

Mc
Graw
Hill Education

学习指导与习题解答

Schaum's Outline of Operating Systems



操作系统 学习指导与习题解答

J. Archer Harris 著

王艳平等译

清华大学出版社



学习指导与习题解答

Schaum's Outline of Operating Systems



操作系统
学习指导与习题解答

清华大学出版社
北京

J. Archer Harris
Schaum's Outline of Operating Systems
EISBN: 978-0-07-136435-5

Copyright © 2010 The McGraw-Hill Companies, Inc.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education (Asia) and Tsinghua University Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2010 by McGraw-Hill Education (Asia), a division of the Singapore Branch of The McGraw-Hill Companies, Inc. and Tsinghua University Press.

版权所有。未经出版人事先书面许可,对本出版物的任何部分不得以任何方式或途径复制或传播,包括但不限于复印、录制、录音,或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。

版权© 2010 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社所有。

北京市版权局著作权合同登记号 图字: 01-2009-5146 号

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。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

操作系统学习指导与习题解答/(美)哈里斯(Harris, J. A.)著;王艳平等译. —北京:清华大学出版社, 2010. 8

书名原文: Schaum's Outline of Operating Systems

ISBN 978-7-302-22880-6

I. ①操… II. ①哈… ②王… III. ①操作系统—高等学校—教学参考资料
IV. ①TP316

中国版本图书馆 CIP 数据核字(2010)第 099222 号

责任编辑: 龙啟铭

责任校对: 李建庄

责任印制: 孟凡玉

出版发行: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

投稿与读者服务: 010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京市清华园胶印厂

经 销: 全国新华书店

开 本: 185×230 印 张: 13

版 次: 2010 年 8 月第 1 版

印 数: 1~3000

定 价: 29.00 元

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

字 数: 278 千字

印 次: 2010 年 8 月第 1 次印刷

产品编号: 034572-01

作者简介

J. ARCHER HARRIS 博士于 1973 年在丹尼森(Denison)大学获得数学学士学位,1975 年和 1978 年在纽约州立石溪大学获得计算机科学硕士学位和博士学位。现在,他是詹姆斯麦迪逊大学计算机科学系的副教授,自 1985 年一直教学至今。他所关注的方向有操作系统、Unix、网络以及计算机科学教育。Harris 博士还在南伊利诺伊大学任教,并为许多公司做咨询活动,比如 Tandon 公司、U. S. Immigration and Naturalization Service、BDS 公司、TransAmerica Telemarketing 和 Shenandoah Electronic Intelligence。Harris 博士开发的软件包括 NAMS(Network Account Management System)、hed 编辑器和 sim 微程序模拟器。

译者序

操作系统是计算机系统的资源管理者,其主要作用就是 CPU 管理、存储管理、设备管理和文件管理,对各种资源进行分配,改善资源的共享和利用程度,最大限度地发挥计算机系统的工作效率,提高计算机系统处理工作的能力。

本书介绍软件工程的基础知识,内容实用而丰富,主要内容有:

- 第 1 章概括说明计算机硬件以及操作系统的结构。
- 第 2 章和第 3 章介绍与进程相关的操作系统设计问题。
- 第 4 章和第 5 章讨论进程的内存分配,主要讲解实现不同存储管理方案所要求的硬件支持以及操作系统的设计问题。
- 第 6 章讨论整个文件系统的结构以及文件系统内单个文件的结构。
- 第 7 章讲解重要输入输出设备的物理特征,以及控制这些设备时对操作系统的要求。
- 第 8 章说明安全措施及其所处理的威胁类型。

本书特点是,提纲挈领地说明各个重要知识点,用丰富的例子,让读者全面透彻地理解和掌握概念。每章最后都有大量习题和补充题,并给出详尽答案,使读者能举一反三,灵活运用理论知识解决实际问题,并检验知识的掌握程度。

