区中。一旦数据传送完成，设备控制器就通知 CPU——它完成了这个操作。这是通过中断方式实现通信。

一旦 CPU 决定响应中断，就把程序计数器和 PSW 的内容压入当前栈中，CPU 切换到核心态。设备号作为进入内存寻找中断处理程序入口地址的索引。这部分内存中存放的信息一般称做中断向量。中断处理程序完成工作后，又回到刚才终止的用户程序继续执行。

中断在操作系统中非常重要，每当硬件或软件产生事件时（如时钟到时、I/O 完成、执行系统调用等），都会激活中断机制进行相应处理。

③ **DMA 方式**用于高速 I/O 设备，如磁盘、光盘、高速打印机等。这种方式使用一个特殊的 DMA 芯片，它可以控制数据直接在内存和某个控制器之间进行传送，不需 CPU 频繁地进行干预。CPU 对 DMA 芯片进行设置时，需要提供传送的字节数、设备和内存地址，以及传送方向。设置完成以后，DMA 控制器就启动 I/O 操作。当 DMA 控制器执行数据传送时，CPU 可以执行其他任务。当数据传送完成时，DMA 控制器中断 CPU，然后进行如上所述的中断处理过程。

1.1.4 总线

按照在总线上传送的信息所起的作用，系统总线基本上分为如下三部分。

① **数据总线**。这是计算机各部件之间传送数据的通道，其宽度随字长而定。如 32 位结构的数据总线应是 32 根。数据总线是双向总线，即两个方向都能传送数据。

② **地址总线**。从 CPU 送来地址的地址线，它可以是存储器的地址，也可以是 I/O 设备控制器中控制寄存器或数据寄存器的地址。

③ **控制总线**。在该线上出现的信号是各个模块之间传送数据时所需的全部控制信号。

在系统中有多设备要向总线发信号时，在传送数据之前，先要监听总线是否有空闲，空闲时才能占用总线，使用之后要释放总线。

实际系统的总线不止这三条，如 Pentium 系统有 8 条总线，即 Cache 总线，本地总线，内存总线，PCI（Peripheral Component Interconnect，外围部件互连）总线，SCSI（Small Computer System Interface，小型计算机系统接口）总线，USB（Universal Serial Bus）总线，IDE（Integrated Drive Electronics）总线和 ISA（Industry Standard Architecture）总线。这些总线各有不同的传输速率和功能。操作系统必须知道它们的全部信息，以便进行配置和管理。

1.2 什么是操作系统

大家都知道一些操作系统的名称，如 Windows 98, Windows 2000, UNIX, Linux 等，也有使用操作系统的体验，但什么是操作系统呢？对此至今也没有形成一个统一的标准化定义。出现这个问题一方面由于操作系统实现两项相对独立的功能——扩展机器和管理资源，另一方面取决于从什么角度来看待操作系统——用户观点还是系统观点。

1.2.1 操作系统概念

1. 操作系统作为扩展机器

裸机（仅有硬件的计算机）提供的机器语言难记、难用又难懂。设想一下：如果用户所用的机器上没有装操作系统，就无法使用现在已熟悉的 I/O 命令，不能利用鼠标打开或复制文件，甚至连内存的分配和保护、设备的读写驱动都需要用机器指令编程——机器指令不仅晦涩难懂，更麻烦的是必须确切地明白相关硬件设备是如何动作的，对每个具体细节都不能遗漏。很显然，一般程序员是不愿意在这种环境下编程的。

在裸机之上安装操作系统之后，就把硬件细节与程序员隔离开。用户可以使用系统提供的各种命令，直接打开文件、读写文件、更改目录、将文件复制到软盘上，等等。在做这些事情时，我们只关心自己要实现的目标，并未考虑硬件如何动作，从而隐藏了底层硬件的特性。经过操作系统的加工，呈现在用户面前的机器是功能更强、使用更方便的机器。通常把裸机之上覆盖各种软件，从而形成功能更强的机器称为**扩展机器**或**虚拟机**。

这种功能扩展可以重叠。在裸机之上覆盖一层软件后，得到第一层扩展；在此基础上再加一层软件，就得到第二层扩展，依此类推。

2. 操作系统作为资源管理器

操作系统为用户提供方便的接口，使计算机的应用更加容易。这是一种自顶向下的观点。另外一种观点是自底向上的观点，它考察操作系统如何管理一个复杂系统的各个部分。大家知道，现代计算机由处理器、内存、时钟、磁盘、鼠标、网络接口、打印机及各种其他设备组成。操作系统的功能就是管理这些硬件资源和数据、程序等软件资源，控制、协调各个程序对这些资源的利用，尽可能地充分发挥各种资源的作用。

设想一下：当多个用户的程序都想在系统中运行时，如何为它们分配内存？何时调度哪个程序在 CPU 上执行？要打开某个文件时，怎样到磁盘中查找？多个用户都要在同一台打印机上输出计算结果时，如何解决彼此的竞争问题？诸如此类的资源分配、管理、保护以及程序活动的调度、协调种种事项都需要操作系统负责。因此，作为资源管理者，操作系统主要做以下工作：

- ① 监视各种资源，随时记录它们的状态；
- ② 实施某种策略以决定谁获得资源，何时获得，获得多少；
- ③ 分配资源供需求者使用；
- ④ 回收资源，以便再分配。