



高等学校计算机类“十二五”规划教材

操作系统原理教程

◎主 编 胡元义 马俊宏
◎副主编 梁 琨 毕如田
金海燕 杨凯峰
何文娟

CAOZUOXITONGYUANLI
JIAOCHENG



西安电子科技大学出版社
<http://www.xduph.com>

高等学校计算机类“十二五”规划教材

操作系统原理教程

主 编 胡元义 马俊宏

副主编 梁 琨 毕如田 金海燕 杨凯峰 何文娟

参 编 杨 艳 周淑琴 柴西林

西安电子科技大学出版社

内 容 简 介

操作系统主要涉及对计算机软、硬件资源的控制和管理。本书对操作系统的实现原理进行了详细和深入的分析,力求做到对操作系统阐述的全面性、系统性、准确性和通俗性,以便透彻理解操作系统的设计思想,深化对基本概念的掌握。全书共分 6 章,主要包括:引论、处理器管理、进程同步与通信、存储管理、设备管理和文件管理。

本书结构清晰、内容丰富、取材新颖,既强调知识的实用性,也注重理论的完整性,可作为高等院校计算机及相关专业的操作系统课程教材,也可作为从事计算机工作及报考研究生人员的参考资料。

图书在版编目(CIP)数据

操作系统原理教程/胡元义,马俊宏主编. —西安:西安电子科技大学出版社,2014.7

高等学校计算机类“十二五”规划教材

ISBN 978-7-5606-3438-8

I. ① 操… II. ① 胡… ② 马… III. ① 操作系统—高等学校—教材 IV. ① TP316

中国版本图书馆 CIP 数据核字(2014)第 134804 号

策 划 胡华霖

责任编辑 毛红兵 胡华霖

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2014 年 7 月第 1 版 2014 年 7 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 18.5

字 数 438 千字

印 数 1~3000 册

定 价 38.00 元

ISBN 978-7-5606-3438-8/TP

XDUP 3730001-1

如有印装问题可调换

前 言

操作系统为计算机的使用提供了一个方便灵活、安全可靠的运行环境，特别是 Windows 操作系统的出现，使用计算机只需单击鼠标就可以了。

操作系统是计算机专业的一门核心课程，在计算机本科教学中占有十分重要的地位。操作系统主要涉及对计算机软硬件资源的控制和管理，其理论性强，内容抽象。特别是进程管理，需要通过缜密、细致的逻辑思维来想象微观时间世界中处理器的调度与运行，这种抽象、复杂的内容不易理解，也难以掌握。本书对操作系统的实现原理进行了详细和深入的分析，力求做到对操作系统进行全面性、系统性、准确性和通俗性的阐述，以便透彻理解操作系统的设计思想，深化对基本概念的掌握。

操作系统是现有软件系统中最复杂的软件之一，代码多达几亿条。著名计算机科学家 P. Denning 和他的助手及同事们认为，在操作系统方面取得了进程、内存管理、信息的保护与安全性、调度与资源管理以及系统结构等 5 种主要的成就。现今，信息的保护与安全性已作为一门独立的课程——计算机信息安全来讲授；而系统结构的部分内容已经转化为硬件的内容在计算机系统结构课程中讲授。本书围绕着操作系统主要成就(除上述已经独立设课的内容)，从原理出发详细介绍了操作系统有关内容，讲述中注重操作系统理论的发展与传承，注意知识的连贯性和拓展性，并通过精选的示例和图例来帮助读者理解和掌握操作系统知识，同时还设计了一些不同于其他操作系统教材的实现算法，如睡眠理发师问题算法等，对理解操作系统原理提供了更好的帮助。

全书共分 6 章。第 1 章引论，主要介绍操作系统的基本概念、操作系统的发展过程、操作系统的分类、操作系统运行的硬件环境以及操作系统与用户的接口；第 2 章处理器管理，主要介绍进程的引入和描述、进程状态及其转换、进程调度及调度算法、进程控制和线程；第 3 章进程同步与通信，主要介绍了进程同步与互斥、临界区的使用、实现进程同步与互斥的工具(P、V 操作和管程)、进程通信和进程死锁；第 4 章存储管理，主要介绍了存储管理的基本概念和功能、各种存储管理技术、虚拟存储的思想及实现方法；第 5 章设备管理，主要介绍了 I/O 设备的硬件结构和软件组成、I/O 设备控制方式、设备管理使用的有关技术及设备分配；第 6 章文件管理，主要介绍了文件的概念、文件的逻辑结构和物理结构、文件目录、文件存储空间的组织和管理、文件的共享和保护。

本书结构清晰、内容丰富、取材新颖，既强调知识的实用性，也注重理论的完整性。本书是编者二十多年在操作系统方面获得教学实践成果的总结，同时也汲取了国内外优秀操作系统教材的精华。学习中使用与本书配套的辅助教材《操作系统原理教程习题解析与上机实践》(西安电子科技大学出版社 2014 年出版，作者：胡元义)将会得到更好的效果。此外，本书还配有教学用的电子教案。本书可作为高等院校计算机及相关专业的操作系统课程教材，也可作为从事计算机工作及报考研究生人员的参考资料。

