

操作系统 习题与解答

Schaum's Outline of Operating Systems

最佳的复习资料，实用的辅助教材

与国外高校计算机水平保持同步

为考研和出国深造奠定坚实基础

J. Archer Harris 著
须德 等译

本系列丛书
全球销售超过
3000万册！

全美经典
学习指导系列

485

TP316.44

H11

操作系统 习题与解答

Stallin's Outline of Operating Systems

J. Archer Harris 著

须德 等译

本书将操作系统原理纲要和题解综合为一体,讲述了进程管理、进程间的通信和同步、存储器管理、虚拟存储器、文件系统管理、设备管理和安全等方面的内容。每章都是首先概要地叙述了操作系统中该主题的重要知识点,接着给出了适当数量的习题及其解答,然后给出了和习题相似内容的初充习题,以便读者能举一反三,灵活运用学到的知识,而且每章最后还给出了补充题的答案,可供读者检验自己掌握知识的程度。

本书内容全面,重点突出,叙述简洁,习题包括了基本题和较深入的题,可供读者选用。本书可作为高校计算机专业操作系统课程的教学参考书,也可作为自学人员的配套教材。

J. Archer Harris: Operating Systems (ISBN:0-07-136435-8).

Copyright © 2002 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书版权登记号:图字:01-2002-1826

图书在版编目(CIP)数据

操作系统习题与解答/(美)哈里斯(Harris, J. A.)著;须德等译. —北京:机械工业出版社, 2003. 1

(全美经典学习指导系列)

书名原文: Operating Systems

ISBN 7-111-11380-2

I. 操… II. ①哈…②须… III. 操作系统(软件)—解题 IV. TP316-44

中国版本图书馆 CIP 数据核字(2002)第 102865 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:朱 劼

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·12 印张

印数:0 001-5 000 册

定价:19.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

前 言

我第一次接触的操作系统是在 DEC PDP-11 计算机上运行的 Unix V6。阅读了 Unix 的源代码,我认识到操作系统是引人入胜的软件,它复杂但精美。对于一个程序员来说,操作系统是最大的挑战,它包含了从底层设备管理到并发控制再到面向对象设计的方方面面。

本书探讨现代操作系统的设计原理,面向希望更多地了解操作系统或者对某个具体操作系统感兴趣,希望更广泛地了解操作系统行为的读者。本书每章包括课程内容的简介和大量的题解。本书适合与相应的操作系统教材配套使用,作为课程的补充读物。

本书重点介绍操作系统的设计原理,而不介绍某个具体操作系统的详细特性。虽然书中的例子都是引自各个具体操作系统,如 DOS、Windows,引用得最多的是 Unix,但是其中的内容适用于普遍的设计原理,不包括一些比较专门的议题,如实时系统或分布式系统。

我要感谢在成书过程中帮助过我的所有 McGraw-Hill 的员工,也要感谢 James Madison 大学的学生和同事,感谢他们的理解和支持。特别要感谢 Dr. Ramón Mata Toledo,感谢他为这本书问世所做的工作。

最后,我要对我的妻子兼同事 Mancy Harris 表示由衷的谢意,除了给予我感情上的支持外,她对本书的关键性的评阅所起的作用是无法估量的。

虽然我希望书中的内容准确无误,但是错误之处难免。错误、疏漏之处或建议改进意见请发邮件到 schaumos@mailgate.cs.jmu.edu。本书的更新部分可查阅 URL <http://www.cs.jmu.edu/users/harrisja/schaumos>。

