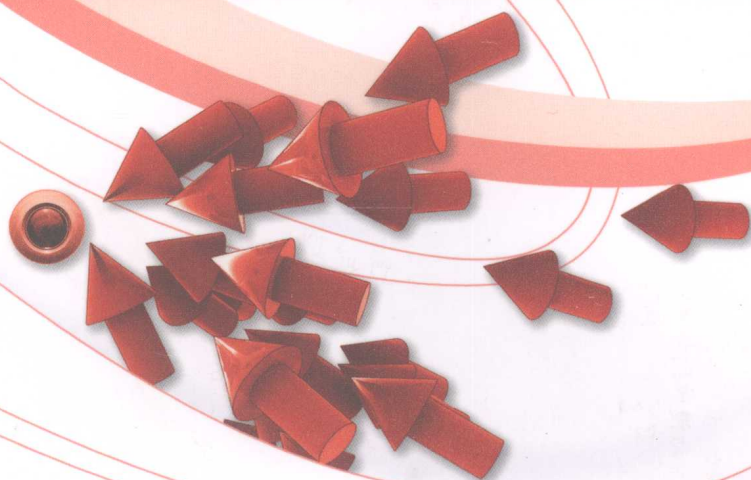




普通高等教育“十一五”国家级规划教材



北京高等教育精品教材
BEIJING GAODENG JIAOYU JINGPIN JIAOCAI



高等学校计算机规划教材

操作系统 (第2版)

■ 孟庆昌 牛欣源 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

普通高等教育“十一五”国家级规划教材
北京高等教育精品教材

高等学校计算机规划教材

操作系统

(第2版)

孟庆昌 牛欣源 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材，北京高等教育精品教材。全书共分14章，分别介绍操作系统引论，进程和线程，死锁，调度，存储管理，文件系统，输入/输出管理，用户接口服务，嵌入式操作系统，分布式操作系统，安全性与保护机制，实例研究1：UNIX，实例研究2：Linux和实例研究3：Windows 2000。附录分别给出实验指导、Linux常用系统调用、部分习题参考答案，并为教师免费提供电子教案。

本书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, 牛欣源编著. —2版. —北京: 电子工业出版社, 2009.11
高等学校计算机规划教材
ISBN 978-7-121-08858-2

I. 操… II. ①孟… ②牛… 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(2009)第077018号

策划编辑: 童占梅

责任编辑: 童占梅

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

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

开 本: 787×1092 1/16 印张: 25.25 字数: 643千字

印 次: 2009年11月第1次印刷

印 数: 4000册 定价: 34.00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 zlt@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

“操作系统”课程是计算机科学与技术及相关专业的必修专业基础课。前一版《操作系统》教材是2002年北京市高等教育精品教材建设重点项目之一，2004年由电子工业出版社出版，2006年评为**北京高等教育精品教材**。

本书是**普通高等教育“十一五”国家级规划教材**，在前一版《操作系统》的基础上认真修订而成。在修订时，我们一方面总结教学中的经验和体会，归纳了众多师生的反馈意见，学习了兄弟院校的教学大纲和教学经验；另一方面，分析了国内外操作系统理论、技术和应用的快速发展的现状，了解了各部门、行业对软件人员（包括毕业生）的需求情况。与前一版相比，本书进行了一系列重大修改，有删有增。主要修改包括以下几方面：

(1) 将前一版第8章“中断和信号机制”改为“用户接口服务”，讲述一般操作系统中为用户提供的三种接口的管理方式及其应用。将中断和信号机制的内容并入第4章。

(2) 将前一版第9章“网络操作系统”改为“嵌入式操作系统”，以适应当代信息技术广泛应用的潮流。

(3) 对前一版的第10、12、13和14各章进行适当压缩，以缩减篇幅。

(4) 在附录A中提供实验指导，以加强学生的实践能力；在附录B中给出Linux常用系统调用，供上机编程时参考。

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8章分别讲述进程和线程管理、死锁、调度、存储管理、文件系统、输入/输出管理和用户接口服务；

