

21世纪高等学校本科计算机专业系列实用教材

操作系统

实 用 教 程

(第2版)

◎ 张献忠 主 编
梁 银 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

21 世纪高等学校本科计算机专业系列实用教材

操作系统实用教程

(第 2 版)

张献忠 主 编

梁 银 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是《操作系统实用教程》的第2版,它是在第1版的基础上,结合目前操作系统新课程教学大纲与考研大纲修订而成的。本书全面系统地介绍了作业管理与用户接口、进程管理、内存管理、设备管理、文件管理、操作系统的安全性、网络和分布式操作系统,从实用的角度把操作系统原理与实践紧密结合起来。本书的最大特色是结合目前主流操作系统 Windows 2000/XP 和 Linux 的相关实现技术进行阐述,举例生动形象,针对性强。另外,附录A还提供了10个操作系统上机实验,实用性强,具有很好的可操作性。

本书既可作为高等院校本、专科计算机及相关专业的教材或参考书,也可作为自考、考研或计算机等级(三级和四级)考试的辅导用书,同时也适合操作系统爱好者与软件开发人员研习。

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

操作系统实用教程 / 张献忠主编. —2版. —北京:电子工业出版社,2010.7

(21世纪高等学校本科计算机专业系列实用教材)

ISBN 978-7-121-10832-7

I. ①操… II. ①张… 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(2010)第083714号

责任编辑:刘海艳(lhy@phei.com.cn)

印 刷:北京市海淀区四季青印刷厂

装 订:涿州市桃园装订有限公司

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:18 字数:484千字

印 次:2010年7月第1次印刷

印 数:4000册 定价:29.80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 zltts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

序 言

21 世纪是“信息”主导的世纪，是崇尚“创新与个性”发展的时代，体现“以人为本”、构建“和谐社会”是社会发展的主流。然而随着全球经济一体化进程的不断推进，市场与人才的竞争日趋激烈。对于国家倡导发展的 IT 产业，需要培养大量的、适应经济和科技发展的计算机人才。

众所周知，近年来，一些用人单位对部分大学毕业生到了工作岗位后，需要 1~2 年甚至多年的训练才能胜任工作的“半成品”现象反应强烈。从中反映出单位对人才的需求越来越讲究实用，社会要求学校培养学生的标准应该和社会实际需求的标准相统一。对于 IT 业界来讲，一方面需要一定的科研创新型人才，从事高端的技术研究，占领技术发展的高地；另一方面，更需要计算机工程应用、技术应用及各类服务实施人才，这些人才可统称“应用型”人才。

应用型本科教育，简单地讲就是培养高层次应用型人才的本科教育。其培养目标应是面向社会的高新技术产业，培养在工业、工程领域的生产、建设、管理、服务等第一线岗位，直接从事解决实际问题、维持工作正常运行的高等技术应用型人才。这种人才，一方面掌握某一技术学科的基本知识和基本技能，另一方面又具有较强的解决实际问题的基本能力，他们常常是复合性、综合性人才，受过较为完整的、系统的、有行业应用背景的“职业”项目训练，其最大的特色就是有较强的专业理论基础支撑，能快速地适应职业岗位并发挥作用。因此，可以说“应用型人才”培养既有本科人才培养的一般要求，又有强化岗位能力的内涵，它是在本科基础之上的以“工程师”层次培养为主的人才培养体系，人才培养模式必须吸取一般本科教育和职业教育的长处，兼收并蓄。“计算机科学与技术”专业教学指导委员会已经在研究并指导实施计算机人才的“分类”培养，这需要我们转变传统的教育模式和教学方法，明确人才培养目标，构建课程体系，在保证“基础的前提”下，重视素质的培养，突出“工程性”、“技术应用性”、“适应性”概念，突出知识的应用能力、专业技术的应用能力、工程实践能力、组织协调能力、创新能力和创业精神，较好地体现与实施人才培养过程的“传授知识、训练能力、培养素质”三者的有机统一。

在规划本套教材的编写时，我们遵循专业教学委员会的要求，针对“计算机工程”、“软件工程”、“信息技术”专业方向，以课群为单位选择部分主要课程，以计算机应用型人才培养为宗旨，确定编写体系，并提出以下编写原则。

(1) 本科平台：必须遵循专业基本规范，按照“计算机科学与技术”专业教学指导委员会的要求构建课程体系，覆盖课程教学知识点。

(2) 工程理念：在教材体系编写时，要贯穿“系统”、“规范”、“项目”、“协作”等工程理念，内容取舍上以“工程背景”、“项目应用”为原则，尽量增加一些实例教学。

(3) 能力强化：教学内容的举例，结合应用实际，力争有针对性；每种教材要安排课程实践教学指导，在课程实践环节的安排上，要统筹考虑，提供面向现场的设计性、综合性的实践教学指导内容。

(4) 国际视野：本套教材的编写要做到兼收并蓄，吸收国内、国外优秀教材的特点，人才培养要有国际背景和视野。

本套教材的编委会成员及每本教材的主编都有着丰富的教学经验，从事过相关的工程项目（软件开发）的规划、组织与实施，希望本套教材的出版能为我国计算机应用型人才的培养尽一点微薄之力。

