



中国计算机学会教育专业委员会 推荐
全国高等学校计算机教育研究会 出版
高等学校规划教材

现代操作系统

汤小丹 梁红兵 哲凤屏 汤子瀛 编著

软件 工程 课程 群



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

TP316/440

2008

高等学校规划教材

现代操作系统

汤小丹 梁红兵 编著
哲凤屏 汤子瀛

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本教材对传统和现代操作系统进行了较为全面的介绍。本书共分 17 章：第 1 章为操作系统引论，介绍了操作系统的发展，传统和现代操作系统的特征和功能；第 2~4 章深入地阐述了进程（线程）管理、进程同步、处理机调度和死锁；第 5 章和第 6 章介绍连续和离散存储器管理方式及虚拟存储器；第 7 章自下而上地对 I/O 系统的各个层次，进行了较系统的阐述；第 8 章和第 9 章介绍了文件系统和磁盘存储器管理；第 10 章介绍用户接口，以及接口的实现方法；第 11~15 章，其内容都是与现代操作系统直接相关的，分别介绍多处理机操作系统、网络操作系统、多媒体操作系统和操作系统结构设计及系统安全性；第 16 章和第 17 章介绍了一个 OS 实例 Linux。

本书可作为高等学校计算机等专业本科生、研究生教材，也可作为计算机及通信领域相关科技人员的参考书。

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

现代操作系统/汤小丹等编著. —北京：电子工业出版社，2008.4

高等学校规划教材

ISBN 978-7-121-05658-1

I. 现… 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 196799 号

责任编辑：韩同平 特约编辑：李佩乾

印 刷：北京季蜂印刷有限公司

装 订：三河市万和装订厂

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：26.25 字数：758.5 千字

印 次：2008 年 4 月第 1 次印刷

印 数：4000 册 定价：38.00 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

《现代操作系统》教材的前身是《计算机操作系统》，它于 1981 年由西安电讯工程学院（现为西安电子科技大学）出版社出版，1984 年对它进行了重大修改，作为电子工业部规划教材出版。该教材经过多次修编，曾获得原电子工业部优秀教材二等奖、一等奖，以及国家级教学成果二等奖。

尽管该教材获得广泛的好评，但由于操作系统（OS）是计算机领域中最活跃的学科之一，其发展速度又极为迅速，特别是 20 世纪 90 年代以后，在信息工程领域中，发生了以下几方面的巨大的变化：

（1）计算机硬件的发展，特别是处理机芯片的集成度有了突飞猛进的发展，相应的运行速度也有极大提高，其时钟频率从 20 世纪 80 年代的数 MHz 提高到如今的数 GHz，内存从 20 世纪 80 年代的数十 KB 扩大到如今的数 GB。它为 OS 的进一步发展奠定了坚实的硬件基础。

（2）计算机网络的高速发展，特别是 Internet 的兴起，它为政府机关、企事业单位等，提供了许多有用而高效的服务，为科技的进步和经济的发展起到了巨大的推动作用。为使人们能方便地获取 Internet 所提供的各种服务，在 OS 中又增加了面向网络的功能。

（3）多媒体的应用。各种数字化的多媒体设备，如 DVD、扫描仪等设备的出现，极大地丰富了人们的文化生活，增强了用户欲利用一台计算机来对多种媒体进行播放的强烈愿望，这促使在 OS 上又增添了能播放音频和视频信号的软件。

（4）系统安全性。21 世纪人类已进入了信息社会，把越来越多的、极其重要的机密信息保存在计算机系统中，于是也就引发了别有用心的人，采用各种方法和通过多种途径，对计算机系统进行攻击和破坏。因此，确保系统的安全已成为 OS 责无旁贷的任务。

（5）软件开发方法的发展。随着软件规模的急剧扩大，为使所开发出的软件产品具有良好的质量和合理的费用，开发方法也在迅速地发展。而利用不同的开发方法所开发出的 OS 具有不同的结构，从早期的模块化结构、分层结构，发展为微内核结构。现代 OS 广泛采用了微内核结构。

为了能适应上述的变化，在进入 20 世纪 90 年代后，在操作系统中广泛引入了线程，来进一步提高程序执行的并发性，增加了很强的面向网络的功能，配置了能播放音频和视频信号的多媒体软件，在 OS 结构上广泛采用微内核结构；此外还采取了多种有效措施来确保系统的安全可靠。可见，OS 在功能和特征上都发生了巨大的变化。人们把具有上述功能和特征的 OS 称为现代 OS。

不谋而合的是，电子工业出版社韩中平编辑也看到了 OS 的快速发展，很希望出版一本能较及时地反映 OS 学科的前沿，对现代 OS 做较全面的阐述，以便较全面地反映当今 OS 状况的操作系统教材。这就是本教材的编写背景和基础。

本书共分 17 章，从内容上可分为三个部分：

第一部分，即第 1~10 章，主要介绍传统的操作系统。第 1 章为操作系统引论，介绍了传统 OS 的作用与发展过程，其所具有的特征和功能等，以及现代 OS 中引入的新特征和功能。第 2 章深入地阐述了进程和线程的基本概念，进程控制，线程的实现方法。第 3 章对进程同步的基本概念，与用于实现进程同步和通信的多种方法与机制，做了详细的描述。第 4 章分为两个部分：第一部分是关于处理机的调度，其内容包含了处理机调度中的三个层次，即高级调度、低级调度和中级调度，以及用于实时系统中的实时调度；第二部分是死锁问题，介绍了死锁的基本概念，如何预防和避免死锁，以及如何对死锁进行检测与解除。第 5 章对三种基本的存储器管理方式，即连续分配存储管理方式、分页存储管理方式和分段存储管理方式进行了详细的介绍；第 6 章在介绍了虚拟存储器的基本概念之后，