本书主要由王艳平老师翻译,参加翻译工作的人员还有李志云、李晓春、陈安华、侯佳宜、许伟、戴文雅、于樊鹏、刘朋、王嘉佳、邓卫、邓凡平、李波、程云建、许晓哲、朱珂、韦笑、孙宏、李腾、陈磊、魏宇、周京平、徐冬、冯哲、李绯、李强、赵东辉等人。

前言

笔者第一次使用的操作系统是在 DEC PDP-11 计算机上运行的 Unix 第 6 版。后来,我阅读了 Unix 的源代码,认识到,操作系统是如此富有魅力,它精美而复杂。对于程序员来说,操作系统是最大的挑战,它涵盖了从底层设备管理到并发控制,再到面向对象设计的各个方面。

本书讲解现代操作系统的设计原理,适用于这样的读者:希望更多地了解操作系统,或对某个操作系统感兴趣,希望更广泛地了解操作系统行为。本书每章都有课程内容的简介和大量题解。本书适于与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,不介绍具体操作系统的详细特性。虽然书中的例子均来自各个具体操作系统,如 DOS、Windows,还有 Unix,但其中的内容适用于普遍的设计原理,不包括比较专门的议题,如实时系统或分布式系统。

感谢在编写本书过程中帮助过我的所有 McGraw-Hill 的工作人员,感谢詹姆斯·麦迪逊(Jalnes Madison)大学的学生和同事,感谢他们的理解和支持。特别感谢 Ramo'n Mata Toledo 博士为本书所做的所有工作。

最后,感谢我的妻子也是我的同事 Mancy Harris,感谢她的爱,她对本书的审查,价值无法估量。

虽然我希望本书内容准确无误,但错误疏漏在所难免。如有意见和建议,请发邮件到 schaumos@mailgate.cs.jmu.edu。本书更新内容可查阅:<http://w3.cs.jmu.edu/arch/schol/schaumos>。



| | |
|-------------------------|----|
| 第 1 章 概述 | 1 |
| 1.1 计算机硬件 | 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 | 4 |
| 1.2.1 操作系统类型 | 4 |
| 1.2.2 操作系统内核 | 5 |
| 1.2.3 引导进程 | 6 |
| 1.3 本书概要 | 7 |
| 本章习题与答案 | 7 |
| 补充题 | 9 |
| 补充题答案 | 9 |
| | |
| 第 2 章 进程管理 | 11 |
| 2.1 进程调度 | 11 |
| 2.2 进程状态 | 12 |
| 2.3 调度条件 | 14 |
| 2.4 调度算法 | 15 |
| 2.4.1 先来先服务算法 | 15 |
| 2.4.2 最短作业优先算法 | 16 |
| 2.4.3 最短剩余时间算法 | 16 |
| 2.4.4 循环算法 | 16 |
| 2.4.5 优先级 | 17 |
| 2.4.6 多级反馈队列算法 | 17 |
| 2.5 调度算法性能 | 18 |
| 2.6 进程的属性 | 19 |
| 2.6.1 运行态和调度 | 20 |

| | | |
|------------|------------------------|-----------|
| 2.6.2 | 存储管理 | 20 |
| 2.6.3 | 硬件状态 | 20 |
| 2.6.4 | 信号机制 | 20 |
| 2.6.5 | 访问控制 | 20 |
| 2.6.6 | 输入和输出 | 20 |
| 2.6.7 | 其他 | 21 |
| 2.7 | 进程管理程序调用 | 21 |
| | 本章习题与答案 | 22 |
| | 补充题 | 31 |
| | 补充题答案 | 34 |
| 第3章 | 进程间通信与同步 | 37 |
| 3.1 | 进程间通信 | 37 |
| 3.2 | 进程同步 | 40 |
| 3.2.1 | 临界区 | 40 |
| 3.2.2 | 禁止中断 | 41 |
| 3.2.3 | TestAndSet 指令 | 42 |
| 3.2.4 | Swap 指令 | 43 |
| 3.2.5 | Wait 和 Signal 原语 | 44 |
| 3.2.6 | 信号量 | 45 |
| 3.2.7 | Dekker 算法 | 46 |
| 3.2.8 | Peterson 算法 | 47 |
| 3.2.9 | Bakery 算法 | 47 |
| 3.2.10 | 监控程序 | 48 |
| 3.3 | 死锁 | 49 |
| 3.3.1 | 预防死锁 | 50 |
| 3.3.2 | 避免死锁 | 51 |
| 3.3.3 | 死锁检测 | 52 |
| 3.3.4 | 死锁恢复 | 55 |
| 3.3.5 | 鸵鸟算法 | 56 |
| | 本章习题与答案 | 56 |
| | 补充题 | 65 |
| | 补充题答案 | 68 |