编委会

前 言

操作系统是计算机系统中最重要系统软件，是计算机的“管家”，是计算机和用户之间的接口。操作系统这门课程是计算机及其相关专业的核心课程之一，它主要讲述操作系统的基本原理和实现方法。操作系统课程的特点是概念多，内容抽象，理论性强。长期以来，高校普遍反映这门课难讲、难学、难做实验。除了课程本身的原因，事实上，教材的选用也是一个非常重要的因素。目前已出版的操作系统教材中，偏重理论的较多，而能结合教学实际、注重实践性的教材很少。本教材就是作者根据多年教学研究经验，针对课程本身的特点，充分结合目前的教学实际而编写的。

与传统操作系统教材相比，本教材采用理论与实践相结合的方式，以达到学以致用目的。本教材的特色如下：

(1) 传统经典理论与实际应用相结合，实现感性认识与理性认识的统一。结合目前主流操作系统 Windows 2000/XP 和 Linux 进行实例分析，体现形象、生动的教学模式，避免了抽象、空洞的纯理论教学。

(2) 一切从教学实际出发，体现“以读者和学习为中心”的原则。尽量采用通俗易懂的语言，举例生动形象。在内容章节安排上注重读者的阅读习惯，做到由表及里，前后贯通，思路清晰。

全书共 8 章。第 1 章绪论，简要介绍了操作系统的概念、基本特征、类型、发展历史和硬件环境。第 2 章作业管理与用户接口，介绍了作业的概念、作业管理的功能与用户接口知识。第 3 章进程管理，详细介绍了进程的基本概念、进程调度、进程同步与死锁问题，对多线程概念进行了说明。第 4 章内存管理，介绍了内存管理的基本概念和功能，详细讨论了内存管理的机制与算法，重点讨论了虚拟内存的管理机制。第 5 章设备管理，介绍了设备管理的基本概念和功能，讨论了设备管理的相关技术和处理过程。第 6 章文件管理，介绍了文件系统的基本概念和功能，详细讨论了文件系统的管理策略。第 7 章操作系统的安全性，介绍了操作系统安全性的基本概念，讨论了操作系统的安全策略。第 8 章网络和分布式操作系统，简单介绍了它们的基本概念和实现特点。附录 A 中提供了操作系统的相关实验安排。

本书由张献忠主编，梁银副主编，季洪波、何中胜参编。其中，张献忠编写了第 1、2、5、6 章，梁银编写了第 3、4 章，季洪波编写了第 7 章，何中胜编写了第 8 章。全书最后由张献忠统一定稿，参加本书编写的人员还有李成、李云、谢光前、唐土生、徐强等。全书在编写过程中得到了电子工业出版社的大力支持，在此表示衷心感谢。

由于编者水平有限，不足之处在所难免，敬请广大读者批评指正。本书的相关资料可与出版社编辑刘海艳（lhy@phei.com.cn）或作者联系，联系信箱：zhangxz@czu.cn。

编 者

目 录

第 1 章 绪论	1	1.6.5 时钟	28
1.1 什么是操作系统	1	1.7 如何学习操作系统	29
1.1.1 引入操作系统的目标	1	1.7.1 学习操作系统的目的	29
1.1.2 操作系统的定义与功能	2	1.7.2 学习操作系统的几个观点	30
1.1.3 操作系统的逻辑结构	5	1.7.3 学习操作系统的方法	32
1.2 操作系统的基本特征	5	本章小结	32
1.2.1 并发性	5	习题 1	33
1.2.2 共享性	6	第 2 章 作业管理与用户接口	34
1.2.3 虚拟性	6	2.1 作业的概念	34
1.2.4 异步性	6	2.1.1 作业与作业步	34
1.3 操作系统的类型	7	2.1.2 作业的类型	35
1.3.1 批处理操作系统	7	2.1.3 作业控制语言与作业说明书	36
1.3.2 分时操作系统	8	2.1.4 Linux 中的作业示例	36
1.3.3 实时操作系统	8	2.2 作业管理的功能	37
1.3.4 单用户操作系统	9	2.2.1 作业的建立	37
1.3.5 网络操作系统	9	2.2.2 作业控制块	39
1.3.6 分布式操作系统	9	2.2.3 作业的状态变迁	39
1.3.7 并行操作系统	10	2.2.4 作业调度	40
1.3.8 嵌入式操作系统	10	2.3 操作系统的用户接口	42
1.4 操作系统的发展历史	11	2.3.1 用户接口的功能与类型	42
1.4.1 手工操作时期	11	2.3.2 命令接口	42
1.4.2 监控程序时期	11	2.3.3 程序接口	44
1.4.3 多道程序与操作系统成熟 时期	12	2.3.4 用户接口的发展	46
1.4.4 软件工程与操作系统的发展 时期	14	2.4 Windows 2000/XP 的用户接口	47
1.5 实用操作系统介绍	15	2.4.1 Windows 2000/XP 的系统命令	47
1.5.1 Windows 操作系统	15	2.4.2 Windows 2000/XP 的 GUI	48
1.5.2 UNIX 操作系统	17	2.4.3 Win32 API 函数	50
1.5.3 Linux 操作系统	19	2.5 Linux 的用户接口	52
1.6 操作系统的运行环境	21	2.5.1 Linux 的 shell	52
1.6.1 Intel x86 CPU	21	2.5.2 Linux 的系统调用	56
1.6.2 IA32 寄存器	22	2.6 DOS 系统调用	58
1.6.3 存储器	24	2.6.1 BIOS 调用	58
1.6.4 中断	26	2.6.2 DOS 系统功能调用	58
		本章小结	59
		习题 2	59