第9章介绍嵌入式操作系统；

第10章讲述分布式操作系统；

第11章讲述系统的安全性与保护机制；

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。

三个附录分别给出**实验指导、Linux常用系统调用、部分习题参考答案**，并为教师提供**电子教案**。任课老师可登录华信教育资源网 <http://www.huaxin.edu.cn> **免费注册下载**。

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C中给出了各章的部分习题（涉及重点、难点问题）的参考答案，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，自觉主动地学习，避免对它的依赖性。

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、牛欣源编写，参加编写、整理、录入工作的还有刘振英、孟欣、肖林、袁薇、孟平等。

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

于北京信息科技大学

目 录

第 1 章 操作系统引论	(1)	2.2.1 进程的状态及其转换	(30)
1.1 计算机硬件结构	(1)	2.2.2 进程描述	(32)
1.1.1 处理器	(2)	2.2.3 进程队列	(34)
1.1.2 存储器	(2)	2.3 进程管理	(36)
1.1.3 I/O 设备	(3)	2.3.1 进程图	(36)
1.1.4 总线	(4)	2.3.2 进程创建	(36)
1.2 什么是操作系统	(4)	2.3.3 进程终止	(38)
1.2.1 操作系统概念	(4)	2.3.4 进程阻塞	(39)
1.2.2 操作系统的主要功能	(6)	2.3.5 进程唤醒	(40)
1.2.3 操作系统的地位	(9)	2.4 线程	(40)
1.3 操作系统的发展历程	(10)	2.4.1 线程概念	(40)
1.3.1 操作系统的形成	(10)	2.4.2 线程的实现	(43)
1.3.2 操作系统的发展	(12)	2.5 进程的同步和通信	(44)
1.3.3 推动操作系统发展的动力	(13)	2.5.1 进程的同步与互斥	(44)
1.4 操作系统的类型	(13)	2.5.2 临界资源和临界区	(46)
1.4.1 批处理系统	(13)	2.5.3 互斥实现方式	(47)
1.4.2 分时系统	(14)	2.5.4 信号量	(49)
1.4.3 实时系统	(15)	2.5.5 信号量的一般应用	(53)
1.4.4 网络操作系统	(17)	2.6 经典进程同步问题	(55)
1.4.5 分布式操作系统	(18)	2.7 管程	(61)
1.4.6 其他操作系统	(19)	2.8 进程通信	(63)
1.5 操作系统的特征	(20)	2.8.1 消息传递系统	(63)
1.6 操作系统结构设计	(21)	2.8.2 客户-服务器系统中的 通信	(66)
1.6.1 整体结构	(21)	2.9 本章小结	(67)
1.6.2 层次结构	(22)	习题 2	(68)
1.6.3 虚拟机结构	(22)	第 3 章 死锁	(71)
1.6.4 客户-服务器结构	(24)	3.1 资源	(71)
1.7 本章小结	(25)	3.1.1 资源使用模式	(71)
习题 1	(26)	3.1.2 可剥夺资源与不可剥夺 资源	(72)
第 2 章 进程和线程	(27)	3.2 死锁概念	(72)
2.1 进程概念	(27)	3.2.1 什么是死锁	(73)
2.1.1 多道程序设计	(27)	3.2.2 死锁的条件	(75)
2.1.2 进程概念	(28)	3.2.3 资源分配图	(75)
2.2 进程的状态和组成	(30)		

