



21世纪高等院校计算机科学与技术系列教材

主编 许曰滨 孙英华 程 亮



操作系统

C A O Z U O X I T O N G



北京邮电大学出版社
www.buptpress.com

操 作 系 统

许曰滨 孙英华 程 亮 主编

北京邮电大学出版社
·北 京·

内 容 简 介

本书以现代操作系统的功能模块为主线,系统介绍了操作系统的原理与技术。内容包括:作业管理、进程管理、进程同步与通信、调度与死锁、存储管理、虚拟存储器技术、设备管理、文件管理和 Linux 系统实例等部分。

作为一本操作系统原理的基础教材,本书覆盖面广,内容丰富,技术性强,可读性好,可作为普通高等院校计算机科学与技术专业的教学用书,也可供广大计算机科学工作者和从事相关专业的研究人员参考。

图书在版编目(CIP)数据

操作系统/许曰滨,孙英华,程亮主编. —北京:北京邮电大学出版社,2005
ISBN 7-5635-1070-2

I. 操... II. ①许...②孙...③程... III. 操作系统 IV. TP316

中国版本图书馆 CIP 数据核字(2005)第 096734 号

书 名: 操作系统

主 编: 许曰滨 孙英华 程 亮

策 划 人: 章 剑

责任编辑: 章 剑

出 版 者: 北京邮电大学出版社(北京市海淀区西土城路 10 号) 邮编:100876

发行部电话:(010)62282185 62283578(传真)

电子信箱: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京通州皇家印刷厂印刷

开 本: 787 mm×1 092 mm 1/16

印 张: 19

字 数: 468 千字

印 数: 1—3 000 册

版 次: 2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

ISBN 7-5635-1070-2/TP·181

定价: 29.00 元

·如有印装质量问题,请与北京邮电大学出版社发行部联系·

21 世纪高等院校计算机科学与技术系列教材

编 委 会

主 任：金怡濂

委 员：(按姓氏笔划排名)

王士同 王明严 刘 弘

朱其亮 何炎祥 汪厚祥

金 海 徐 涛 潘振宽

序

计算机科学技术是科学性与工程性并重的一门学科。它的迅猛发展除了源于微电子学等相关学科的发展外,更主要源于其应用需求的广泛性不断增长,它已渗透到人类社会的各个领域,成为经济发展的倍增器,科学文化与社会进步的催化剂。计算机与通信的融合和全球联网,更显示出它无可限量的发展前景。任何一个领域的发展都离不开计算机已成为无可否认的事实。应用是计算机科学技术发展的动力、源泉和归宿,而计算机科学技术又不断为应用提供先进的方法、设备与环境。

近年来,计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地推进了经济信息化和社会信息化的进程。计算机科学技术对一个国家在政治、经济、科技、文化、国防等方面的催化作用和强化作用都具有难以估量的意义。计算机知识与能力已成为21世纪人才素质的基本要求之一,因此,计算机科学技术的教育在世界各国都备受重视,我国政府和教育部门对计算机科学技术的教育及人才培养也非常重视。为了适应社会发展对计算机科学技术人才的强烈要求,各高校均在着力培养基础扎实、知识面广、综合素质高、实践能力强、富有创新精神,且具有较强的科学技术运用、推广、转化能力的高层次人才。

由北京邮电大学出版社联合北京邮电大学、武汉大学、华中理工大学及山东、江苏等多所高校的计算机专业教学负责人组成的“21世纪高等院校计算机科学与技术系列教材编委会”按照《中国计算机科学与技术学科教程2002》的要求组织编写的系列教材,体现了近年计算机学科的新理论、新技术。内容涵盖计算机专业学生所应掌握的相关知识,并根据目前计算机科学技术的发展趋势与实际应用相结合,能够满足目前高校计算机专业教学的需要,也可作为计算机专业人员的自学参考材料。

本系列教材作者均为多年从事教学、科研的一线教师,有着丰富的教学和科研实践经验,所编写的这套教材具有结构严谨,内容丰富、理论与实际结合紧密的特点,是他们的教学经验和科研成果的结晶。