J. Archer Harris

目 录

译者序	
前言	
第 1 章 引言	1
1.1 机器硬件	1
1.1.1 陷阱和中断	2
1.1.2 多态执行	3
1.2 操作系统结构	3
1.2.1 操作系统的类型	4
1.2.2 操作系统的内核	5
1.2.3 引导进程	5
1.3 本书其余部分概要	6
第 2 章 进程管理	11
2.1 进程调度	11
2.2 进程状态	12
2.3 调度标准	14
2.4 调度算法	15
2.4.1 先来先服务	15
2.4.2 最短作业优先	15
2.4.3 最短剩余时间	16
2.4.4 循环	16
2.4.5 优先级	16
2.4.6 多级反馈队列	16
2.5 调度算法的性能	17
2.6 进程的属性	18
2.6.1 运行状态和调度	18
2.6.2 存储器管理	19
2.6.3 硬件状态	19
2.6.4 信号机制	19
2.6.5 存取控制	19
2.6.6 输入和输出	19
2.6.7 其他	19
2.7 进程管理程序调用	20
第 3 章 进程间的通信和同步	35
3.1 进程间的通信	35
3.2 进程同步	38
3.2.1 临界区	38
3.2.2 禁止中断	39
3.2.3 TestAndSet 指令	39
3.2.4 Swap 指令	40
3.2.5 Wait 和 Signal 原语	41
3.2.6 信号量	42
3.2.7 Dekker 算法	43
3.2.8 Peterson 算法	43
3.2.9 Bakery 算法	44
3.2.10 管程	45
3.3 死锁	46
3.3.1 预防死锁	46
3.3.2 避免死锁	47
3.3.3 死锁检测	49
3.3.4 死锁恢复	51
3.3.5 鸵鸟算法	52
第 4 章 存储器管理	71
4.1 单一绝对分区	71
4.2 单一重定位分区	72
4.3 多道程序	72
4.4 多重分区	72
4.4.1 多重固定分区	72
4.4.2 多重可变分区	73
4.4.3 伙伴系统	74
4.5 简单分页	75
4.6 简单分段	76
4.7 段页式	76
4.8 段表和页表	77
4.8.1 联想存储器	77
4.8.2 逆页表	78
4.9 对换	78
4.10 覆盖	78

第 5 章 虚拟存储器	97	7.1.3 模块寄存器	146
5.1 请求分页	97	7.1.4 忙等待 I/O	146
5.1.1 访问局部性	97	7.1.5 轮询 I/O	146
5.1.2 页面锁定	98	7.1.6 中断 I/O	146
5.1.3 页面大小	99	7.1.7 直接存储器访问	147
5.1.4 页面替换算法	99	7.2 软件组织	147
5.1.5 算法性能	100	7.2.1 网络 I/O	148
5.1.6 分配策略	100	7.2.2 逻辑 I/O	148
5.1.7 工作集	101	7.2.3 缓冲	149
5.1.8 预先分配页	101	7.2.4 高速缓存机制	149
5.2 分段	101	7.2.5 设备驱动程序	150
第 6 章 文件系统管理	119	7.3 设备	150
6.1 目录和名字	119	7.3.1 图形	151
6.1.1 分区	120	7.3.2 基于文本的显示	152
6.1.2 每一个进程的根目录	120	7.3.3 存储磁盘	152
6.1.3 目录结构	120	7.3.4 硬盘的性能	154
6.1.4 目录项	122	7.3.5 硬盘调度	154
6.2 文件系统对象的类型	123	7.3.6 格式化	155
6.3 文件系统功能	123	7.3.7 RAID	156
6.4 信息类型	123	7.3.8 RAM 磁盘	156
6.5 文件系统体系结构	124	第 8 章 安全	169
6.5.1 访问方法	125	8.1 鉴别	169
6.5.2 访问控制	125	8.1.1 口令	169
6.5.3 文件锁定	126	8.1.2 物理鉴别	170
6.5.4 分块	126	8.2 阻止	170
6.5.5 分配	126	8.3 检测	171
6.5.6 空闲空间	129	8.4 校正	171
第 7 章 设备管理	143	8.5 身份识别	171
7.1 硬件 I/O 组织	143	8.6 威胁的种类	171
7.1.1 I/O 控制	144	8.7 程序威胁	172
7.1.2 端口和存储器映射 I/O	145	参考文献	177

计算机系统提供收集数据、执行计算、存储信息以及与其他计算机系统通信和产生输出报告的能力。这些能力有的用硬件实现,有的用软件实现。操作系统是操纵硬件并为程序执行建立一个更实用的平台的软件。操作系统管理着硬件资源、提供存取这些资源的服务并创建文件、目录和进程等较高层次的抽象概念。

一个典型的计算机系统包括5个主要部分:硬件、操作系统、系统程序、应用程序和用户(图1-1)。硬件包括存储器、中央处理单元(CPU)和输入输出(I/O)设备,完成所有实际的工作。操作系统为程序提供了一系列的服务,用户不直接与操作系统打交道,而是通过程序使用操作系统。系统程序是操作系统提供的一组实用程序,它为用户提供基本的服务。例如图形用户接口(GUI)的窗口管理器,命令解释器,重命名、复制、删除文件的程序都是系统程序。应用程序为计算机提供了用户所需的功能。例如税收准备软件、财务计划软件、文字处理软件、电子制表软件等都是应用程序。

