

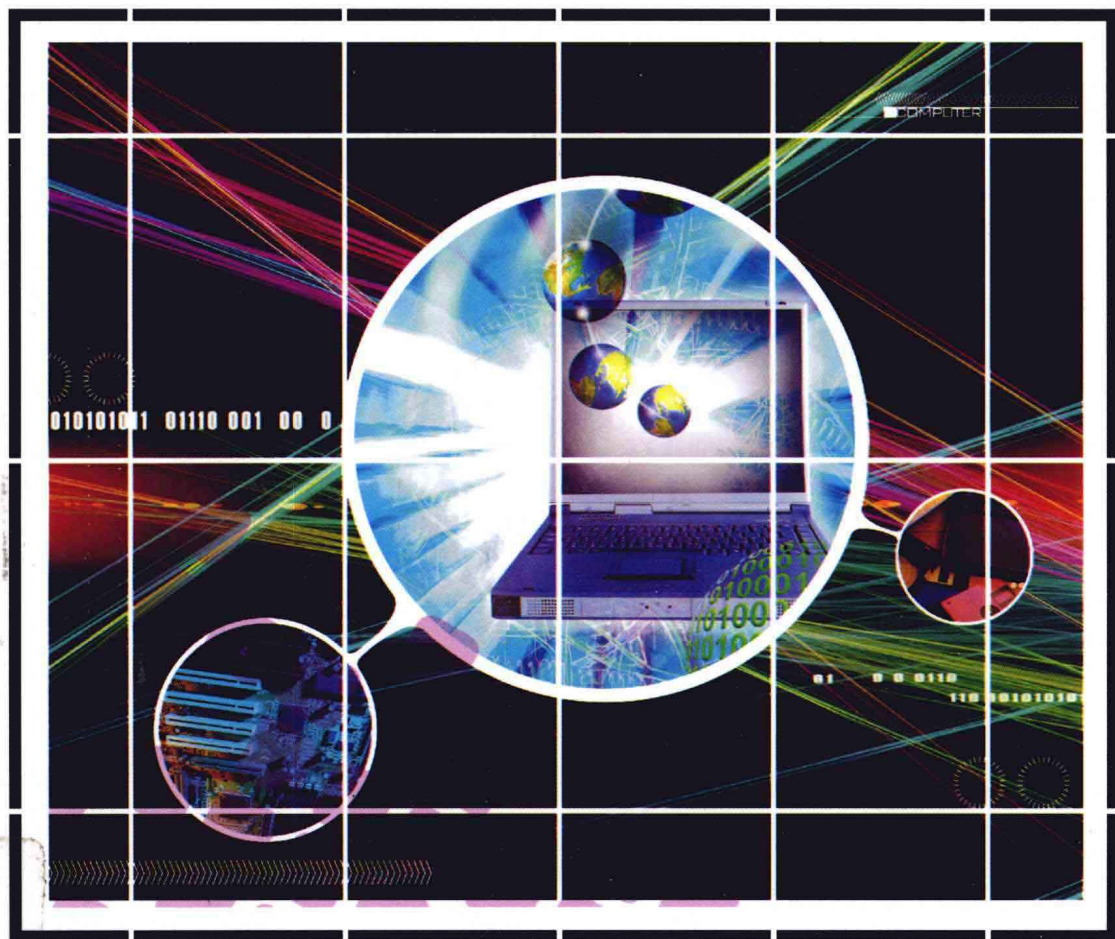
新世纪计算机类本科规划教材
COMPUTER

部级优秀教材

计算机操作系统

(第三版)

汤小丹 梁红兵 哲凤屏 汤子瀛 编著



西安电子科技大学出版社
<http://www.xduph.com>

新世纪计算机类本科规划教材

部级优秀教材

计算机操作系统

(第三版)

汤小丹 梁红兵
哲凤屏 汤子瀛 编著

西安电子科技大学出版社

内 容 简 介

本书全面介绍了计算机系统中的一个重要软件——操作系统(OS)，本书是第三版，对 2001 年出版的修订版的各章内容均作了较多的修改，基本上能反映当前操作系统发展的现状，但章节名称基本保持不变。全书仍分为 10 章，第一章介绍了 OS 的发展、特征、功能以及 OS 结构；第二、三章深入地阐述了进程和线程的基本概念、同步与通信、调度与死锁；第四章对连续和离散存储器管理方式及虚拟存储器进行了介绍；第五章为设备管理，对 I/O 软件的层次结构作了较深入的阐述；第六、七章分别是文件管理和用户接口；第八章介绍了计算机网络、网络体系结构、网络提供的功能和服务以及 Internet；第九章对保障系统安全的各种技术和计算机病毒都作了较详细的介绍；第十章是一个典型的 OS 实例——UNIX 系统内核结构。

本书可作为计算机硬件和软件以及计算机通信专业的本科生教材，也可作为从事计算机及通信工作的相关科技人员的参考书。

图书在版编目 (CIP) 数据

计算机操作系统 / 汤小丹等编著. — 3 版.