3.2.4 处理死锁的方法	(77)	4.5.3 最短剩余时间优先法.....	(103)
3.3 死锁的预防	(77)	4.5.4 优先级法.....	(103)
3.3.1 破坏互斥条件	(77)	4.5.5 轮转法.....	(105)
3.3.2 破坏占有且等待条件	(78)	4.5.6 多级队列法.....	(106)
3.3.3 破坏非抢占条件	(78)	4.5.7 多级反馈队列法.....	(107)
3.3.4 破坏循环等待条件	(79)	4.5.8 高响应比优先法.....	(108)
3.4 死锁的避免	(79)	4.5.9 公平共享法.....	(109)
3.4.1 安全状态	(80)	4.5.10 几种常用调度算法的 比较	(109)
3.4.2 资源分配图算法	(81)	4.6 线程调度	(109)
3.4.3 银行家算法	(82)	4.7 多处理器调度	(110)
3.5 死锁的检测和恢复	(84)	4.7.1 多处理器系统的类型.....	(110)
3.5.1 对单体资源类的死锁检测	(84)	4.7.2 多处理器调度方法.....	(111)
3.5.2 对多体资源类的死锁检测	(85)	4.8 实时调度	(112)
3.5.3 从死锁中恢复	(86)	4.8.1 实时任务类型.....	(112)
3.5.4 “饥饿”状态	(88)	4.8.2 实时调度算法.....	(112)
3.6 处理死锁的综合方式	(88)	4.9 UNIX/Linux 进程调度.....	(113)
3.7 本章小结	(90)	4.9.1 UNIX 进程调度.....	(113)
习题 3	(91)	4.9.2 Linux 进程调度.....	(115)
第 4 章 调度	(93)	4.10 中断处理	(116)
4.1 调度类型	(93)	4.10.1 中断概述.....	(116)
4.2 作业调度	(95)	4.10.2 中断的处理过程.....	(118)
4.2.1 作业状态	(95)	4.10.3 中断优先级和多重中断	(121)
4.2.2 作业控制块和作业调度的 功能	(95)	4.11 信号机制	(124)
4.3 进程调度	(97)	4.11.1 信号机制概念.....	(124)
4.3.1 进程调度的功能	(97)	4.11.2 信号的分类、产生和 传送	(125)
4.3.2 进程调度的时机	(97)	4.11.3 信号的处理方式.....	(126)
4.3.3 进程调度的基本方式	(98)	4.11.4 信号的检测和处理.....	(127)
4.3.4 交互式系统中常用的调度 算法	(98)	4.12 本章小结	(128)
4.3.5 两级调度模型	(98)	习题 4	(129)
4.4 调度准则	(99)	第 5 章 存储管理	(131)
4.4.1 影响调度算法选择的主要 因素	(99)	5.1 引言	(131)
4.4.2 调度性能评价准则	(100)	5.1.1 用户程序的地址空间.....	(131)
4.5 调度算法	(101)	5.1.2 重定位.....	(133)
4.5.1 先来先服务法	(101)	5.1.3 对换技术.....	(134)
4.5.2 短作业优先法	(102)	5.2 分区法	(135)
		5.2.1 固定分区法.....	(136)