| | |
|-------------------------|-----|
| 第 4 章 存储管理 | 77 |
| 4.1 单一绝对分区 | 77 |
| 4.2 单一重定位分区 | 78 |
| 4.3 多道程序 | 78 |
| 4.4 多重分区 | 78 |
| 4.4.1 多重固定分区 | 78 |
| 4.4.2 多重可变分区 | 79 |
| 4.4.3 伙伴系统 | 80 |
| 4.5 简单分页 | 81 |
| 4.6 简单分段 | 82 |
| 4.7 段页式 | 83 |
| 4.8 页表和段表 | 84 |
| 4.8.1 相联内存 | 84 |
| 4.8.2 逆页表 | 84 |
| 4.9 交换 | 85 |
| 4.10 覆盖 | 85 |
| 本章习题与答案 | 86 |
| 补充题 | 98 |
| 补充题答案 | 101 |
| | |
| 第 5 章 虚拟内存 | 106 |
| 5.1 请求分页 | 106 |
| 5.1.1 访问局部性 | 106 |
| 5.1.2 页面锁定 | 108 |
| 5.1.3 页面大小 | 108 |
| 5.1.4 页面替换算法 | 108 |
| 5.1.5 算法性能 | 110 |
| 5.1.6 分配策略 | 110 |
| 5.1.7 工作集 | 110 |
| 5.1.8 预分页 | 111 |
| 5.2 分段 | 111 |
| 本章习题与答案 | 111 |
| 补充题 | 123 |
| 补充题答案 | 127 |

| | |
|-------------------------|-----|
| 第 6 章 文件管理 | 131 |
| 6.1 目录和名称 | 131 |
| 6.1.1 分区 | 132 |
| 6.1.2 每进程的根目录 | 132 |
| 6.1.3 目录结构 | 132 |
| 6.1.4 目录项 | 134 |
| 6.2 文件系统对象类型 | 134 |
| 6.3 文件系统功能 | 135 |
| 6.4 信息类型 | 135 |
| 6.5 文件系统结构 | 136 |
| 6.5.1 访问方法 | 136 |
| 6.5.2 访问控制 | 137 |
| 6.5.3 文件锁定 | 138 |
| 6.5.4 分块 | 138 |
| 6.5.5 分配 | 139 |
| 6.5.6 空闲空间 | 140 |
| 本章习题与答案 | 142 |
| 补充题 | 152 |
| 补充题答案 | 154 |
| | |
| 第 7 章 设备管理 | 157 |
| 7.1 硬件 I/O 组织 | 157 |
| 7.1.1 I/O 控制 | 158 |
| 7.1.2 端口和内存映像 I/O | 159 |
| 7.1.3 模块寄存器 | 160 |
| 7.1.4 忙等待 I/O | 160 |
| 7.1.5 轮询 I/O | 160 |
| 7.1.6 中断 I/O | 161 |
| 7.1.7 直接内存访问 | 161 |
| 7.2 软件组织 | 161 |
| 7.2.1 网络 I/O | 162 |
| 7.2.2 逻辑 I/O | 162 |
| 7.2.3 缓冲 | 163 |
| 7.2.4 高速缓存 | 164 |

