

普通高等院校“十一五”规划教材

操作系统 原理

CAOZUO XITONG YUANLI

王旭阳 李睿 编著



国防工业出版社

National Defense Industry Press

普通高等院校“十一五”规划教材

操作系统原理

王旭阳 李睿 编著

国防工业出版社

·北京·

内 容 简 介

操作系统是计算机系统的核心系统软件,操作系统课程是计算机教育的必修课程,是高等院校计算机相关专业必修的核心课,是从事计算机应用专业人员必须掌握的专业知识。

本书详细阐述了操作系统的基本原理、概念和应用,共分8章,第1章操作系统概论。第2章~第6章分别介绍操作系统的进程管理、处理机调度、存储管理、设备管理和文件系统,第7章探讨操作系统的安全与保护问题。前7章每章中都分析了操作系统相关设计原理在Windows系统上的实现技术。第8章以Linux系统为例详细分析了其进程管理、内存管理、设备管理和文件系统。

本书可作为高等学校本科操作系统相关课程的教材或参考书,也可供广大从事操作系统原理研究与系统开发的工程技术人员参考。

图书在版编目(CIP)数据

操作系统原理/王旭阳,李睿编著. —北京:国防工业出版社,2009.1

普通高等院校“十一五”规划教材

ISBN 978-7-118-06055-3

I. 操... II. ①王...②李... 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(2008)第 189562 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

天利华印刷装订有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 21½ 字数 498 千字

2009 年 1 月第 1 版第 1 次印刷 印数 1—4000 册 定价 35.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)68428422

发行邮购: (010)68414474

发行传真: (010)68411535

发行业务: (010)68472764

前 言

操作系统是连接计算机系统硬件和用户的桥梁,是计算机系统的核心组成部分。同时,操作系统课程也是计算机专业教学中的重点专业基础课,不但高等院校计算机相关专业学生必须学习,而且从事计算机行业的从业人员也需要深入了解它。

自从 20 世纪 40 年代首台计算机出现以来,硬件技术和体系结构始终在以惊人的速度发展。其中多数变化是可以量化的,例如,处理器、内存和外设的速度在不断地提升,但它们的体积、价格和能耗却一直在下降。除此之外,计算机的很多方面也发生了质的变化。例如,带有精密输入、输出和存储设备的个人计算机现已无处不在,并且大都已经与局域网或者因特网相联。这些进步给世界带来了激动人心的新局面。20 世纪 90 年代以来,特别是进入 21 世纪之后,操作系统作为计算机领域最活跃的分支之一,许多新的设计要素引入到了新的操作系统中使操作系统发生了本质的变化,其新概念、新技术和新方法层出不穷。为了适应这种发展的趋势,我们结合国内外最新的资料编写了本教材,以适应信息社会计算机科学技术飞速发展的形势和计算机教学内容改革的迫切要求。

本书作者是长期从事操作系统课程教学和科研工作的教师,在结合多年教学经验的基础上,以拓宽基础、注重应用、提高能力为宗旨编写了本教材。本教材的特点之一是在浓缩传统理论精华、保持课堂内容相对稳定的基础上,注重反映当代操作系统发展的最新成果和动向,着眼于操作系统学科知识体系的系统性、先进性和实用性;特点之二是把操作系统成熟的基本原理与当代具有代表性的具体实例紧密结合起来、把操作系统的设计原理与操作系统的实现技术紧密结合起来、把操作系统的理论知识与操作系统的实践紧密结合起来,选择具有代表性的 Windows 主流操作系统作为实例贯穿全书,最后以 Linux 系统为例全面分析操作系统原理在其上的实现技术,这非常有益于学生深入理解操作系统的整体概念并牢固掌握操作系统设计与实现的精髓。

全书共 8 章,每章的最后一节是小结。第 1 章为操作系统概论。第 2 章~第 6 章分别介绍操作系统的进程管理、处理机调度、存储管理、设备管理和文件系统,第 7 章探讨操作系统的安全与保护问题。前 7 章每章中都分析了操作系统的相关设计原理在 Windows 系统上的实现技术。第 8 章以 Linux 操作系统为例详细分析了 Linux 系统的进程管理、内存管理、文件系统和设备管理。

全书由王旭阳主编和统稿。王旭阳编写了第 1 章和第 7 章,李睿编写了第 2 章和第 3 章,刘嘉编写了第 4 章和第 6 章,周小健编写了第 5 章和第 8 章。本书的编写参阅了大量国内外相关文献,在此对所有原作者致以衷心的感谢和深深的敬意!

本书可作为高等学校本科、专科的操作系统相关课程的教材或参考书,也可供计算机等级考试、成人自学考试的考生和广大从事操作系统原理研究与系统开发的工程技术人员参考。

读者在学习本书的过程中遇到疑问或有好的建议和要求,请及时与我们联系,我们将不胜感激。由于水平所限,错误与不足之处在所难免,衷心希望广大读者提出建议和批评意见,联系的电子邮件地址为 wangxy@lut.cn。