5.2.2	动态分区法	(137)	5.9.2	抖动问题	(171)
5.2.3	可重定位分区分配	(140)	5.10	请求分段技术	(174)
5.3	分页技术	(142)	5.10.1	请求分段存储管理的硬件支持	(174)
5.3.1	分页存储管理的基本概念	(142)	5.10.2	动态链接和链接中断处理	(174)
5.3.2	分页系统中的地址映射	(144)	5.11	Linux 系统的存储管理	(176)
5.3.3	页面尺寸	(145)	5.11.1	Linux 的多级页表结构	(176)
5.3.4	硬件支持	(146)	5.11.2	内存页的分配与释放	(177)
5.3.5	保护方式	(147)	5.11.3	内存交换	(178)
5.3.6	页表的构造	(148)	5.12	本章小结	(179)
5.3.7	页面共享	(151)	习题 5	(180)	
5.4	分段技术	(152)	第 6 章 文件系统	(183)	
5.4.1	分段存储管理的基本概念	(152)	6.1	概述	(183)
5.4.2	地址转换	(154)	6.1.1	文件及其分类	(183)
5.4.3	段的共享和保护	(155)	6.1.2	文件命名	(186)
5.5	段页式技术	(156)	6.1.3	文件属性	(186)
5.5.1	段页式存储管理的基本原理	(156)	6.1.4	文件存取方法	(187)
5.5.2	地址转换过程	(157)	6.1.5	文件结构	(189)
5.6	虚拟存储器	(157)	6.2	文件系统的功能和结构	(190)
5.6.1	虚拟存储器的概念	(157)	6.2.1	文件系统的功能	(190)
5.6.2	虚拟存储器的特征	(159)	6.2.2	文件系统的结构	(191)
5.7	请求分页技术	(159)	6.3	目录结构和目录查询	(191)
5.7.1	请求分页存储管理的基本思想	(159)	6.3.1	文件控制块和文件目录	(192)
5.7.2	硬件支持及缺页处理	(160)	6.3.2	单级目录结构	(193)
5.7.3	请求分页技术的性能	(161)	6.3.3	二级目录结构	(194)
5.8	页面置换算法	(163)	6.3.4	树形目录结构	(195)
5.8.1	页面置换	(163)	6.3.5	非循环图目录结构	(196)
5.8.2	先进先出法	(164)	6.3.6	目录查询方法	(197)
5.8.3	最佳置换法	(165)	6.4	文件和目录操作	(198)
5.8.4	最近最少使用置换法	(166)	6.4.1	文件操作	(198)
5.8.5	第二次机会置换法	(167)	6.4.2	目录操作	(200)
5.8.6	时钟置换法	(167)	6.5	文件系统的实现	(201)
5.8.7	最少使用置换法	(168)	6.5.1	文件系统的格式	(201)
5.8.8	页面缓冲算法	(168)	6.5.2	文件存储分配	(203)
5.9	内存块的分配和抖动问题	(169)	6.5.3	空闲存储空间的管理	(208)
5.9.1	内存块的分配	(169)	6.6	管道文件	(211)
			6.7	文件系统的可靠性	(212)

6.7.1 磁盘坏块管理	(212)	8.3.3 shell 程序设计	(257)
6.7.2 后备	(213)	8.4 图形用户界面	(258)
6.7.3 文件系统和一致性	(214)	8.4.1 图形界面简介	(259)
6.8 本章小结	(216)	8.4.2 X Window 系统	(260)
习题 6	(217)	8.5 本章小结	(264)
第 7 章 输入/输出管理	(219)	习题 8	(265)
7.1 I/O 管理概述	(219)	第 9 章 嵌入式操作系统	(266)
7.1.1 I/O 设备分类和标识	(219)	9.1 嵌入式系统概述	(266)
7.1.2 I/O 系统结构	(220)	9.2 嵌入式操作系统概述	(267)
7.1.3 设备控制器	(221)	9.2.1 嵌入式软件系统的体系 结构	(267)
7.1.4 I/O 系统的控制方式	(222)	9.2.2 嵌入式操作系统	(267)
7.1.5 I/O 管理的功能	(226)	9.3 实时内核及其实现	(269)
7.2 设备分配	(228)	9.3.1 任务管理与调度	(269)
7.2.1 与设备分配相关的因素	(228)	9.3.2 中断和时间管理	(271)
7.2.2 设备分配技术	(228)	9.3.3 任务的同步和通信	(271)
7.2.3 设备分配算法	(229)	9.3.4 内存管理	(274)
7.2.4 SPOOLing 系统	(230)	9.3.5 I/O 管理	(276)
7.3 I/O 软件层次	(231)	9.4 实例简介—— μ CLinux	(276)
7.3.1 中断处理程序	(231)	9.5 本章小结	(279)
7.3.2 设备驱动程序	(231)	习题 9	(279)
7.3.3 与设备无关的操作系统 I/O 软件	(235)	第 10 章 分布式操作系统	(280)
7.3.4 用户级 I/O 软件	(238)	10.1 分布式系统概述	(280)
7.4 磁盘调度和管理	(238)	10.1.1 分布式系统特征	(280)
7.4.1 磁盘调度	(239)	10.1.2 分布式系统的优点	(281)
7.4.2 磁盘管理	(242)	10.2 分布式操作系统概述	(281)
7.5 本章小结	(245)	10.2.1 分布式操作系统简介	(281)
习题 7	(245)	10.2.2 4 种多机系统的比较	(282)
第 8 章 用户接口服务	(247)	10.2.3 分布式系统的设计目标	(283)
8.1 用户接口的发展	(247)	10.3 分布式系统的实现	(285)
8.2 系统调用	(248)	10.3.1 通信问题	(285)
8.2.1 系统调用和库函数	(248)	10.3.2 进程管理	(286)
8.2.2 系统调用使用方式	(249)	10.3.3 死锁问题	(288)
8.2.3 系统调用的处理方式	(251)	10.3.4 文件系统	(288)
8.3 命令行接口	(254)	10.3.5 中间件	(290)
8.3.1 命令的一般使用方式	(255)	10.4 本章小结	(292)
8.3.2 命令解释程序	(256)	习题 10	(293)