限于编者水平，书中难免有疏漏之处，敬请读者赐教。

编 者
2014 年 2 月

目 录

第 1 章 引论.....1	2.1.2 程序的并发执行46
1.1 操作系统的概念1	2.1.3 进程48
1.1.1 什么是操作系统1	2.2 进程的状态及转换50
1.1.2 操作系统的主要功能4	2.2.1 两状态进程模型50
1.1.3 操作系统的基本特征6	2.2.2 进程的三态模型51
1.2 操作系统的逻辑结构和运行模型7	2.2.3 进程的五态模型52
1.2.1 用户态和内核态的划分7	2.2.4 进程的挂起53
1.2.2 操作系统的逻辑结构8	2.2.5 进程控制块55
1.2.3 操作系统的运行模型11	2.3 进程控制57
1.3 操作系统的形成与发展12	2.3.1 进程切换58
1.3.1 操作系统的形成时期12	2.3.2 进程控制原语59
1.3.2 操作系统的成熟时期14	2.4 处理器调度62
1.3.3 操作系统的进一步发展时期17	2.4.1 作业与进程的关系63
1.4 主要操作系统的类型18	2.4.2 CPU 的三级调度.....64
1.4.1 批处理操作系统18	2.4.3 处理器调度队列模型66
1.4.2 分时操作系统19	2.4.4 进程调度的方式和时机67
1.4.3 实时操作系统21	2.5 单处理器调度算法70
1.4.4 微机操作系统22	2.5.1 调度原则70
1.4.5 网络操作系统23	2.5.2 常用调度算法71
1.4.6 多 CPU 操作系统.....25	2.5.3 实时调度76
1.4.7 分布式操作系统26	2.6 线程78
1.4.8 嵌入式操作系统28	2.6.1 线程的引入78
1.5 操作系统安全性概述29	2.6.2 线程的概念79
1.5.1 操作系统安全的重要性29	2.6.3 线程与传统进程的比较80
1.5.2 操作系统的安全观点30	2.6.4 线程实现原理81
1.5.3 实现操作系统安全性的基本技术30	习题 282
1.6 操作系统运行基础31	
1.6.1 处理器及工作模式31	第 3 章 进程同步与通信88
1.6.2 中断技术34	3.1 进程同步的基本概念88
1.6.3 系统调用39	3.1.1 并发进程的关系88
习题 141	3.1.2 进程的互斥与同步89
	3.1.3 临界资源与临界区90
第 2 章 处理器管理.....45	3.2 进程互斥方法91
2.1 进程的概念45	3.2.1 实现进程互斥的硬件方法91
2.1.1 程序的顺序执行45	3.2.2 实现进程互斥的软件方法94

3.3 信号量机制	96	4.3 分区式存储管理	160
3.3.1 信号量	96	4.3.1 单一连续分区存储管理	160
3.3.2 使用信号量实现进程互斥	98	4.3.2 固定分区存储管理	162
3.3.3 使用信号量实现进程同步	99	4.3.3 可变分区存储管理	164
3.4 经典互斥与同步问题	103	4.3.4 覆盖与交换技术	169
3.4.1 生产者-消费者问题	103	4.4 分页存储管理	171
3.4.2 哲学家进餐问题	105	4.4.1 分页存储管理的基本原理	171
3.4.3 读者-写者问题	109	4.4.2 分页存储管理的地址转换与 存储保护	173
3.4.4 睡眠理发师问题	111	4.4.3 两级页表和多级页表	176
3.5 经典互斥与同步问题的应用	113	4.4.4 内存物理块的分配与回收	178
3.5.1 缓冲区数据传送问题	113	4.5 分段存储管理	179
3.5.2 吃水果问题	115	4.5.1 分段存储管理的基本原理	179
3.5.3 汽车过桥问题	116	4.5.2 地址转换与存储保护	180
3.6 管程机制	117	4.5.3 分段存储管理的优缺点	183
3.6.1 条件变量与管程结构	118	4.6 段页式存储管理	184
3.6.2 管程在进程同步中的应用	120	4.6.1 段页式存储管理的基本原理	184
3.7 进程通信	123	4.6.2 段页式存储管理的 地址转换与特点	185
3.7.1 进程通信的概念	123	4.7 虚拟存储管理	187
3.7.2 共享内存通信方式	124	4.7.1 虚拟存储器的概念	187
3.7.3 消息缓冲通信方式	125	4.7.2 请求分页存储管理	189
3.7.4 信箱通信方式	128	4.7.3 请求分段存储管理	199
3.7.5 管道通信方式	130	4.7.4 请求段页式存储管理	202
3.8 死锁	131	习题 4	204
3.8.1 产生死锁的原因和必要条件	132		
3.8.2 死锁的预防	135	第 5 章 设备管理	211
3.8.3 死锁的避免	137	5.1 设备管理概述	211
3.8.4 死锁的检测与解除	141	5.1.1 设备的分类	211
习题 3	145	5.1.2 设备管理的目标和功能	212
第 4 章 存储管理	151	5.2 I/O 设备管理系统的组成	213
4.1 程序的链接和装入	151	5.2.1 I/O 系统的硬件组织	214
4.1.1 逻辑地址和物理地址	152	5.2.2 I/O 系统的软件组织	218
4.1.2 程序链接	152	5.3 I/O 设备控制方式	223
4.1.3 程序装入	153	5.3.1 程序直接 I/O 控制方式	223
4.2 存储器及存储管理的基本功能	156	5.3.2 程序中断 I/O 控制方式	224
4.2.1 多级存储器体系	156	5.3.3 直接存储器存取 I/O 控制方式	225
4.2.2 内存的分配与回收以及 地址转换	157	5.3.4 I/O 通道控制方式	228
4.2.3 内存的共享、保护及扩充	158	5.4 缓冲技术与虚拟设备技术	230