计算机科学技术日新月异,所以教材也要不断推陈出新,我希望本系列教材能为我国高校计算机专业教育做出新的贡献。

中国工程院院士

金怡廉

前 言

操作系统是现代计算机系统中不可缺少的系统软件。它处于硬件和应用软件的结合部,是最接近硬件的一层软件,其他软件都运行在操作系统所构筑的平台上。对于绝大多数用户来说,没有操作系统的计算机简直就是一堆“废铁”,根本无法使用。

如同操作系统在计算机系统中所处的地位那样,《操作系统》课程也在计算机科学教育中扮演着极其重要的角色。学好操作系统课程,掌握好操作系统的基本概念,学会操作系统的管理方法,对理解计算机系统的运作机制、建立整机概念具有不可替代的作用。我们编写本书的目的是奉献给读者一本注重系统整体描述的、内容新颖的教科书。

本书是我们参考了国内外有关操作系统的技术资料 and 论著,并结合多年来的教学实践与切身体会编纂而成的。书中对大多数操作系统普遍使用的基本概念和技术方法进行了全面、详细地介绍。内容共包括 4 个主要部分,分列为 9 章。

- 第 1 部分是绪论,对应书中第 1 章。概述了操作系统的发展过程、基本结构、功能与分类。为了帮助读者对整体概念的把握,书中详细描述了系统的启动过程和应用程序的运行过程。
- 第 2 部分是处理器管理,对应第 2~4 章。其中,第 2 章讲述作业概念及其管理方法;第 3 章讲述进程和线程概念、进程控制功能、进程同步与通信方法;第 4 章讲述进程调度原理与技术、进程死锁的应对措施等。这一部分中对线程概念、线程与进程之间关系做了突出描述。为了结合实际,此处还引入了 Linux 的共享存储器通信方式——断接和附接概念。
- 第 3 部分是资源管理,对应第 5~8 章。其中,第 5 章与第 6 章讲述存储管理和虚拟存储器的原理与技术,并专辟一节对目前较为流行的伙伴系统给予介绍。第 7 章介绍设备管理和虚拟设备的实现技术,详细说明了





Spooling 系统的原理和构造方法。第 8 章介绍文件管理的方法与技术。

- 第 4 部分是一个操作系统实例,对应第 9 章。概述了 Linux 操作系统的功能特性,并简要介绍了进程管理、存储管理、设备管理和文件管理等子系统的结构与实现方法。

本书力求把握操作系统的精华、突出原理与实例的结合。在写法上,既体现操作系统的完整性和严谨性,又注重深入浅出、简明扼要,力争全面地论述操作系统原理及其实现技术。书中涉及到操作系统发展中多个具有历史影响的系统,特别是 UNIX 和 IBM360 系统,还有一些流行的操作系统例子。

作为一本操作系统原理的基础教材,本书覆盖面较广,内容较丰富,技术性较强,可读性较好,可作为普通高等院校计算机科学与技术专业的教学用书,也可供广大计算机科学工作者和从事相关专业的研究人员参考。

由于作者水平所限,加之时间仓促,书中必定存在一些阐述不当甚至疏漏之处,恳请广大读者批评指正。

编 著
2005 年 7 月



目 录

第1章 绪论

1.1 操作系统的形成与发展	1
1.1.1 手工操作方式	2
1.1.2 单任务操作系统发展阶段	3
1.1.3 多任务操作系统发展阶段	4
1.1.4 规范化和微型化发展阶段	5
1.1.5 并行与分布式系统发展阶段	5
1.2 I/O 控制技术的发展	6
1.2.1 早期的 I/O 方式	7
1.2.2 基于设备控制器的 I/O 技术	7
1.2.3 基于中断机制的 I/O 技术	7
1.2.4 基于 DMA 的 I/O 技术	8
1.2.5 基于通道的 I/O 技术	9
1.3 系统管理方式	10
1.3.1 单道批处理方式	10
1.3.2 多道批处理方式	11
1.3.3 分时处理	12
1.3.4 实时处理	13
1.4 操作系统的组成	14
1.4.1 各管理模块的功能	15
1.4.2 操作系统内核	16
1.5 通用操作系统	17
1.5.1 操作系统的基本特征	18
1.5.2 用户程序运行过程	18
1.5.3 系统启动过程	19
习 题	20