第 11 章 安全性与保护机制	(294)	12.6.1 UNIX 系统的缓冲技术	(321)
11.1 安全性概述	(294)	12.6.2 块设备管理	(322)
11.1.1 信息安全问题	(294)	12.6.3 字符设备管理	(323)
11.1.2 安全环境	(294)	12.7 本章小结	(324)
11.2 常见的安全性攻击	(296)	第 13 章 实例研究 2:Linux	(325)
11.2.1 常见的攻击点	(296)	13.1 Linux 系统的特点	(325)
11.2.2 网络威胁	(296)	13.2 Linux 体系结构	(325)
11.2.3 计算机病毒	(297)	13.3 进程管理	(326)
11.3 一般性安全机制	(300)	13.3.1 进程状态及运行模式	(326)
11.3.1 安全措施	(300)	13.3.2 进程的结构	(327)
11.3.2 一般性安全机制	(300)	13.3.3 进程的操作	(327)
11.4 保护机制	(302)	13.3.4 进程的调度	(328)
11.4.1 保护域	(302)	13.4 文件系统	(328)
11.4.2 存取控制表	(304)	13.4.1 虚拟文件系统	(328)
11.4.3 权力	(305)	13.4.2 ext2 文件系统	(331)
11.4.4 可信系统	(306)	13.4.3 管道文件	(332)
11.4.5 安全性能评测标准	(308)	13.5 I/O 管理	(333)
11.5 本章小结	(309)	13.5.1 设备管理概述	(333)
习题 11	(309)	13.5.2 设备驱动程序和内核之 间的接口	(334)
第 12 章 实例研究 1:UNIX	(310)	13.6 网络系统	(336)
12.1 UNIX 历史简介	(310)	13.6.1 socket	(336)
12.2 UNIX 核心结构	(311)	13.6.2 网络分层结构	(337)
12.3 进程管理	(311)	13.7 本章小结	(337)
12.3.1 进程及其映像	(311)	第 14 章 实例研究 3:Windows 2000	(339)
12.3.2 UNIX 进程管理	(314)	14.1 Windows 2000 的历史和设计 原则	(339)
12.3.3 线程管理中的系统调用	(314)	14.1.1 Windows 2000 的历史	(339)
12.3.4 进程调度	(315)	14.1.2 Windows 2000 的设计 原则	(339)
12.4 文件系统	(315)	14.2 Windows 2000 体系结构	(340)
12.4.1 UNIX 文件系统结构	(315)	14.2.1 硬件抽象层	(341)
12.4.2 文件的打开与关闭	(316)	14.2.2 内核	(341)
12.4.3 主要数据结构之间的 联系	(317)	14.2.3 执行体	(344)
12.4.4 管道文件	(317)	14.3 环境子系统	(346)
12.5 存储管理	(318)	14.4 虚拟存储管理	(347)
12.5.1 进程的数据结构	(318)	14.4.1 进程的虚拟地址空间	(347)
12.5.2 对换	(318)		
12.5.3 请求分页	(319)		
12.6 I/O 管理	(321)		