—西安: 西安电子科技大学出版社, 2007.5 (2010.10 重印)

新世纪计算机类本科规划教材

ISBN 978-7-5606-0496-1

I. 计… II. 汤… 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 (2007) 第 039199 号

责任编辑 李惠萍

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2007 年 5 月第 3 版 2010 年 10 月第 36 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 25.25

字 数 508 千字

印 数 354 001~394 000 册

定 价 30.00 元

ISBN 978-7-5606-0496-1 / TP·0232

XDUP 0766A13-36

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜，谨防盗版。

第三版前言

本书从 1996 年出版至今，这是第二次修订(2001 年修订过一次)。在本次修订中，仍将全书分为 10 章，虽然每章的标题没有改变，但却增、删或更新了其中的一部分内容，使其能更好地反映操作系统的发展现状和前沿技术。

本书 10 章内容的具体安排如下：

第一章为操作系统引论，介绍了 OS 的发展、特征、功能等。在 OS 的发展中增加了微处理机的发展，对 OS 结构设计的内容进行了更新，篇幅也有较大的扩充。第二章深入地阐述了进程和线程的基本概念以及同步与通信，对进程的内容进行了适当增添，对管程以及线程的内容进行了较大的更新。第三章为处理机调度与死锁，对其中的作业调度、进程调度以及实时调度的内容都有一定的增加和修改。由于篇幅的限制，删除了多处理机调度的内容。第四章对连续和离散存储器管理方式及虚拟存储器进行了介绍，对其中的分配算法方面的内容有一定的增加和修改，并增添了对存储器的层次结构的介绍。第五章为设备管理，对 I/O 软件的层次结构作了较深入的阐述。第六章是文件管理，对其中的外存分配方式以及磁盘容错技术等内容进行了更新和扩充。第七章是用户与操作系统的接口，对接口方式以及系统调用等部分内容进行了较大的更新及修改。第八章为网络操作系统，在本章中我们对各节都作了修改，并增加了许多内容，使其能反映 21 世纪网络在硬件和软件方面的发展现状。同样是因为篇幅的限制而删除了 Windows NT 的内容。第九章较详细地介绍了保障系统安全的各种技术。由于近年来计算机病毒已严重地威胁到系统的安全，故我们特增加了一节，对计算机病毒作了较全面的介绍。第十章是一个典型的 OS 实例——UNIX 系统内核结构。

在本书的编写过程中，得到了西安电子科技大学出版社，特别是责任编辑李惠萍老师的大力支持与合作。此外，王侃雅和汤蓓莉等同志在校对、整理等工作中，都付出了辛勤的劳动。在此谨向以上各位表以衷心的感谢。

，虽然本书经过了反复修改，我们也希望能把它写得更好，但限于编者的水平，书中仍难免会有错误和不当之处，恳请读者批评指正。

编者

2007 年 2 月

第二版前言

操作系统(OS)是最重要的计算机系统软件,同时也是最活跃的学科之一,其发展极为迅速。为使本教材内容能紧跟时代潮流,从1981年至今,我们已对本教材做过多次修改。2000年我们又对1996年出版的《计算机操作系统》教材进行了重写。为了适当压缩篇幅,我们调整了该教材的结构,从原来的15章改为10章。即将原来的第二、三章合并为“进程管理”一章;原来的第五、六章合并为“存储器管理”一章;第八、九章合并为“文件管理”一章;第十一、十二章合并为“网络操作系统”一章。另外,考虑到在大学低年级的教学实践中,学生已经学习过Windows OS的使用,故本次修订时删去了原版第十五章。

我们在本教材中,介绍了许多在20世纪90年代引入或广泛使用的技术,如微内核OS结构、线程的控制与通信、数据一致性、系统容错技术等,又因为20世纪90年代是计算机网络特别是Internet大发展的年代,故我们对网络操作系统一章做了较大的修改。还应强调说明的是,随着网络的广泛应用,系统安全性问题提到了头等重要的地位。事实上,若不能确保系统(网络)的安全性,则系统(网络)是难以被人接受的。故在国内外的OS教科书中,大多都增加了一章或几章内容用于介绍系统的安全性保障。我们在第九章中对系统安全性做了较全面的阐述。

本次再版的《计算机操作系统》一书共分10章。第一章仍为操作系统引论,介绍OS的发展过程、基本特征和功能,新增加了OS的结构设计;第二、三章详细地阐述了进程和线程的基本概念、进程控制、同步与通信以及调度与死锁,增加了线程的控制、线程的同步与通信;第四章为存储器管理,内容有连续分配、离散式分配存储管理方式和虚拟存储器;第五、六章分别为设备管理和文件管理;第七章介绍操作系统接口,其中,增加了UNIX系统的Shell语言和系统调用的实现方法;第八章为网络操作系统,扼要地介绍了计算机网络的基本概念,网络OS的工作模式、功能和提供的服务,以及Internet/Intranet;第九章对保障系统和网络安全的存取控制、认证、数据加密和防火墙四大技术做了较详细的阐述;第十章介绍了当前广泛使用的OS实例——UNIX系统内核结构。

本教材在编写过程中,得到了西安电子科技大学出版社的大力支持与合作。此外,汤蓓莉、王侃雅等同志在整理、校对、绘图等工作中,都付出了艰辛的劳动,使本教材能如期地与读者见面。在此谨向以上各位表示衷心感谢。

本教材虽经多次修改,突出了操作系统的基本概念,反映了当代操作系统的新技术,但限于编者水平,在本次编写的教材中,仍难免会有错误和不当之处,恳请读者批评指正。

编者
2000年12月

目 录

第一章 操作系统引论

1.1 操作系统的目标和作用	1	1.3.3 虚拟技术	16
1.1.1 操作系统的目标	1	1.3.4 异步性	17
1.1.2 操作系统的作用	2	1.4 操作系统的主要功能	18
1.1.3 推动操作系统发展的主要动力	4	1.4.1 处理机管理功能	18
1.2 操作系统的发展过程	5	1.4.2 存储器管理功能	19
1.2.1 无操作系统的计算机系统	5	1.4.3 设备管理功能	21
1.2.2 单道批处理系统	6	1.4.4 文件管理功能	21
1.2.3 多道批处理系统	7	1.4.5 操作系统与用户之间的接口	22
1.2.4 分时系统	9	1.5 OS 结构设计	24
1.2.5 实时系统	11	1.5.1 传统的操作系统结构	24
1.2.6 微机操作系统的发展	12	1.5.2 客户/服务器模式	26
1.3 操作系统的基本特性	14	1.5.3 面向对象的程序设计	27
1.3.1 并发性	14	1.5.4 微内核 OS 结构	29
1.3.2 共享性	15	习题	33