第 3 章 进程管理	61
3.1 进程的引入	61
3.1.1 顺序程序与并发程序	61
3.1.2 进程的定义及特性	63
3.1.3 进程的状态及其转换	65
3.1.4 Linux 的进程状态	67
3.2 进程的结构	68
3.2.1 进程的实体	68
3.2.2 进程控制块	69
3.2.3 Linux 中的进程	71
3.2.4 Windows 2000/XP 中的进程	73
3.3 进程控制	75
3.3.1 进程的创建与撤销	75
3.3.2 进程的阻塞与唤醒	77
3.3.3 进程的挂起与激活	78
3.3.4 Linux 的进程控制	78
3.4 进程的同步与互斥	79
3.4.1 基本概念	79
3.4.2 实现进程互斥的硬件方法	81
3.4.3 实现进程互斥的软件方法	83
3.4.4 信号量与 P、V 操作	84
3.4.5 用 P、V 操作实现互斥	85
3.4.6 用 P、V 操作实现同步	86
3.4.7 经典同步与互斥问题	87
3.5 进程间通信	90
3.5.1 消息缓冲	90
3.5.2 信箱方式	92
3.5.3 共享内存	93
3.5.4 管道通信	94
3.5.5 Linux 的进程通信	95
3.6 进程调度	98
3.6.1 进程调度的方式	98
3.6.2 进程调度算法	99
3.6.3 Linux 的进程调度	104
3.7 死锁	104
3.7.1 死锁的概念	104
3.7.2 死锁的预防	106
3.7.3 死锁的避免	107
3.7.4 死锁的检测和解除	112

3.8 线程	115
3.8.1 线程的引入	115
3.8.2 线程的基本概念	115
3.8.3 线程的实现	117
3.8.4 Windows 2000/XP 的线程	118
本章小结	121
习题 3	122
第 4 章 内存管理	125
4.1 内存管理功能	125
4.1.1 内存的分配与回收	125
4.1.2 地址重定位	126
4.1.3 内存的共享和保护	128
4.1.4 虚拟存储器	128
4.2 分区管理	129
4.2.1 单分区	129
4.2.2 固定分区	130
4.2.3 可变分区	131
4.2.4 碎片问题及其解决方法	134
4.2.5 覆盖与交换	135
4.3 页式管理	136
4.3.1 页式管理概述	136
4.3.2 静态页式管理	137
4.3.3 页式虚存管理	142
4.3.4 Windows 2000/XP 的页式虚存管理	149
4.4 段式管理	151
4.4.1 段式管理概述	151
4.4.2 地址转换	153
4.4.3 段的共享和保护	153
4.4.4 段式虚存管理	154
4.4.5 段式、页式管理的比较	155
4.5 段页式管理	155
4.5.1 基本思想	155
4.5.2 地址转换	156
4.5.3 段页式虚存管理	157
4.5.4 Linux 的内存管理	157
本章小结	160
习题 4	161

