

国家精品课程配套教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

计算机 操作系统

Computer Operating System

庞丽萍 编著

- 注重基础性、实质性、先进性
- 以并行处理技术和虚拟技术为纲
- 剖析操作系统原理、实现策略和方法



名家系列



人民邮电出版社
POSTS & TELECOM PRESS

国家精品课程配套教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

计算机操作系统

Computer Operating System

庞丽萍 编著

TP316

P314



名家系列

人民邮电出版社

北京

图书在版编目(CIP)数据

计算机操作系统 / 庞丽萍编著. — 北京: 人民邮电出版社, 2010.8
21世纪高等学校计算机规划教材
ISBN 978-7-115-23249-6

I. ①计… II. ①庞… III. ①操作系统—高等学校—教材 IV. ①TP316

中国版本图书馆CIP数据核字(2010)第122371号

内 容 提 要

本书全面、系统地阐述了现代操作系统的基本原理、主要功能及实现技术。重点论述多用户、多任务操作系统的运行机制; 系统资源管理的策略和方法; 操作系统提供的用户界面。讨论了现代操作系统采用的并行处理技术和虚拟技术。本书以Linux系统为实例, 剖析了其特点和具体的实现技术。

本书既可作为高等院校计算机和信息类本科、专科各专业和其他相关专业教材, 亦可供从事计算机科学、工程、应用等方面工作的科技人员参考使用。

国家精品课程配套教材

21世纪高等学校计算机规划教材

计算机操作系统

-
- ◆ 编 著 庞丽萍
责任编辑 邹文波
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
中国铁道出版社印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.75 2010年8月第1版
字数: 465千字 2010年8月北京第1次印刷

ISBN 978-7-115-23249-6

定价: 32.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前言

操作系统是计算机系统的核心软件，它管理和控制整个计算机系统，使之能正确、有效地运转，为用户提供方便的服务。

操作系统复杂且神秘，使人们感觉它威力无比，能量无限。学习操作系统就是要揭开它神秘的面纱，剖析它的复杂性，理解并掌握它，为深入学习计算机专业、信息类专业知识，进一步提升软件开发能力、乃至系统软件开发能力打下坚实的基础。

要学懂操作系统，必须了解操作系统的特点；要写好操作系统教材，也必须根据操作系统的特点确定教材内容的选取和教材的编写方法。操作系统的特点是：① 内容庞杂、涉及面广。操作系统是计算机系统的管理软件，它对计算机系统中的所有硬件和软件实施管理和控制，为用户提供良好的接口；② 动态性、并行性。现代操作系统都是多用户、多任务操作系统，支持大量的活动同时运行，各种活动都处在不断变化的过程中；③ 实践性强。现在所有运转着的计算机都配置了操作系统。各种类型的操作系统正在运转，为用户提供服务；④ 技术发展快。操作系统的实现技术和方法在不断地变化。

针对操作系统的特点，本书在内容的选取上注重基础性、实质性、先进性；框架的设计上注重逻辑性、完整性，力图将操作系统内容组织成一个逻辑清晰的整体。在这一整体中始终贯穿着并发、共享的主线。在这一主线下，有一条动态的、进程活动轨迹，还有一个系统资源管理的剖面。针对动态的进程活动，本书论述了操作系统需要提出的重要的概念——进程；支持多进程运行必需的机制（包括数据结构、实施进程控制与进程调度的设施）及功能。对系统资源管理则根据多用户、多任务环境的特点，讨论系统资源的共享，资源管理的策略与方法。为解决操作系统实践性强的问题，本书探讨了操作系统原理与实际运行的操作系统之间的关系：原理是实际操作系统采用的各种技术的提炼；实际操作系统采用的技术和方法是原理内容的具体体现。本书以当前流行的 Linux 操作系统为实例，剖析其特点和实现技术，使操作系统原理中的理论知识与操作系统实例的具体实现方法有机地结合、相互印证。随着操作系统技术的不断发展，操作系统教材在抓住基础性的同时，也需要不断地更新。本书提出了实现现代操作系统的关键技术是并行处理技术和虚拟技术，并力图以这种思想方法引领读者思考、理解操作系统的原理和它实施的策略和方法。