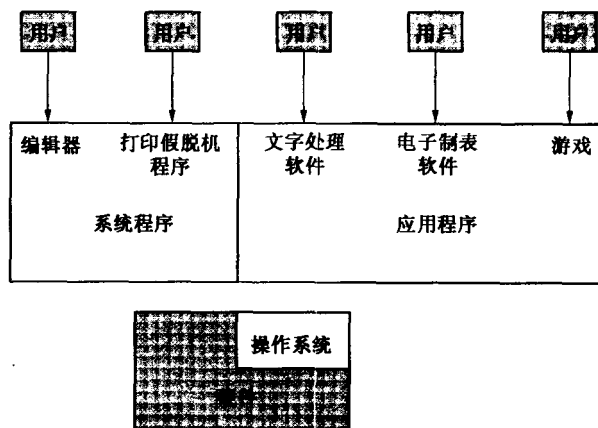


图 1-1 计算机系统组成

1.1 机器硬件

CPU 是计算机系统的大脑和心脏。它包括若干专用寄存器、一个算术逻辑单元(ALU)以及解码与执行指令所必需的控制逻辑(图 1-2)。存储器和 I/O 设备通过通信总线与 CPU 连接。CPU 的操作由 CPU 从存储器中取出的指令来控制,而 I/O 设备由 CPU 控制。

我们可以用简单的循环来描述 CPU 的操作,每次通过这个循环就有一个指令被执行。指

令执行的基本处理是固定不变的。

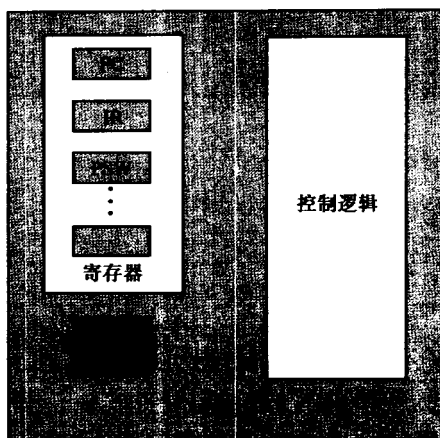


图 1-2 CPU 的组成

- 从被称为程序计数器的专用寄存器所指定的存储器地址中取出一条指令。所有将要执行的指令都从主存储器中取得。
- 指令取出后放在被称为指令寄存器的专用寄存器中。
- 程序计数器是递增的,所以它指向将要执行的下一条指令。
- 把指令解码从而决定将要执行的动作。执行的动作由指令 **opcode**(操作码)的位确定。机器的体系结构定义了哪些二进制位包含操作码。
- 根据被执行操作,从存储器中获得一个或多个操作数。
- 执行由操作码指定的操作。有以下 5 类基本操作:
 - 1) **移动**:将一个值从一个地址移动到另一个地址。这里所说的地址可以是寄存器地址,也可以是存储器地址。
 - 2) **计算**:把一个或多个操作数的值送入 ALU 并进行计算。
 - 3) **条件转移**:如果转移条件为真,则重置程序计数器使其指向转移地址。对于无条件转移,转移条件永远为真。
 - 4) **过程调用**:保存程序计数器的当前值,然后重置程序计数器使其指向过程的起始位置。在过程的末尾,指明保存的程序计数器的转移指令,使程序可以回到当前的执行点。保存的程序计数器存储在寄存器、存储器或堆栈中。
 - 5) **输入/输出**:传送 CPU 和 I/O 设备之间有关输入/输出操作的信息。
- 如果需要,会向主存储器中写回一个值。

1.1.1 陷阱和中断

陷阱和中断是干扰 CPU 执行的指令的正常顺序的事件。陷阱是 CPU 检测到的一个非正常状态,这个状态通常表示出错。陷阱的例子有:除数为零、试图访问不存在的或程序无权访问的存储器地址、执行带有未定义操作码的指令、试图访问不存在的 I/O 设备。