第 5 章 设备管理	163		
5.1 设备管理概述	163	6.2.1 文件的逻辑结构	198
5.1.1 设备及其分类	163	6.2.2 文件的物理结构	200
5.1.2 设备控制器	164	6.2.3 文件的存取方法	205
5.1.3 通道	166	6.2.4 文件的存储设备	206
5.1.4 设备管理的目标与功能	167	6.3 文件目录管理	208
5.2 I/O 控制方式	167	6.3.1 文件控制块与文件目录	208
5.2.1 程序直接控制方式	167	6.3.2 索引节点	209
5.2.2 中断控制方式	168	6.3.3 目录结构	210
5.2.3 DMA 方式	169	6.4 文件存储空间管理	213
5.2.4 通道方式	170	6.4.1 空闲块表法	213
5.3 设备管理技术	171	6.4.2 空闲块链法	213
5.3.1 缓冲技术	171	6.4.3 成组链接法	214
5.3.2 磁盘驱动调度技术	174	6.4.4 位示图	215
5.3.3 虚拟设备与 SPOOLing 技术	176	6.5 文件的共享	216
5.3.4 Windows 2000/XP 中的共享打 印机	177	6.5.1 文件共享	216
5.4 设备的分配	178	6.5.2 Linux 文件的共享	217
5.4.1 设备分配的数据结构	178	6.6 文件系统实例分析	218
5.4.2 设备分配策略	179	6.6.1 Windows 2000/XP 文件系统	218
5.4.3 设备独立性	181	6.6.2 Linux 文件系统	219
5.4.4 独占设备的分配过程	182	本章小结	223
5.5 I/O 软件	183	习题 6	223
5.5.1 用户级 I/O 软件	183	第 7 章 操作系统的安全性	225
5.5.2 设备无关软件	184	7.1 安全性概述	225
5.5.3 设备驱动程序	185	7.1.1 安全性含义	225
5.5.4 中断处理程序	186	7.1.2 影响系统安全性的因素	225
5.6 实用系统中的设备驱动程序	186	7.1.3 操作系统的安全机制	227
5.6.1 Windows 2000/XP 设备驱动 程序	186	7.2 实现系统安全性的基本策略	228
5.6.2 Linux 设备驱动程序	189	7.2.1 身份鉴别策略	228
本章小结	193	7.2.2 文件保护策略	230
习题 5	194	7.2.3 内存保护策略	231
第 6 章 文件管理	195	7.2.4 恶意代码防御策略	232
6.1 文件与文件系统	195	7.3 Linux 的安全性	234
6.1.1 文件的概念	195	7.3.1 Linux 的安全策略	234
6.1.2 文件的类型	196	7.3.2 Linux 的安全漏洞	236
6.1.3 文件的操作	196	7.4 Windows 2000/XP 的安全策略	236
6.1.4 文件系统的概念和功能	197	7.4.1 Windows 2000/XP 安全模型	236
6.2 文件结构与存储设备	198	7.4.2 Windows 2000/XP 的注册表	238
		7.4.3 Windows 2000/XP 的组策略	241
		本章小结	243
		习题 7	243

第 8 章 网络和分布式操作系统	244
8.1 网络操作系统	244
8.1.1 网络拓扑结构	244
8.1.2 网络操作系统的功能	245
8.1.3 网络操作系统的实现方法	246
8.1.4 资源共享技术	247
8.2 分布式操作系统	248
8.2.1 分布式系统概述	248
8.2.2 分布式操作系统的特点及实现策略	250
8.2.3 分布式系统的资源管理	252
8.2.4 分布式系统的通信	254
本章小结	257
习题 8	257

附录 A 操作系统上机实验	258
A.1 用 VMware 虚拟安装 Linux	258
A.2 用户接口实验	260
A.3 作业调度实验	262
A.4 进程创建与控制实验	262
A.5 进程调度实验	264
A.6 进程间通信实验	264
A.7 页式虚拟存储管理实验	266
A.8 文件系统实验	268
A.9 设备驱动程序实验	271
A.10 Linux 部分源代码分析	272
参考文献	273

第1章

绪论

计算机系统由硬件（Hardware）系统和软件（Software）系统两部分组成。其中，硬件是用户直接可见的部分，它是所有软件运行的物质基础。软件是对硬件功能的扩展，它能完成各种系统及应用任务。软件对用户来说直接观察不到，具有一定的抽象性。软件有系统软件和应用软件之分，而操作系统是最重要的一种系统软件，是对硬件系统的第一次扩充，它是所有软件运行的基础。操作系统不仅体现了计算机软件技术的研究成果，也体现了计算机硬件技术及系统结构改进的发展成果。

本章从操作系统的概念出发，主要介绍了操作系统的功能、特征、类型及发展历史，分析了操作系统的运行环境，对当前的主流操作系统 UNIX、Windows 和 Linux 的相关特征与应用做了阐述，最后对初学者如何学习操作系统提出了若干建议。

1.1 什么是操作系统

1.1.1 引入操作系统的目标

操作系统（Operating System, OS），对于一个初学者来讲，可能说不出它到底是什么，但只要用过计算机，就一定使用过操作系统。为了加强对操作系统的感性认识，不妨来判断一下，在下面的这些软件中，哪些属于操作系统呢？

Windows 2000/XP、Visual FoxPro 6.0、ASP.NET、UNIX、Linux、Pascal、Photoshop

其实不难发现，在上述软件中属于操作系统的有 Windows 2000/XP、UNIX、Linux，其余的属于系统软件或应用程序。当然，操作系统的种类远不止这些，只不过有些操作系统一般用户很少接触，如一些专用领域的操作系统。因此，为便于理解，本书选择目前比较流行的 Windows 2000/XP 和 Linux 作为实例进行分析。

为什么在计算机系统中要安装操作系统呢？通常，把没有安装任何软件的计算机称为裸机（Bare Machine）。直接使用裸机十分困难，对用户的要求很高，此时计算机硬件的效率也很难发挥。为此，人们在硬件的基础上，通过增加一层软件来实现对硬件功能的扩充，使它能自动管理计算机系统内的软、硬件资源，这种在硬件上扩充的第一层软件就是操作系统。操作系统紧贴系统硬件之上，所有其他软件的运行都必须以它为基础。