5.4.1 缓冲技术	230	6.2.1 文件的逻辑结构	261
5.4.2 虚拟设备技术	233	6.2.2 文件的物理结构	263
5.5 设备的分配与回收	235	6.3 文件目录	269
5.5.1 用于设备分配的数据结构	235	6.3.1 文件控制块与索引节点	269
5.5.2 设备分配	237	6.3.2 目录结构	270
5.5.3 设备回收	241	6.3.3 文件目录查找	272
5.6 磁盘存储器管理	242	6.4 文件存储空间管理	273
5.6.1 存储设备概述	242	6.4.1 空闲分区表法	274
5.6.2 磁盘调度	243	6.4.2 空闲块链法	274
5.6.3 提高磁盘 I/O 速度的方法	247	6.4.3 位示图法	275
5.6.4 磁盘阵列(RAID)	249	6.5 文件共享与文件安全	276
习题 5	251	6.5.1 早期的文件共享方法	276
第 6 章 文件管理	255	6.5.2 目前常用的文件共享方法	277
6.1 文件系统基本概念	255	6.5.3 文件系统的安全	279
6.1.1 文件系统的引入	255	6.5.4 文件系统的可靠性	281
6.1.2 文件与文件系统	256	6.5.5 文件保护	282
6.1.3 文件操作	259	习题 6	284
6.2 文件的组织结构	261	参考文献	288

第1章 引 论

计算机技术发展到今天,从个人计算机到巨型计算机,无一例外都配置了一种或多种操作系统(Operating System, OS)。计算机系统由硬件和软件两部分组成,操作系统是计算机系统最重要的系统软件,也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;它作为计算机硬件和计算机用户之间的中介,为应用程序提供基础,并成为整个计算机系统的控制中心。在现代计算机系统中如果不安装操作系统,很难想象如何使用计算机。操作系统不仅将仅有硬件的裸机改造成为功能强、使用方便灵活、运行安全可靠的虚拟机来为用户提供良好的使用环境,而且采用有效的方法组织多个用户共享计算机系统内的各种资源,最大限度地提高了系统资源的利用率。

1.1 操作系统的概念

1.1.1 什么是操作系统

1. 引子

计算机程序是如何运行的呢?首先,需要先进行编程,而编写程序是需要以计算机程序设计语言作为基础的。对大多数编写程序的人来说,使用的编程语言称为高级程序设计语言,如 C、C++、Java 等。但由于计算机并不认识用高级语言编写的程序,所以对编写好的程序还需要将它编译成计算机能够识别的机器语言程序,而这需要编译程序或汇编程序的帮助才能完成。其次,编译好的机器语言程序需要加载(调入内存并将程序中的逻辑地址变成可以执行的内存物理地址)到内存形成一个可执行的程序,即进程(见第2章),而这需要操作系统的帮助。进程需要在计算机芯片 CPU(中央处理器)上执行才是真正的执行,而将进程调度到 CPU 上运行也是由操作系统完成的。最后,在 CPU 上执行的机器语言指令需要变成能够在一个个时钟脉冲周期里执行的基本操作,这需要基本硬件指令系统及相关计算机硬件的支持,而整个程序的执行过程还需要操作系统提供服务和程序语言提供运行环境。这样,一个从程序到微指令执行的过程就完成了,图 1-1 给出了程序演变的全过程。

我们只是从一个线性角度来看待程序的演变,而没有考虑各种因素之间的穿插和交互过程。事实上,程序可以直接用机器语言或汇编语言这种称为“低级”语言编写。用机器语言编写的程序无需经过编译程序的翻译就可以直接在 CPU 上运行,而用汇编语言编写的程序则还需经过汇编程序的翻译后才能加载执行。