| | |
|-----------------------|------------|
| 7.2.5 设备驱动程序..... | 164 |
| 7.3 设备 | 165 |
| 7.3.1 图形..... | 165 |
| 7.3.2 基于文本的显示..... | 166 |
| 7.3.3 存储磁盘..... | 167 |
| 7.3.4 硬盘性能..... | 168 |
| 7.3.5 硬盘调度..... | 169 |
| 7.3.6 格式化..... | 169 |
| 7.3.7 RAID | 170 |
| 7.3.8 RAM 磁盘 | 172 |
| 本章习题与答案..... | 173 |
| 补充题..... | 179 |
| 补充题答案..... | 181 |
| 第 8 章 安全 | 184 |
| 8.1 鉴别 | 184 |
| 8.1.1 密码..... | 184 |
| 8.1.2 物理鉴别..... | 185 |
| 8.2 预防 | 185 |
| 8.3 检测 | 186 |
| 8.4 修正 | 186 |
| 8.5 身份识别 | 186 |
| 8.6 威胁类别 | 187 |
| 8.7 程序威胁 | 187 |
| 本章习题与答案..... | 188 |
| 补充题..... | 190 |
| 补充题答案..... | 191 |
| 参考文献 | 192 |

第 1 章 概 述

计算机系统具有收集数据、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。有的能力用硬件实现,有的用软件实现。操作系统是一种软件,它操纵硬件,并为程序执行建立更实用的平台。操作系统管理硬件资源,提供访问资源的服务,并具有诸如文件、目录和进程的高级抽象。

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(参见图 1-1)。硬件包括存储器、中央处理单元(Central Processing Unit,CPU)和输入输出(Input and Output,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,为用户提供基本服务。如图形用户界面(Graphical User Interface,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。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。

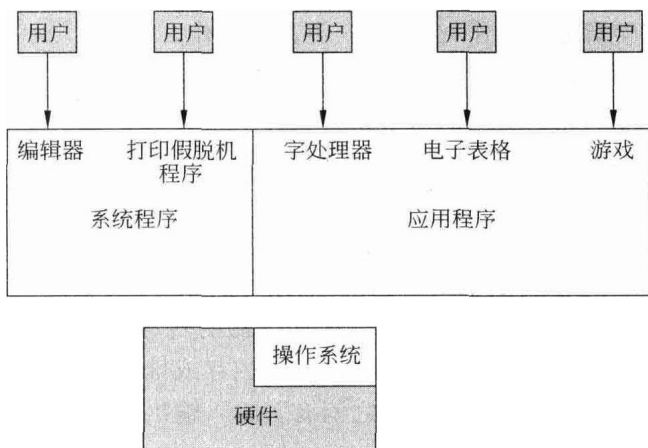


图 1-1 计算机系统组成

1.1 计算机硬件

CPU 是计算机系统的核心。它包括一些专用寄存器、一个算术逻辑单元(Arithmetic Logic Unit,ALU)以及解码与执行指令所必需的控制逻辑(参见图 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进行控制, I/O 设备由 CPU 控制。

下面用简单的循环来描述 CPU 操作, 每通过一次这个循环, 就执行一个指令。指令执行的基本处理是不变的。

(1) 从程序计数器这种专用寄存器指定的内存地址中取出一条指令。所有将要执行的指令都从主存中取得。

(2) 指令放在指令寄存器这种专用寄存器中。

(3) 程序计数器加 1, 指向将要执行的下一条指令。

(4) 指令解码, 以确定将要执行的动作。执行的动作由指令操作码位确定。计算机体系结构确定了哪些二进制位包含操作码。

(5) 根据执行的操作, 从内存中获得一个或多个操作数。

(6) 执行由操作码指定的操作, 有以下 5 类基本操作。

- 移动: 将一个值从一个位置移动到另一个位置。位置可以是寄存器地址或内存地址。
- 计算: 把一个或多个操作数值送入 ALU 并进行计算。
- 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。
- 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, 指定所保存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、内存或堆栈中。
- 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。