又分别介绍了基于分页存储管理方式和分段存储管理方式实现的虚拟存储器。第7章为I/O管理,自下而上地对I/O系统的各个层次,如中断处理程序、设备驱动程序、与设备无关的I/O软件 and 用户层的I/O软件等做了较系统的阐述。第8章是文件管理,介绍了文件的逻辑结构、文件目录、文件共享和保护。第9章对文件在外存上的组织方式、文件存储空间的管理,以及如何提高磁盘的I/O速度和可靠性等进行了详细的描述。第10章是OS接口,对用户命令接口和该接口的实现方法,程序接口及该接口的实现方法,做了系统的介绍。

第二部分内容都是与现代OS直接相关的,分为5章,即第11~15章。第11章介绍配置在多处理机(器)系统中的OS,其中包含多处理器系统的结构、多处理机操作系统的特征、实现进程同步方法和机制,以及多处理器系统中的进程(线程)调度。第12章介绍网络OS,先介绍了计算机网络的若干基本知识和概念,然后介绍网络操作系统的功能和它所提供的网络服务。第13章是多媒体OS,在对多媒体系统和多媒体文件中的各种媒体做了扼要的介绍后,对多媒体进程管理、存储器管理和多媒体服务器中的接纳控制,多媒体进程的实时调度,多媒体存储器的分配方法等,进行了详细的阐述。第14章为OS结构设计,介绍了引入OS结构设计的必要性,几种传统的操作系统结构设计,以及面向对象的程序设计,最后重点描述了在现代OS中常用的微内核OS结构。第15章是保护和安全的,介绍了保障系统安全的几种常用技术,如数据加密技术、用户验证技术、防范病毒的技术,以及可信系统。

第三部分包括第16章和第17章两章,介绍了一个目前广为流行的OS实例,即Linux。第16章主要介绍了进程和存储器管理子系统,介绍了Linux系统的进程机制、同步、通信和调度,以及Linux OS的存储管理。第17章介绍Linux系统中的文件管理和设备管理子系统。

本教材的内容较为丰富,它既可作为计算机硬件、软件、计算机应用和计算机通信等专业的本科生教材,也可作为研究生教材,以及从事计算机及通信工作的相关科技人员的参考书。在使用该教材时可分为两种情况:(1)若OS的总课时数达到100~110学时,可全面讲授本教材;(2)如果OS的课时数只有70学时左右,则只需重点介绍其中的第1~10章和第15章,后面部分供学生在老师的指导下自行阅读,以便能了解现代OS所具有的特征和功能。

本教材在编写过程中,得到电子工业出版社,特别是韩同平编辑的大力支持与合作。此外,王佩雅和汤蓓莉等同志,在校对、整理等工作中都付出了辛勤的劳动,使本教材能如期出版。在此谨向以上各位表示衷心感谢。

虽然本教材与作者经过多次修订的教材——《计算机操作系统》有着一定的关系,但本教材无论是从结构上还是内容上都做了重大的修改和更新,特别是本教材又增加或加强了与现代OS相关的章节,如第11~15章,以及广为流行的Linux OS实例,即第16章和第17章。我们希望并尽力把这一本新的现代OS教材写得更好,但限于编者的水平,加上时间紧迫,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。

(作者的电子邮件地址: danxiaotang@163.com)

(本书配有免费电子课件,可登录华信教育资源网www.huaxin.edu.cn下载)

编者

目 录

第 1 章 操作系统引论	1	3.1 进程同步的基本概念	48
1.1 操作系统的目标和作用	1	3.2 信号量机制	52
1.2 操作系统的发展过程	4	3.2.1 整型信号量和记录型信号量	52
1.3 操作系统的基本特性	9	3.2.2 AND 型信号量和信号量集	54
1.4 现代操作系统的新特征	12	3.2.3 信号量的应用	55
1.4.1 传统 OS 演变为现代 OS 的 主要因素	12	3.3 管程机制	57
1.4.2 多线程	13	3.3.1 管程的定义	57
1.4.3 微内核	14	3.3.2 条件变量	58
1.4.4 网络化	15	3.3.3 利用管程解决生产者-消费者 问题	59
1.4.5 多媒体	16	3.4 经典进程的同步问题	60
1.5 传统操作系统的主要功能	17	3.4.1 生产者-消费者问题	60
1.6 现代操作系统新增功能	21	3.4.2 哲学家进餐问题	61
1.6.1 系统安全管理	21	3.4.3 读者-写者问题	63
1.6.2 面向网络的功能和服务	22	3.5 进程通信	65
1.6.3 多媒体功能	23	3.5.1 进程通信的类型	65
习题	24	3.5.2 直接消息传递系统	66
第 2 章 进程的描述与控制	26	3.5.3 信箱通信	67
2.1 前趋图和程序执行	26	3.5.4 直接消息传递系统实例	68
2.2 进程的描述	29	3.5.5 线程间的同步和通信	70
2.2.1 进程的定义和特征	29	习题	71
2.2.2 进程的基本状态及转换	30	第 4 章 处理机调度与死锁	73
2.2.3 挂起操作和进程状态的转换	31	4.1 处理机调度的层次和调度算法的目标	73
2.2.4 进程管理中的数据结构	32	4.2 作业与作业调度	75
2.3 进程控制	34	4.2.1 批处理系统中的作业及调度	75
2.4 线程的基本概念	39	4.2.2 先来先服务和短作业优先调度 算法	76
2.4.1 线程的引入	39	4.2.3 优先级调度算法和高响应比优先 调度算法	77
2.4.2 线程与进程的比较	40	4.3 进程调度	78
2.4.3 线程的状态和线程控制块	41	4.3.1 进程调度的任务、机制和方式	78
2.4.4 线程的类型和多线程模型	42	4.3.2 轮转调度算法	79
2.5 线程的实现	43	4.3.3 优先级调度算法	80
2.5.1 线程的实现方式	43	4.3.4 多级反馈队列调度算法	81
2.5.2 线程的实现	44	4.3.5 基于公平原则的调度算法	82
2.5.3 线程的创建和终止	46	4.4 实时调度	83
习题	47		
第 3 章 进程的同步与通信	48		