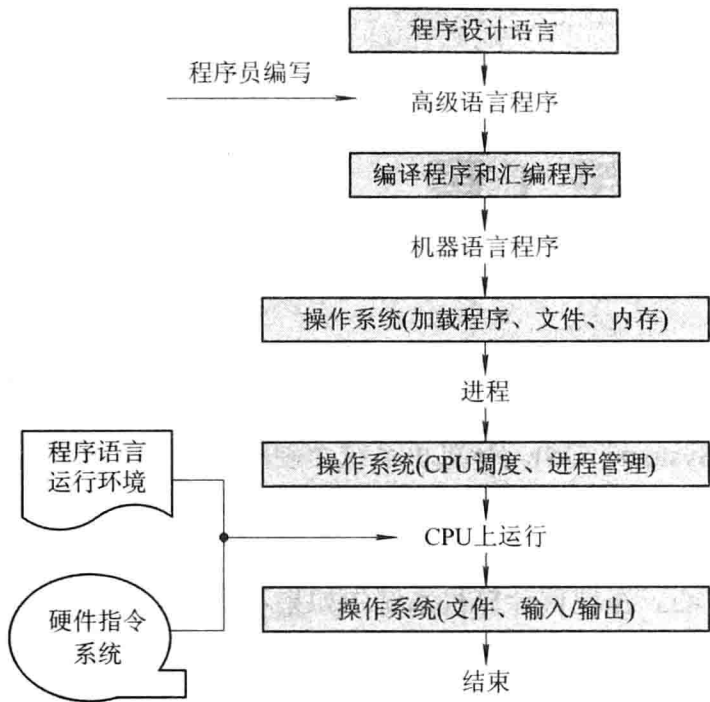


图 1-1 由程序到结果的演变

由图 1-1 的描述可以看出，程序的运行涉及四个方面：① 程序设计语言；② 编译程序；③ 操作系统；④ 硬件指令系统(计算机硬件系统)。而操作系统在程序的执行过程中具有关键作用。

2. 操作系统的定义

计算机系统是由硬件系统和软件系统两大部分组成的，硬件系统是计算机赖以工作的实体，软件系统则保证了计算机系统的硬件部分按用户指定的要求协调地工作。

计算机硬件系统由中央处理器(Central Processing Unit, CPU)、内存储器、外存储器和各种输入输出设备组成，它提供了基本的计算机资源。只有硬件的计算机称为裸机。

计算机硬件由软件来控制。按与硬件相关的密切程度，通常将计算机软件分为系统软件和应用软件两类。用户直接使用的软件通常为应用软件，而应用软件一般需借助系统软件来指挥计算机的硬件完成其功能。

那么，操作系统是什么呢？英文中的 Operating System 意为掌控局势的一种系统，也就是说，计算机里的一切事情均由 Operating System 来掌控(管理)。现在面临两个问题：第一个问题是，操作系统到底是什么东西？第二个问题是，操作系统到底操控(管理)什么事情？

由图 1-1 可以大致得到第一个问题的答案：操作系统是介于计算机硬件和应用软件之间的一个软件系统，即操作系统的下面是硬件平台，而上面则是应用软件，如图 1-2 所示。

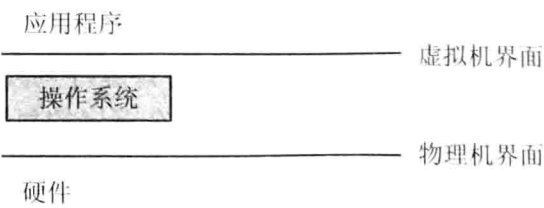


图 1-2 操作系统上下界面

现在分析第二个问题。操作系统掌控的事情当然是计算机上或计算机里发生的一切事情。最早的计算机并没有操作系统，而是直接由人来操控。随着计算机复杂性的增加，人

已经不能胜任直接掌控计算机了，于是编写出操作系统这个“软件”来掌控和管理计算机。这个掌控有着多层深远的意义：

(1) 由于计算机的功能和复杂性不断趋向更加完善和复杂，操作系统所掌控的事情也就越来越多，越来越复杂，即操作系统必须掌控计算机的所有软硬件资源，并使计算机的工作变得有序。这是早期驱动操作系统不断改善的根本原因。

(2) 既然操作系统是专门掌控计算机的，那么计算机上发生的所有事情必须得到操作系统的允许，但由于所设计的操作系统不可能做到尽善尽美，因此就给攻击者造成可乘之机，而操作系统设计人员和攻击者之间的博弈是当前驱动操作系统改善的一个重要动力。

(3) 为了更好地掌握计算机上发生的所有事情，同时也为了更好地满足人们对操作系统越来越苛刻的要求，操作系统必须不断地完善自己。

也即，从计算机管理的角度看，操作系统的引入是为了更加充分、更加有效地使用计算机系统资源，也就是合理地组织计算机的工作流程，有效地管理和分配计算机系统的硬件和软件资源，同时注意操作系统自身的安全与完善。从用户使用的角度看，操作系统的引入是为了给用户使用计算机提供一个良好的界面，使之既方便又安全。

因此，操作系统是掌控(管理)计算机上所有事情的系统软件，它需要完成如下功能：

- (1) 控制和管理计算机系统的所有硬件和软件资源。
- (2) 合理地组织计算机的工作流程，保证计算机资源的公平竞争和使用。
- (3) 方便用户使用计算机。
- (4) 防止对计算机资源的非法侵占和使用。
- (5) 保证操作系统自身的正常运转。

长期以来，(1)~(3)项一直是操作系统定义的内容，但随着恶意攻击计算机系统的事件和计算机病毒出现的越来越多，操作系统自身的安全越来越受到人们的重视。所以，现在的操作系统的定义又增加了(4)和(5)两项。