本书的论述力求深入浅出，通俗易懂，使读者便于阅读和理解。为了让读者能建立操作系统整体轮廓，抓住操作系统的主线，在第 1 章绪论中讨论了操作系统在计算机系统中的地位、操作系统应解决的基本问题、操作系统采用的关键技术。这些内容在读者还没有了解操作系统时是不能很好地理解的。但这些内容可以作为指导引领读者去理解后面各章的相关内容。当读者学完了全书内容，再回头看第 1 章，就会理解现代操作系统的最基本的、最关键的实现技术。

书中所有算法采用类 C 的伪码来描述。因为，这种语言与 PDL 语言十分相似，它含有更多的自然语言，这样使读者容易掌握算法的功能。

本书的编写考虑了目前高等学校计算机以及信息类各专业教学工作的实际需要。本书用于高等学校计算机本科教学时，原则上应讲授第1章~第9章的全部内容，其授课时数建议按55~60学时安排；若用于高校计算机专科教学时，应选择第1章~第9章的基本内容讲授，其授课时数建议按45~50学时安排；本书用于高校其他有关专业本科或研究生教学时，其讲授内容和学时数可由任课教师根据具体情况确定。

我在教学和编写本书过程中，学习、参考了有关操作系统、UNIX、Linux系统方面的教材，这些书都给了我很大的帮助。在此书出版之际，我要感谢指导、帮助过我的专家、作者、老师和我的朋友们，和他们的讨论、交流使我受益匪浅。