第2章 作业管理

2.1 概 述	22
2.1.1 作业分类	23





2.1.2 作业管理的功能	24
2.2 用户管理	25
2.2.1 添加和删除用户	26
2.2.2 用户组管理及用户登录管理	27
2.3 操作系统接口	28
2.3.1 脱机命令接口	28
2.3.2 联机命令接口	29
2.3.3 程序级接口	31
2.4 作业控制	32
2.4.1 后备状态	32
2.4.2 执行状态与完成状态	34
2.4.3 作业控制块	35
2.4.4 作业控制方式	36
2.4.5 作业卸出	37
2.5 作业调度	37
2.5.1 作业调度时机	38
2.5.2 调度算法	38
2.5.3 作业调度准则	40
2.6 作业调度的讨论	43
2.6.1 作业类型对作业调度的影响	43
2.6.2 优先级	45
2.6.3 程序道数对作业调度的影响	46
2.6.4 示例	47
习 题	49

第3章 进程管理

3.1 进程管理的基本概念	50
3.1.1 程序的运行方式	50
3.1.2 进程概念	52
3.1.3 进程管理的主要功能	53
3.2 进程控制块	53
3.2.1 进程控制块的内容	53
3.2.2 进程基本状态及状态变迁	54
3.2.3 扩展状态	56
3.2.4 PCB 组织结构	57
3.3 进程控制	59
3.3.1 进程创建与撤销原语	59



3.3.2 阻塞与唤醒原语	60
3.3.3 挂起与激活原语	62
3.4 进程调度	63
3.4.1 进程调度分类	63
3.4.2 轮转算法	65
3.4.3 多级队列调度算法	67
3.4.4 多级反馈队列调度	68
3.5 实时系统的进程调度	69
3.5.1 实时任务的分类及其调度方法	70
3.5.2 周期性任务调度	72
3.6 线程	73
3.6.1 线程的引入	73
3.6.2 线程结构	74
3.6.3 内核级线程	75
3.6.4 用户级线程	76
3.7 调度	77
3.7.1 中级调度	77
3.7.2 公平共享调度	78
3.7.3 多处理器系统调度	80
习 题	82

第4章 并发与互斥

4.1 基本概念	84
4.1.1 临界资源和临界区	85
4.1.2 互斥准则	86
4.2 实现互斥的软件方法	86
4.2.1 Dekker 方法	86
4.2.2 Peterson 算法	89
4.3 实现互斥的硬件方法	90
4.3.1 交换指令	90
4.3.2 测试与设置指令	91
4.4 信号量机制	92
4.4.1 整型信号量	92
4.4.2 记录型信号量	94
4.4.3 信号量集机制	99
4.5 管程	101
4.6 进程通信	104





4.6.1 共享存储器系统	104
4.6.2 管道通信	106
4.6.3 消息传递	108
4.7 死锁	111
4.7.1 产生死锁的原因	111
4.7.2 死锁预防	112
4.7.3 死锁避免	114
4.7.4 死锁检测	117
4.7.5 死锁消除	118
习 题	119

第5章 存储管理

5.1 存储器管理概述	122
5.1.1 内存储器的组成	122
5.1.2 存储管理的主要功能	123
5.1.3 存储管理的数据结构	125
5.1.4 存储管理的主要方法	127
5.2 单一连续区管理	127
5.3 分区管理	128
5.3.1 固定分区	128
5.3.2 动态分区	130
5.3.3 分配算法	132
5.3.4 可重定位动态分区管理	134
5.3.5 伙伴系统	134
5.4 分页存储管理	135
5.4.1 页面	136
5.4.2 页面分配与地址变换	138
5.4.3 页面共享	143
5.5 分段管理技术	145
5.5.1 分段管理的基本原理	145
5.5.2 段表	146
5.5.3 地址变换	148
5.5.4 分段保护与共享	149
5.5.5 段页式管理	150
习 题	152