第二章 进程管理

2.1 进程的基本概念	34	2.4 经典进程的同步问题	58
2.1.1 程序的顺序执行及其特征	34	2.4.1 生产者—消费者问题	58
2.1.2 前趋图	35	2.4.2 哲学家进餐问题	61
2.1.3 程序的并发执行及其特征	36	2.4.3 读者—写者问题	63
2.1.4 进程的特征与状态	37	2.5 进程通信	65
2.1.5 进程控制块	41	2.5.1 进程通信的类型	65
2.2 进程控制	43	2.5.2 消息传递通信的实现方法	66
2.2.1 进程的创建	43	2.5.3 消息传递系统实现中的若干问题	68
2.2.2 进程的终止	45	2.5.4 消息缓冲队列通信机制	69
2.2.3 进程的阻塞与唤醒	46	2.6 线程	71
2.2.4 进程的挂起与激活	47	2.6.1 线程的基本概念	72
2.3 进程同步	47	2.6.2 线程间的同步和通信	75
2.3.1 进程同步的基本概念	47	2.6.3 线程的实现方式	77
2.3.2 信号量机制	50	2.6.4 线程的实现	78
2.3.3 信号量的应用	53	习题	81
2.3.4 管程机制	55		

第三章 处理机调度与死锁

3.1 处理机调度的层次	84	3.4.2 实时调度算法的分类	99
3.1.1 高级调度	84	3.4.3 常用的几种实时调度算法	100
3.1.2 低级调度	86	3.5 产生死锁的原因和必要条件	103
3.1.3 中级调度	87	3.5.1 产生死锁的原因	103
3.2 调度队列模型和调度准则	88	3.5.2 产生死锁的必要条件	105
3.2.1 调度队列模型	88	3.5.3 处理死锁的基本方法	105
3.2.2 选择调度方式和调度算法的若干 准则	90	3.6 预防死锁的方法	106
3.3 调度算法	91	3.6.1 预防死锁	106
3.3.1 先来先服务和短作业(进程)优先 调度算法	91	3.6.2 系统安全状态	107
3.3.2 高优先权优先调度算法	93	3.6.3 利用银行家算法避免死锁	108
3.3.3 基于时间片的轮转调度算法	95	3.7 死锁的检测与解除	111
3.4 实时调度	97	3.7.1 死锁的检测	111
3.4.1 实现实时调度的基本条件	97	3.7.2 死锁的解除	113
		习题	114

第四章 存储器管理

4.1 存储器的层次结构	116	4.4.3 两级和多级页表	133
4.1.1 多级存储器结构	116	4.5 基本分段存储管理方式	135
4.1.2 主存储器与寄存器	117	4.5.1 分段存储管理方式的引入	135
4.1.3 高速缓存和磁盘缓存	117	4.5.2 分段系统的基本原理	136
4.2 程序的装入和链接	118	4.5.3 信息共享	138
4.2.1 程序的装入	118	4.5.4 段页式存储管理方式	140
4.2.2 程序的链接	120	4.6 虚拟存储器的基本概念	141
4.3 连续分配方式	121	4.6.1 虚拟存储器的引入	142
4.3.1 单一连续分配	121	4.6.2 虚拟存储器的实现方法	143
4.3.2 固定分区分配	122	4.6.3 虚拟存储器的特征	144
4.3.3 动态分区分配	123	4.7 请求分页存储管理方式	144
4.3.4 伙伴系统	126	4.7.1 请求分页中的硬件支持	144
4.3.5 哈希算法	126	4.7.2 内存分配策略和分配算法	147
4.3.6 可重定位分区分配	127	4.7.3 调页策略	148
4.3.7 对换	129	4.8 页面置换算法	149
4.4 基本分页存储管理方式	130	4.8.1 最佳置换算法和先进先出置换算法	150
4.4.1 页面与页表	130	4.8.2 最近最久未使用(LRU)置换算法	151
4.4.2 地址变换机构	131	4.8.3 Clock 置换算法	153

4.8.4 其它置换算法	154	4.9.2 分段的共享与保护	157
4.9 请求分段存储管理方式	155	习题	159
4.9.1 请求分段中的硬件支持	155		

第五章 设备管理

5.1 I/O 系统	160	5.4.1 I/O 软件的设计目标和原则	177
5.1.1 I/O 设备	160	5.4.2 中断处理程序	179
5.1.2 设备控制器	162	5.4.3 设备驱动程序	181
5.1.3 I/O 通道	164	5.4.4 设备独立性软件	184
5.1.4 总线系统	166	5.4.5 用户层的 I/O 软件	186
5.2 I/O 控制方式	167	5.5 设备分配	186
5.2.1 程序 I/O 方式	167	5.5.1 设备分配中的数据结构	186
5.2.2 中断驱动 I/O 控制方式	168	5.5.2 设备分配时应考虑的因素	187
5.2.3 直接存储器访问(DMA)I/O 控制 方式	169	5.5.3 独占设备的分配程序	188
5.2.4 I/O 通道控制方式	170	5.5.4 SPOOLing 技术	189
5.3 缓冲管理	171	5.6 磁盘存储器的管理	191
5.3.1 缓冲的引入	171	5.6.1 磁盘性能简述	191
5.3.2 单缓冲和双缓冲	172	5.6.2 磁盘调度	194
5.3.3 循环缓冲	174	5.6.3 磁盘高速缓存	197
5.3.4 缓冲池	175	5.6.4 提高磁盘 I/O 速度的其它方法	199
5.4 I/O 软件	177	5.6.5 廉价磁盘冗余阵列	200
		习题	202