作 者

2008 年 10 月

目 录

第 1 章 操作系统概论.....	1	2.1.3 进程的状态	31
1.1 操作系统的概念.....	1	2.1.4 进程的管理	34
1.1.1 用户观点	2	2.2 进程的描述与控制.....	35
1.1.2 系统观点	3	2.2.1 进程的描述	35
1.1.3 软件观点	3	2.2.2 进程的控制	37
1.2 操作系统的形成.....	4	2.3 线程.....	39
1.2.1 手工操作	4	2.3.1 线程的引入	39
1.2.2 监控程序	5	2.3.2 线程的状态	41
1.2.3 多道程序设计	5	2.3.3 线程的并发执行	42
1.2.4 分时系统	6	2.3.4 用户级线程和内核级 线程	42
1.3 操作系统的特征与功能.....	7	2.3.5 线程的描述与控制	44
1.3.1 操作系统的特征	7	2.4 进程同步.....	44
1.3.2 操作系统的功能	8	2.4.1 进程同步的基本概念	45
1.4 用户与操作系统的接口.....	12	2.4.2 进程同步的解决方法	47
1.4.1 操作接口	13	2.4.3 线程同步	56
1.4.2 程序接口	15	2.4.4 多处理机同步	57
1.5 操作系统的分类.....	17	2.5 经典进程同步问题.....	58
1.5.1 批处理系统	17	2.5.1 生产者—消费者问题	58
1.5.2 分时系统	18	2.5.2 哲学家进餐问题	60
1.5.3 实时系统	19	2.5.3 读者—写者问题	63
1.5.4 微机操作系统	20	2.6 进程通信.....	66
1.5.5 网络操作系统	21	2.6.1 信号通信机制	66
1.5.6 分布式操作系统	22	2.6.2 共享文件通信机制	67
1.5.7 嵌入式操作系统	23	2.6.3 共享存储器通信机制	68
1.5.8 智能卡操作系统	23	2.6.4 消息传递通信机制	68
1.5.9 操作系统的其他分类	24	2.7 死锁问题.....	70
1.6 小结.....	26	2.7.1 死锁的形成与定义	71
习题一	26	2.7.2 死锁的预防	73
第 2 章 进程管理.....	28	2.7.3 死锁的避免	75
2.1 进程的引入.....	28	2.7.4 死锁的检测与恢复	78
2.1.1 程序的顺序执行与 并发执行	28	2.7.5 鸵鸟算法	80
2.1.2 进程的概念	30	2.7.6 一种综合的死锁策略	81
		2.7.7 饥饿与活锁	81
		2.8 实例:Windows 进程管理	83

2.8.1 Windows 进程和线程	83	4.1.2 地址转换	127
2.8.2 Windows 进程和线程 的同步	87	4.1.3 程序装入	129
2.8.3 Windows 进程通信	88	4.2 基本存储空间管理	129
2.9 小结	90	4.2.1 单用户存储管理	129
习题二	91	4.2.2 固定分区存储管理	131
第3章 处理机调度	94	4.2.3 可变分区存储管理	133
3.1 概述	94	4.2.4 交换	137
3.1.1 调度层次	94	4.3 分页存储管理	138
3.1.2 调度准则	95	4.3.1 分页存储管理的基本 原理	138
3.2 作业调度	97	4.3.2 硬件结构	140
3.2.1 作业控制块	97	4.3.3 页表结构	144
3.2.2 作业状态	97	4.4 分段存储管理	147
3.2.3 作业调度功能	98	4.4.1 程序分段	147
3.2.4 作业调度时机	98	4.4.2 分段存储管理的基本 原理	148
3.2.5 作业调度算法	99	4.4.3 段的共享与保护	150
3.3 进程调度	102	4.4.4 段页式存储管理	150
3.3.1 进程调度功能	102	4.5 虚拟存储器	152
3.3.2 进程调度时机	103	4.5.1 虚拟存储器的基本 原理	152
3.3.3 进程调度方式	103	4.5.2 请求分页虚拟存储 管理	153
3.3.4 进程调度算法	104	4.5.3 请求分段虚拟存储 管理	164
3.3.5 进程调度过程	109	4.6 实例:Windows 存储管理	167
3.3.6 线程调度	109	4.6.1 Windows 存储管理的 特点	167
3.4 实时调度	110	4.6.2 Windows 存储管理的 内存分配	168
3.4.1 实时调度的要求	110	4.6.3 Windows 内存管理 实现	171
3.4.2 实时任务的分类	111	4.7 小结	172
3.4.3 实时调度算法	111	习题四	174
3.5 多处理机调度	117	第5章 设备管理	176
3.5.1 不相关进程的调度	117	5.1 概述	176
3.5.2 相关进程的调度	119	5.1.1 设备分类	176
3.5.3 群调度	120	5.1.2 设备控制器	177
3.6 实例:Windows 调度	121	5.1.3 设备通道	180
3.6.1 进程和线程优先级	121	5.2 I/O 控制	183
3.6.2 多处理机调度	123		
3.7 小结	123		
习题三	124		
第4章 存储管理	126		
4.1 概述	126		
4.1.1 存储层次	126		