4.4.1	实现实时调度的基本条件	83	5.6.2	分段系统的基本原理	121
4.4.2	实时调度算法的分类	84	5.6.3	信息共享	123
4.4.3	最早截止时间优先算法	84	5.6.4	段页式存储管理方式	124
4.4.4	最低松弛度优先算法	86	习题		125
4.4.5	优先级倒置	87	第6章 虚拟存储器		127
4.5	死锁概述	88	6.1	虚拟存储器概述	127
4.5.1	资源问题	88	6.1.1	传统存储管理方式的特征和 局部性原理	127
4.5.2	计算机系统死锁	89	6.1.2	虚拟存储器的定义和特征	128
4.5.3	死锁的定义、必要条件和 处理方法	90	6.1.3	虚拟存储器的实现方法	129
4.6	预防死锁	91	6.2	请求分页存储管理方式	130
4.7	避免死锁	93	6.2.1	请求分页中的硬件支持	130
4.7.1	系统安全状态	93	6.2.2	请求分页中的内存分配	131
4.7.2	利用银行家算法避免死锁	94	6.2.3	页面调入策略	133
4.8	死锁的检测与解除	96	6.3	页面置换算法	134
4.8.1	死锁的检测	96	6.3.1	最佳置换算法和先进先出 置换算法	135
4.8.2	死锁的解除	98	6.3.2	最近最久未使用和最少使用 置换算法	136
习题		99	6.3.3	Clock 置换算法	137
第5章 存储器管理		101	6.3.4	页面缓冲算法	139
5.1	存储器的层次结构	101	6.4	“抖动”与工作集	139
5.2	程序的装入和链接	103	6.5	请求分段存储管理方式	142
5.2.1	程序的装入	103	习题		145
5.2.2	程序的链接	104	第7章 输入/输出系统		147
5.3	连续分配存储管理方式	105	7.1	I/O 系统的功能、模型和接口	147
5.3.1	单一连续分配	105	7.1.1	I/O 系统的基本功能	147
5.3.2	固定分区分配	106	7.1.2	I/O 系统的层次结构和模型	148
5.3.3	动态分区分配	107	7.1.3	I/O 系统接口	150
5.3.4	基于顺序搜索的分区分配 算法	108	7.2	I/O 设备和设备控制器	151
5.3.5	基于索引搜索的分区分配 算法	109	7.3	中断机构和中断处理程序	154
5.3.6	可重定位分区分配	111	7.4	设备驱动程序	156
5.4	对换	112	7.4.1	设备驱动程序概述	157
5.5	分页存储管理方式	115	7.4.2	设备驱动程序的处理过程	157
5.5.1	分页存储管理的几个基本 概念	115	7.4.3	对 I/O 设备的控制方式	159
5.5.2	地址变换机构	116	7.5	与设备无关的 I/O 软件	161
5.5.3	两级和多级页表	118	7.5.1	基本概念	161
5.5.4	反置页表工程	119	7.5.2	实现与设备无关性的软件	162
5.6	分段存储管理方式	120	7.5.3	设备分配	163
5.6.1	分段存储管理方式的引入	120	7.5.4	逻辑设备名到物理设备名 映射的实现	165

7.6 用户层的 I/O 软件	166	9.2 文件存储空间的管理	212
7.6.1 系统调用与库函数	166	9.3 提高磁盘 I/O 速度的途径	215
7.6.2 假脱机系统	167	9.4 提高磁盘可靠性的技术	218
7.6.3 守护进程	168	9.5 数据一致性控制	221
7.7 缓冲区管理	169	习题	225
7.7.1 缓冲的引入	169	第 10 章 操作系统接口	227
7.7.2 单缓冲区和双缓冲区	170	10.1 用户接口	227
7.7.3 环形缓冲区	171	10.2 Shell 命令语言	230
7.7.4 缓冲池	172	10.2.1 简单命令	231
7.8 磁盘存储器的性能和调度	173	10.2.2 其他常用命令	233
7.8.1 磁盘性能简述	174	10.3 联机命令接口的实现	235
7.8.2 早期的磁盘调度算法	176	10.3.1 键盘终端处理程序	236
7.8.3 基于扫描的磁盘调度算法	177	10.3.2 命令解释程序	237
习题	178	10.4 系统调用的概念和类型	241
第 8 章 文件管理	180	10.5 UNIX 系统调用	244
8.1 文件和文件系统	180	10.6 系统调用的实现	246
8.2 文件的逻辑结构	183	10.6.1 系统调用的实现方法	247
8.2.1 文件逻辑结构的类型	184	10.6.2 UNIX 系统调用的实现	248
8.2.2 顺序文件	184	10.6.3 Linux 系统调用	250
8.2.3 索引文件	186	10.6.4 Win32 的应用程序接口	250
8.2.4 索引顺序文件	187	习题	251
8.2.5 直接文件和哈希文件	189	第 11 章 多处理机操作系统	253
8.3 文件目录	189	11.1 多处理机系统的基本概念	253
8.3.1 文件控制块和索引结点	189	11.2 多处理器系统的结构	254
8.3.2 简单的文件目录	191	11.2.1 UMA 多处理机系统的结构	255
8.3.3 树形结构目录	192	11.2.2 NUMA 多处理机系统结构	257
8.3.4 目录查询技术	194	11.3 多处理机操作系统的特征与分类	260
8.4 文件共享	195	11.4 进程同步	264
8.5 文件保护	198	11.4.1 集中式与分布式同步方式	264
8.5.1 保护域	198	11.4.2 自旋锁	265
8.5.2 访问矩阵	199	11.4.3 读-复制修改锁	266
8.5.3 访问矩阵的修改	200	11.4.4 二进制指数补偿算法和待锁 CPU 等待队列机构	267
8.5.4 访问矩阵的实现	201	11.4.5 定序机构	268
习题	202	11.5 多处理机系统进程调度	269
第 9 章 磁盘存储器管理	204	11.5.1 用于评价调度性能的若干 因素	269
9.1 外存的组织方式	204	11.5.2 进程分配方式	270
9.1.1 连续组织方式	204	11.5.3 进程(线程)调度方式	271
9.1.2 链接组织方式	205	习题	274
9.1.3 FAT 技术	206	第 12 章 网络操作系统	275
9.1.4 NTFS 的文件组织方式	208		
9.1.5 索引组织方式	209		