此书出版后，恳切地希望能继续得到同行和读者们的批评和帮助，以便使本书的质量能不断地提高。

庞丽萍

2010年6月于武汉华中科技大学

目 录

第 1 章 绪论 1

1.1 操作系统在计算机系统中的地位 1

1.1.1 存储程序式计算机的结构和特点 1

1.1.2 操作系统与计算机系统各层次的关系 3

1.1.3 操作系统与计算机体系结构的关系 4

1.2 操作系统的形成和发展 6

1.2.1 操作系统发展的初级阶段 7

1.2.2 操作系统的形成 9

1.2.3 操作系统的进一步发展 11

1.3 操作系统的基本概念 12

1.3.1 操作系统的定义和特性 12

1.3.2 操作系统的资源管理功能 14

1.3.3 操作系统应解决的基本问题 16

1.4 操作系统的基本类型 17

1.4.1 批量操作系统 17

1.4.2 分时操作系统 18

1.4.3 实时操作系统 18

1.4.4 个人计算机操作系统 19

1.4.5 网络操作系统 20

1.4.6 分布式系统 21

1.5 操作系统采用的关键技术 23

1.5.1 操作系统采用的并行处理技术 24

1.5.2 操作系统采用的虚拟技术 24

1.6 Linux 系统概述 25

1.6.1 UNIX 系统及其特点 25

1.6.2 Linux 系统形成和发展的基础 27

1.6.3 Linux 系统及其特点 29

习题 1 31

第 2 章 操作系统的结构和硬件支持 33

2.1 操作系统虚拟机 33

2.2 操作系统的组织结构 34

2.2.1 操作系统的结构 34

2.2.2 运行时的组织结构 37

2.2.3 操作系统与计算机系统各层次的接口 38

2.3 处理机的特权级 38

2.3.1 处理机的状态及分类 38

2.3.2 特权指令 39

2.4 中断及其处理 39

2.4.1 中断的概念及类型 39

2.4.2 向量中断和探询中断 41

2.4.3 中断进入 42

2.4.4 软件中断的处理过程 44

2.5 Linux 系统的内核结构 46

2.6 Linux 系统的特权级与中断处理 46

2.6.1 Linux 系统的特权级 46

2.6.2 中断处理的上半部和下半部 47

2.6.3 中断处理下半部的实现机制 48

习题 2 49

第 3 章 操作系统的用户接口 50

3.1 用户工作环境 50

3.1.1 操作系统提供的环境 50

3.1.2 操作系统的生成和系统初启 51

3.1.3 应用程序的处理 54

3.2 用户接口 55

3.2.1 用户接口的定义 55

3.2.2 操作系统提供的用户接口 56

3.3 系统功能调用 58

3.3.1 系统功能调用的定义 58

3.3.2 系统功能调用的实现 59

3.3.3 应用程序的编程接口 60

3.4 Linux 系统功能调用 60

3.4.1 Linux 系统功能调用的过程 60

3.4.2 Linux 系统功能调用的实现机制 62

3.4.3 增加一个新的系统调用的方法 63

3.4.4 从用户空间访问新的系统调用 64

习题 3	65	4.9.6 Linux 系统的进程等待与唤醒	102
第 4 章 进程及进程管理	66	4.9.7 Linux 系统中线程的实现	103
4.1 进程的引入	66	习题 4	104
4.1.1 顺序程序及特点	66	第 5 章 资源分配与调度	107
4.1.2 并发程序及特点	67	5.1 资源管理概述	107
4.1.3 与时间有关的错误	70	5.1.1 资源管理的目的和任务	107
4.2 进程的概念	71	5.1.2 虚拟资源	108
4.2.1 进程的定义	71	5.2 资源管理的机制和策略	109
4.2.2 进程的状态	72	5.2.1 资源分配机制	109
4.2.3 进程控制块	74	5.2.2 资源分配策略	110
4.3 进程控制	75	5.3 死锁	115
4.3.1 进程控制的概念	75	5.3.1 死锁的定义与例子	115
4.3.2 进程创建与撤销	76	5.3.2 产生死锁的原因和必要条件	116
4.3.3 进程阻塞与唤醒	77	5.3.3 系统模型和死锁的处理	118
4.4 进程之间的约束关系	79	5.3.4 解决死锁问题的策略	121
4.4.1 进程竞争与合作	79	5.3.5 死锁的预防	122
4.4.2 进程互斥的概念	79	5.3.6 死锁的避免	123
4.4.3 进程同步的概念	81	5.3.7 死锁的检测与忽略	125
4.5 同步机构	83	习题 5	125
4.5.1 锁和上锁、开锁操作	83	第 6 章 处理机调度	127
4.5.2 信号灯和 P、V 操作	84	6.1 处理机的多级调度	127
4.6 进程互斥与同步的实现	86	6.2 针对作业的调度	128
4.6.1 上锁原语和开锁原语实现进程互斥	86	6.2.1 作业的状态	128
4.6.2 信号灯实现进程互斥	87	6.2.2 作业调度的功能与作业控制块	128
4.6.3 进程同步的实现	88	6.2.3 作业调度算法性能的衡量	130
4.6.4 生产者——消费者问题	91	6.2.4 作业调度算法	131
4.7 进程通信	92	6.3 进程调度	132
4.7.1 进程通信的概念	92	6.3.1 调度/分派结构	132
4.7.2 进程通信方式	92	6.3.2 进程调度的功能	133
4.8 线程概念及特点	94	6.3.3 调度方式	134
4.8.1 线程的概念	94	6.3.4 进程优先数调度算法	135
4.8.2 线程的特点与状态	95	6.3.5 循环轮转调度	136
4.9 Linux 系统的进程管理	97	6.3.6 多级反馈队列调度	137
4.9.1 Linux 系统的进程与线程	97	6.3.7 调度用的进程状态变迁图	138
4.9.2 进程描述符及其主要内容	97	6.4 线程调度	139
4.9.3 进程描述符的获得	99	6.5 Linux 系统的进程调度	140
4.9.4 Linux 系统进程状态的变迁	100	6.5.1 进程调度程序的设计目标和特点	141
4.9.5 Linux 系统的进程创建和终止	101		