5.2.1	程序直接控制 I/O 方式	183	6.1.1	文件的基本概念	225
5.2.2	中断驱动方式	184	6.1.2	文件系统模型	226
5.2.3	DMA 方式	185	6.1.3	文件分类	227
5.2.4	通道方式	187	6.1.4	文件属性	228
5.3	I/O 软件层次	189	6.1.5	文件操作	229
5.3.1	I/O 软件的目标	189	6.2	文件结构	230
5.3.2	I/O 中断处理程序	189	6.2.1	文件与记录	230
5.3.3	I/O 设备驱动程序	190	6.2.2	文件的逻辑结构	231
5.3.4	与设备无关的 I/O 软件	192	6.2.3	文件的物理结构	231
5.3.5	用户空间的 I/O 软件	193	6.3	文件目录	233
5.4	缓冲管理	195	6.3.1	分区与目录	233
5.4.1	缓冲的引入	195	6.3.2	一级目录	234
5.4.2	单缓冲	195	6.3.3	二级目录	235
5.4.3	双缓冲	196	6.3.4	树形目录	235
5.4.4	循环缓冲	197	6.3.5	无环图目录和通用图目录	236
5.4.5	缓冲池	198	6.4	文件系统的实现	238
5.5	设备分配	200	6.4.1	文件系统调用的实现	238
5.5.1	设备分配中的数据 结构	200	6.4.2	文件共享	239
5.5.2	设备独立性	201	6.4.3	文件的安全与保护	239
5.5.3	设备分配技术	202	6.5	辅存空间管理	241
5.6	磁盘调度和管理	205	6.5.1	辅存的分配方法	241
5.6.1	磁盘的物理性能	205	6.5.2	空闲空间管理	244
5.6.2	磁盘调度算法	209	6.6	实例:Windows 文件系统	245
5.6.3	磁盘调度算法的比较	212	6.6.1	Windows 文件系统 的基本概念	245
5.6.4	磁盘的错误处理	213	6.6.2	Windows 文件系统 模型与结构	246
5.6.5	独立磁盘冗余阵列	213	6.6.3	Windows 文件系统 的实现	249
5.7	实例:Windows 设备管理	218	6.7	小结	253
5.7.1	I/O 系统结构	218	习题六		254
5.7.2	设备管理的数据 结构	219	第 7 章	操作系统的安全与保护	255
5.7.3	I/O 处理过程	221	7.1	安全	255
5.7.4	中断处理	221	7.1.1	安全问题	255
5.8	小结	222	7.1.2	安全威胁	256
习题五		223	7.1.3	安全目标	258
第 6 章	文件系统	225	7.1.4	操作系统安全	259
6.1	概述	225	7.2	保护	261

7.2.1	操作系统保护层次·····	261		及其转换·····	297
7.2.2	内存存储器的保护·····	261	8.2.3	Linux 进程控制·····	298
7.2.3	面向用户的访问控制···	262	8.2.4	Linux 进程调度·····	301
7.2.4	面向数据的访问控制···	262	8.2.5	Linux 进程通信·····	302
7.3	入侵者·····	263	8.2.6	Linux 线程·····	307
7.3.1	入侵技术·····	263	8.3	Linux 内存管理·····	309
7.3.2	口令保护·····	264	8.3.1	Linux 内存管理体系···	309
7.3.3	入侵检测·····	271	8.3.2	Linux 物理内存管理···	309
7.4	恶意软件·····	276	8.3.3	Linux 虚拟内存管理···	313
7.4.1	病毒·····	279	8.3.4	Linux 内核空间管理···	316
7.4.2	反病毒方法·····	281	8.4	Linux 文件系统·····	318
7.5	实例:Windows 安全机制·····	284	8.4.1	Linux 的虚拟文件 系统·····	318
7.5.1	安全性系统组件·····	285	8.4.2	Linux 的 EXT2 文件 系统·····	321
7.5.2	保护对象·····	285	8.4.3	Linux 文件系统管 理的缓冲机制·····	324
7.5.3	访问控制策略·····	286	8.4.4	Linux 的文件管理和 操作·····	325
7.5.4	访问令牌·····	286	8.5	Linux 设备管理·····	327
7.5.5	安全描述符·····	287	8.5.1	设备驱动程序·····	327
7.6	小结·····	289	8.5.2	设备的识别与设备 文件·····	328
习题七	·····	290	8.5.3	中断处理·····	328
第 8 章 Linux 系统	·····	291	8.5.4	字符设备驱动·····	330
8.1	Linux 系统概述·····	291	8.5.5	块设备驱动·····	331
8.1.1	Linux 的诞生·····	291	8.5.6	网络设备驱动·····	332
8.1.2	Linux 的版本·····	291	习题八	·····	333
8.1.3	Linux 内核体系 结构·····	293	参考文献	·····	335
8.1.4	Linux 系统组成·····	293			
8.2	Linux 进程管理·····	294			
8.2.1	Linux 进程描述·····	294			
8.2.2	Linux 系统进程状态				

第 1 章 操作系统概论

操作系统是计算机硬件和用户之间的中介程序,其目的是为用户提供方便有效的执行程序的环境。操作系统完成这些任务的方式多种多样,如大型机操作系统设计的目的是为了充分优化硬件的利用率,而个人计算机操作系统是为了能支持复杂游戏、商业应用或位于两者之间的事务等。在所有的系统软件中,操作系统是一种首要的、最基本的、最重要的系统,也是最庞大、最复杂的系统软件。几十年来,软件界花费了大量的时间和资金来研究、开发、扩展和完善计算机操作系统,使其获得了飞速的发展。

本章主要讲述什么是操作系统及其发展与研究现状。首先跟踪操作系统的发展历史,介绍操作系统形成的过程;然后详细介绍操作系统的特征及其功能,最后讨论目前比较典型的几种操作系统。

1.1 操作系统的概念

操作系统(Operating System, OS)的出现、使用和发展是 40 余年来计算机软件的一个重大进展。计算机发展到现在,从个人机到巨型机,无一例外都配置一种或多种操作系统。操作系统也可被看作是用户和计算机硬件之间的一种接口,它已经成为现代计算机系统不可分割的重要组成部分,它为人们建立各种各样的应用环境奠定了重要基础。配置操作系统的主要目标可归结为:

- (1) 方便用户使用。OS 通过提供用户与计算机之间的友善接口来方便用户使用。
- (2) 扩大机器功能。OS 通过扩充改造硬件设施和提供新的服务来扩大机器功能。
- (3) 管理系统资源。OS 有效管理好系统中所有硬件和软件资源使之得到充分利用。
- (4) 提高系统效率。OS 合理组织好计算机的工作流程以改进系统效能和提高系统效率。
- (5) 构筑开放环境。OS 遵循有关国际标准来设计和构造以构筑出一个开放环境。其含义主要是指:遵循有关国际标准(如开放的通信标准、开放的用户接口标准、开放的线程库标准等);支持体系结构的可伸缩性和可扩展性;支持应用程序在不同平台上的可移植性和互操作性。