中断是外部设备(典型的是 I/O 设备)发给 CPU 的信号。它是页面调度程序的 CPU 等价

物,一个请求 CPU 中断现行活动去处理中断设备需求的信号。CPU 仅在完成一条指令处理之后以及在获取下一条指令之前检查中断。

CPU 通过保存程序计数器的当前值并重置程序计数器的值为新地址来响应陷阱和中断。这将使 CPU 执行完陷阱和中断的过程后能返回陷阱和中断发生点。CPU 跳转的地址由硬件结构决定。在一些机器上,有惟一的地址与每个陷阱和中断相关。更一般的情况是,硬件结构把存储器中的某一地址定义为一个**中断向量**的位置。每个陷阱和中断与一个向量索引相关联。陷阱和中断的索引所指向的向量表示的存储器位置上的内容决定了转移地址。

当陷阱或中断发生时,其他的状态信息也将被保存起来。在许多机器上,这些信息保存在专用的**程序状态字(PSW)**寄存器中。

中断可以与一个**硬件优先级**关联起来。CPU 也可以与优先级联系起来。只有中断的优先级高于 CPU 的优先级时中断才被执行。低优先级的中断要等到 CPU 降低了自己的优先级时才会得到响应。CPU 的优先级保存在 PSW 中,并且可以通过改变 PSW 中相应的二进制位来重置优先级。

陷阱和中断的区别之一是,陷阱是同步的,而中断是异步的。如果给定相同的机器状态和输入数据,每次程序运行时陷阱就会发生在程序执行的同一点上,而中断的发生依赖于中断设备和 CPU 之间的相对时序。由于受中断时序影响的错误不容易重复,因此中断给调试过程带来了难度。

1.1.2 多态执行

为了给操作系统提供一些应用程序未被授予的特权,硬件必须支持执行的多态性。一般来讲,执行支持两种状态:**核心(管理)态**和**用户态**。在 PSW 中用一个二进制位来记录系统执行态。当在用户态下试图执行某些动作时,就会导致陷阱。这些动作通常限制为只能由操作系统来完成,例如某些指令(**特权指令**)的执行、访问某些寄存器和访问 I/O 设备。

系统可以通过三种方式从用户态进入核心态。一种是**管理程序调用(SVC)**或**系统调用**,它与过程调用相似,不同的是,它把系统状态设置为核心态。与过程调用指令不同,管理程序调用指令并不提供转移地址。指令的操作数是一个用做向量索引的数值,这个向量类似于中断向量。转移地址由管理程序调用操作数指向的存储器单元的内容决定。该向量位于操作系统控制的存储器中,所以进入核心态的切换与跳转到操作系统入口点相一致。

陷阱和中断是另外两种进入核心态的切换机制。类似于管理程序调用,这样实现的向核心态的切换与跳转到操作系统内核入口点相一致。应用程序不能使系统进入核心态,它们只能执行自己的代码。

在 Unix 系统中,**超级用户**享有专门的访问特权。超级用户可以读写任何文件并可以终止任何进程。不能混淆超级用户与核心态执行。虽然作为超级用户运行的应用程序被内核赋予了特殊的访问权限,但这些程序并不在核心态下运行,它们必须使用管理程序调用请求操作系统服务。

1.2 操作系统结构

操作系统为应用程序提供了一台虚拟机。操作系统实现的管理程序调用扩充了原始硬件提供的指令集的功能。管理程序调用支持新的抽象概念,例如进程和文件系统。

除了提供系统调用接口外,操作系统还负责管理基础硬件资源。应用程序不能访问 I/O 设备或执行特权指令,操作系统代表应用程序执行这些任务。在执行这些任务时,操作系统力图有效地利用可用资源并保护共享这些资源的应用程序的完整性。

操作系统执行的任务可分为以下 4 个方面:

- **进程管理:**进程是一个正在执行的程序。和进程相关的是代码、数据、分配给它的一组资源和执行其代码的一个或多个执行“流”。操作系统使用管理程序调用管理进程,并管理进程所需资源的分配。如果同时存在多进程,操作系统必须为每个进程提供它运行所需的适当的虚拟环境。
- **存储器管理:**存储器至少应被一个应用程序和操作系统共享。在更复杂的系统中,存储器可被多个进程共享。操作系统必须管理进程的存储器分配,并控制存储器管理硬件,存储器管理硬件决定一个进程可以访问哪些存储器单元。
- **文件系统管理:**计算机处理信息。信息必须被传输、处理和存储。文件系统对象是存储和传输信息集合的抽象实体。文件系统是经过组织的文件系统对象的集合。操作系统必须提供操纵这些对象的原语。
- **设备管理:**计算机通过输入输出设备交换信息。进程通过操作系统的管理程序调用访问这些设备。操作系统力图通过某种方式管理输入输出设备,使那些需要输入输出设备的进程有效地共享这些设备。