12.1 计算机网络概述	275	14.5.1 概述	339
12.1.1 网络的拓扑结构	275	14.5.2 系统核心态组件	340
12.1.2 计算机广域网	277	14.5.3 执行体	342
12.1.3 计算机局域网	278	14.5.4 用户态进程	343
12.1.4 Internet 简介	279	习题	344
12.2 网络体系结构	280	第 15 章 保护和安全的	345
12.2.1 网络体系结构的基本概念	280	15.1 安全环境	345
12.2.2 OSI 七层模型	282	15.2 数据加密技术	347
12.2.3 TCP/IP 网络体系结构	283	15.2.1 数据加密原理	347
12.2.4 LAN 网络体系结构	284	15.2.2 对称加密算法与非对称加密 算法	349
12.3 客户-服务器模式	286	15.2.3 数字签名和数字证明书	349
12.3.1 客户-服务器模式的基本 概念	286	15.3 用户验证	351
12.3.2 三层结构的客户-服务器 模式	287	15.3.1 使用口令验证	351
12.3.3 浏览器-服务器模式	289	15.3.2 基于物理标志的验证技术	353
12.4 网络操作系统的功能	289	15.3.3 生物识别验证技术	354
12.5 网络操作系统提供的服务	293	15.4 来自系统内部的攻击	355
12.5.1 传统信息服务	294	15.5 来自系统外部的攻击	358
12.5.2 Web 服务	294	15.6 可信系统	363
12.5.3 域名系统	295	15.6.1 访问矩阵模型和信息流控制 模型	363
12.5.4 目录服务	297	15.6.2 可信计算基	364
习题	298	15.6.3 设计安全操作系统的原则	365
第 13 章 多媒体操作系统	300	习题	366
13.1 多媒体系统简介	300	第 16 章 Linux 的进程控制子系统及 存储管理	368
13.2 多媒体文件中的各种媒体	303	16.1 Linux 概述	368
13.3 多媒体进程管理中的问题和 接纳控制	306	16.2 进程的描述	371
13.4 多媒体实时调度	309	16.3 进程的控制和调度	373
13.5 媒体服务器的特征和接纳控制	313	16.3.1 进程的控制	373
13.6 多媒体存储器的分配方法	316	16.3.2 进程调度与切换	376
13.7 高速缓存与磁盘调度	322	16.4 进程的同步	377
13.7.1 高速缓存	322	16.4.1 锁机制	377
13.7.2 磁盘调度	323	16.4.2 信号量机制	378
习题	325	16.5 进程通信	379
第 14 章 OS 结构设计	327	16.5.1 信号	379
14.1 OS 设计概述	327	16.5.2 共享内存	380
14.2 传统结构的操作系统	329	16.5.3 消息队列	381
14.3 面向对象的程序设计	333	16.5.4 管道	382
14.4 微内核 OS 结构	335	16.6 Linux 存储管理的主要数据结构	383
14.5 Windows 2000 的体系结构	339	16.6.1 物理内存结构	383

16.6.2 虚拟内存结构	384	17.1.2 VFS 的文件模型的结构及其 功能	393
16.6.3 页表结构	386	17.2 VFS 的主要数据结构	394
16.7 内存的申请和释放	386	17.3 Linux 高速缓存	399
16.8 交换空间、页面置换和缓存机制	388	17.4 文件系统的安装及有关的操作	400
习题	391	17.5 设备管理	403
第 17 章 Linux 系统的文件管理和设备管理	392	习题	406
17.1 Linux 文件系统概述	392	参考文献	407
17.1.1 Linux 文件系统的结构	392		

第 1 章 操作系统引论

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。其主要作用是管理好这些设备, 提高它们的利用率和系统的吞吐量, 并为用户和应用程序提供一个简单的接口, 便于用户使用。OS 是现代计算机系统中最基本和最重要的系统软件, 而其他的软件, 如编译程序、数据库管理系统等系统软件, 以及大量的应用软件, 都直接依赖于操作系统的支持, 取得它所提供的服务。事实上 OS 已成为现代计算机系统、多处理机系统、计算机网络中都必须配置的系统软件。

1.1 操作系统的目标和作用

操作系统的目标与应用环境有关。例如, 在查询系统中所用的 OS, 希望能提供好的人机交互性; 对于应用于工业控制、武器控制, 以及多媒体环境下的 OS, 要求其具有实时性; 而对于微机上配置的 OS, 则更看重的是其使用的方便性。

1. 操作系统的目标

在计算机系统上配置操作系统, 其主要目标是:

(1) 方便性。一个未配置 OS 的计算机系统是极难使用的。用户如果想直接在计算机硬件 (裸机) 上运行自己所编写的程序, 就必须用机器语言书写程序。但如果在计算机硬件上配置了 OS, 用户便可以使用编译命令, 将用户采用高级语言书写的程序翻译成机器代码, 或者直接通过 OS 所提供的各种命令, 操纵计算机系统, 极大地方便了用户, 使计算机变得易学易用。

(2) 有效性。有效性所包含的第一层含意是提高系统资源的利用率。在早期未配置 OS 的计算机系统中, 诸如处理机、I/O 设备等都经常处于空闲状态, 各种资源无法得到充分利用, 所以在当时, 提高系统资源利用率是推动 OS 发展最主要的动力。有效性的另一层含意是, 提高系统的吞吐量。OS 可以通过合理地组织计算机的工作流程, 加速程序的运行, 缩短程序的运行周期, 从而提高了系统的吞吐量。

方便性和有效性是设计 OS 时最重要的两个目标。在过去很长的一段时间内, 由于计算机系统非常昂贵, 有效性显得特别重要。然而, 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, 似乎更加重视如何提高用户使用计算机的方便性。因此, 在微机操作系统中, 都配置了深受用户欢迎的图形用户界面, 并为程序员提供了大量的系统调用, 方便了用户对计算机的使用和编程。

(3) 可扩充性。为适应计算机硬件、体系结构, 以及应用发展的要求, OS 必须具有很好的可扩充性。可扩充性的好坏, 与 OS 的结构有着十分紧密的联系, 由此推动了 OS 结构的不断发展。从早期的无结构发展成模块化结构, 进而又发展成层次化结构, 近年来 OS 已广泛采用了微内核结构, 该结构能方便地增添新的功能和模块, 以及对原有的功能和模块进行修改, 具有良好的可扩充性。

(4) 开放性。随着计算机应用的日益普及, 计算机硬件和软件的兼容性问题, 便提到议事日程上来了。世界各国相应地制定了一系列的软、硬件标准, 使得不同厂家按照标准生产的软、硬件, 都能在本国范围内很好地相互兼容。这无疑给用户带来了极大的方便, 也给产品的推广、应用铺平了道路。后来, 又相继颁布了许多国际性的软、硬件标准, 进一步提高了软、硬件的开放性。近年来, 随着

Internet 的迅速发展,使计算机 OS 的应用环境,由单机环境转向了网络环境。其应用环境就必须更为开放,进而对 OS 的开放性提出了更高的要求。

所谓开放性,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。事实上,凡遵循国际标准所开发的硬件和软件,都能彼此兼容,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,也是衡量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。

2. 操作系统的作用

操作系统在计算机系统中所起的作用,可以从用户、资源管理及资源抽象等不同的观点来分析。

(1) OS 作为用户与计算机硬件系统之间的接口。其含义是: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、可靠地操纵计算机硬件和运行自己的程序。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,即通过命令方式、系统调用方式和图形、窗口方式来实现与操作系统的通信,并取得它的服务。

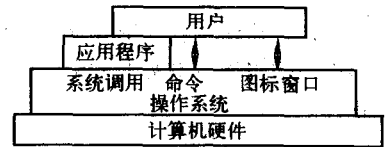


图 1-1 OS 作为接口的示意图

(2) OS 作为计算机系统资源的管理者。在一个计算机系统中,通常都含有多种硬件和软件资源。归纳起来可将这些资源分为四类:处理器、存储器、I/O 设备,以及信息(数据和程序)。相应地,OS 的主要功能也正是对这四类资源进行有效的管理。处理器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(回收)与操纵;文件管理用于实现对文件的存取、共享和保护。可见,OS 的确是计算机系统资源的管理者。

值得进一步说明的是,当一台计算机系统同时供多个用户使用,诸多用户对系统中共享资源的需求(包括数量和时间)有可能发生冲突。为此,操作系统必须对使用资源的请求进行授权,以协调诸用户对共享资源的使用。

(3) OS 实现了对计算机资源的抽象。对于一台完全无软件的计算机系统(即裸机),由于它向用户提供的仅是硬件接口(物理接口),因此,用户必须对物理接口的实现细节有充分的了解,致使该物理机器是难以使用的。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,并向上将 I/O 设备抽象为一组数据结构,以及提供一组 I/O 操作命令,如 read 和 write 命令,这样用户可利用这些数据结构及操作命令,来进行数据输入或输出,而无须关心 I/O 是如何具体实现的。此时用户所看到的机器,是一台比裸机功能更强、使用更方便的机器。换言之,在裸机上铺设的 I/O 软件,隐藏了 I/O 设备的具体细节,向上提供了一组抽象的 I/O 设备。

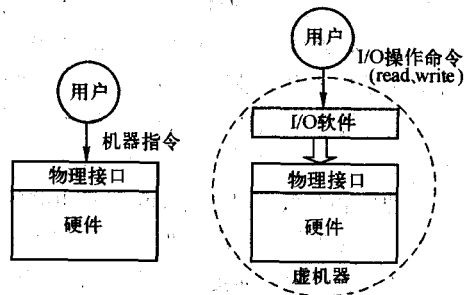


图 1-2 I/O 软件隐藏了 I/O 操作实现的细节

通常把覆盖了上述软件的机器,称为扩充机器或虚拟机:它向用户提供了一个对硬件操作的抽象模型。用户可利用该模型提供的接口使用计算机,无须了解物理接口实现的细节,从而使用户更容易地去使用计算机硬件资源。亦即,I/O 设备管理软件实现了对计算机硬件操作的第一个层次的抽象。

同理,为了方便用户使用文件系统,又可在第一层软件(I/O 管理软件)上,再覆盖一层用于文件管理的软件,由它来实现对文件操作的细节,并向上层提供一组实现对文件进行存取操作的数据结