计算机系统可以粗分为四个部分:硬件、操作系统、应用程序和用户。图 1-1 给出了操作系统的定位。

系统最底层是硬件,提供了基本的可计算性资源,包括处理器、寄存器、存储器以及各种 I/O 设备。在很多情形下硬件可以分成两层或更多层,最底层是物理设备,包括集成芯片、连线、电源、阴极射线管以及类似的设备等。接着是微体系结构,这层中有 CPU 的专用内部寄存器以及包含算术逻辑单元的数据通道。再往上是由相关硬件和对汇编程序员可见的指令构成的指令集体系结构(Instruction Set Architecture, ISA)层,这一层通常称为机器语言。

应用程序,如字处理程序、电子制表软件、编译器、网络浏览器等,规定了按何种方式使用

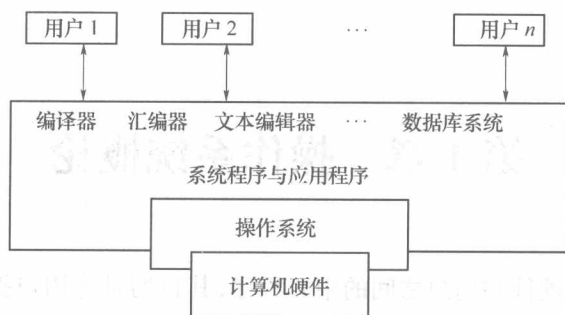


图 1-1 计算机系统组成

这些资源来解决用户的计算机问题。

操作系统的作用类似于政府,用来控制和协调各用户的应用程序对硬件的使用。与政府一样,操作系统本身并不能实现任何有用的功能,只不过提供了一个方便其他程序工作的环境。因此,操作系统性能的高低决定着计算机的潜在硬件性能能否发挥出来,操作系统本身的安全可靠程度决定了整个计算机系统的安全性和可靠性。操作系统是软件技术含量最大、附加值最高的部分,是软件技术的核心,是软件的基础运行平台。

尽管“操作系统”这个名称诞生至今已有了几十年的时间,计算机使用人员也都知道它,但要对其下一个精确的定义并非轻而易举。很多论述操作系统的书籍从不同的角度对操作系统下了不同的定义,综合起来,通常把操作系统定义为用来控制和管理计算机系统资源、方便用户使用的程序和数据结构的集合。

操作系统在计算机系统中的作用可以从三个方面来理解。

(1) 从用户角度看,OS 是其与计算机硬件之间的接口。

(2) 从系统角度看,OS 是计算机系统的资源管理者。

(3) 从软件角度看,OS 就像一台虚拟计算机。

下面分别从这三个方面来介绍操作系统。

1.1.1 用户观点

用户观点根据所使用界面的不同而异。绝大多数计算机用户坐在个人计算机前,个人计算机有显示器、键盘、鼠标和主机。这类系统设计是为了让单个用户独立使用其资源,优化用户所进行的工作(或游戏),设计目的主要是为了用户使用方便,性能是次要的,而且不在乎资源利用率。

有些用户坐在与大型机或小型机相连的终端前,所有用户通过其终端访问同一计算机。这些用户共享资源并可交换信息。这类系统设计的目的是使资源利用率最大化,确保所有的 CPU 时间、内存和 I/O 设备都能得到充分利用,并且确保没有用户使用超过其限额以外的资源。

另一些用户坐在工作站前,工作站与其他工作站和服务器相连。这些用户不但可以使用专用的资源,而且可以使用共享资源,如网络和服务器提供的文件、计算和打印服务等。因此,这类操作系统设计的目的是个人可用性和资源利用率的折中。

近年来,许多类型的手持计算机成为时尚。这些设备绝大多数为单个用户独立使用,有的也通过有线或(更为常见)无线与网络相连。由于受电源和接口所限,它们只能执行相对少的

远程操作。这类操作系统的设计目的主要是个人可用性,当然如何在有限的电池容量中发挥最大的效用也很重要。

有的计算机几乎或根本没有用户管理。例如,在家电和汽车中所使用的嵌入式计算机可能只有一个数值键盘,只能打开或关闭灯来显示状态,这些设备及其操作系统通常都设计成无需用户干预就能执行的模式。

操作系统改造和扩充过的计算机不但功能更强,使用也更为方便,用户可直接调用系统提供的各种功能,而无需了解软硬件本身的细节,因此,从用户的角度看,操作系统是用户与计算机硬件之间的接口,用户一般可从以下三种方式获得计算机系统提供的服务。

(1) 命令方式。用户通过键盘输入有关命令来直接使用计算机,以获得计算机所能提供的服务。用户也可将若干个命令集中存储在一个文件中,以批处理的方式连续运行命令,获得多种服务。