操作系统有什么作用？从操作系统的使用经验可知，操作系统能提供程序运行的环境并控制其运行过程，如在 Windows 系统中，用鼠标双击程序图标就能启动程序的运行；操作系统能提供友好的用户接口，很容易对文件进行操作，如创建、复制和删除文件等；操作系统能很好地控制 I/O 设备的工作，如打印机的工作等。总之，操作系统能做的事很多，一下子

很难概括,它既像个“管家”,对系统中的一切资源要进行管理,又像个“服务员”,要为用户使用计算机提供服务,但有些服务又无能为力。例如,操作系统不是编译程序,它不能对用户的源程序进行编译,这时需要专门的编译程序才能完成;操作系统不能提供财务管理功能,这要靠相应的财务软件才能实现;操作系统不会进行天气预报,这需要专门的天气预报软件来完成;等等。事实上,操作系统提供的服务是与“硬件相关”和“应用无关”的服务,它不直接处理与实际应用有关的问题,但所有与硬件相关的问题都由它解决,也就是说,所有软件要在计算机上运行,必须要有操作系统的支持。因此,操作系统是所有软件运行的基础。操作系统在整个计算机系统中的地位如图 1-1 所示。

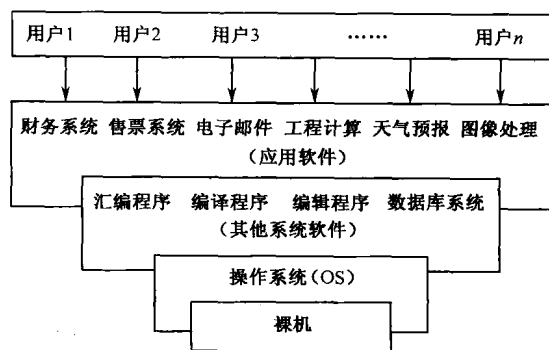


图 1-1 操作系统在整个计算机系统中的地位

经过以上分析可以得出,引入操作系统的目标有以下 4 个。

- (1) 有效性。操作系统能有效管理、分配和调度软、硬件资源,合理地组织计算机的工作流程。
- (2) 方便性。操作系统提供良好的、统一的用户接口,使用户使用计算机变得十分方便。
- (3) 可扩充性。操作系统通过扩充改造硬件类型和规模来扩充计算机的功能。
- (4) 开放性。操作系统支持体系结构的可伸缩性和可扩展性,能实现在多个系统之间的资源共享和互操作,使应用程序便于在不同平台上移植。

1.1.2 操作系统的定义与功能

1. 操作系统的定义

到目前为止,操作系统的产生、使用和发展已经有四十多年了,然而,到底什么是操作系统,人们至今对它的定义仍无统一的标准。但综合各种观点和说法来看,一种被普遍认同的定义是:操作系统是一组控制和管理计算机硬件和软件资源,合理地各类作业进行调度,以及方便用户使用计算机的程序的集合。从它的定义可以看出,操作系统的作用主要体现在两个方面:一是管理员,二是服务员。作为管理员,操作系统要负责对各种硬件和软件资源进行分配和管理。其中,硬件资源主要包括 CPU、存储器、寄存器、堆栈和 I/O 设备等;软件资源主要包括系统程序、应用程序和各种数据及其数据结构等。作为服务员,操作系统是用户和计算机之间的接口(Interface),操作系统通过用户接口来为用户提供服务。

2. 操作系统的功能

操作系统的功能主要体现在以下 5 个方面。

(1) 处理器管理

处理器（CPU）管理主要完成处理器的分配、调度和释放等功能。由于处理器是计算机硬件中最宝贵的资源，因此，必须使处理器尽可能地忙起来，减少其等待时间，从而提高它的利用率。在单用户单任务的环境下，处理器仅为一个用户的任务服务，虽然其管理工作简单，但是它的使用效率却很低。为了尽量提高处理器的利用率，操作系统采用了多道程序技术。在多道程序环境下，为了保证这些程序能并发执行，采用合理、有效的处理器管理策略是十分必要的。

为了便于处理器的分配、调度和管理，更好地描述多道程序的并发执行，操作系统引入了进程（Process）和线程（Thread）的概念。进程是对程序并发执行的动态描述，是系统内资源分配的基本单位，在多进程操作系统中，进程也是处理器调度的基本单位。而线程是进程内部的一个控制流，是对进程的有效细化。引入线程的目的是为了减少操作系统对进程调度的开销，进一步提高系统的并发程度，使并发执行的代价降为最低。在多线程系统中，线程是处理器调度的基本单位。实际上，在多进程多线程系统中，对处理器的管理和调度最终归结为对进程和线程的管理和调度，它主要包括：

- ① 进程和线程控制。完成进程或线程的创建、撤销、挂起和改变执行优先级等功能。
- ② 进程和线程的同步。协调并发进程或线程之间的执行顺序，方便进行资源共享。
- ③ 进程间通信。进程之间相互传输数据，以协调进程间的协作。
- ④ 进程的死锁。包括对死锁的预防、避免、检测和解除等功能。
- ⑤ 处理器调度。包括作业调度、中级调度、进程或线程调度。通过作业和进程的执行切换，来充分利用处理器资源和提高系统性能。