1.2.1 操作系统的类型

只有少数用户直接使用过早期的计算机系统。早期计算机的输入(无论是程序还是数据)都存放在输入介质上,典型的输入介质是纸带或穿孔卡片。用户将他们的输入交给操作员,然后几分钟(在幸运的情况下)或几小时后来取回他们的程序或作业的输出。操作员把相似的作业组织成一批,并让计算机运行这一批作业。每个作业完全控制着计算机直到它执行完毕。

批处理方式的操作系统就是以上述方式管理计算机的运行。由于批处理操作系统不需要处理多进程共享资源的复杂情况,所以操作系统提供了极少的功能。

在**多道程序批处理系统**中,作业被读入存放在磁盘上的作业池中。当一个作业因为等待 I/O 操作的完成而不能执行时,其他作业可以运行。并发执行的进程对计算机资源的共享大大增加了操作系统的复杂性。

分时操作系统允许用户和进程之间进行交互。在批处理系统中,当程序输入时提供所有的数据。对于获取薪水信息和打印每周薪水支票的程序来说,批处理系统是很合适的,但是对于必须和用户交互的程序来说,操作系统需要提供一个环境,使得程序对用户的输入在合理的时间内做出响应。操作系统不仅必须在不同的进程间分享资源,而且必须创造一种假象好像多个进程同时运行。这是通过在所有活动进程间迅速地轮流执行来实现的。

计算机越来越多地不再作为独立个体而是作为计算机网络的一部分存在。这对操作系统底层的影响是非常有限的。I/O 设备允许计算机在网络上通信。然而,网络通信涉及复杂的协议,而且为了网络的可靠性,在网络操作系统中建立了对这些协议的支持。当今通用的操作系统都是网络分时系统,这种操作系统也是本书的重点。这样的操作系统也可能支持某种形式的批处理。

实时操作系统用来支持在限定的时间内执行任务。通常,用户希望计算机系统尽快执行

他们的程序,但没有提供确切的时间安排。在实时系统中,处理任务的正确性依赖于处理发生的时间。例如,感知核反应堆冷却剂损失的实时系统需要在一秒以内启动备份系统。操作系统必须保证任务在指定的时间限制内执行。实时系统大多限于专门的应用,如工业控制系统、武器系统和计算机控制的产品使用。

分时系统中所需要的特性,如资源共享和操作系统对 I/O 设备的管理,在实时系统中成为一种不利因素。完成任务的最小化延时阻碍了资源共享,但支持了对硬件底层的访问。出于这个原因,通用操作系统的设计不同于实时操作系统的设计。本书不涉及实时操作系统的设计。

另外一种特殊形式的操作系统是分布式操作系统。在网络操作系统中,网络中每台机器的资源由机器自身的操作系统管理。操作系统使机器间很容易进行通信。在分布式操作系统中,所有机器的操作系统一起管理共同的网络资源,而每台网络计算机或节点提供的网络资源由单一共用的分布式操作系统管理。本节也不讨论分布式操作系统。

1.2.2 操作系统的内核

操作系统内核是硬件在核心态执行时被执行的代码。内核不应当被认为是一个程序,更确切地说它是一个子例程库。在陷阱、中断或管理程序调用后执行库中一个或多个过程。在某一点上,硬件被重置为用户态,并且控制返回到用户程序(有可能不是进入内核时执行的同一个程序)。

在某种程度上,内核就是操作系统。操作系统管理的 4 个方面都是在内核中实现的,即用管理程序调用触发相关函数的执行。另外,设备中断导致设备管理函数的执行,并且存储器管理硬件产生的陷阱激活存储器管理代码。每次进入内核时,在把控制返回到用户程序之前,进程调度程序可以作为最后一个步骤被调用。