(2) 系统调用。用户在应用程序中调用操作系统向用户提供的服务程序以获得系统服务。

(3) 图形界面。用户在窗口环境中通过鼠标、按键、菜单和对话框等方式操纵计算机,这是现代操作系统向多媒体转化的标志之一。

1.1.2 系统观点

从计算机的角度来看,操作系统是与硬件最为密切的程序,可以将操作系统看作是资源分配器。计算机系统有许多资源:处理器(CPU)、存储器(内存和外存)、外部设备和文件。现代计算机系统都支持多个用户、多道作业共享。那么,面对众多的程序争夺处理器、存储器、设备和共享软件资源,如何协调,从而有条不紊地进行分配呢?操作系统就负责登记谁在使用什么样的资源,系统中还有哪些资源空闲以及当前响应谁的资源请求,应该收回哪些不再使用的资源等。面对许多甚至冲突的资源请求,操作系统要提供一些机制去协调进程间的竞争与同步,提供机制决定如何为各个程序和用户分配资源,以便使计算机系统能有效而公平地运行。

现代计算机硬件设备种类越来越多,功能越来越强,控制和操作起来也越来越复杂。如果一个程序员要直接与打印机、磁盘等 I/O 设备打交道,那么就要对每一种设备编制几千、几万条机器指令,这不仅是用户所不及的,对系统存储的信息来说也是极其不安全的。因此,操作系统向用户提供了高级而调用简单的服务,掩盖了绝大部分硬件设备复杂的特性和差异,使用户可以免除大量的令人乏味的杂务,而把精力集中在自己所要处理的任务上。

操作系统的重要任务之一是对资源进行抽象研究,找出各种资源的共性和个性,有序地管理计算机中的硬件、软件资源,跟踪资源使用情况,监视资源的状态,满足用户对资源的需求,协调各程序对资源的使用冲突;研究使用资源的统一方法,让用户简单、有效地使用资源,最大限度地实现各类资源的共享,提高资源利用率,从而使得计算机系统的效率有很大提高。

1.1.3 软件观点

许多年以前,人们就认识到必须找到某种方法把硬件复杂性与用户隔离开来,经过不断的探索和研究,目前采用的方法是在计算机裸机上加上一层又一层软件来组成整个计算机系统,同时为用户提供一个容易理解和便于程序设计的接口。每当在计算机上覆盖一层软件,提供了一种抽象,系统的功能便增加一点,使用就更加方便一点,用户可用的运行环境就更加好一点。操作系统是紧靠硬件的第一层软件,计算机上覆盖操作系统后,可以扩展基本功能,为

用户提供一台功能显著增强,使用更加方便,安全性、可靠性好,效率明显提高的机器,称为虚拟计算机,或称操作系统虚拟机(Virtual Machine)。

从软件的观点看,操作系统有其作为软件的外在特性和内在特性。

所谓外在特性是指,操作系统是一种软件,它的外部表现形式,即它的操作命令定义集和界面,完全确定了操作系统这个软件的使用方式。例如,操作系统的各种命令、各种系统调用及其语法定义等。我们需要从操作系统的使用界面上学习和研究操作系统,才能从外部特性上把握住每一个操作系统的性能。

所谓内在特性是指,操作系统作为一个软件,它具有一般软件的结构特点,然而这种软件又不是一般的应用软件,它还具有一般软件所不具备的特殊结构。因此,学习和研究操作系统时就需要探讨其特殊性,从而更好地把握其结构上的特点。例如,操作系统是直接与硬件打交道的,那么与硬件交互的软件是怎么组成的?每个组成部分的功能和各部分之间的关系是什么?等等,即要研究其内部算法。

1.2 操作系统的形成

操作系统的发展是随着计算机硬件技术的发展和应用需求的增长而不断发展的。同其他任何事物一样,操作系统也有它的诞生、成长和发展的过程。了解操作系统的发展过程,有助于我们更深刻地认识操作系统基本概念的内在含义。

计算机的发展经历了第一代电子管时代(1946年—1957年)、第二代晶体管时代(1958年—1964年)、第三代集成电路时代(1965年—1970年)以及第四代大规模/超大规模集成电路时代(1971年至今)等阶段。本节将沿着这个线索介绍操作系统的发展历史。

1.2.1 手工操作

由于第二次世界大战对武器装备设计的需要,美国、英国和德国等国家陆续开始了电子数字计算机的研究工作。20世纪40年代中期,哈佛大学的Howard Aiken、普林斯顿高等研究院的John Neumann(冯·诺依曼)、宾西法尼亚大学的J. Presper Eckert和William Mauchley、德国电话公司的Konraad Zuse以及其他一些人,都使用真空管成功地建造了运算机器。这些非常巨大的机器,使用了数万个真空管,占据了几个房间,然而其运算速度却比现在最便宜的家用计算机还要慢得多。

在这个阶段,程序设计全部采用机器语言,通过在一些插板上的硬连线来控制其基本功能,没有程序设计语言(甚至没有汇编语言),更谈不上操作系统。使用机器的方式是程序员提前在墙上计时表上预约一段时间,然后到机房将他的插件板插到计算机里,在接下来的几小时里计算自己的题目,并期盼着在这段时间中几万个真空管不会有烧断的。这时,实际上所有的题目都是数值计算问题。

到了20世纪50年代早期,出现了穿孔卡片,可以将程序写在卡片上然后读入计算机,此时不再用插板了,计算过程如下:

(1) 人工把源程序用穿孔机穿制在卡片或纸带上。

(2) 将准备好的汇编解释程序或编译系统装入计算机。

(3) 汇编程序或编译系统读入人工装在输入机上的穿孔卡或穿孔带。

(4) 执行汇编过程或编译过程,产生目标程序,并输出目标卡片叠或纸带。

(5) 通过引导程序把装在输入机上的目标程序读入计算机。

(6) 启动目标程序执行,从输入机上读入人工装好的数据卡或数据带。

(7) 产生计算结果,执行结果从打印机上或卡片机上输出。

在一个程序员上机期间,整台计算机连同附属设备全部被其占用。程序员实际上兼职操作员,效率低下。其特点是手工操作,独占方式。后来人们开发了汇编语言及其汇编编译程序以及其他一些控制外设的程序等,但工作方式仍属于这一阶段。