(2) 内存管理

内存储器简称内存（主存）。内存管理的主要任务是对内存资源的分配和回收，实现地址转换，实现内存的逻辑扩充以及提供内存共享和保护机制等功能。内存管理的目标是提高内存的利用率，方便用户使用内存，为用户提供透明服务。众所周知，计算机运行程序的前提是先将程序装入内存，但受成本等方面条件的制约，内存的容量是有限的。在多道程序系统中，用户使用内存时需考虑其他程序的影响。每个用户程序如何申请内存，如何装入内存中的指定区域，如何保护好自己的程序不受干扰或影响等问题，这都是用户程序设计人员无法事先知道的事情。因此，操作系统中的内存管理模块就是用于解决这些问题的。内存管理的具体功能如下。

① 内存的分配与回收。为需要运行的程序分配必要的内存资源，当程序运行结束后再回收其分配的内存资源，这需要在操作系统中设置相应的数据结构，用于登记当前内存的使用情况。

② 地址转换。由于内存和外存编址的不同，当程序从外存装入内存时，需将程序中的逻辑地址（相对地址）转换为内存中的物理地址（绝对地址）。这一过程可通过地址重定位完成。根据重定位时机的不同，可以分为静态重定位和动态重定位。静态重定位是在程序装入内存时进行；而动态重定位是在程序运行时进行，动态重定位通常需要硬件的支持。

③ 内存的共享和保护。内存管理要实现用户程序对内存资源的共享，即当多个程序包含同一个公用代码段时，这个公用代码段在内存中只保留一个副本。内存的共享提高了内存资源的利用率。内存的保护指的是要保证不同用户的程序之间互不干扰、相互保密，尤其是不允许访问操作系统的程序和数据，保证用户和系统程序在内存中的信息不被破坏。

④ 内存的逻辑扩充。内存容量的有限性直接制约了进程的并发执行效率，而且也限制

了在内存中能运行的应用程序的规模。为了解决内存的“瓶颈”问题,必须实现内存的扩充。内存扩充可采用物理扩充和逻辑扩充两种方式。由于受客观因素的影响,无限制地进行物理扩充是不现实的,也是不经济的。借助于软件技术,可以对内存进行逻辑扩充。操作系统主要采用覆盖、交换和虚拟存储等软件技术来实现对内存的逻辑扩充,使进程的内存空间扩大,从而提高内存利用率。

(3) 设备管理

设备管理的主要任务是对各种外围设备进行管理,响应用户进程提出的 I/O 请求,为用户进程分配 I/O 设备,控制 CPU 与外设间的数据交换等。设备管理的目标:一是方便用户使用设备,二是提高 CPU 与 I/O 设备的利用率。设备管理的具体功能如下。

① 完成设备的分配与回收。根据用户的 I/O 请求分配相应的设备,并在使用完后回收该设备,尤其要解决在多用户间共享 I/O 设备资源等问题。

② 实现设备独立性。设备独立性是指提供统一的 I/O 设备接口,使应用程序独立于物理设备,提高可适应性。

③ 控制设备操作。利用设备驱动程序和控制程序完成对设备的操作。

④ 实现虚拟设备。虚拟设备 (Virtual Device) 是为实现独享设备的共享而提出的。当多个进程要共享某个独享设备时,可通过虚拟技术 (如 SPOOLing 技术) 将其转换为几个逻辑设备,分别为多个进程服务。对于每个进程来说,就如同独占该物理设备一样。

⑤ 实现缓冲区管理。为了解决 CPU 和 I/O 设备之间的速度不匹配问题,必须使用缓冲区。可通过单缓冲区、双缓冲区、多缓冲区和缓冲池技术来实现,这样可提高 CPU 和 I/O 设备的利用率。

(4) 文件管理

文件管理又称为外存管理。外存是存放系统中的信息资源的场所,信息资源主要包括各类程序和数据等软件资源,它们主要以文件的形式存放。文件管理的主要任务是对各种文件 (包括用户文件和系统文件) 进行有效管理,实现对文件的按名存取;解决文件在外存上的存储、共享、保密和保护问题,采用合理的分配策略来提高外存资源的利用率。文件管理的具体功能如下。

① 实现文件的按名存取。通过文件名找到该文件的具体存储区域并进行访问。

② 文件存储空间管理。解决如何分配和回收文件空间,以提高外存空间利用率和对文件的读/写性能。

③ 文件和目录管理。解决文件的分类与检索问题,采用合适的目录结构将提高文件的访问效率。提供对文件和目录操作的手段,如文件的读、写、复制、删除等。可通过系统调用方式和命令方式来实现。

④ 文件的存取控制。解决文件的安全和保护问题,文件信息的安全在多用户系统中显得尤为重要。

(5) 用户接口

为了方便用户使用计算机,操作系统提供了用户接口。用户接口的功能是提供一个友好的用户使用计算机的途径。操作系统的用户接口有以下两种:

① 命令接口。命令接口是向一般操作级用户提供的,它由字符方式下的键盘命令和图形方式下的鼠标命令组成。

② 程序接口。程序接口是向编程人员提供的,又称为应用程序编程接口 (API),它由一系列的系统调用组成。库函数是系统调用的高级形式,通过程序接口,编程人员可以在程

序中调用操作系统的内核代码，完成一些与硬件相关的复杂功能。

1.1.3 操作系统的逻辑结构

操作系统是个十分庞大的系统软件，它由几万行到几百万行不等的代码组成，这些代码被分散在操作系统的多个功能模块中。那么，这些功能模块是如何集成在一起的呢？这就是操作系统的逻辑（或整体）结构问题。事实上，每个操作系统从逻辑的角度来看都可以分为两个部分：一是内核，二是外壳。其中，内核（Kernel）部分是紧挨着硬件的部分，主要完成与硬件相关的功能，如操作系统的几大功能模块；而外壳（Shell）部分主要完成与硬件无关的功能，如提供实用程序和命令解释功能等，如图 1-2 所示。如果应用程序要使用内核的功能，则必须通过系统调用才能完成。

随着操作系统设计技术和思想的发展，内核本身的组织结构也产生了多样性，如整体模块结构、分层结构、客户机/服务器结构和微内核结构等。图 1-3 所示为内核分层的结构图。

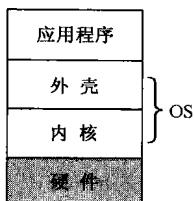


图 1-2 操作系统的逻辑结构

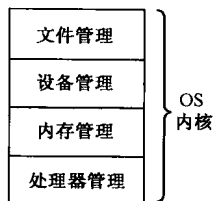


图 1-3 分层式的内核结构

1.2 操作系统的基本特征

1.2.1 并发性

并发性（Concurrency）是指若干个事件或活动在同一时间段内发生。操作系统是一个并发系统，并发性是它的最基本特征。操作系统的并发性是指在某一段时间内，计算机系统中若干程序都处于已经启动但尚未结束的状态。也就是说，计算机系统的内存中允许同时有多个用户程序，或内存中同时有操作系统程序和用户程序被交替、穿插地执行。操作系统具有处理和调度多个程序并发执行的能力。在单处理器系统中，程序的并发执行是指宏观上有多个程序在同时运行，微观上多个程序只能轮流交替执行。

并行（Parallel）是与并发相似的另一个概念，它是指若干个事件在同一时刻发生，因此，并行是并发的一个特例。在多处理器系统中，由于多个程序同一时刻可在不同 CPU 上运行，因此程序的运行是并行的。但在单处理器环境下，由于每一时刻 CPU 只能运行一个程序，所以单处理器系统中并不存在多个程序真正的并行。举个例子，可以把体育比赛中的接力赛看成是一种并发活动，接力棒相当于 CPU，选手相当于程序，跑道相当于内存。单组接力（一个接力棒）就是单处理器下的并发活动；多组接力（多个接力棒）就是多处理器下的并发活动，此时会出现并行的情况（多名选手同时持棒在跑）。通常情况下，我们讨论的并发活动主要指单处理器下的并发活动；而多处理器下的情况比较复杂，关注的多是并行活动。

那么，单处理器下的并发是如何实现的呢？这要从程序的逻辑结构说起。通常，每个程序都由两部分组成：一是 CPU 处理部分，二是 I/O 处理部分，不同的程序中这两部分所占比

重也不同。在通道和中断技术的支持下,一个程序的 CPU 处理与另一个程序的 I/O 处理可以同时进行,这也使得多道程序实现并发成为可能。事实上,单处理器系统中实现并发的本质思想:当一个程序发生事件(如等待 I/O)时让出其占用的 CPU 而由另一个程序运行,即一个程序在进行 I/O 处理的同时让另一程序在 CPU 上运行。因此,实现并发技术的关键之一在于如何对系统内的多个运行程序进行调度和切换。

操作系统的并发性能能够消除计算机系统中各程序之间的相互等待,有效地改善系统资源的利用率,从而提高了系统效率。但是并发也会引发一系列的问题,使操作系统的设计和实现变得复杂化。例如,在系统中的多个程序之间如何分配资源、如何切换运行状态、如何不对其他运行的程序造成影响等。

1.2.2 共享性

共享性(Sharing)是操作系统的另一个主要特性。共享是指计算机系统资源能被内存中的多个并发执行的系统程序和用户程序共同使用,而不是让某一个程序独享。如果一次性向每个用户程序分别提供它所需的全部资源不但是浪费的,有时也是不现实的。一种有效的做法是让操作系统程序和其他用户程序公用计算机系统的全部资源,这样可以减少浪费,从而提高系统资源的利用率。从实现的角度来看,资源共享有两种方式:互斥共享和同时共享。

(1) 互斥共享是指系统中的某些资源,如打印机、磁带机等,虽然它们可以被多个程序共同使用,但在某一段时间内只允许一个程序使用,其他程序不能使用,即程序之间要互斥使用该资源,只有等到一个程序使用完后其他程序才能使用。一次只允许被一个程序使用的资源称为临界资源。计算机系统许多物理设备,主要是字符设备,以及某些软件中的数据、表格、变量等都可能是临界资源,它们只能被互斥访问。