尽管以巨大的单片内核为特征的系统很普遍,但也有人对微内核操作系统感兴趣。在微内核系统中,内核包括极少的功能。内核提供了诸如存储器管理硬件的控制、进程间的通信机制、中断和陷阱处理等必不可少的基本功能。服务器进程提供了诸如文件系统服务等高级功能。寻找文件系统服务的程序使用进程间通信的系统调用向文件系统服务器进程发送请求。

使用服务器进程增强了灵活性。甚至不用重启系统,新的服务器就能被激活。微内核很小,这样就简化了新系统的实现和移植。微内核非常适合分布式系统,因为当使用进程间通信的设备时,无论进程在本地系统还是在远程系统中都无关紧要。

如果操作系统的定义扩充到包括用户级进程提供的某些服务时,那么系统程序和操作系统的区别就变得模糊了,例如用于打印的假脱机系统或窗口管理器,这样的服务就很难归类。有一种观点认为它们是操作系统设备管理能力的一部分。本书对通常在单片内核中可以见到的设计特性进行了研究。尽管这些特性的一部分可能在内核之外实现,但设计原则不会改变。打印假脱机系统通常被认为是系统程序的高级服务,不在本书讨论范围之内。

1.2.3 引导进程

在计算机系统刚通电时,执行只读存储器(ROM)中的程序。执行完诊断检查后,执行零级引导程序。该程序检查引导设备是一个还是多个。在 PC 机上,引导设备可以是软盘、CD-ROM 或硬盘。一旦找到引导设备,零级引导程序把引导设备首扇区读到主存。引导设备的

首扇区称为引导扇区,它应该包含一级引导程序。一旦首扇区被读入存储器,零级引导程序转移到一级引导程序。

引导扇区不能保证一定包含引导程序。如果引导扇区没有被初始化,则未定义的值将被复制到存储器中。控制转移到引导进程的下一级之前,当前引导程序将检查引导扇区末尾的一个或多个字节,在其中寻找一个“幻数”位模式,这个模式用来说明该扇区包含了有效的引导程序。

在某些系统中,一级引导程序接着读入操作系统。在另一些操作系统中,必须把一系列的引导陷阱程序复制到存储器并执行。无论在哪种情况下,操作系统最终被复制到存储器中并且最终引导程序会转移到操作系统初始化入口点。

在从硬盘引导的 PC 上,引导扇区中包含一个分区表。硬盘最多可以被分成 4 个主分区。分区表包含了每个分区的起始和结束位置,它把某一个分区定义为活动分区。一级引导程序读分区表并把活动分区的首扇区复制到存储器中。活动分区的首扇区包含了一个二级引导程序,这个引导程序用来和这个分区中的操作系统协同工作。活动分区的首扇区也能从其他分区装载引导程序或操作系统。在最简单情况下,二级引导程序把分区中的操作系统装入存储器中。

在引导进程即将结束时,操作系统已经装入内存并且引导程序也转移到操作系统,机器在核心态执行。任何在操作系统装载时没被设置的数据结构(如中断向量)必须初始化,系统寄存器和设备也要按需要初始化。

一旦操作系统为进程的服务做好了准备,操作系统就可以创建提供操作系统服务的进程,并且创建一个或多个进程执行初始化系统程序。在 DOS 这样的简单系统上,操作系统可能只产生一个进程去执行一个命令解释程序。在一个比较复杂的系统上,可以提供系统程序 init 来创建在配置数据库中指定的另外一些进程。无论在哪种情况下,一旦系统进入用户态,并且控制切换到用户态进程,操作系统就完成了它的引导任务,系统就开始运行了。

1.3 本书其余部分概要

第 2 章“进程管理”和第 3 章“进程间的通信和同步”研究与进程相关的设计问题。第 2 章包括了进程特征、进程调度和用来控制进程的管理程序调用。第 3 章讲述了进程合作运行时产生的问题,主要讨论了同步和通信原语,这些原语是用来处理死锁的。

第 4 章“存储器管理”和第 5 章“虚拟存储器”讨论进程的存储器分配。主要讲解实现不同存储器管理方案时所要求的硬件支持以及操作系统的设计问题。第 5 章的重点讨论了那些使计算机运行时仿佛有比实际存储器空间更大的存储器管理方案。