任何计算机，只有在安装了相应的操作系统后才构成一台可以使用的计算机系统。只有安装了操作系统，用户才能方便地使用计算机，计算机的各种资源才能分配给用户使用。只有在操作系统的支撑下，其他软件(如编译程序、数据库程序和网络程序等)才能因获得运行条件而执行。操作系统性能的高低直接决定着计算机整体硬件性能能否充分发挥。操作系统本身的安全性和可靠程度在一定程度上决定了整个计算机系统的安全性和可靠性。操作系统在整个计算机系统中的地位如图 1-3 所示。

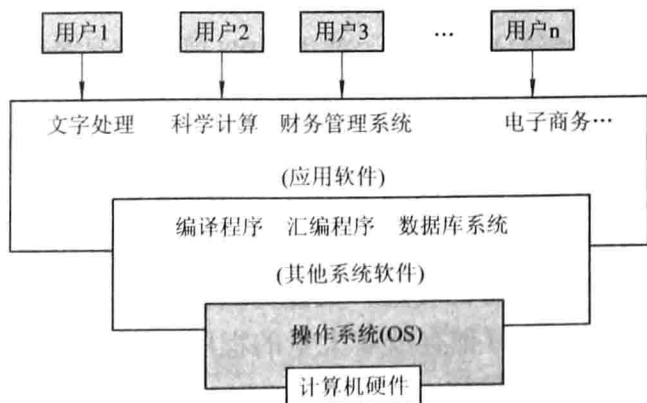


图 1-3 操作系统在计算机系统中的地位

由图 1-3 可以看出, 计算机系统具有层次结构, 其中操作系统是在硬件基础上的第一层软件, 是其他软件和硬件之间的接口。因此, 操作系统是最重要的系统软件, 它控制和协调各用户程序对硬件的使用。实质上, 用户在使用计算机时直接面对的并不是计算机的硬件, 而是应用软件, 由应用软件在“幕后”与操作系统打交道, 再由操作系统指挥计算机完成相应的工作。

3. 操作系统的设计目标

目前存在多种类型的操作系统, 不同类型的操作系统其设计目标各有侧重。一般来说, 设计的操作系统应达到如下几个目标:

(1) 方便性。操作系统应提供统一且界面友好的用户接口, 以方便用户使用计算机。

(2) 有效性。操作系统应能有效地分配和管理计算机的软、硬件资源, 使系统资源得到充分利用。此外, 操作系统应能合理地组织计算机的工作流程, 改善系统性能并提高系统运行效率。

(3) 可扩展性。操作系统应具备较好的可扩展性, 以适应计算机硬件和计算机网络等发展的需要。可扩展性表现在能否很容易地增加新的模块。

(4) 开放性。操作系统应遵循国际标准进行设计, 构造一个统一的开放环境, 以便不同厂家生产的计算机和设备能通过网络集成, 且能正确、有效地协调工作, 实现应用程序的可移植性和互操作性。

(5) 可靠性。可靠性包括正确性、健壮性和安全性。操作系统除了应满足正确性这一基本要求外, 还应满足确保操作系统自身正常工作的健壮性和如何防止非法操作和入侵的安全性。

(6) 可移植性。可移植性是指将程序从一个计算机环境转移到另一个计算机环境中仍能正常运行的特性。由于操作系统开发是一项非常庞大的工程, 为了避免重复工作及缩短软件研发周期, 现在操作系统设计已将可移植性作为一个重要目标。

1.1.2 操作系统的主要功能

为了高效地使用计算机软、硬件资源, 提高计算机系统资源的利用率和方便用户使用, 在计算机系统中都采用了多道程序设计技术(内存中同时放入多道程序交替运行), 操作系统也正是随着多道程序设计技术的出现而逐步发展起来的。要保证多道程序的正常运行, 在技术上需要解决如下问题:

(1) 在多道程序之间应如何分配 CPU, 使得 CPU 既能满足各程序运行的需要, 又能有较高的利用率。此外, 一旦将 CPU 分配给某程序后, 应何时回收。

(2) 如何为每道程序分配必要的内存空间, 使它们各得其所但又不会因相互重叠而丢失信息; 此外, 还要防止因某道程序出现异常情况而破坏其他程序。

(3) 系统中可能有多种类型的 I/O(输入/输出)设备供多道程序共享, 应如何分配这些 I/O 设备, 如何做到既方便用户对设备的使用, 又能提高设备的利用率。

(4) 在现代计算机系统中通常都存放着大量的程序和数据, 应如何组织它们才便于用户使用并保证数据的安全性和一致性。

(5) 系统中的各种应用程序有的属于计算型, 有的属于 I/O 型, 有些既重要又紧迫, 有

些又要求系统能及时响应,这时系统应如何组织这些程序(作业)的工作流程。

实际上,这些问题的全体就是操作系统的核心内容。因此,操作系统应具有以下几个方面的管理功能:处理器管理、存储管理、设备管理及文件管理。此外,随着网络技术的不断发展,操作系统还应具备相应的网络功能。同时,为了方便用户使用计算机,操作系统还应向用户提供方便、友好的用户接口。