第6章 虚拟存储器

6.1 基础知识	154
----------------	-----



6.1.1 覆盖技术	154
6.1.2 交换技术	155
6.1.3 局部性原理	157
6.2 请求分页存储管理	158
6.2.1 地址变换	158
6.2.2 置换算法	162
6.2.3 驻留集和工作集	167
6.2.4 抖动的产生和预防	170
6.3 请求分段存储管理	172
6.3.1 实现方法	172
6.3.2 分段共享	174
6.3.3 请求段页式存储管理	175
习 题	176

第7章 设备管理

7.1 基础知识	178
7.1.1 外部设备	178
7.1.2 I/O控制方式	179
7.1.3 设备管理的功能与结构	185
7.2 逻辑 I/O	187
7.2.1 设备分配	188
7.2.2 安全性	191
7.3 设备 I/O	191
7.3.1 I/O 任务	192
7.3.2 缓冲管理	192
7.4 I/O 调度与驱动	196
7.4.1 I/O 调度	196
7.4.2 通道程序	200
7.4.3 设备驱动程序	201
7.4.4 设备中断处理程序	202
7.5 虚拟设备	202
7.5.1 Spooling 的基本概念	203
7.5.2 Spooling 虚拟输入设备	204
7.5.3 Spooling 虚拟输出设备	207
习 题	209

第8章 文件管理

8.1 文件管理系统概述	211
--------------------	-----





8.1.1 文件的概念	211
8.1.2 文件的组成	212
8.1.3 文件管理系统的功能与特点	212
8.1.4 文件管理系统的组成	214
8.1.5 常见文件管理系统	214
8.2 文件的逻辑结构和物理结构	215
8.2.1 文件的逻辑结构及存取方式	215
8.2.2 文件的物理结构	217
8.3 文件存储空间管理	222
8.3.1 空闲区表	223
8.3.2 空闲块链表	223
8.3.3 位示图	224
8.3.4 成组链接结构	224
8.4 文件目录管理	226
8.4.1 文件目录	226
8.4.2 文件目录结构	226
8.4.3 目录管理实例	229
8.4.4 UNIX 的目录和索引结点	230
8.5 文件的共享、保护与保密	232
8.5.1 文件共享	232
8.5.2 文件保护	235
8.5.3 文件保密	237
8.6 文件的操作和使用	237
8.6.1 文件操作类系统调用	237
8.6.2 文件的使用	239
习 题	239

第 9 章 Linux 系统概述

9.1 Linux 系统的发展和特性	242
9.1.1 Linux 的诞生	242
9.1.2 Linux 的发展	243
9.1.3 Linux 的版本	244
9.1.4 Linux 系统的功能特性	244
9.2 Linux 系统的用户接口	245
9.2.1 Linux 的命令接口	246
9.2.2 Linux 的图形接口	246
9.2.3 Linux 的系统调用	247



9.3 Linux 系统的进程管理	247
9.3.1 进程控制块	248
9.3.2 进程状态及转换	249
9.3.3 进程的创建与撤销	250
9.3.4 Linux 的线程机制	251
9.3.5 Linux 的进程调度	251
9.3.6 进程的切换	253
9.4 进程的同步与通信	254
9.4.1 Linux 进程的同步	254
9.4.2 Linux 进程的通信机制	255
9.5 存储器管理	263
9.5.1 物理内存的管理	263
9.5.2 虚拟内存	265
9.5.3 Linux 虚存的保护	267
9.5.4 Linux 使用的缓存机制	267
9.6 设备管理	268
9.6.1 设备和设备特殊文件	268
9.6.2 相关数据结构	269
9.6.3 中断和异常	271
9.6.4 Linux 的设备驱动程序	271
9.6.5 设备文件的存取权限和系统安全	274
9.7 文件管理	274
9.7.1 虚拟文件系统	274
9.7.2 EXT2 文件系统	278
9.8 系统安全性	281
9.8.1 标识与鉴别	281
9.8.2 存取控制	282
9.8.3 审计与加密	282
9.8.4 网络安全	283
9.8.5 备份和恢复	284
习题	284
参考文献	286



第1章 绪论

电子计算机系统分为硬件和软件两部分,其中,硬件部分称为“裸机”,由若干物理设备连接而成。现代计算机的硬件功能很强,用途很多。然而,硬件本身提供给外界的界面却是十分粗糙的,许多信息都令人费解、难以使用。为此,人们研制了一种能够管理和控制这种裸机的软件,这就是“操作系统”(Operating System, OS)。