1.2.2 监控程序

20 世纪 50 年代晶体管的发明改变了现状,晶体管时代的计算机比较可靠,厂商可以成批地生产并将其销售给用户,用户可以指望计算机长时间运行来完成一些有用的工作,FORTRAN 高级语言于 1954 年提出,1956 年正式设计完成。ALGOL 高级语言于 1958 年引入,COBOL 高级语言于 1959 年引入。此时,设计人员、生产人员、操作人员、程序人员和维护人员之间第一次有了明确的分工。

计算机安装在专门的空调房间里,有专业人员操作。要运行一个作业(job),程序员首先将程序写在纸上(用高级语言或汇编语言),然后穿孔成卡片,再将卡片盒带到输入室交给操作员。当计算机运行完当前任务后,将结果从打印机输出,操作员到打印机上撕下运算结果并送到输出室,程序员稍后就能取到结果。然后,操作员从已经送到输入室的卡片盒中读入另一个任务,计算机就开始运行下一任务。如果需要 FORTRAN 编译器,操作员还要从文件柜把它取来读入计算机。当操作员在机房里走来走去时许多机时被浪费掉了。

由于处理器速度的提高,使得手工操作设备进行输入/输出与计算机的计算速度不匹配,因而人们设计了监督程序(也称管理程序)来实现作业的自动转换处理。这期间每道作业由程序提供一组在某种介质上准备好的作业信息(文件),包括用作业控制语言书写的作业说明书,相应的程序和数据。作业控制语言被穿孔成一叠作业卡片,由程序员提交给系统操作员,而操作员将作业成批地输入到计算机中,由监督程序识别一个作业进行处理后再取下一个作业。这种自动定序的处理方式称为“批处理”方式,由于是串行执行作业,因此这种早期的批处理方式又称为单道批处理,也被认为是操作系统的雏形。

1.2.3 多道程序设计

在第二代计算机后期,特别是进入第三代以后,系统软件有了很大发展,其作用也日益显著,同时,硬件也有了很大发展,主存容量增大,出现了大容量的辅助存储器——磁盘以及代理 CPU 来管理设备的通道,使得计算机体系结构发生了很大变化。由以中央处理器为中心的结构改变为以主存为中心,通道的产生使得输入/输出操作与 CPU 之间的并行工作成为可能。软件系统也随之相应变化,实现了在硬件提供并行处理之上的多道程序设计。

所谓多道程序设计是指允许多个程序同时存在于主存之间,由中央处理器以切换方式为之服务,使得多个程序可以同时执行。如图 1-2 所示,主存中多个相互独立的程序均处于开始和结束之间,从宏观上看是并行的,多道程序都处于运行而未运行结束的过程中;从微观上看是串行的,各道程序轮流占用 CPU 交替地执行。

引入多道程序设计技术有以下优点:一是提高 CPU、主存和设备的利用率;二是提高系统的吞吐率,使单位时间内完成的作业数增加;三是充分发挥计算机硬部件的并行性,设备与设备之间、CPU 与设备之间均可并行工作。其主要缺点是延长作业的周转时间。这个阶段的计

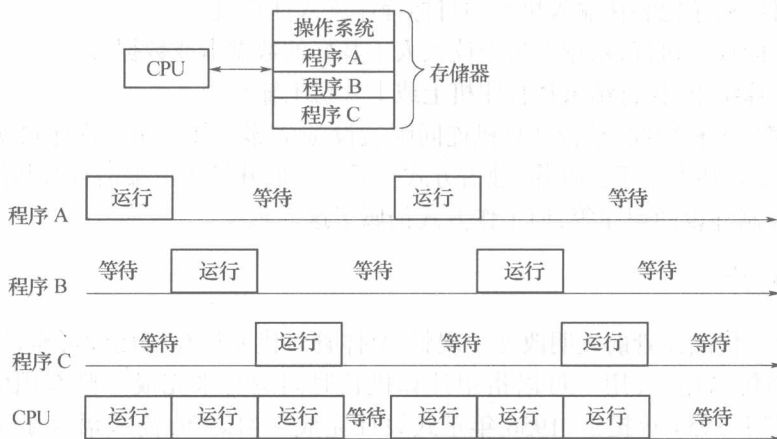


图 1-2 多道程序设计

计算机资源不再是“串行”地被一个个用户独占,而可以同时被几个用户共享,从而极大地提高了系统在单位时间内处理作业的能力。这时管理程序已迅速地发展成为一个重要的软件分支——操作系统。

这一代计算机典型的操作系统是 FMS(FORTRAN Monitor System, FORTRAN 监控系统)和 IBM 为 7094 机配备的操作系统 IBSYS。这些操作系统由监控程序、特权指令、存储保护和简单的批处理构成(如图 1-3 所示)。

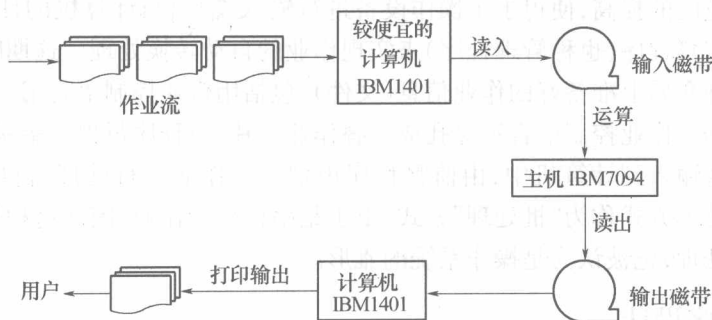


图 1-3 批处理操作系统示意图

1.2.4 分时系统

第三代计算机适用于大型科学计算和繁忙的商务数据处理,但其实质上仍旧是批处理系统。许多程序员很怀念第一代计算机的使用方式,那时他们可以几个小时独占一台机器,可以及时地调试他们的程序。而对第三代计算机,从一个作业提交到运算结果取回往往长达数小时,更有甚者,一个逗号的误用就会导致编译失败从而可能浪费程序员半天的时间。