6.5.2 可变优先级	141	8.1.1 设备管理的功能	186
6.5.3 可变时间片	143	8.1.2 设备独立性	187
6.5.4 进程调度用的数据结构	144	8.1.3 设备控制块	189
6.5.5 Linux 系统的进程调度算法	145	8.2 缓冲技术	190
习题 6	146	8.2.1 缓冲概述	190
第 7 章 主存管理	148	8.2.2 常用的缓冲技术	191
7.1 主存管理概述	148	8.2.3 UNIX 系统的缓冲区管理	192
7.1.1 主存分片共享	148	8.3 设备分配	199
7.1.2 程序的逻辑组织	149	8.3.1 设备分配概述	199
7.2 主存管理的功能	149	8.3.2 独享分配	200
7.2.1 虚拟存储器	149	8.3.3 共享分配	200
7.2.2 地址映射	150	8.3.4 虚拟分配	200
7.2.3 主存分配	153	8.4 输入/输出控制	202
7.2.4 存储保护	153	8.4.1 输入/输出硬件	202
7.3 分区存储管理及其存在的问题	154	8.4.2 输入/输出控制方式	204
7.3.1 动态分区存储管理技术	155	8.4.3 输入/输出子系统	206
7.3.2 分区分配机构	156	8.4.4 输入/输出控制的例子	208
7.3.3 分区的分配与放置策略	156	8.5 Linux 系统的设备驱动	210
7.3.4 碎片问题及拼接技术	159	8.5.1 Linux 系统设备的分类	211
7.4 页式存储管理	160	8.5.2 设备文件及其标识	211
7.4.1 页式系统应解决的问题	160	8.5.3 Linux 块设备的处理	213
7.4.2 页式地址变换	161	8.5.4 用于块设备处理的数据结构	215
7.4.3 请调页面的机制	164	8.5.5 输入/输出调度程序	217
7.4.4 淘汰机制与策略	166	8.5.6 策略例程	218
7.4.5 几种置换算法	167	习题 8	219
7.5 段式系统和段页式系统	171	第 9 章 文件系统	221
7.5.1 段式地址结构	171	9.1 文件系统概述	221
7.5.2 段式地址变换	171	9.1.1 文件	221
7.5.3 扩充段表功能	172	9.1.2 文件系统	223
7.5.4 段页式存储管理	172	9.1.3 文件的组织	224
7.6 Linux 系统的存储管理	174	9.2 文件的逻辑结构和存取方法	225
7.6.1 主存寻址	174	9.2.1 文件的逻辑结构	225
7.6.2 Linux 系统段页式地址变换	176	9.2.2 文件的存取方法	226
7.6.3 Linux 系统动态内核管理	178	9.3 文件的物理结构	226
7.6.4 Linux 系统的进程地址空间	181	9.3.1 连续文件	227
习题 7	183	9.3.2 串联文件	228
第 8 章 设备管理	186	9.3.3 索引文件	229
8.1 设备管理概述	186	9.3.4 文件物理结构比较	232
		9.3.5 UNIX 系统的索引文件结构	232

9.4	文件目录	237
9.4.1	文件目录及其内容	237
9.4.2	一级文件目录及缺点	237
9.4.3	多级文件目录	238
9.5	共享与安全	239
9.5.1	文件共享与安全性的关系	239
9.5.2	文件共享的实现方法	240
9.5.3	存取权限的类型及其验证	240
9.6	文件操作与文件备份	243
9.6.1	文件操作	243
9.6.2	文件备份	244
9.7	Linux 文件系统	246
9.7.1	虚拟文件系统 (virtual file system, VFS) 概述	246
9.7.2	VFS 通用文件系统模型与 VFS 对象	246
9.7.3	与进程相关的数据结构	250

9.7.4	VFS 系统调用的实现	251
9.7.5	Ext2 文件系统概述	253
9.7.6	Ext2 磁盘数据结构	255
9.7.7	Ext2 磁盘空间的管理	257
9.7.8	Ext2 主存数据结构	260

习题 9	260
------	-----

模拟试题 1	263
--------	-----

模拟试题 2	265
--------	-----

模拟试题 1 答案	267
-----------	-----

模拟试题 2 答案	271
-----------	-----

参考文献	275
------	-----

第 1 章

绪论

1.1 操作系统在计算机系统中的地位

1.1.1 存储程序式计算机的结构和特点

随着科学技术的飞速发展,人类生活质量的不断提高,生产实践和社会活动的水平不断地提升,计算机应用随之深入且广泛。在大量的计算机应用中,如金融、航天、电信、信息家电等领域,都有大量的问题需要计算机来解决。