第六章 文件管理

6.1 文件和文件系统	203	6.3.3 FAT 和 NTFS 技术	216
6.1.1 文件、记录和数据项	203	6.3.4 索引分配	221
6.1.2 文件类型和文件系统模型	205	6.4 目录管理	223
6.1.3 文件操作	206	6.4.1 文件控制块和索引结点	224
6.2 文件的逻辑结构	208	6.4.2 目录结构	226
6.2.1 文件逻辑结构的类型	208	6.4.3 目录查询技术	229
6.2.2 顺序文件	209	6.5 文件存储空间的管理	231
6.2.3 索引文件	210	6.5.1 空闲表法和空闲链表法	231
6.2.4 索引顺序文件	211	6.5.2 位示图法	232
6.2.5 直接文件和哈希文件	212	6.5.3 成组链接法	233
6.3 外存分配方式	213	6.6 文件共享与文件保护	234
6.3.1 连续分配	213	6.6.1 基于索引结点的共享方式	234
6.3.2 链接分配	215	6.6.2 利用符号链实现文件共享	236

6.6.3 磁盘容错技术	237	6.7.3 并发控制	243
6.7 数据一致性控制	240	6.7.4 重复数据的数据一致性问题	243
6.7.1 事务	241	习题	246
6.7.2 检查点	242		

第七章 操作系统接口

7.1 联机用户接口	248	7.3.3 POSIX 标准	265
7.1.1 联机用户接口	248	7.3.4 系统调用的实现	266
7.1.2 联机命令的类型	250	7.4 UNIX 系统调用	268
7.1.3 键盘终端处理程序	252	7.4.1 UNIX 系统调用的类型	269
7.1.4 命令解释程序	254	7.4.2 被中断进程的环境保护	271
7.2 Shell 命令语言	255	7.4.3 系统调用陷入后需处理的 公共问题	272
7.2.1 简单命令	255	7.5 图形用户接口	273
7.2.2 重定向与管道命令	258	7.5.1 图形化用户界面	273
7.2.3 通信命令	259	7.5.2 桌面、图标和任务栏	274
7.2.4 后台命令	260	7.5.3 窗口	276
7.3 系统调用	260	7.5.4 对话框	277
7.3.1 系统调用的基本概念	261	习题	279
7.3.2 系统调用的类型	263		

第八章 网络操作系统

8.1 计算机网络概述	281	8.4.1 两层结构客户/服务器模式的 局限性	304
8.1.1 计算机网络的拓扑结构	281	8.4.2 三层结构的客户/服务器模式	305
8.1.2 计算机广域网络	284	8.4.3 两层客户/服务器与三层客户/服务器的 比较	306
8.1.3 计算机局域网	287	8.4.4 浏览器/服务器(Browser/Server) 模式	307
8.1.4 网络互连	288	8.5 网络操作系统的功能	308
8.2 网络体系结构	290	8.5.1 数据通信功能	308
8.2.1 网络体系结构的基本概念	290	8.5.2 网络资源共享功能	309
8.2.2 OSI/RM 中的低三层	292	8.5.3 应用互操作功能	312
8.2.3 OSI/RM 中的高四层	294	8.5.4 网络管理功能	314
8.2.4 TCP/IP 网络体系结构	295	8.6 网络操作系统提供的服务	315
8.2.5 LAN 网络体系结构	297	8.6.1 域名系统(DNS)	315
8.3 Internet 与 Intranet	299	8.6.2 目录服务	317
8.3.1 Internet 简介	300	8.6.3 支持 Internet 提供的服务	319
8.3.2 Internet 提供的传统信息服务	301		
8.3.3 Web 服务	303		
8.4 客户/服务器模式	304		

习题	320
----------	-----

第九章 系统安全性

9.1 系统安全的基本概念	322	9.3.3 基于生物标志的认证技术	337
9.1.1 系统安全性的内容和性质	322	9.3.4 基于公开密钥的认证技术	339
9.1.2 系统安全威胁的类型	323	9.4 访问控制技术	340
9.1.3 信息技术安全评价公共准则	324	9.4.1 访问矩阵	340
9.2 数据加密技术	325	9.4.2 访问矩阵的修改	342
9.2.1 数据加密的基本概念	325	9.4.3 访问控制矩阵的实现	343
9.2.2 对称加密算法与非对称加密算法	328	9.5 计算机病毒	345
9.2.3 数字签名和数字证书	329	9.5.1 计算机病毒的基本概念	345
9.2.4 网络加密技术	331	9.5.2 计算机病毒的类型	346
9.3 认证技术	332	9.5.3 病毒的隐藏方式	348
9.3.1 基于口令的身份认证	333	9.5.4 病毒的预防和检测	350
9.3.2 基于物理标志的认证技术	335	习题	351

第十章 UNIX 系统内核结构

10.1 UNIX 系统概述	353	10.4.1 请求调页管理的数据结构	370
10.1.1 UNIX 系统的发展史	353	10.4.2 换页进程	372
10.1.2 UNIX 系统的特征	355	10.4.3 请求调页	373
10.1.3 UNIX 系统的内核结构	356	10.5 设备管理	374
10.2 进程的描述和控制	357	10.5.1 字符设备缓冲区管理	374
10.2.1 进程控制块	357	10.5.2 块设备缓冲区管理	375
10.2.2 进程状态与进程映像	359	10.5.3 内核与驱动程序接口	377
10.2.3 进程控制	361	10.5.4 磁盘驱动程序	379
10.2.4 进程调度与切换	363	10.5.5 磁盘读/写程序	380
10.3 进程的同步与通信	364	10.6 文件管理	381
10.3.1 sleep 与 wakeup 同步机制	364	10.6.1 UNIX 文件系统概述	381
10.3.2 信号机制	365	10.6.2 文件的物理结构	383
10.3.3 管道机制	365	10.6.3 索引结点的管理	385
10.3.4 消息机制	367	10.6.4 空闲磁盘空间的管理	386
10.3.5 共享存储区机制	368	10.6.5 文件表的管理	388
10.3.6 信号量集机制	369	10.6.6 目录管理	389
10.4 存储器管理	370	习题	390
参考文献	392		