程序员们的希望很快得到了响应,这种需求导致了分时系统(Timesharing System)的出现。所谓分时系统是指多个用户通过终端设备与计算机交互作用来运行自己的作业,共享一个计算机系统而互不干扰,就好像自己有一台计算机。它实际上是多道程序设计的变种。在分时系统中,假设有 20 个用户登录,其中 17 个用户在思考、谈论或喝咖啡,则 CPU 可分配给其他三个需要服务的作业轮流执行。由于调试程序的用户常常只发出简短的命令(如编译一

个5页的源程序),而很少有长的费时命令(如上百万条记录的文件排序),所以计算机能够为许多用户提供高速的交互式服务,同时在CPU空闲时还可能后台运行一大批作业。分时系统的思想于1959年在麻省理工学院提出,第一个真正的分时系统CTSS(Compatible Time Sharing System,兼容分时系统)是麻省理工学院的Fernando Corbato等人于1961年在一台改装过的IBM 7094/94机上开发成功的,当时有32个交互式用户。但直到第三代计算机广泛采用了必需的保护硬件之后,分时系统才逐渐流行开来。这个时候人们认为操作系统逐渐步入了成熟阶段。

1.3 操作系统的特征与功能

1.3.1 操作系统的特征

操作系统作为一种系统软件,有着与其他一些软件所不同的特征。

1. 并发性(Concurrence)

并发是指在某一时间间隔内计算机系统中运行着多个程序。并发与并行是有区别的,并行是指在同一时刻计算机内有多个程序都在运行,这只有在多CPU系统中才能实现。在单CPU系统中,程序的并发性具体体现在用户程序与用户程序之间以及用户程序与操作系统程序之间的并发执行两个方面。

在CPU环境下,从宏观上看,这些程序是同时向前推进的,从微观上看,这些并发执行的程序交替地在这个单CPU上运行着。而在多CPU系统中,多个程序的并发特性表现为不仅在宏观上是并行的,而且在微观上程序也是并发执行的。在分布式系统中,多个计算机的并存使程序的并发特征得到了更充分的体现。

2. 共享性(Sharing)

共享是指多个用户或程序共享系统的软件、硬件资源。共享首先出于经济方面的考虑,因为向每一个用户分别提供所有的资源是非常浪费的。共享可以提高各种系统设备和系统软件的使用效率。在合作开发某一项目时,同组用户共享软件和数据库可以大大提高开发效率和速度。

共享一般有两种方式:互斥共享和同时共享。

(1) 互斥共享。系统中的有些资源如打印机、磁带机、扫描仪等,虽然可以供多个用户程序同时使用,但在一段特定的时间内只能由某一个用户程序使用。当这类资源中的一个正在被使用的时候,其他请求该资源的程序必须等待,并且在这个资源被使用完了以后才由操作系统根据一定的策略再选择一个用户程序占有该资源。通常把这样的资源称为临界资源。许多操作系统维护的重要系统数据都是临界资源,它们都要求被互斥共享。

(2) 同时共享。系统中的有些快速设备如磁盘,尽管也只能允许多个作业串行地访问,但由于作业访问和释放该资源的时间极短,在宏观上可看作允许多个程序同时访问,这类设备的访问方式被认为是同时访问。

软件的共享方式也可分为互斥共享和同时共享。一般来说,只读的数据、数据结构、只读的文件以及纯可执行的文件可同时共享,而可写数据、数据结构和文件只能互斥共享。

3. 异步性(Asynchronism)

操作系统的第三个特性是异步性,也称随机性。在多道程序环境中,允许多个进程并发执

行,由于资源有限而进程众多,多数情况下,进程的执行不是“一气呵成”,而是“走走停停”。例如,一个进程在 CPU 上运行一段时间后,由于等待资源满足或事件发生,它被暂停执行,CPU 转让给另一个进程执行。系统中的进程何时执行?何时暂停?以什么样的速度向前推进?进程总共要花多少时间才能完成?这些问题都是不可预知的,或者说进程是以异步方式运行的,其导致的直接后果是程序执行结果可能不唯一。异步性给系统带来了潜在的危险,有可能导致进程产生与时间有关的错误,可以说操作系统运行在随机的环境下。但只要运行环境相同,操作系统必须保证多次运行进程都会获得完全相同的结果。

操作系统中的随机性处处可见。例如,作业到达系统的类型和时间是随机的;操作员发出命令或按钮的时间是随机的;运行程序发生错误或异常的时间是随机的;各种各样硬件和软件中断事件发生的时间是随机的;等等。随机性并不意味着操作系统就无法控制使用资源的程序的执行,操作系统内部产生的事件序列有许许多多可能,而操作系统的一个重要任务是必须确保捕捉任何一种随机事件,正确处理可能发生的随机事件,正确处理已经产生的随机事件序列,否则将会导致严重的后果。

4. 虚拟性(Virtual)