本章主要介绍操作系统的发展过程、操作系统的处理方式、操作系统的功能与构造,以及通用操作系统的特点等。

1.1 操作系统的形成与发展

计算机从1946年问世至今,已有半个多世纪的发展时期。它的应用从单纯的数值运算,发展到过程控制、数据处理和网络通信等各个方面。在工农业生产、国防建设、文化教育及社会生活的许多领域发挥着重要的作用。

众所周知,从1946年至20世纪50年代中期,计算机处于以电子管为特征的“第1代”发展时期。计算机的运算速度慢、存储容量小、软件功能低。当时的计算机主要用于数值计算方面,操作方式基本上采用手工方式,计算机上没有配置操作系统。

从20世纪50年代中、后期开始,计算机进入到以晶体管为特征的“第2代”发展时期。在这一时期内,除了硬件上的功能得到扩充外,软件方面出现了高级程序设计语言和用于管理计算机的软件——“监控程序”(也就是操作系统前身)。操作系统的基本概念已经形成。此后,操作系统进入快速发展时期。

从20世纪60年代中期开始,计算机又进入到以集成电路为特征的“第3代”发展时期。计算机开始向许多应用领域渗透,因而出现了大量适用于某些专业的高级语言。一些用于过程控制的实时系统和通用的操作系统开始在计算机上运行。

20世纪七八十年代,计算机进入“第4代”发展时期。超大规模集成电路的出现使得计算机的体积越来越小、功能越来越强、价格越来越低,计算机结构出现了微型化的趋势。微型机的出现促使了结构化和小型化的操作系统的研究。

20世纪80年代末,随着通信技术的发展以及大型数据库管理系统、远程处理系统和计算机网络的成熟与推广,操作系统的研究开始向并行计算与分布式方向发展。

从上述介绍来看,操作系统的发展大体经历以下4代:

- (1) 20世纪50年代中期至60年代初期为单任务操作系统时代;
- (2) 20世纪60年代中期至70年代中期为多任务和多用户操作系统时代;
- (3) 20世纪70年代末期至80年代末期为微型操作系统时代;





(4) 20 世纪 90 年代开始至今为分布式操作系统时代。

1.1.1 手工操作方式

早期的计算机由于运算速度低、外部设备少,加之没有配置操作系统,用户使用计算机是相当困难的。用户编制好的程序被输入到计算机内后,直接通过控制台上的一排排机械式开关和指示灯来监督控制程序的运行。当时,尚未研制出现操作系统,因此用户使用计算机的繁琐程度可想而知。

这一阶段大体对应计算机诞生后的前 10 年(1946—1955),这 10 年又可分为两个时期:前一个时期称为手工交互时期,后一个时期是手工批处理时期。

1. 手工交互时期

手工交互时期,程序设计者需要用机器语言,也就是由二进制数形式构成的指令系统来编制程序,一个程序的运行通常需要 3 个步骤。

(1) 通过控制台开关设定计算机的初始状态,将存有程序和数据的卡片装到输入设备上。设定程序和数据存放在内存的位置后,按下“作业装入”开关,使程序和数据由输入机上输入。

(2) 通过控制台开关设定程序的开工地址,按下“作业运行”开关,作业开始运行。操作员可根据控制台上的状态显示灯确定运行是否正常,若出现错误则停机处理。

(3) 运行结束后,将结果存于某一存储位置上。通过控制台开关设定好结果的存储位置,按下“结果输出”开关,将该位置的运算结果从输出机(比如,卡片机)输出,然后卸下卡片交给用户。

这种操作方式的特点是:

- 没有“源程序”概念,提交给计算机的程序都是直接可运行的目标程序。
- 用户自己设计程序,自己上机操作,一个用户同时也兼作程序员和操作员。
- 程序员的操作以交互方式进行,用户可通过控制台的指示灯观看运行结果,并及时通过控制台按钮对内存的目标程序进行修改。
- 程序的执行过程得不到系统的任何帮助,每个作业都通过自己的引导程序装入计算机。