构及命令。这样，用户可利用该软件提供的数据结构及命令，对文件进行存取。此时用户所看到的是一台功能更强、使用更方便的虚拟机。亦即，文件管理软件实现了对硬件资源操作的第二个层次的抽象。以此类推，如果在文件管理软件上，再覆盖一层面向用户的窗口软件，则用户便可在窗口环境下，方便地使用计算机，从而形成一台功能更强的虚拟机。

由此可知，OS 是铺设在计算机硬件上的多层软件的集合，它们不仅增强了系统的功能，还隐藏了对硬件操作的具体细节，实现了对计算机硬件操作的多个层次的抽象模型。值得说明的是，一个硬件不仅可在底层加以抽象，还可以在高层，对该资源底层已抽象的模型，再次进行抽象，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强，用户使用起来也越方便。

3. 推动操作系统发展的主要动力

OS 自 20 世纪 50 年代诞生后，经历了由简单到复杂、由低级到高级的发展。在短短 40 多年间，OS 在各方面都有了长足的进步，能够很好地适应计算机硬件和体系结构的快速发展，以及应用需求的不断变化。下面我们对推动 OS 发展的主要推动力做具体阐述。

(1) 不断提高计算机系统资源的利用率。在 20 世纪 50 年代计算机系统特别昂贵，提高计算机系统中各种资源的利用率就成为了 OS 最初发展的推动力。由此促成了由单道到多道批处理系统的演变，它通过减少计算机的空闲时间，来提高系统中 CPU 和 I/O 设备的利用率。在 20 世纪 60 年代，又出现了可以支持多个用户同时使用一台计算机的分时系统，使系统资源利用率得到了极大的提高，推动了计算机的第一次大普及。与此同时还推出了能改善 I/O 设备和 CPU 利用率的 SPOOLing 系统。

在 20 世纪 70 年代有效地提高存储器系统利用率，并能从逻辑上扩大内存的虚拟存储器技术，获得了广泛的应用。此后在网络环境下，通过在服务器上配置网络文件系统和数据库系统的方法，将资源提供给全网用户共享，又进一步提高了资源的利用率。

(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成为主要矛盾。这又成为继续推动 OS 发展的主要因素。在 20 世纪 60 年代分时系统的出现，不仅提高了系统资源的利用率，还能实现人机交互，使用户能像早期使用计算机时一样，感觉自己独占全机资源，对其进行直接操控的，这极大地方便了程序员对程序进行调试和修改。在 20 世纪 90 年代初，图形用户界面的出现，受到用户广泛的欢迎，进一步方便了用户对计算机的使用，这无疑又加速推动计算机的迅速普及和广泛应用。

(3) 器件的不断更新换代。随着 IT 技术的迅猛发展，尤其是微芯片的不断更新换代，使得计算机的性能迅速提高，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，OS 的功能和性能也都有显著的增强和提高。

与此同时，外部设备也在迅速发展，OS 所能支持的外部设备也越来越多。例如，现在的微机 OS 已能够支持种类繁多的外部设备，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、数码相机等。

(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 OS 类型。例如，当计算机由单处理机系统发展成为多处理机系统时，相应地，OS 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，它不仅有效地管理好网络中的共享资源，而且还能向用户提供许多网络服务。

(5) 不断提出新的应用需求。操作系统能如此迅速发展的另一个重要原因是，人们不断提出新的应用需求。例如：为了提高产品的质量和数量，需要将计算机应用于工业控制中，此时在计算机上，就需要配置能进行实时控制的 OS，由此产生了实时 OS。为了能满足用户在计算机上播放 DVD 碟片、

从网上下载数字电影的需求,在 OS 中又增添了多媒体功能。另外,由于在计算机系统中,保存了越来越多的宝贵信息,致使能够确保系统的安全性,也成为 OS 必须具备的功能。尤其是随着 VLSI 的发展,体积越来越小,价格也变得越来越便宜,大量智能设备应运而生,这样,嵌入式操作系统的产生和发展,也成了一种必然。

1.2 操作系统的发展过程

在 20 世纪 50 年代中期,出现了第一个简单的批处理 OS; 20 世纪 60 年代中期开发出多道程序批处理系统,不久又推出分时系统。与此同时,用于工业和武器控制的实时 OS 也相继问世。20 世纪 70 到 90 年代,是 VLSI 和计算机体系结构大发展的年代,导致了微型机、多处理机和计算机网络的诞生和发展,与此相应地,也相继开发出了微机 OS、多处理机 OS 和网络 OS,并得到极为迅速地发展。

1. 未配置操作系统的计算机系统

从 1945 年诞生的第一台计算机,到 20 世纪 50 年代中期的计算机,都属于第一代计算机。这时还未出现 OS,对计算机的全部操作,都是由用户采取人工操作方式完成的。

(1) 人工操作方式

早期的操作方式是由程序员将事先已穿孔的纸带(或卡片),装入纸带输入机(或卡片输入机),再启动它们将纸带(或卡片)上的程序和数据输入计算机,然后启动计算机运行。仅当程序运行完毕并取走计算结果后,才允许下一个用户上机。这种人工操作方式有以下两方面的缺点:①用户独占全机。即一台计算机的全部资源由上机用户所独占。②CPU 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。虽然 CPU 的速度在迅速提高,但 I/O 设备的速度却提高缓慢,这使 CPU 与 I/O 设备之间速度不匹配的矛盾更加突出。为此曾先后出现了通道技术、缓冲技术,然而都未能很好地解决上述矛盾,直至后来引入了脱机输入/输出技术,才获得了较为令人满意的结果。

(2) 脱机输入/输出 (Off-Line I/O) 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 I/O 技术。该技术是事先将装有用户程序 and 数据的纸带,装入纸带输入机,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,再从磁带上高速地调入内存。