操作系统向用户提供了比直接使用裸机简单方便的高级服务,从而对程序员隐藏了对硬件操作的复杂性,相当于在原先的物理计算机上覆盖了一层至多层系统软件,将其改造成一台功能更强大而且易于使用的扩展机或虚拟机。例如,在多道程序系统中,物理 CPU 可以只有一个,每次也仅能执行一道程序,但通过多道程序和分时使用 CPU 技术,宏观上有多个程序在执行,就好像有多个 CPU 在为各道程序工作一样,物理上的一个 CPU 变成了逻辑上的多台虚拟设备;窗口技术可把一个物理屏幕变成逻辑上的多个虚拟屏幕;通过时分或频分多路复用技术可以把一个物理信道变成多个逻辑信道;IBM 的 VM 技术把物理上的一台计算机变成逻辑上的多台计算机。虚拟存储器则是把物理上的多个存储器(主存和辅存)变成逻辑上的一个存储器(虚存)的例子。

从另一方面说,操作系统是一个极其庞大、复杂的软件系统,它也不是十全十美的。在某些特殊的情况下,一些较为隐蔽的问题可能暴露出来,这种错误同样也不易复现和难以测试。就像一个城市的交通问题,由于受到很多不可预测因素的影响,交通事故也不可能复现。一个大型的操作系统在经过测试和交付用户使用后,还可能隐含有几百上千个潜在的缺陷(Bug),而这种缺陷往往需要几年、几十年的时间来修复,并且难以彻底消除。

1.3.2 操作系统的功能

操作系统有两个基本的任务。

(1) 创建一个多自治抽象组件的虚拟机环境,其中的大多数组件可以并发运行。例如,操作系统使用多道程序设计来为每个进程创建一个虚拟机。

(2) 根据计算机管理员的策略来协调组件的使用。例如,调度器决定什么时机、选择什么进程为它分配处理器。

操作系统的创建部分提供了程序员使用的各种抽象资源(如进程、线程和资源)。协调部分管理这些资源的并发使用,并使得一组进程可以协同工作。

资源管理是操作系统的一项主要任务,而控制程序执行、扩充机器功能、提供各种服务、方便用户使用、组织工作流程、改善人机界面等都可以从资源管理的角度去理解。从资源管理的观点来看操作系统应具备五类功能:处理机管理、存储管理、设备管理、文件管理和用户接口。

1. 处理机管理

处理机是整个计算机系统的核心硬件资源,它的性能和使用情况对整个计算机系统的性能有着关键的影响。处理机也是计算机系统最昂贵的资源,它的速度一般比其他硬件设备的工作速度要快得多,其他设备的正常运行往往也离不开 CPU。因此,有效地管理 CPU,充分利用 CPU 资源也是操作系统最重要的管理任务。

为了提高处理机的利用率,现代操作系统都采用了多道程序技术。如果一个程序因等待某一条件而不能运行下去时,就把处理机占用权转交给另外一个可运行程序,或者当出现了一个比当前运行程序更重要的可运行程序时,后者可能强占 CPU,为了描述多道程序的并发运行操作系统引入了进程的概念。所以,处理机管理也可以认为就是进程管理和作业管理,主要包括以下几个方面。

(1) 进程控制。在多道程序环境下,进程是系统进行资源分配的单位。进程控制的主要任务是创建进程、撤消结束的进程以及控制进程运行时的各种状态转换。

(2) 进程同步。进程同步的主要任务是协调各并发进程的运行过程,保证各进程可以安全、有效地访问系统资源或者实现它们之间的相互合作。因此,操作系统一般提供两种进程同步机制来协调进程的执行:同步和互斥。互斥指多个进程对临界资源访问时采用互斥的形式;同步则是在相互协作共同完成任务的进程之间协调其执行顺序。

(3) 进程通信。在多道程序环境下,系统为一个应用程序建立多个进程,在这些进程相互合作去完成一个共同任务的过程中,往往需要交换信息,如何实现和保证这些交换就是进程通信的任务。

(4) 调度。调度通常包括进程(线程)调度和作业调度等。调度的任务是采用适当的调度策略(算法)来调度作业进入内存或者调度进程(线程)获得 CPU 开始执行。不同类型的操作系统所需的环节会有所差异。

2. 存储管理

存储器可以说是操作系统中最重要的系统资源,存储器对作业的重要程度就如同土地对于人类。存储管理主要管理内存资源,虽然内存芯片的集成度不断地提高,价格不断地下降,但是由于计算机对于内存的需求量大,所以相对来看内存整体的价格仍然较昂贵,而且受 CPU 寻址能力的限制,内存的容量也是有限的。因此,当多个程序共享有限的内存资源时就需要考虑如何为它们分配内存空间,同时,既要使用户存放在内存中的程序和数据彼此隔离、互不干扰,又要能保证在一定条件下共享。尤其是当内存不够用时解决内存扩充问题,即将内存和外存结合起来管理,为用户提供一个容量比实际内存大得多的虚拟存储器。操作系统这一部分的功能与存储器的硬件组织结构密切相关。

存储管理主要提供以下功能。

(1) 内存的分配与回收。一个有效的存储分配机制,能够以每个存储单元的状态作为依据,在用户提出需求时予以快速响应,为之分配相应的存储区域;在用户进程不再需要它及时回收,以供其他用户使用。内存分配的主要任务是要为每个进程分配空间并要尽可能地提高内存资源利用率。内存分配有静态分配和动态分配两种方式。内存分配机制具有以下结构和功能。

① 内存分配数据结构。用于记录内存空间的使用情况,是内存分配的依据。

② 内存分配功能。按照一定的内存分配算法为用户程序分配内存空间。

③ 内存回收功能。对用户不再需要的内存根据释放请求进行回收。