(2) 同时共享是指系统中的某些资源能够在一段时间内被多个程序同时使用。但在单处理器系统中,这里的同时只是个宏观上的概念,因为实际上这些程序是在一个时间段内并发使用这些资源,如对磁盘文件的同时访问其实也是交替进行的。

1.2.3 虚拟性

虚拟性(Virtual)是指操作系统中的一种管理技术,它是把一个物理设备转换为若干个对应的逻辑设备,或者把多个物理设备转换成一个逻辑设备的技术。采用虚拟技术的目的是为用户提供易于使用、方便管理的操作环境。例如,在分时系统中,虽然物理 CPU 只有一个,但通过使用分时技术,在宏观上还是有多个程序在同时运行,就好像一个物理 CPU 变成逻辑上的多个 CPU 一样,每个用户程序就在属于它自己的逻辑 CPU 上运行。还有,共享打印机技术也是利用某种虚拟 I/O 技术(SPOOLing 技术)实现的。同样,利用窗口技术可把物理上的一个屏幕变成逻辑上的多个虚拟屏幕。在操作系统中,虚拟性表现的另一个重要方面就是普遍采用的虚拟存储技术。为了能运行比物理内存大得多的应用程序,虚拟存储技术将内存和外存有机地结合起来,向用户提供足够大的虚拟内存空间。总之,虚拟技术的应用既方便用户使用计算机,也提高了系统的性能和资源利用率。

1.2.4 异步性

异步性(Asynchronism)也称随机性或不确定性。在多道系统中,允许多个进程并发执行。但由于系统内资源数量有限,而进程数量又往往多于资源数量,因此,进程必须相互竞争使用资源。当一个进程申请的资源被另一个进程占用时,该进程只能暂停执行,直到该进

程拥有这个资源后才能继续执行。因此,进程的执行顺序和执行时间存在着不确定性,即进程是以异步的方式运行,“走走停停”。很显然,这种异步性给系统带来了潜在的危险,若处理不好,可能会产生与时间有关的错误。可以说,操作系统是运行在随机的环境下。但只要环境相同,操作系统必须保证同一进程的多次运行,都会获得完全相同的结果。

在操作系统中,随机性的例子到处可见。例如,作业进入系统的时间和类型是随机的,操作员使用命令的时刻是随机的,程序运行中发生错误或异常的时刻是随机的,各种硬件中断信号产生的时刻是随机的,等等。面对这些随机发生的事件,操作系统必须确保捕捉任何一种随机事件并做出正确处理,否则将会导致严重后果。

1.3 操作系统的类型

在计算机发展的不同时期产生了不同类型的操作系统,以适应在特定的硬件环境下满足用户对计算机的使用要求。对操作系统的分类角度不同,就会有不同的分法。本节所介绍的几种类型,主要是从操作系统内部功能上进行划分的。随着操作系统技术的不断发展,操作系统本身的功能也出现了综合化和多元化,尤其体现在目前的主流操作系统中。因此,基于功能的划分界限现在已变得越来越模糊,这一点需要引起注意。

1.3.1 批处理操作系统

在计算机应用的早期,批处理是一般计算中心的主机(小型机以上)工作的主要方式。这些主机上所配置的操作系统就是批处理操作系统(Batch Processing Operating System)。在批处理操作系统中,常把用户交给计算机做的工作称为作业,一个作业通常由程序、数据和作业说明书组成。为了减少作业建立和结束过程中的时间浪费,当用户把作业提交给操作员后,操作员将把作业按照其性质分组(或分批),然后再成组(或成批)地提交给计算机系统,由计算机自动完成这批作业并输出结果。采用这种批量化处理作业的操作系统称为批处理操作系统。

根据在内存中允许存放的作业个数,批处理操作系统又分为单道批处理操作系统和多道批处理操作系统。早期的批处理操作系统属于单道批处理操作系统,它的特征是内存中只允许存放一个作业,即当前正在执行的作业才能驻留内存,作业的执行是按顺序执行。在单道批处理操作系统中,一个作业独占系统全部资源,当作业进行 I/O 操作时,由于内存中无其他作业,CPU 只能处于等待状态。因此,CPU 的利用率较低,尤其是对于 I/O 操作时间较长的作业。为了提高 CPU 的利用率,在单道批处理操作系统的基础上引入了多道程序设计(Multiprogramming)技术,这就形成了多道批处理操作系统,即在内存中可同时存放多个作业,作业执行的次序与进入内存的次序无严格的对应关系,因为作业的执行由作业调度算法来决定。一个作业在等待 I/O 处理时,系统调度另外一个作业执行,因此 CPU 的利用率显著地提高了。单道批处理系统与多道批处理系统的区别见表 1-1。

表 1-1 单道批处理系统与多道批处理系统的区别

类 别	单道批处理系统	多道批处理系统
内存利用情况	只存放单个作业,内存利用率低	可存放多个作业,内存利用率高
作业执行次序	先进先出	无确定次序