14.4.2 虚拟分页地址转换机构	(348)	B.2 有关进程控制的系统调用	(365)
14.5 本章小结	(348)	B.3 有关进程通信的函数	(366)
附录 A 实验指导	(350)	B.4 有关内存管理的函数	(368)
实验一 进程同步和互斥	(350)	附录 C 部分习题参考答案	(370)
实验二 进程及其资源管理	(353)	第 1 章	(370)
实验三 存储管理	(357)	第 2 章	(371)
实验四 页面置换算法	(358)	第 3 章	(378)
实验五 进程调度	(359)	第 4 章	(380)
实验六 银行家算法	(360)	第 5 章	(383)
实验七 磁盘调度算法	(361)	第 6 章	(384)
实验八 设备处理程序设计	(361)	第 7 章	(386)
实验九 文件系统	(362)	第 8 章	(388)
附录 B Linux 常用系统调用	(364)	第 9 章	(391)
B.1 有关文件操作的系统调用	(364)	第 10 章	(391)
		第 11 章	(392)

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，从第一代的手工系统，到多道程序和分时系统，直到当今的掌上型系统和网络系统。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体系结构，从早期的整体结构，到现代的客户-服务器结构，其内部实现有很大差别，从而加深对操作系统设计的理解。

1.1 计算机硬件结构

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，而软件是对硬件进行管理和功能扩充。**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。

现代计算机体系结构基本上仍沿用 Von Neumann（冯·诺依曼）体系结构，采用存储程序工作原理，即：把计算过程描述为由许多条命令按一定顺序组成的程序，然后把程序和所需的数据一起输入计算机存储器中保存起来，工作时控制器执行程序，控制计算机自动连续进行运算。

大家知道，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。从功能上讲，是由五大功能部件组成，即**运算器、控制器、存储器、输入设备和输出设备**。这五大功能部件相互配合，协同工作。其中，运算器和控制器集成在一片或几片大规模或超大规模集成电路中，称为中央处理器（CPU）。

图 1-1 是现代计算机系统硬件结构的示意图。图中示出的控制器是设备控制器。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。

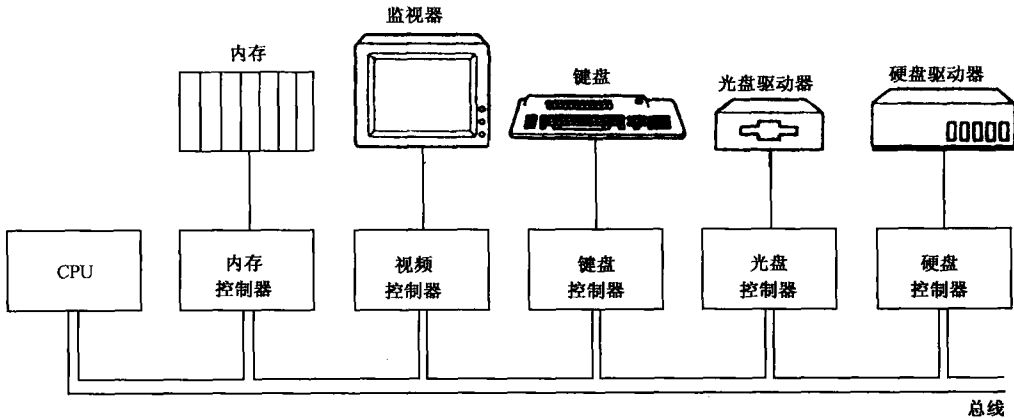


图 1-1 现代计算机硬件结构

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：提取指令，译码分析，执行指令。对后面的指令按类似步骤进行处理。