第一章 操作系统引论

计算机系统由硬件和软件两部分组成。操作系统(OS, Operating System)是配置在计算机硬件上的第一层软件,是对硬件系统的首次扩充。它在计算机系统中占据了特别重要的地位;而其它的诸如汇编程序、编译程序、数据库管理系统等系统软件,以及大量的应用软件,都将依赖于操作系统的支持,取得它的服务。操作系统已成为现代计算机系统(大、中、小及微型机)、多处理机系统、计算机网络、多媒体系统以及嵌入式系统中都必须配置的、最重要的系统软件。

1.1 操作系统的目标和作用

在计算机系统上配置操作系统的主要目标,首先与计算机系统的规模有关。通常对配置在大、中型计算机系统上的 OS,由于计算机价格昂贵,因此都比较看重机器使用的有效性,而且还希望 OS 具有非常强的功能;但对于配置在微机中的操作系统,由于微机价格相对较便宜,此时机器使用的有效性也就显得不那么重要了,而人们更关注的是使用的方便性。

影响操作系统的主要目标的另一个重要因素是操作系统的应用环境。例如,对于应用在查询系统中的操作系统,应满足用户对响应时间的要求;又如对应用在实时工业控制和武器控制环境下的 OS,则要求其 OS 具有实时性和高度可靠性。

1.1.1 操作系统的目标

目前存在着多种类型的 OS,不同类型的 OS,其目标各有所侧重。一般地说,在计算机硬件上配置的 OS,其目标有以下几点。

1. 有效性

在早期(20世纪50~60年代),由于计算机系统非常昂贵,操作系统最重要的目标无疑是有效性。事实上,那时有效性是推动操作系统发展最主要的动力。正因如此,现在的大多数操作系统书籍,都着重于介绍如何提高计算机系统的资源利用率和系统的吞吐量问题。操作系统的有效性可包含如下两方面的含意:

(1) 提高系统资源利用率。在未配置 OS 的计算机系统中,诸如 CPU、I/O 设备等各种资源,都会因它们经常处于空闲状态而得不到充分利用;内存及外存中所存放的数据太少或者无序而浪费了大量的存储空间。配置了 OS 之后,可使 CPU 和 I/O 设备由于能保持忙碌状态而得到有效的利用,且可使内存和外存中存放的数据因有序而节省了存储空间。

(2) 提高系统的吞吐量。操作系统还可以通过合理地组织计算机的工作流程,而进一步改善资源的利用率,加速程序的运行,缩短程序的运行周期,从而提高系统的吞吐量。

2. 方便性

配置 OS 后可使计算机系统更容易使用。一个未配置 OS 的计算机系统是极难使用的，因为计算机硬件只能识别 0 和 1 这样的机器代码。用户要直接在计算机硬件上运行自己所编写的程序，就必须用机器语言书写程序；用户要想输入数据或打印数据，也都必须自己用机器语言书写相应的输入程序或打印程序。如果我们在计算机硬件上配置了 OS，用户便可通过 OS 所提供的各种命令来使用计算机系统。比如，用编译命令可方便地把用户用高级语言书写的程序翻译成机器代码，大大地方便了用户，从而使计算机变得易学易用。

方便性和有效性是设计操作系统时最重要的两个目标。在过去的很长一段时间内，由于计算机系统非常昂贵，因而其有效性显得比较重要。但是，近十多年来，随着硬件越来越便宜，在设计配置在微机上的 OS 时，人们似乎更重视如何使用户能更为方便地使用计算机，故在微机操作系统中都配置了受到用户广泛欢迎的图形用户界面，提供了大量的供程序员使用的系统调用。

3. 可扩充性

随着 VLSI 技术和计算机技术的迅速发展，计算机硬件和体系结构也随之得到迅速发展，相应地，它们也对 OS 提出了更高的功能和性能要求。此外，多处理机系统、计算机网络，特别是 Internet 的发展，又对 OS 提出了一系列更新的要求。因此，OS 必须具有很好的可扩充性，方能适应计算机硬件、体系结构以及应用发展的要求。这就是说，现代 OS 应采用新的 OS 结构，如微内核结构和客户服务器模式，以便于方便地增加新的功能和模块，并能修改老的功能和模块。关于新的 OS 结构将在本章最后一节中介绍。

4. 开放性

自 20 世纪 80 年代以来，由于计算机网络的迅速发展，特别是 Internet 的应用的日益普及，使计算机操作系统的应用环境已由单机封闭环境转向开放的网络环境。为使来自不同厂家的计算机和设备能通过网络加以集成化，并能正确、有效地协同工作，实现应用的可移植性和互操作性，要求操作系统必须提供统一的开放环境，进而要求 OS 具有开放性。

开放性是指系统能遵循世界标准规范，特别是遵循开放系统互连(OSI)国际标准。凡遵循国际标准所开发的硬件和软件，均能彼此兼容，可方便地实现互连。开放性已成为 20 世纪 90 年代以后计算机技术的一个核心问题，也是一个新推出的系统或软件能否被广泛应用的至关重要的因素。

1.1.2 操作系统的作用

可以从不同的观点(角度)来观察 OS 的作用。从一般用户的观点，可把 OS 看做是用户与计算机硬件系统之间的接口；从资源管理的观点看，则可把 OS 视为计算机系统资源的管理者。另外，OS 实现了对计算机资源的抽象，隐藏了对硬件操作的细节，使用户能更方便地使用机器。