类似地,当 CPU 需要输出时,可先由 CPU 把数据直接从内存高速地送到磁带上,然后再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,把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 方式的主要优点为:①减少了 CPU 的空闲时间。装带、卸带,以及将数据从低速 I/O 设备,送到高速磁带上(或反之)的操作,都是在脱机情况下由外围机完成的,并不占用主机时间,从而有效地减少了 CPU 的空闲时间。②提高了 I/O 速度。当 CPU 在运行中需要输入数据时,是直接从高速的磁带上将数据输入到内存的,极大地提高了 I/O 速度,从而进一步减少了 CPU 的空闲时间。

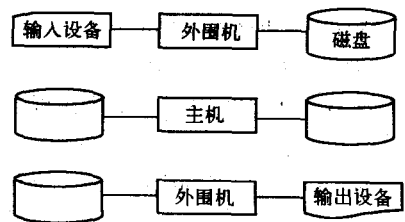


图 1-3 脱机 I/O 示意图

2. 单道批处理系统

20 世纪 50 年代中期出现了第二代晶体管计算机, 此时计算机虽已具有推广应用的价值, 但计算机系统仍然非常昂贵。为了能充分地提高它的利用率, 应尽量保持系统的连续运行, 即在处理完一个作业后, 紧接着处理下一个作业, 以减少机器的空闲时间。

(1) 单道批处理系统 (Simple Batch Processing System) 的处理过程

为对作业连续处理, 需要先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上的第二个作业调入内存。计算机系统就是这样自动地一个作业紧接一个作业进行处理的, 直至磁带上的所有作业全部完成, 这样便形成了早期的批处理系统。虽然系统对作业的处理, 是成批地进行的, 但在内存中始终只保持一道作业, 故称为单道批处理系统。

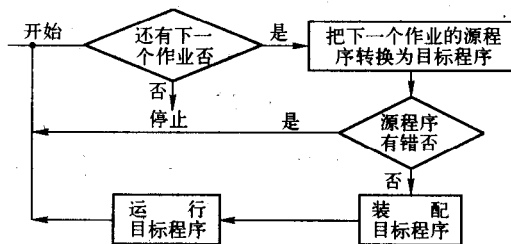


图 1-4 单道批处理系统的处理流程

图 1-4 示出了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和 CPU 与 I/O 设备速度不匹配矛盾的过程中形成的。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充分地利用系统资源, 故现已很少使用。

(2) 单道批处理系统的缺点

单道批处理系统最主要的缺点是, 系统中的资源得不到充分的利用。这是因为在内存中仅有一道程序, 每当该程序在运行中发出 I/O 请求后, CPU 便处于等待状态, 必须在其 I/O 完成后才继续运行。又因 I/O 设备的低速性, 更使 CPU 的利用率显著降低。图 1-5 示出了单道程序的运行情况, 从图可以看出: 在 $t_2 \sim t_3$ 、 $t_6 \sim t_7$ 时间间隔内 CPU 空闲。

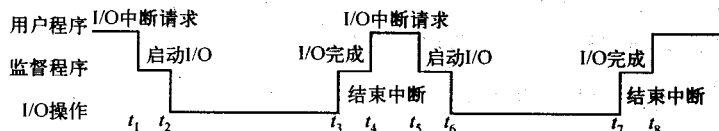


图 1-5 单道程序的运行情况

为了能在系统中运行较大的作业, 通常在计算机中都配置了较大容量的内存。但实际情况是, 有 80% 以上的作业都属于中小型, 因此在单道程序环境下, 也必定造成内存的浪费。类似地, 为了满足各种类型作业的需要, 在系统中将会配置多种类型的 I/O 设备, 显然这些设备在单道程序环境下也不能得到充分利用。

3. 多道批处理系统 (Multiprogrammed Batch Processing System)

20 世纪 60 年代中期, 由 IBM 公司生产的第一台小规模集成电路计算机——IBM 360 (第三代计算机系统), 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, 使该机器获得极大的成功, 并为它开发出了 OS/360 操作系统, 这是第一个能运行多道程序的批处理系统。

(1) 多道程序设计的基本概念

为了进一步提高资源的利用率和系统吞吐量，在 20 世纪 60 年代中期引入了多道程序设计技术，由此形成了多道批处理系统。在该系统中，用户所提交的作业先存放在外存上，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，使它们共享 CPU 和系统中的各种资源。由于同时在内存中装有若干道程序，这样便可以在运行程序 A 时，利用其因 I/O 操作而暂停执行时的 CPU 空闲时间，再调度另一道程序 B 运行。同样可以利用程序 B 在 I/O 操作时的 CPU 空档时间，再调度程序 C 运行，使多道程序交替地运行，这样便可以保持 CPU 处于忙碌状态。图 1-6 示出了四道程序时的运行情况。

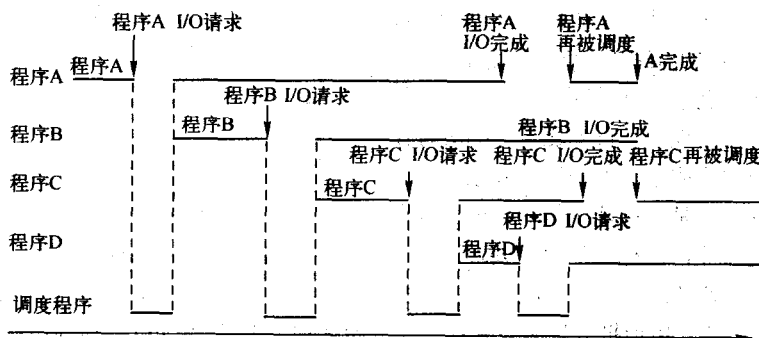


图 1-6 多道程序的运行情况

(2) 多道批处理系统的优缺点

多道批处理系统的优点为：①资源利用率高。引入多道程序能使多道程序交替地运行，以保持 CPU 处于忙碌状态；在内存中装入多道程序，来提高内存的利用率；此外还可以提高 I/O 设备的利用率。②系统吞吐量大。能提高系统吞吐量的主要原因可归结为：CPU 和其他资源保持“忙碌”状态；仅当作业完成时或运行不下去时才进行切换，系统开销小。