第 6 章“文件系统管理”既讨论了整个文件系统的结构也讨论了文件系统内单个文件的结构。

第 7 章“设备管理”讨论了重要输入输出设备的物理特征和控制这些设备时对操作系统的要求。磁盘是讨论的重点,还讨论了磁盘调度和 RAID 系统。

第 8 章“安全”是本书的最后一章。在许多方面,与其说是操作系统设计问题,不如说是管理和策略问题。然而,至少操作系统设计和安全是相关的。第 8 章简要讨论了安全措施和这些安全措施所处理的威胁类型。

习题与解答

1.1 操作系统的两个主要功能是什么？

解答：

操作系统的两个主要功能是管理系统资源和为应用程序提供一组原语,这些原语用来提供高级服务。

1.2 当没有程序运行时 CPU 干什么？

解答：

一直有程序在运行(只要计算机开机)。取指令、解码、执行指令这个循环从不停止。当没有用户程序运行时,操作系统将执行不做任何事情的循环(称之为忙等待循环或空闲循环)直到有中断发生。

1.3 陷阱、中断、管理程序调用和子例程调用的共同特征是什么？

解答：

陷阱、中断、管理程序调用和子例程调用都保存程序计数器的当前值并转移到存储器中的一个新地址。

1.4 哪些特征是陷阱、中断和管理程序调用都有而子例程调用没有的？

解答：

陷阱、中断和管理程序调用都使机器转移到核心态,而子例程调用不改变执行态。

1.5 下面哪些指令是特权指令(只能在核心态执行)？

(a) 改变存储器管理寄存器

(b) 写程序计数器

(c) 读日历钟

(d) 设置日历钟

(e) 改变处理器优先级

解答：

(a) 是。改变存储器管理寄存器将会允许进程访问它无权访问的存储器单元。

(b) 不是。写程序计数器和执行无条件转移是一样的。

(c) 不是。尽管直接访问设备通常是不明智的,但读系统时钟没有什么害处。

(d) 是。改变时钟会中断已被调度的事件,因此用户进程通常不具有这项权限。

(e) 是。改变处理器的优先级会引起中断丢失。

1.6 操作系统可以实现存储器 I/O 设备。设备的 I/O 操作会使相应的存储器单元被读写。那么提供这种设备的缺点是什么？这样的设备应该可以被普通用户访问还是仅限于系统管理员访问？

解答：

如果普通用户可以访问存储器设备,那么他们就能读或写存储器的任何位置,这样就破坏了操作系统的保护机制。如果操作系统让用户程序具有在核心态执行指令的能力,那么操作系统就可能被覆盖。把这种能力提供给某个用户,甚至系统管理员,都将会威胁操作系统的完整性。

1.7 将下列应用程序分为交互性和面向批处理两类：

- (a) 字处理
- (b) 按月生成的银行报表
- (c) 计算精确到百万分位的圆周率(π)

解答:

- (a) 交互的
- (b) 面向批处理的
- (c) 面向批处理的

这个作业几乎达到了 CPU 的极限范围。在没有多道程序设计的批处理系统中运行这个程序, CPU 的利用率几乎能达到 100%。

- 1.8 当电源刚接通时, 计算机为什么必须从核心态引导?

解答:

当电源刚接通时, 存储器的内容没有被定义。ROM 中的引导程序必须从引导设备中读代码并装入到内存中。为了执行这一输入操作, 硬件必须在核心态。由于操作系统未被装载, 所以在用户态就没有 ROM 引导程序可以使用的管理程序调用。

- 1.9 假定引导陷阱幻数位模式为 2 字节、扇区大小为 512 字节, 那么在硬盘开始的一级引导程序的最大尺寸是多少?

解答:

一级引导程序必须适合初始扇区的大小。如果扇区尺寸是 512 字节、引导幻数位模式为 2 字节, 则引导程序最大尺寸是 510 字节(在 PC 系统中, 首扇区中通常存储一个分区表, 所以引导程序最大为 446 字节)。

补充习题

- 1.10 请列举出明显是为了支持操作系统而设计的硬件特征。
- 1.11 哪些特征是陷阱、管理程序调用和子例程调用都有而中断没有的?
- 1.12 哪些特征是子例程调用和管理程序调用都有而陷阱和中断没有的?
- 1.13 为什么不允许用户程序向包含中断向量的存储器单元写入数据?
- 1.14 下面哪些指令是特权指令(只能在核心态执行)?
- (a) 设置模式为核心态
 - (b) 重新引导
 - (c) 读程序状态字(PSW)
 - (d) 禁止中断
 - (e) 写指令寄存器
- 1.15 将下列应用程序分为交互性和面向批处理两类。
- (a) 生成个人的电子邮箱注册表单
 - (b) 生成雇员的月工资报表
 - (c) 飞行模拟器
- 1.16 操作系统和复制操作系统到存储器的引导程序必须位于同一磁盘吗? 为什么?