1. OS 作为用户与计算机硬件系统之间的接口

OS 作为用户与计算机硬件系统之间接口的含义是：OS 处于用户与计算机硬件系统之间，用户通过 OS 来使用计算机系统。或者说，用户在 OS 帮助下，能够方便、快捷、安全、可靠地操纵计算机硬件和运行自己的程序。应注意，OS 是一个系统软件，因而这种接口是

软件接口。图 1-1 是 OS 作为接口的示意图。由图可看出,用户可通过以下三种方式使用计算机。

(1) 命令方式。这是指由 OS 提供了一组联机命令接口,以允许用户通过键盘输入有关命令来取得操作系统的服务,并控制用户程序的运行。

(2) 系统调用方式。OS 提供了一组系统调用,用户可在自己的应用程序中通过相应的系统调用,来实现与操作系统的通信,并取得它的服务。

(3) 图形、窗口方式。这是当前使用最为方便、最为广泛的接口,它允许用户通过屏幕上的窗口和图标来实现与操作系统的通信,并取得它的服务。

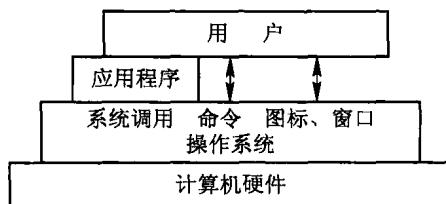


图 1-1 OS 作为接口的示意图

2. OS 作为计算机系统资源的管理者

在一个计算机系统中,通常都含有各种各样的硬件和软件资源。归纳起来可将资源分为四类:处理器、存储器、I/O 设备以及信息(数据和程序)。相应地,OS 的主要功能也正是针对这四类资源进行有效的管理,即:处理机管理,用于分配和控制处理机;存储器管理,主要负责内存的分配与回收;I/O 设备管理,负责 I/O 设备的分配与操纵;文件管理,负责文件的存取、共享和保护。可见,OS 的确是计算机系统资源的管理者。事实上,当今世界上广为流行的一个关于 OS 作用的观点,正是把 OS 作为计算机系统的资源管理者。

值得进一步说明的是,当一个计算机系统同时供多个用户使用,用户对系统中共享资源的需求(包括数量和时间)可能发生冲突,为了管理好这些共享资源(包括硬件和信息)的使用,操作系统必须记录下各种资源的使用情况,对使用资源的请求进行授权,协调诸用户对共享资源的使用,避免发生冲突,并计算使用资源的费用等。

3. OS 实现了对计算机资源的抽象

对于一个完全无软件的计算机系统(即裸机),它向用户提供的是实际硬件接口(物理接口),用户必须对物理接口的实现细节有充分的了解,并利用机器指令进行编程,因此该物理机器必定是难以使用的。为了方便用户使用 I/O 设备,人们在裸机上覆盖上一层 I/O 设备管理软件,如图 1-2 所示,由它来实现对 I/O 设备操作的细节,并向上提供一组 I/O 操作命令,如 Read 和 Write 命令,用户可利用它来进行数据输入或输出,而无需关心 I/O 是如何实现的。此时用户所看到的机器将是一台比裸机功能更强、使用更方便的机器。这就是说,在裸机上铺设的 I/O 软件隐藏了对 I/O 设备操作的具体细节,向上提供了一组抽象的 I/O 设备。

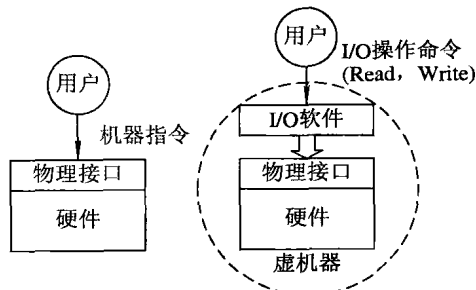


图 1-2 I/O 软件隐藏了 I/O 操作实现的细节

通常把覆盖了上述软件的机器称为扩充机器或虚机器。它向用户(进程)提供了一个对硬件操作的抽象模型,用户可利用抽象模型提供的接口使用计算机,而无需了解物理接口实现的细节,从而使用户更容易地使用计算机硬件资源。由该层软件实现了对计算机硬件操作的第一个层次的抽象。

为了方便用户使用文件系统,人们又在第一层软件上再覆盖上一层用于文件的管理软件,同样由它来实现对文件操作的细节,并向上提供一组对文件进行存取操作的命令,用户可利用这组命令进行文件的存取。此时,用户所看到的是一台功能更强、使用更方便的虚机器。该层软件实现了对硬件资源操作的第二个层次的抽象。而当人们又在文件管理软件上再覆盖一层面面向用户的窗口软件后,用户便可在窗口环境下方便地使用计算机,形成一台功能更强的虚机器。

由此可知,OS是铺设在计算机硬件上的多层系统软件,它们不仅增强了系统的功能,而且还隐藏了对硬件操作的细节,由它们实现了对计算机硬件操作的多个层次的抽象。值得说明的是,对一个硬件在底层进行抽象后,在高层还可再次对该资源进行抽象,成为更高层的抽象模型。随着抽象层次的提高,抽象接口所提供的功能就越来越强,用户使用起来也更加方便。

1.1.3 推动操作系统发展的主要动力

在出现 OS 后的短短 50 多年中,操作系统取得了重大的发展,操作系统的功能已变得非常强大,所提供的服务也十分有效,用户使用更加方便。相应地,操作系统的规模也由早期的数十 KB 发展到如今的数千万行代码,甚至更多。推动操作系统发展的主要动力,可归结为如下所述的四个方面。