1. 处理器管理

处理器管理主要是指对计算机系统的中央处理器(CPU)进行管理,其主要任务是对CPU进行分配,并对其运行进行有效的控制与管理。

为了提高计算机的利用率,操作系统采用了多道程序技术。为了描述多道程序的并发执行引入了进程的概念,进程可看作是正在执行的程序,通过进程管理来协调多道程序之间的关系,以使CPU资源得到最充分的利用。在多道程序环境下,CPU的分配与运行是以进程为基本单位的。随着并行处理技术的发展,为了进一步提高系统并行性,使并发执行单位的粒度更细以降低并发执行的代价,操作系统又引入了线程(Thread)的概念。对CPU的管理和调度最终归结为对进程和线程的管理和调度,它的主要功能包括进程控制和管理、进程同步与互斥、进程通信、进程死锁、线程控制和管理以及CPU调度。

2. 存储管理

存储管理是指对内存空间的管理。程序要运行就必须由外存装入内存,当多道程序被装入内存共享有限的内存资源时,存储管理的主要任务就是为每道程序分配内存空间,使它们彼此隔离互不干扰,尤其是当内存不够用时要通过虚拟技术来扩充物理内存,把当前不运行的程序和数据及时调出内存,需要运行时再将其由外存调入内存。存储管理的主要功能包括内存分配、内存保护、地址映射和内存扩充。

3. 设备管理

设备管理是指计算机中除了CPU和内存之外的所有输入输出设备(I/O设备,也称外部设备,简称外设)的管理。其首要任务是为这些设备提供驱动程序或控制程序,以使用户不必详细了解设备及接口的细节就可以方便地对设备进行操作。设备管理的另一个任务就是通过中断技术、通道技术和缓冲技术使外部设备尽可能与CPU并行工作,以提高设备的使用效率。为了完成这些任务,设备管理应该具有以下功能:外部设备的分配与释放,缓冲区管理,共享型外部设备的驱动调度,虚拟设备管理等。

4. 文件管理

文件是计算机系统中除CPU、内存、外部设备等硬件设备之外的另一类资源,即软件资源。程序和数据以文件的形式存放在外存储器(如磁盘、光盘、磁带、优盘)上,需要时再把它们装入内存。文件管理系统的主要任务是有效地组织、存储和保护文件,以使用户能方便、安全地访问它们。文件管理的主要功能包括文件存储空间管理、文件目录管理、文件存取控制和文件操作等。

5. 用户接口

为了方便用户使用,操作系统向用户提供了使用接口。接口通常以命令、图形和系统调用等形式呈现给用户;前两个供用户通过键盘、鼠标或屏幕操作,后一个供用户在编程

时使用。用户接口的主要功能包括命令接口管理、图形接口管理(图形实际上是命令的图形化表现形式)和程序接口管理。用户通过这些接口能方便地调用操作系统功能。

6. 网络与通信管理

随着计算机网络的迅速发展,网络功能已成为操作系统的重要组成部分。现代操作系统都注重为用户提供便捷、可靠的网络通信服务。为此,网络操作系统至少应具有以下管理功能:

(1) 网络资源管理。计算机联网的主要目的之一是共享资源,网络操作系统应能够实现网上资源共享,管理用户程序对资源的访问,保证网络信息资源的安全性和完整性。

(2) 数据通信管理。计算机联网后,站点之间可以互相传送数据。数据通信管理为网络应用提供必要的网络通信协议,处理网络信息传输过程中的物理细节;通过通信软件,按照网络通信协议完成网络上计算机之间的信息传输。

(3) 网络管理。包括网络性能管理、网络安全管理、网络故障管理、网络配置管理和日志管理等。

1.1.3 操作系统的基本特征

目前存在多种类型的操作系统,不同类型的操作系统有各自的特征,但它们都具有并发、共享、虚拟和不确定性等共同特征。在这些共同特征中,并发是操作系统中最重要的特征,而其他 3 个特征都是以并发为前提的。

1. 并发性

并发性(Concurrence)是操作系统最重要的特征。并发性是指两个或两个以上的事件或活动在同一时间间隔内发生(注意,不是同一时刻)。也即,在计算机系统中同时存在多个进程,从宏观上看,这些进程是同时运行并向前推进着;从微观上讲,任何时刻只能有一个进程执行,如果在单 CPU 条件下,则这些进程是在 CPU 上交替执行。

操作系统的并发性能够有效地改善系统资源的利用率,提高系统的效率。例如,一个进程等待 I/O 时,就让出 CPU 并调度另一个进程占用 CPU 执行。这样,在一个进程等待 I/O 时 CPU 就不会空闲,这就是并发技术。

操作系统的并发性会使操作系统的设计和实现变得更加复杂。例如,以何种策略选择下一个可执行的进程?怎样从正在执行的进程切换到另一个等待执行的进程?如何将各交替执行的进程隔离开来,使之互不干扰?怎样让多个交替执行的进程互通消息并协作完成任务?如何协调多个交替执行的进程对资源的竞争?多个交替执行的进程共享文件数据时,如何保证数据的一致性?为了更好地解决这些问题,操作系统必须具有控制和管理进程并发执行的能力,必须提供某种机制和策略进行协调,以使各并发进程能够顺利推进并获得正确的运行结果。此外,操作系统要充分发挥系统的并行性,就要合理地组织计算机的工作流程,协调各类软、硬件资源的工作,充分提高资源的利用率。