任何问题的求解都需要给出其形式化定义和求解方法的形式描述。问题的形式化定义称为数学模型,问题求解方法的形式描述称为算法,通常将一个算法的实现叫做一次计算,而对问题的求解还必须有实现算法的工具或设施。

实现算法的工具或设施从最初的算盘,后来使用的计算器,直到现代的、几乎无所不能的计算机,发生了巨大的变化。然而,这些工具的计算方法的本质特征是相同的。算盘和计算器可以进行加、减、乘、除运算。人们要解决某一问题,只有将问题的求解方法归结为四则运算问题后,才可以用算盘之类的工具进行计算。当遇到一个复杂的算法时,如求解一个微分方程,就必须将微分方程的解法转化为数值解法。这种计算方法称为手工计算方式,算盘或计算器是手工计算的一种工具。在这种计算方式下,人们按照预先确定的一种计算方案,先输入原始数据,然后按操作步骤做第一步计算,记下中间结果,再做第二步计算,直到算出最终结果,并把结果记录在纸上。在这一过程中,输入原始数据、执行运算操作、中间结果的存储和最终结果的抄录都是依靠人的操作来完成的,所以这一计算过程是手工操作过程。

现代计算机归根到底还是进行四则运算,不过,最重要的是它还具有自动计算和逻辑判断能力。著名数学家冯·诺依曼(Von Neumann)总结了手工操作的规律以及前人研究计算机的经验,于20世纪40年代提出了“存储程序式计算机”方案,即冯·诺依曼计算机体系结构,实现了计算的自动化。计算机要进行自动计算,需要有计算方案或称计算机程序存放在机器内,计算机还必须能“理解”程序语言的含义并顺序执行指定的操作,能及时取得初始数据和中间数据,并自动地输出结果。根据这样的分析,冯·诺依曼计算机必须有一个存储器用来存储程序和数据,有一个运算器用来执行指定的操作,有一个控制部件用来实现操作的顺序,还要有输入/输出(简称I/O)设备,以便输入数据和输出计算结果。

存储程序式计算机由中央处理器(CPU)、存储器和输入/输出设备组成。所有的单元都通过

总线连接，总线分为地址总线和数据总线，分别连接不同的部件。冯·诺依曼计算机体系结构如图 1.1 所示。从 20 世纪 40 年代至今，计算机体系结构不断地发展变化，但对于单 CPU 的计算机而言，其结构仍然是存储程序式计算机的体系结构。