1. 不断提高计算机资源的利用率

在计算机发展的初期,计算机系统特别昂贵,人们必须千方百计地提高计算机系统中各种资源的利用率,这就是 OS 最初发展的推动力。由此形成了能自动地对一批作业进行处理的多道批处理系统。在 20 世纪 60 和 70 年代,又分别出现了能有效提高 I/O 设备和 CPU 利用率的 SPOOLing 系统和改善存储器系统利用率的虚拟存储器技术,以及在网络环境下,在服务器上配置了允许所有网络用户访问的文件系统和数据库系统。

2. 方便用户

当资源利用率不高的问题得到基本解决后,用户在上机、调试程序时的不方便性便又成为主要矛盾。于是人们又想方设法改善用户上机、调试程序时的环境,这又成为继续推动 OS 发展的主要因素。随之便形成了允许进行人机交互的分时系统,或称为多用户系统。在 20 世纪 90 年代初出现了受到用户广泛欢迎的图形用户界面,极大地方便了用户使用计算机,使中小學生都能很快地学会上机操作,这无疑会更加推动计算机的迅速普及。

3. 器件的不断更新换代

微电子技术的迅猛发展,推动着计算机器件,特别是微芯片的不断更新,使得计算机的性能迅速提高,规模急剧扩大,从而推动了 OS 的功能和性能也迅速增强和提高。例如,当微芯片由 8 位发展到 16 位、32 位,进而又发展到 64 位时,相应的微机 OS 也就由 8 位发展到 16 位和 32 位,进而又发展到 64 位,此时相应 OS 的功能和性能也都有显著的增

强和提高。

在多处理机快速发展的同时，外部设备也在迅速发展。例如，早期的磁盘系统十分昂贵，只能配置在大型机中。随着磁盘价格的不断降低且小型化，很快在中、小型机以及微型机上也无一例外地配置了磁盘系统，而且其容量还远比早期配置在大型机上的大得多。现在的微机操作系统(如 Windows XP)能支持种类非常多的外部设备，除了传统的外设外，还可以支持光盘、移动硬盘、闪存盘、扫描仪等。

4. 计算机体系结构的不断发展

计算机体系结构的发展，也不断推动着 OS 的发展并产生新的操作系统类型。例如，当计算机由单处理机系统发展为多处理机系统时，相应地，操作系统也就由单处理机 OS 发展为多处理机 OS。又如，当出现了计算机网络后，配置在计算机网络上的网络操作系统也就应运而生，它不仅能有效地管理好网络中的共享资源，而且还向用户提供了许多网络服务。

1.2 操作系统的发展过程

OS 的形成迄今已有 50 多年的时间。在上世纪 50 年代中期出现了单道批处理操作系统；60 年代中期产生了多道程序批处理系统；不久又出现了基于多道程序的分时系统，与此同时也诞生了用于工业控制和武器控制的实时操作系统。20 世纪 80 年代开始至 21 世纪初，是微型机、多处理机和计算机网络高速发展的年代，同时也是微机 OS、多处理机 OS 和网络 OS 以及分布式 OS 的形成和大发展的年代。本节主要介绍早期的操作系统发展，在本章 1.5 节中再对微机 OS、多处理机 OS、网络 OS 和分布式 OS 等作简单的阐述。

1.2.1 无操作系统的计算机系统

1. 人工操作方式

从第一台计算机诞生(1945 年)到 20 世纪 50 年代中期的计算机，属于第一代计算机。此时的计算机是利用成千上万个真空管做成的，它的运行速度仅为每秒数千次，但体积却十分庞大，且功耗也非常高。这时还未出现 OS。计算机操作是由用户(即程序员)采用人工操作方式直接使用计算机硬件系统，即由程序员将事先已穿孔(对应于程序和数据)的纸带(或卡片)装入纸带输入机(或卡片输入机)，再启动它们将程序和数据输入计算机，然后启动计算机运行。当程序运行完毕并取走计算结果之后，才让下一个用户上机。这种人工操作方式有以下两方面的缺点：

(1) 用户独占全机。此时，计算机及其全部资源只能由上机用户独占。

(2) CPU 等待人工操作。当用户进行装带(卡)、卸带(卡)等人工操作时，CPU 及内存等资源是空闲的。

可见，人工操作方式严重降低了计算机资源的利用率，此即所谓的人机矛盾。随着 CPU 速度的提高和系统规模的扩大，人机矛盾变得日趋严重。此外，随着 CPU 速度的迅速提高而 I/O 设备的速度却提高缓慢，这又使 CPU 与 I/O 设备之间速度不匹配的矛盾更加突出。为了缓和此矛盾，曾先后出现了通道技术、缓冲技术，但都未能很好地解决上述矛盾，直至后来又引入了脱机输入/输出技术，才获得了较为令人满意的结果。

2. 脱机输入/输出方式

为了解决人机矛盾及 CPU 和 I/O 设备之间速度不匹配的矛盾, 20 世纪 50 年代末出现了脱机输入/输出(Off-Line I/O)技术。该技术是事先将装有用户程序和数据的纸带(或卡片)装入纸带输入机(或卡片机), 在一台外围机的控制下, 把纸带(卡片)上的数据(程序)输入到磁带上。当 CPU 需要这些程序和数据时, 再从磁带上将其高速地调入内存。

类似地, 当 CPU 需要输出时, 可由 CPU 直接高速地把数据从内存送到磁带上, 然后在另一台外围机的控制下, 将磁带上的结果通过相应的输出设备输出。图 1-3 示出了脱机输入/输出过程。由于程序和数据的输入和输出都是在外围机的控制下完成的, 或者说, 它们是在脱离主机的情况下进行的, 故称为脱机输入/输出方式; 反之, 在主机的直接控制下进行输入/输出的方式称为联机输入/输出(On-Line I/O)方式。这种脱机 I/O 方式的主要优点如下:

(1) 减少了 CPU 的空闲时间。装带(卡)、卸带(卡)以及将数据从低速 I/O 设备送到高速磁带(或盘)上, 都是在脱机情况下进行的, 并不占用主机时间, 从而有效地减少了 CPU 的空闲时间, 缓和了人机矛盾。

(2) 提高了 I/O 速度。当 CPU 在运行中需要数据时, 是直接从高速的磁带或磁盘上将数据调入内存的, 不再是从低速 I/O 设备上输入, 极大地提高了 I/O 速度, 从而缓和了 CPU 和 I/O 设备速度不匹配的矛盾, 进一步减少了 CPU 的空闲时间。

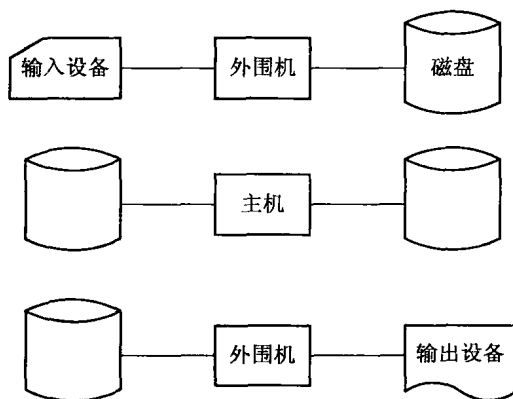


图 1-3 脱机 I/O 示意图

1.2.2 单道批处理系统

1. 单道批处理系统的处理过程

上世纪 50 年代中期发明了晶体管, 人们开始用晶体管替代真空管来制作计算机, 从而出现了第二代计算机。它不仅使计算机的体积大大减小, 功耗显著降低, 同时可靠性也得到大幅度提高, 使计算机已具有推广应用的价值, 但计算机系统仍非常昂贵。为了能充分地利用它, 应尽量让该系统连续运行, 以减少空闲时间。为此, 通常是把一批作业以脱机方式输入到磁带上, 并在系统中配上监督程序(Monitor), 在它的控制下使这批作业能一个接一个地连续处理。其自动处理过程是: 首先, 由监督程序将磁带上的第一个作业装入内存, 并把运行控制权交给该作业。当该作业处理完成时, 又把控制权交还给监督程序, 再由监督程序把磁带(盘)上的第二个作业调入内存。计算机系统就这样自动地一个作业一个作业地进行处理, 直至磁带(盘)上的所有作业全部完成, 这样便形成了早期的批处理系统。由于系统对作业的处理都是成批地进行的, 且在内存中始终只保持一道作业, 故称此系统为单道批处理系统(Simple Batch Processing System)。图 1-4 示出了单道批处理系统的处理流程。

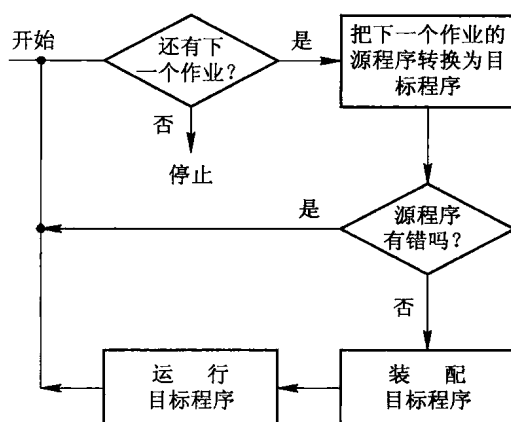


图 1-4 单道批处理系统的处理流程

由上所述不难看出，单道批处理系统是在解决人机矛盾以及 CPU 与 I/O 设备速度不匹配问题的过程中形成的。换言之，批处理系统旨在提高系统资源的利用率和系统吞吐量。但这种单道批处理系统仍然不能很好地利用系统资源，故现已很少使用。

2. 单道批处理系统的特征

单道批处理系统是最早出现的一种 OS。严格地说，它只能算是 OS 的前身而并非是目前人们所理解的 OS。尽管如此，该系统比起人工操作方式的系统已有很大进步。该系统的主要特征如下：

(1) 自动性。在顺利情况下，在磁带上的一批作业能自动地逐个地依次运行，而无需人工干预。

(2) 顺序性。磁带上的各道作业是顺序地进入内存，各道作业的完成顺序与它们进入内存的顺序，在正常情况下应完全相同，亦即先调入内存的作业先完成。

(3) 单道性。内存中仅有一道程序运行，即监督程序每次从磁带上只调入一道程序进入内存运行，当该程序完成或发生异常情况时，才换入其后继程序进入内存运行。

1.2.3 多道批处理系统

20 世纪 60 年代中期，人们开始利用小规模集成电路来制作计算机，生产出第三代计算机。由 IBM 公司生产的第一台小规模集成电路计算机——360 机，较之于晶体管计算机，无论在体积、功耗、速度和可靠性上，都有了显著的改善。虽然在开发 360 机器使用的操作系统时，为能在机器上运行多道程序而遇到了极大的困难，但最终还是成功地开发出能在一台机器中运行多道程序的操作系统 OS/360。

1. 多道程序设计的基本概念

在单道批处理系统中，内存中仅有一道作业，它无法充分利用系统中的所有资源，致使系统性能较差。为了进一步提高资源的利用率和系统吞吐量，在 20 世纪 60 年代中期又引入了多道程序设计技术，由此而形成了多道批处理系统(Multiprogrammed Batch Processing System)。在该系统中，用户所提交的作业都先存放在外存上并排成一个队列，称为“后备队列”；然后，由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存，使它