注意,并发性与并行性是两个不同的概念。并发性是指两个或多个程序在同一时间段内同时执行,即宏观上并行(同时执行),微观上串行(交替执行);而并行性则是指同时执行,如不同硬件(CPU 与 I/O 设备)在同时执行。

2. 共享性

共享性(Sharing)是操作系统的另一个重要特征。在内存中并发执行的多个进程可以共同使用系统中的资源(包括硬件资源和信息资源)。资源共享的方式可以分为以下两种。

(1) 互斥使用方式。该方式是指当一个进程正在使用某种资源时,其他欲使用该资源的进程必须等待,仅当这个进程使用完该资源并释放后,才允许另一个进程使用该资源,即它们只能互斥地共享该资源,因此这类资源也称互斥资源。系统中的有些资源,如打印机、磁带机的使用就只允许互斥使用。

(2) 同时使用方式。系统中有些资源允许在同一段时间内被多个进程同时使用,这里的“同时”是宏观意义上的。典型的可供多个进程同时使用的资源是磁盘,可重入程序(可供多个用户同时运行的程序)也是可被同时使用的,如编译程序。

共享性和并发性是操作系统两个最基本的特征,它们互为依存。一方面,资源的共享是因为程序的并发执行而引起的,若系统不允许程序并发执行,自然也就不存在资源共享的问题。另一方面,如果系统不能对资源共享实施有效的管理,则必然会影响到程序的并发执行,甚至程序无法并发执行,操作系统也就失去了并发性,导致整个系统的效率低下。

3. 虚拟性

虚拟性(Virtual)的本质含义是指将一个物理实体映射为多个逻辑实体。前者是实际存在的;后者是虚拟的,是一种感觉性的存在。例如,在单 CPU 系统中虽然只有一个 CPU 存在,且每一时刻只能执行一道程序,但操作系统采用了多道程序技术后,在一段时间间隔内从宏观上看有多个程序在运行,就好像有多个 CPU 支持每一道程序在运行;这种情况就是将一个物理的 CPU 虚拟为多个逻辑的 CPU。

4. 不确定性

在多道程序设计环境下,不确定性(Nondeterminacy)主要有以下表现:

(1) 在多道程序环境中允许多个进程并发执行,但由于资源等因素的限制,每个进程的运行并不是“一气呵成”,而是以“走走停停”的方式执行。内存中的每个进程在何时开始执行,在何时暂停,以什么速度向前推进,每道程序需要多少时间才能完成都是不可预知的。

(2) 并发程序的执行结果也可能不确定,即对同一程序和同样的初始数据,其多次执行的结果可能是不一样的。因此,操作系统必须解决这个问题,即保证在初始条件和运行环境相同时,同一程序的多次运行都将获得完全相同的结果。

(3) 外部设备中断、I/O 请求、程序运行时发生中断的时间等都是不可预测的。

1.2 操作系统的逻辑结构和运行模型

1.2.1 用户态和内核态的划分

设想一下,如果操作系统的可靠性和安全性出了问题,可能的结果是造成整个系统的崩溃,这将影响系统的所有用户程序;但如果一个用户程序的可靠性和安全性出了问题,所造成的损失只不过是该用户程序崩溃而操作系统将继续运行,这就保证了系统下的其他

用户程序不受影响。

不言而喻，操作系统的重要性要远远大于用户程序。那么如何保证操作系统的重要性呢？通常的办法是采用内核态和用户态两种模式。内核态是指操作系统程序运行的状态，在该状态下可以执行系统的所有指令(包括特权指令)，并能够使用系统的全部资源。用户态是指用户程序运行的状态，在该状态下所能执行的指令和访问的资源都将受到限制。

内核态和用户态各有优势：运行在内核态的程序可以访问的资源多，但对可靠性、安全性的要求高，维护管理比较复杂；用户态程序可以访问的资源有限，但对可靠性、安全性的要求低，编写程序和维护起来都比较简单。

一般来说，如果一个程序能够运行于用户态，那么就应该让它在用户态运行，只有在迫不得已的情况下才让程序在内核态运行。凡是涉及计算机本身运行的事情都应该在内核态运行，凡是只与用户数据和应用相关的部分则放在用户态运行。另外，对时序要求特别高的操作，如中断等也应在内核态完成。

那么，什么样的功能应在内核态下实现呢？首先，从保障计算机安全的角度来说，CPU和内存的管理必须在内核态实现。诊断与测试程序也应在内核态下实现，因为诊断和测试需要访问计算机的所有资源，否则如何判断计算机是否正常呢？I/O 管理也是一样，因为要访问各种设备和底层数据结构，所以也必须在内核态实现。

对于文件管理来说，可以一部分放在用户态，一部分放在内核态。文件系统本身的管理必须放在内核态，否则任何人都可能破坏文件系统的结构；用户文件(程序和数据)的管理则可放在用户态。编译程序、编辑程序、网络管理的部分功能等，自然都可以放在用户态下执行。图 1-4 描述了 Windows 操作系统的内核态与用户态的界线。