其缺点为：①平均周转时间长。由于作业要排队依次进行处理，因而作业的周转时间较长，通常需几个小时，甚至几天。②无交互能力。用户一旦把作业提交给系统后，直至作业完成，用户都不能与自己的作业进行交互，这对修改和调试程序是极不方便的。

(3) 多道批处理系统需要解决的问题

多道批处理系统是一种十分有效，但又非常复杂的系统，为使系统中的多道程序间能协调地运行，系统必须解决下述一系列问题：①争用处理机问题。既要能满足各道程序运行的需要，又要能提高处理机的利用率。②内存分配和保护问题。系统应能为每道程序分配必要的内存空间，使它们“各得其所”，且不会因某道程序出现异常情况，而破坏其他程序等问题。③I/O 设备分配问题。系统应采取适当的策略来分配系统中的 I/O 设备，以达到既能方便用户对设备的使用，又能提高设备利用率的目的。④文件的组织和管理问题。系统应能有效地组织存放在系统中的大量的程序和数据，使它们既便于用户使用，又能保证数据的安全性。⑤作业管理问题。系统中存在着各种作业（应用程序），系统应对系统中所有的作业，进行合理的组织，以满足这些作业用户的不同要求。⑥用户与系统的接口问题。为使用户能方便地使用操作系统，OS 还应提供用户与 OS 之间的接口。

为此，应在计算机系统中增加一组软件，用以对上述问题进行妥善、有效的处理。这组软件应包括：能有效地组织和管理四大资源的软件，合理地对各类作业进行调度和控制它们运行的软件，以及

方便用户使用计算机软件。正是这样一组软件构成了操作系统。据此，我们可把操作系统定义为：**操作系统是一组能有效地组织和管理计算机硬件和软件资源，合理地各类作业进行调度，以及方便用户使用的程序的集合。**

4. 分时系统 (Time Sharing System)

(1) 分时系统的引入

如果说，推动多道批处理系统形成和发展的主要动力，是提高资源利用率和系统吞吐量，那么，推动分时系统形成和发展的主要动力，则是为了满足用户对人机交互的需求，所形成的一种新型 OS。用户的需求具体表现在以下几个方面：

① 人机交互。每当程序员写好一个新程序时，都需要上机进行调试。由于新编程序难免存在一些错误或不当之处，需要进行修改，因此用户希望能像早期使用计算机那样，独占全机并对它进行直接控制，以便能方便地对程序中的错误进行修改。亦即，希望能进行人机交互。

② 共享主机。在 20 世纪 60 年代，计算机还十分昂贵，一台计算机要同时供很多用户共享使用。显然，用户们在共享一台计算机时，每人都希望能像独占时一样，不仅可以随时与计算机进行交互，而且还不会感觉到其他用户的存在。

由上所述不难得知，分时系统是指在一台主机上连接了多个配有显示器和键盘的终端，所组成的系统，该系统允许多个用户，同时通过自己的终端，以交互方式使用计算机，共享主机中的资源。

(2) 分时系统实现中的关键问题

在多道批处理系统中，用户无法与自己的作业进行交互的主要的原因是：作业都先驻留在外存上，即使以后被调入内存，也要经过较长时间的等待后方能运行，用户无法与自己的作业进行交互。为了能够实现人机交互，必须解决的关键问题是，如何使用户能与自己的作业进行交互。为此，系统首先必须能提供多个终端，同时给多个用户使用；其次，当用户在自己的终端上键入命令时，系统应能及时接收，并及时处理该命令，再将结果返回给用户。此后，用户可根据系统返回的响应情况，再继续键入下一条命令，即人机交互。亦即，允许有多个用户同时通过自己的键盘键入命令，系统也应能及时地接收并处理。

1) 及时接收

要做到及时接收多个用户键入的命令或数据，只需在系统中配置一个多路卡即可。例如，如果主机上需要连接 64 个终端时，就配置一个 64 用户的多路卡。多路卡的作用是，实现分时多路复用。即主机以很快的速度，周期性地扫描各个终端，在每个终端处停留一很短的时间（如 30ms 时间），用于接收从终端发来的数据。对于 64 用户的多路卡，用不到 2s 的时间，便可完成一次扫描。即主机能用不到 2s 的时间，分时接收各用户从终端上输入的数据一次。此外为了能使从终端上输入的数据，被依次逐条地进行处理，还需要为每个终端，配置一个缓冲区，用来暂存用户键入的命令（或数据）。

2) 及时处理

人机交互的关键在于，用户键入命令后，能对自己的作业及其运行，及时地实施控制，或进行修改。因此，各个用户的作业，都必须驻留在内存中，并能频繁地获得处理机运行。否则，用户键入的命令，将无法作用到自己的作业上。由此可见，为实现人机交互，必须彻底地改变原来批处理系统的运行方式，转而采用下面的方式：

① 作业直接进入内存。因为作业在磁盘上是不能运行的，所以作业应直接进入内存。

② 采用轮转运行方式。如果一个作业独占 CPU 连续运行，那么其他的作业，就没机会被调度运行。为避免一个作业长期独占处理机，而引入了时间片的概念。一个时间片，就是一段很短的时间（例