2. 手工批处理时期

自从 1951 年出现汇编语言以后,汇编、连接、装入等软件使计算机的操作发生了变化。程序设计者利用汇编语言编制的程序,编译和连接后就可以运行。这样一来,作业的装卸次数比以前增加了。一个程序的运行通常需要 5 步。

(1) 将存有汇编语言的磁带和存有用户程序的磁带装入输入设备上,按启动开关将它们装入。

(2) 运行汇编程序,产生目标磁带。尔后卸下汇编磁带和源程序磁带。

(3) 安装连接程序磁带,装入连接程序。启动连接程序,产生可执行的目标磁带。

(4) 安装数据磁带,装入初始数据。

(5) 装入可执行目标程序,启动可执行目标程序开始运算。产生结果磁带后,用户卸下结果磁带。

这种操作方式的特点是:



- 目标程序不是程序设计者直接设计的,而是由汇编语言产生的。这样,程序设计人员避免了与机器语言打交道之苦,因为用汇编语言设计程序,相对要简单一些。
- 用户设计了程序,不必自己直接上机操作,装卸磁带和人工干预判断,都可以让操作员来完成,程序设计人员可以不必参与进去。
- 程序的执行过程可以得到系统的帮助,比如,系统的标准装入程序(Loading Program)可从磁带上装入程序和数据,不必用户自带装入程序。
- 由于程序运行中的一些环节由机房的操作员完成,程序员与程序之间必然失去了交互性。

手工操作的主要缺点是操作较麻烦。操作员要了解计算机各部分的技术细节才能准确地控制作业运行,而这样做极容易出错,维护工作比较复杂。另一个问题是资源浪费。计算机的所有软硬件资源全部被一个作业独占,不用的资源也不能给其他作业使用。特别是当一个作业完成,另一个作业开始,作业切换过程中的上机、下机将浪费大量时间。

1.1.2 单任务操作系统发展阶段

关于操作系统概念,现在几乎无人不知、无人不晓了。然而,关于操作概念的第一次讨论,是1953年在华盛顿召开的IBM 701计算机用户会议上展开的。到计算机诞生10年后的1955年,计算机上出现了第一个操作系统,当时称为监控程序(Monitor)。该系统是美国通用动力研究实验室(General Motors Research Laboratory, GMRL)为IBM 701计算机研制的,主要目的是实现作业的自动转换、向用户提供方便的操作接口、尽可能高效率地利用计算机,这就是单任务操作系统。

这一时期中,计算机语言的研究也开始萌动。1951年兰宁(J. Halcombe Laning)和齐勒尔(Neal Zierler)建立了一个翻译程序,内含算术和代数语句。其后,苏黎士联邦高等技术学院的卢狄豪斯(H. Rutishauser)设计了一个供科学计算用的相当完善的语言,但终因使用的计算机太小,未能实现翻译程序的工作。1954年,国际商业机器公司开始在纽约研制Fortran语言。

受IBM 701计算机上成功运行监控程序的鼓舞,北美航空公司开始与GMRL实验室合作,共同开发了运行于IBM 704计算机上的操作系统。该系统就是一个支持Fortran语言的监控程序,简记为FMS(Fortran Monitor System)。该系统的研制成功使操作系统达到了一个阶段性高潮。与此同时,计算机硬件也进入了第2代发展时期,计算机的硬件处理能力有了显著的提高。主要表现在:CPU的运行速度明显的加快;内存的容量有了很大增长;外部设备数量增多。计算机硬件的提高为软件的发展奠定了基础。在程序设计方面,人们研制出了一些高级程序设计语言及编译软件,如著名的算法语言Algol、商业通用语言Cobol、工程师代数语言Sale等。硬件和软件的发展,促使操作系统进一步完善。到20世纪60年代初,已有多个自行研制的操作系统在IBM 704计算机上运行,对后来的系统发展产生了良好的影响。

由于高级语言的出现,用户的应用程序可以直接使用高级语言进行设计。然后由管理程序负责调用相关的编译程序将其编译为机器语言程序再投入运行。据当时国际商业机器公司的Fortran研究小组的一份报告称, Fortran的编译程序精妙绝伦,其效率与人工编制的