每种型号 CPU 的指令集都是专用的。另外，CPU 内部都包含若干寄存器，其中，一类是通用寄存器，用来存放关键变量和中间结果；另一类是专用寄存器，如程序计数器（PC）、栈指针寄存器和程序状态字（PSW）。PC 中保存下面要提取指令的内存地址。栈指针寄存器中存放指向当前内存栈的顶端的指针，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。PSW 中包括条件码位、CPU 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器执行状态。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，处理器处于核心态（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。而用户程序（也包括各种应用程序、工具、例程等）是在用户态下执行的，它们的权限较低，只能执行指令集中非特权指令。

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，存储器系统构成一个层次结构，如图 1-2 所示。

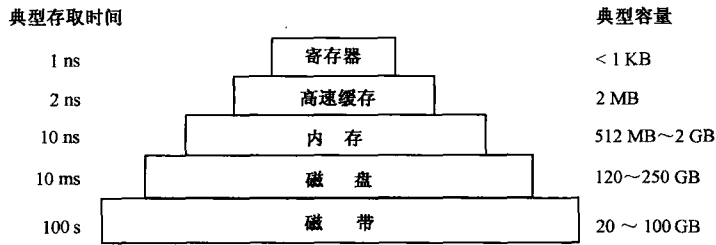


图 1-2 典型的存储器层次结构

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。

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 2MB。

中间一层是**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。

再下一层是**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 120~250 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。

最下层是**磁带**。它记录的数据可以持久保存，而且根据需要可以随时换装新带，故容量很大。但是，它的存取速度很慢，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)、U 盘等。

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，往往以印制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。

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。

在系统中有多多个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。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

大家都用过一种或几种操作系统, 如 Windows XP, Windows Vista, UNIX, Linux 等, 也有应用的体验, 但什么是操作系统呢? 至今尚未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 (即仅有硬件的计算机) 提供的机器语言难记、难用又难懂, 全是“0”、“1”码, 好似“天书”。很显然, 大多数程序员都不愿意在这种环境下编程。

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 U 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要在系统中运行时, 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。

资源管理包含资源复用（或共享），分为时间复用和空间复用两种方式。

时间复用的一个例子是 CPU 的轮流使用，即多道程序在同一个 CPU 上分时运行：第一个程序在 CPU 上运行一小段时间，然后退下来，让第二个程序运行；接着第二个程序退下来，让第三个程序运行，……，轮转一圈后，再让第一个程序继续运行，依此类推，直至程序完成。

另一种复用是空间复用，它不是轮流占用，而是每个客户只占用部分资源。例如，若干程序同时存放在内存中，每个程序只占用部分内存。另外，硬盘也是空间复用的资源。

总之，操作系统确实是计算机系统的资源管理器。当今看待操作系统作用的众多观点中，这种观点仍占主导地位。

3. 操作系统的用户观点和系统观点

从计算机用户的角度来看，操作系统处于用户与计算机硬件系统之间，为用户提供使用计算机系统的接口。因此，操作系统应当使用方便，功能强，效率高，使用安全可靠，易于安装和维护，等等，当然价格应该便宜。这些看法反映了普通用户对操作系统的需求和期望，是从系统外部看待操作系统的作用。

另一种观点是系统观点，从系统内部实现的角度来看待操作系统的作用。操作系统是硬件之上的第一层软件，它要管理计算机系统中各种硬件资源和软件资源的分配问题，如 CPU 时间、内存空间、文件存储空间、I/O 设备，等等，要解决大量对资源请求的冲突问题，决定把资源分配给谁、何时分配、分配多少等，使得资源的利用高效而且公平。这样，操作系统就是资源分配者。