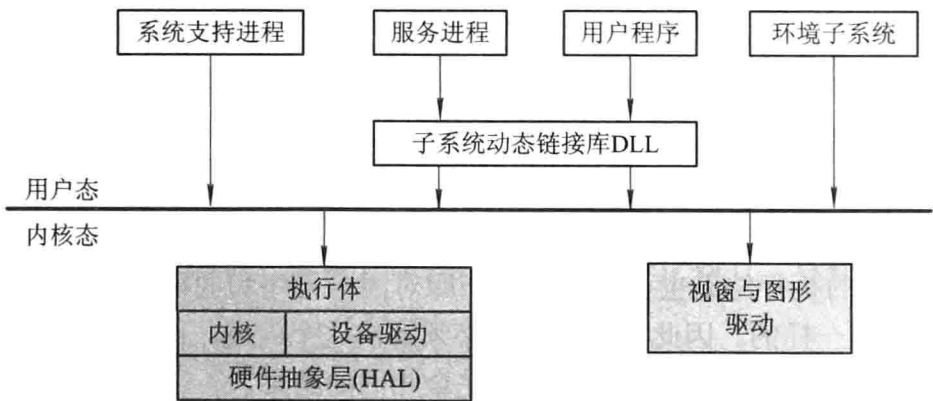


图 1-4 Windows 操作系统的内核态与用户态界线

1.2.2 操作系统的逻辑结构

操作系统结构的发展也和操作系统的历史类似，经历了好几个阶段。在操作系统刚出现时，人们并没有意识到操作系统的存在，也没有将那些库函数称为操作系统。那时候想到什么功能就把这个功能加入进来，并没有对所有的功能进行统筹兼顾的计划。显然，那时候的操作系统是杂乱无章的，没有什么结构可言。

随着操作系统不断地发展与完善，人们对操作系统的认识逐步加深，出现了不同的逻辑结构。根据内核的组织结构，可以将操作系统的逻辑结构划分为单内核、分层式和微内

核三种。

1. 单内核结构

随着操作系统的不断演化而逐渐有了一些结构，各种功能归为不同的功能模块(程序)，每个功能模块相对独立却又通过固定的界面相互联系。任何一个功能模块都可以调用其他功能模块的服务，整个操作系统呈现出单内核结构。虽然有些内核的内部又划分成若干模块或层次，但内核在结构上可以看成是一个整体，如图 1-5 所示。操作系统运行在内核态下，为用户提供服务。在单内核结构中，模块间的交互是通过直接调用相应模块中的函数来实现的，所有模块都在相同的内核空间中运行，内核代码高度集成。

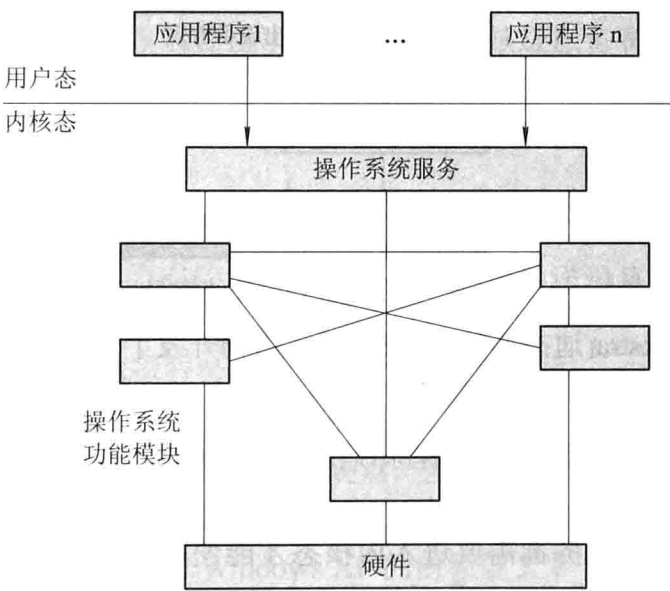


图 1-5 单内核操作系统结构图

单内核结构的优点是：结构紧密，模块间可以方便地进行组合以满足不同的需要，灵活性较好，效率高。单内核结构的缺点是：对模块功能的划分往往不能精确确定，模块的独立性可能较差；模块之间调用关系复杂，导致系统结构不清晰，正确性和可靠性不容易保证，系统维护较困难。

2. 分层式结构

单内核结构的操作系统有很多缺点：功能模块之间的关系复杂，修改任意一个功能模块可能导致其他许多功能模块都要随之修改，从而导致操作系统的设计开发困难。并且，这种没有层次关系的网状联系容易造成循环调用，从而形成死锁，导致操作系统的可靠性降低。于是，人类熟悉的层次关系引入到操作系统的设计中。

分层式结构的设计思想是：操作系统被划分成若干模块，这些模块按照功能调用次序分成若干层；每一层的程序只能使用其底层模块提供的功能和服务，即低层为高层服务，高层可以调用低层的功能，反之则不允许。按照分层结构设计的操作系统不但系统结构清晰，而且不会出现循环调用，如图 1-6 所示。

分层式结构的主要优点是：按照单向调用关系以层为单位组织各模块(程序)，使得模块之间的依赖、调用关系更加清晰和规范，对一个分层进行修改不会影响到其他层次，系统的调试和验证比较容易，系统的正确性更容易得到保证，系统中间的接口也会减少。当