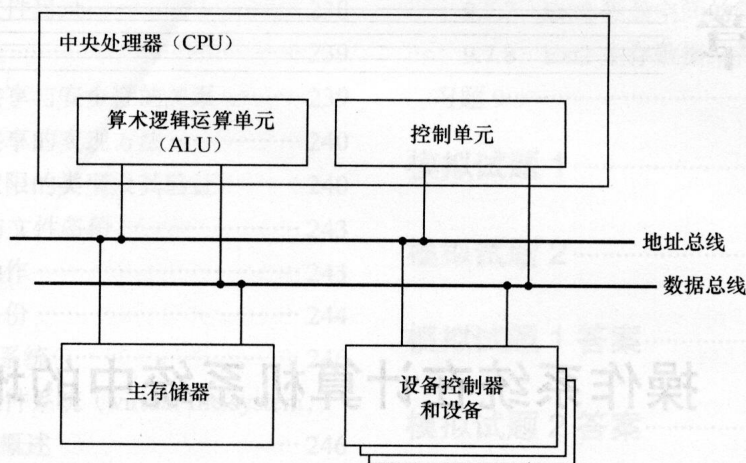


图 1.1 冯·诺依曼计算机体系结构

CPU（又称为中央处理器）是一种能够解释指令，执行指令，并控制操作顺序的硬设备。它由算术逻辑运算单元（ALU）和控制单元构成。ALU 包含一个能完成算术和逻辑操作的功能单元以及通用寄存器和状态寄存器。通用寄存器为功能单元提供操作数，并能接收、保存操作的结果。状态寄存器保存中央处理器运行过程的当前状态。现代的 CPU 一般包含 32~64 个通用寄存器，每个寄存器能够保存一个 32 位（bit）的数值。控制单元负责从主存储器提取指令、分析其类型，并产生信号通知计算机其他部分执行指令所指定的操作。控制单元包含一个程序计数器（program counter, PC）和一个指令寄存器（instruction register, IR）。程序计数器指示下一步应该执行的指令，而指令寄存器包含当前指令的拷贝。

主存储器（main memory）简称主存，是组成计算机的一个重要部件，其作用是存放指令和数据，并能由中央处理器（CPU）直接随机存取。主存接口由存储地址寄存器（memory address register, MAR）、存储数据寄存器（memory data register, MDR）以及命令寄存器（command register, CR）组成。主存的单元数目和每个单元的位数取决于当时的电子制造技术以及硬件设计的考虑。现代计算机为了提高性能，兼顾合理的造价，往往采用多级存储体系。多级存储体系由存储容量小、存取速度高的高速缓冲存储器，存储容量和存取速度适中的主存储器，存储容量大、存取速度较慢的辅存储器组成。

输入/输出设备负责信息的传输，将数据从外部世界传送到计算机内，或将主存中的内容传输到计算机的外部世界。输入/输出设备分为存储设备（如磁盘或磁带）、字符设备（如终端显示器、鼠标）和通信设备（如连接调制解调器的串行端口或网络接口）。每个设备通过设备控制器与计算机的地址总线和数据总线相连。控制器提供一组物理部件，可以通过 CPU 指令操纵它们以完成输入/输出操作。

冯·诺依曼计算机是人类历史上第一次实现自动计算的计算机，它的影响是十分深远的。它具有逻辑判断能力和自动连续运算能力。它的特点是集中顺序过程控制。计算是过程性的，完全模拟人们手工操作过程，即首先取原始数据，执行一个操作，将中间结果保存起来，再取一个数据，与中间结果一起又执行一个操作，如此计算下去，直到计算完成。由系统中的程序计数器体

现其顺序性（在单 CPU 的计算机系统中只有一个程序计数器），它根据程序规定的顺序依次执行每一个操作。集中控制是指机器各部件的工作由 CPU 集中管理和指挥。冯·诺依曼计算机的计算模型是顺序过程计算模型。

1.1.2 操作系统与计算机系统各层次的关系

现代计算机系统拥有丰富的硬件和软件资源。硬件是指组成计算机的机械的、磁性的、电子的装置或部件，也称为硬设备。硬件包括中央处理器（CPU）、存储器和各类外部设备。由这些硬件组成的机器称为裸机。

如果用户直接在裸机上处理程序会寸步难行。因为裸机不包括任何软件，没有程序设计语言及编译系统、没有编辑软件、没有操作系统（不提供数据输入/输出、文件处理等功能）……总之，裸机不提供任何可以帮助用户解决问题的手段，没有方便应用程序运行的环境。用户在使用计算机时希望十分方便，应用程序在处理时需要各方面的支持，而这一切若要求硬件来完成，不仅成本极高，有些功能硬件也不可能实现，而且对用户使用计算机将造成极大的障碍。所以，在裸机上必须配置软件，以满足用户的各种要求，特别是那些复杂而又灵活的要求。

软件由程序、数据和在软件研制过程中形成的各种文档资料组成。程序是方便用户和充分发挥计算机效能的各种程序的总称。软件可分为以下三类。

① 系统软件：操作系统、编译程序、程序设计语言，以及与计算机密切相关的程序。

② 应用软件：各种应用程序、软件包（如数理统计软件包、运筹计算软件包等）。

③ 工具软件：各种诊断程序、检查程序、引导程序。

整个计算机系统的组成可用图 1.2 来描述。

由图 1.2 可见，计算机系统由硬件和软件两部分组成。硬件处于计算机系统的底层；软件在硬件的外围，由操作系统、其他的系统软件、应用软件构成。硬件是计算机系统的物质基础，没有硬件就不能执行指令和实施最基本、最简单的操作，软件也就失去了效用；而若只有硬件，没有配置相应的软件，计算机也不能发挥它的潜在能力，这些硬件资源也就失去了活力。软件和硬件有机地结合在一起构成了计算机系统。