另外，操作系统要对 I/O 设备和用户程序加以控制，保证设备正常工作，防止非法操作，及时诊断设备的故障等。从这个意义上讲，操作系统就是控制程序。

还可以从其他角度来看待操作系统，这里不一一列举。

综上所述，以下几点有助于我们理解操作系统的定义。

- ① 操作系统是软件，而且是系统软件。即它由一整套程序组成。
- ② 它的基本职能是控制和管理系统内各种资源，有效地组织多道程序的运行。想象一下用户编写的程序在计算机上执行的大致过程：程序以文件形式存放在磁盘上，运行之前计算机把它调入内存，然后在 CPU 上运行，产生的结果在屏幕上显示出来。这些工作都由操作系统完成。
- ③ 它提供众多服务，方便用户使用，扩充硬件功能。例如，用户可以使用操作系统提供的上百条命令或者图形界面完成对文件、输入/输出、程序运行等许多方面的控制、管理工作；可在一台机器上完成多项任务，甚至可以多人同时使用一台机器。

通常，可以这样定义操作系统：**操作系统是控制和管理计算机系统内各种硬件和软件资**

源，有效地组织多道程序运行的系统软件（或程序集合），是用户与计算机之间的接口。

1.2.2 操作系统的主要功能

从资源管理的角度看，操作系统要对系统内所有的资源进行有效的管理，优化其使用。从用户的角度来看，操作系统应当使用方便。综合这些因素可以看出，操作系统的主要功能有以下5个方面：存储管理、作业和进程管理、设备管理、文件管理和用户接口服务。

1. 存储管理

存储管理的主要功能包括：内存分配，地址映射，内存保护和内存扩充。

(1) 内存分配

内存分配的主要任务是为每道程序分配一定的内存空间。为此，操作系统必须记录整个内存的使用情况，处理用户提出的申请，按照某种策略实施分配，接收系统或用户释放的内存空间。

由于内存是宝贵的系统资源，并且往往出现这种情况：用户程序和数据对内存需求量的总和大于实际内存可提供的使用空间。为此，在制定分配策略时应该考虑提高内存的利用率，减少内存浪费。

(2) 地址映射

大家都有这种经历：我们在编写程序时并未考虑程序和数据放在内存的什么地方，在程序中设置变量、数组和函数等，只是为了实现这个程序所要完成的任务。源程序经过编译之后，形成若干目标程序，各自的起始地址都是“0”（但它并不是实际内存的开头地址！），各程序中用到的其他地址都分别相对起始地址计算。这样一来，在多道程序环境下，用户程序中所涉及的相对地址与装入内存后实际占用的物理地址就不一样。CPU执行用户程序时，要从内存中取出指令或数据，为此就必须把所用的相对地址（或称逻辑地址）转换成内存的物理地址。这就是操作系统的地址映射功能（需要有硬件支持）。

(3) 内存保护

不同用户的程序都放在一个内存中，就必须保证它们在各自己的内存空间中活动，不能相互干扰，更不能侵占操作系统的空间。为此，必须建立内存保护机制。例如，设置两个界限寄存器，分别存放正在执行的程序在内存中的上界地址值和下界地址值。当程序运行时，所产生的每个访问内存的地址都要做合法性检查。就是说，该地址必须大于或等于下界寄存器的值，并且小于上界寄存器的值。如果地址不在此范围内，则属于地址越界，将发生中断并且进行相应处理。

另外，还要允许不同用户程序共享一些系统的或用户的程序。

(4) 内存扩充

一个系统中内存容量是有限的，不能随意扩充其大小。然而，用户程序对内存的需求越来越大，很难完全满足用户的要求。这样就出现各用户对内存“求大于供”的局面。怎么办？物理上按需扩充内存的办法往往并不妥当，实际上是采取逻辑扩充内存的办法，这就是虚拟存储技术。简单来说，就是把一个程序当前正在使用的部分（不是全体）放在内存，而其余部分放在磁盘上。在这种“程序部分装入内存”的情况下，就启动并执行它。以后根据程序执行时的要求和内存当时使用的情况，随机地将所需部分调入内存；必要时还要把已分出去的内存回收，供其他程序使用（即内存置换）。