(7) 如果需要, 向主存中写回一个值。

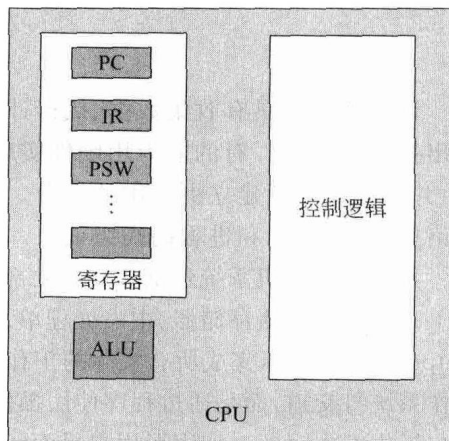


图 1-2 CPU 的组成

1.1.1 陷阱和中断

陷阱和中断是干扰 CPU 执行指令正常顺序的事件。陷阱是 CPU 检测到的非正常状态, 通常表示出错。陷阱的例子有: 除零, 试图访问不存在的内存地址, 或访问程序无权访问的存储器地址, 执行的指令带有未定义操作码, 试图访问不存在的 I/O 设备。

中断是外部设备(通常是 I/O 设备)发给 CPU 的信号。它相当于 CPU 的页面调度程序, 其信号请求 CPU 中断现行活动, 去处理中断设备需求。仅在完成一条指令处理之后及

在获取下一条指令之前,CPU才检查中断。

通过保存程序计数器的当前值,并重置程序计数器的值,CPU为新地址响应陷阱和中断。这样,CPU执行完陷阱和中断的过程中,就能返回陷阱和中断的发生位置。CPU跳转的地址由硬件结构确定。在一些计算机上,与每个陷阱和中断相关的地址唯一。更常见的情况是,硬件结构把内存中的某一地址定义为中断向量的位置。每个陷阱和中断与一个向量索引相关联。陷阱和中断的索引指向的向量所表示的内存位置的内容确定了转移地址。

当发生陷阱或中断时,也将保存其他状态信息。在许多计算机上,这些信息保存在专用的程序状态字(Program Status Word,PSW)寄存器中。

中断可以与硬件优先级关联。CPU也可以与优先级相关。只有中断优先级高于CPU优先级时,才执行中断。要等CPU降低了其优先级时,才会响应低优先级中断。CPU优先级保存在PSW中,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,中断是异步的。如果具有相同的计算机状态,输入相同的数据,那么,每次程序运行时,就会在程序执行的同一点上出现陷阱。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易重复,所以,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未授予应用程序的特权,硬件必须支持执行的多态性。通常,执行支持两种状态:核心(管理)态和用户态。在PSW中,用一个二进制位记录系统执行态。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,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3种方式从用户态进入核心态。一种是管理程序调用(Supervisor Call,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的是,管理程序调用指令并不提供转移地址。指令的操作数是用做向量索引的数值,该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内存单元内容确定。该向量位于操作系统控制的内存中,所以进入核心态的切换与跳转到操作系统入口点一致。

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,超级用户具有特殊访问权。超级用户可以读写任何文件,可以终止任何进程。超级用户与核心态执行的概念是不同的。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,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虚拟机。操作系统实现的管理程序调用扩展了原始硬件提供的指令集功能。管理程序调用支持新的抽象概念,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。

操作系统的任务分为以下 4 方面:

- 进程管理:进程是正在执行的程序。与进程相关的是代码、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。如果同时存在多个进程,操作系统必须为每个进程提供它运行所需的适当虚拟环境。
- 存储管理:内存至少应被一个应用程序和操作系统所共享。在更复杂的系统中,内存可被多个进程所共享。操作系统必须管理进程的内存分配,并控制存储管理硬件。存储管理硬件确定一个进程可以访问哪些内存位置。
- 文件管理:计算机处理信息。信息必须传输、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而交换信息。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 操作系统类型

只有少量用户直接使用过早期的计算机系统。早期计算机的输入(程序和数据)都存放在输入介质上,通常,输入介质是纸带或穿孔卡片。用户将输入交给操作员,然后,几分钟或几小时后,取程序或作业的输出。操作员把相似的作业组织成一批,并让计算机运行这批作业。每个作业都完全控制计算机,直到它执行完成。

批处理方式的操作系统以上述方式管理计算机运行。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很少的功能。