补充习题解答

- 1.10 (1) 核心态/用户态操作
(2) 特权指令
(3) 管理程序调用指令
- 1.11 陷阱、管理程序调用和子例程调用都是同步的。如果程序在每次运行时使用相同的输入数据,则都会在相同的点产生相同的陷阱、管理程序调用和子例程调用。而中断是异步的,程序在每次运行时使用相同的输入数据,不一定在相同的点产生中断。
- 1.12 子例程调用和管理程序调用都是可以被用户程序执行的指令,而陷阱和中断不能,它们是由硬件触发的事件。
- 1.13 如果程序可以向中断向量写入数据,那么中断发生时的转移地址就可以改变。至少,程序能够使系统重定向,从而中断就不能被正确地处理。最坏的情况下,中断能使应用程序控制的存储单元地址发生转移,这样应用程序就会执行核心态下指令。
- 1.14 (a) 是
(b) 是
(c) 不是
(d) 是
(e) 不是
- 1.15 (a) 交互的
(b) 面向批处理的
(c) 交互的
- 1.16 不必。引导程序必须能够读操作系统。无论操作系统和引导程序是否在同一磁盘上都是无关紧要的。把从硬盘中读出操作系统的引导程序安装在软盘中,可以在硬盘引导扇区被破坏时起到有效的保护作用。当操作系统过大不能放在一张软盘中时,分开存放引导程序和操作系统就特别有用。

计算机系统中的所有活动都是以计算机程序指令的执行为中心。进程是指一组机器指令的执行,指令本身是静态的实体,它们存在于纸上、文件中甚至作为一种概念上的想法存在于人的大脑中。进程是处在活动状态的指令集,也可以说是一种执行代码所描述活动的动态实体。

与进程相关联的是它的代码、数据、分配给它的可操作的资源和一个或多个执行线程。线程是进程代码的执行流,它有自己的程序计数器、系统寄存器和堆栈。

操作系统可分为三种:单进程单线程、多进程单线程、多进程多线程。最简单的一种操作系统是单进程单线程,每次只能有一个进程执行。一旦进程得到 CPU,则直到该进程运行完成,其他进程才能执行。DOS 就是单进程单线程操作系统的例子。

多处理系统更加实用。多处理系统允许用户执行多个并发的任务,更有效地使用了计算机的资源。CPU 的执行速度比 I/O 设备快得多,在完成一次 I/O 请求的时间内可以执行数百万条指令。在一个单进程的环境中,当等待完成 I/O 请求时,CPU 保持空闲。在一个典型的程序中,CPU 有 50% 的时间空闲。多处理系统允许 CPU 被多个进程共享,系统交叉执行各个进程,使各个进程有机会周期性地使用 CPU。如果一个进程因等待 I/O 请求的完成而空闲,另一个不等待 I/O 的进程可以得到 CPU。如果有足够的存储器并且协调得当,CPU 利用率可以接近 100%。

在 Unix 这样的多进程单线程系统中,进程是进行资源分配和 CPU 调度的惟一实体。在多处理多线程系统中,资源分配给进程,但 CPU 按线程调度。Windows 2000、Mach 和 Solaris 都是多处理多线程操作系统的例子。为了区分单线程和多线程,任务有时用于表示多线程实体,而重进程 (heavyweight process) 表示只有一个线程的任务,线程有时被称为轻进程 (lightweight process)。

在早期的批处理系统中,作业用来代表一个或多个按顺序执行的程序,并被系统视为单独的实体。正在执行的作业称之为进程,这两个术语有时交替使用。

2.1 进程调度

在多处理系统中,进程调度用来决定哪个进程将得到 CPU 的控制。调度分为三个阶段:长期、中期、短期。长期调度或者说是作业调度决定哪些作业或进程可以竞争系统资源。典型地,一旦作业调度程序使一个作业为活动的,那么这个作业会一直保持活动状态直到作业完成为止。作业调度程序的主要目标是给中期调度程序提供适当数量的作业。如果作业数量太