在多道程序批处理系统中,作业读入存放在磁盘上的作业池中。当作业因为等待 I/O 操作的完成而不能执行时,其他作业可能可以运行。并发执行的进程对计算机资源的共享大幅提升了操作系统的复杂性。

在分时操作系统中,用户和进程之间可进行交互。在批处理系统中,程序输入时提供所

有数据。对于获取工资信息和打印每周工资支票的程序来说,批处理系统很合适,但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,在合理时间内,程序对用户输入做出响应。操作系统不仅必须在不同的进程间分享资源,还必须作出一种“假象”,好像多个进程在同时运行。实际上,在所有活动进程间在迅速地轮流执行。

现在,计算机很少作为独立实体,而是作为计算机网络的一部分。这对操作系统底层的影响非常有限。计算机利用 I/O 设备在网络上通信。网络通信涉及复杂的协议,为了网络可靠性,在网络操作系统中,都内置支持这些协议。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。这种操作系统也可能支持某种形式的批处理。

实时操作系统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程序,但没有具体的时间安排。在实时系统中,任务处理的正确性依赖于处理发生的时间。如感知核反应堆冷却剂损失的实时系统,需要瞬间启动备份系统。操作系统必须保证在指定的时间限制内执行任务。实时系统大多是专用的,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,却是不利因素。完成任务的最小化延时,会阻碍资源共享,却增加了对硬件底层的访问。因此,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,网络中每台计算机的资源由计算机自身的操作系统所管理。有了操作系统,计算机之间就可以很容易地进行通信。在分布式操作系统中,所有计算机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,都由一个共同的分布式操作系统管理。分布式操作系统超出了本书的讨论范围。

1.2.2 操作系统内核

硬件在核心态执行时所执行的代码就是操作系统内核。内核不是程序。实际上,它是子例程库。在陷阱、中断或管理程序调用后,执行库中的一个或多个过程。在某点,硬件重置为用户态,并且控制返回到用户程序,但可能不是进入内核时执行的同一个程序。

从某种程度上来看,内核就是操作系统。操作系统管理的 4 个方面都在内核中实现,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,并且存储管理硬件产生的陷阱激活存储管理代码。每次进入内核时,把控制返回到用户程序之前,进程调度程序可作为最后一个步骤被调用。

尽管大的单片内核系统很常见,但也有人关注微内核操作系统。在微内核系统中,内核包括很少的功能。内核提供了诸如存储管理硬件的控制、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。不用重启系统就能激活新服务器。微内核很小,从而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,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展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模糊了,如打印的假脱机系统或窗口管理器之类的服务就很难归类。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具有的设计特性进行了研究。这些特性的一部分可能在内核之外实现,但设计原则不会变。打印假脱机系统通常被认为是系统程序的高级服务,超出了本书的讨论范围。

1.2.3 引导进程

在计算机系统加电时,执行只读存储器(Read-Only Memory, ROM)中的程序。执行完诊断检查,执行零级引导程序。程序检查引导设备是一个还是多个。在 PC 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第 1 个扇区读到主存。引导设备的第 1 个扇区称为引导扇区,它应该包含一级引导程序。一旦第 1 个扇区读入内存,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,则未定义的值将复制到内存中。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内存并执行。无论在哪种情况下,操作系统最终复制到内存中,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,并把活动分区的第 1 个扇区复制到内存中。活动分区的第 1 个扇区包含了二级引导程序,用来与分区中的操作系统协同工作。活动分区的第 1 个扇区也能从其他分区装载引导程序或操作系统。在最简单的情况下,二级引导程序把分区中的操作系统装入内存中。

在引导进程即将结束时,操作系统已装入内存,并且引导程序也转移到操作系统,计算机在核心态执行。在操作系统装载时没设置的所有数据结构(如中断向量)必须初始化,还要根据需要初始化系统寄存器和设备。

一旦操作系统为进程服务做好了准备,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 这样的简单系统中,操作系统可能只产生一个进程去执行命令解释程序。在比较复杂的系统中,可以提供系统程序 init,创建