在所有软件中，有一个重要的系统软件称为操作系统。它管理系统中的各种软件、硬件资源，控制用户和应用程序的工作流程，将系统各部件有机地融合成一个整体，使计算机真正体现了系统的完整性和可利用性。

在计算机系统中，操作系统的位置处在硬件和其他所有软件之间。它是所有软件中与硬件相连的第一层软件，在裸机上运行。从操作系统在计算机系统的位置可以分析操作系统与各层之间的关系，这对于理解操作系统应具备的功能以及实现这些功能的方法是十分重要的。操作系统与各层的关系表现在两个方面：一是操作系统对各层的管理和控制，二是各层对操作系统的影响和制约。

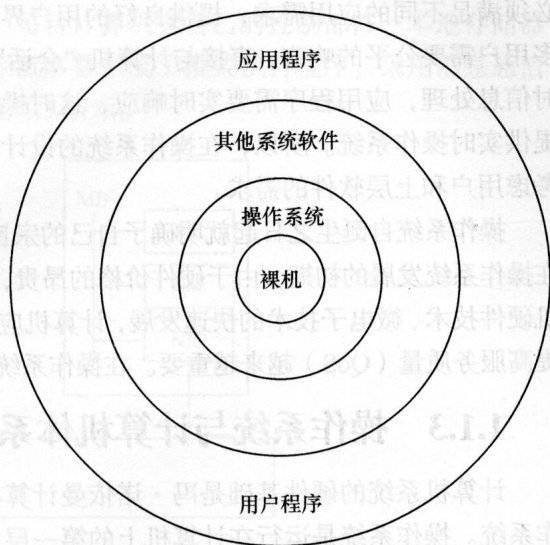


图 1.2 计算机系统的组成

1. 操作系统对各层的管理和控制

(1) 操作系统直接与硬件交互

操作系统控制 CPU 的工作、访问存储器、进行设备驱动和设备中断处理。

(2) 操作系统与用户和应用程序交互

操作系统是其他系统软件 and 应用程序运行的基础，它为上层软件 and 用户提供运行环境，即提供方便、简单的用户接口。

2. 各层对操作系统的制约

(1) 计算机系统结构对操作系统实现技术的制约

硬件提供了操作系统的运行基础，计算机的系统结构对操作系统的实现技术有着重大的影响。例如，单 CPU 计算机的特点是集中顺序过程控制，其计算模型是顺序过程计算模型。而现代操作系统大多数是多用户、多任务操作系统，是一个并行计算模型，这就是一对矛盾。

大家熟知的 Windows 系统就是一个多任务操作系统，用户在 Windows 系统中可以打开很多窗口，一个窗口就是一个任务，系统支持多个任务同时执行。例如，你正在编辑一个图像时，可以听着音乐，还可下载文件。那么单 CPU 计算机如何运行多个任务呢？这是一件十分困难的事，需要许多技术来支持。为此，操作系统提出并实现了后续各章节要讨论的内容（如多道程序设计技术、分时技术，进程概念、进程控制及同步，资源分配与调度的机制和策略），使得在单 CPU 的计算机上能实现多任务操作系统。这就是计算机的系统结构对操作系统的实现技术的影响和制约。

(2) 用户和应用程序的需求对操作系统实现技术的制约

用户和上层软件运行在操作系统提供的环境上，他们对操作系统会提出各种要求，操作系统必须满足不同的应用需求，提供良好的用户界面，为此，需要设计不同类型的操作系统。例如，多用户需要公平的响应，直接与计算机“会话”，这就需要分时操作系统。若要进行过程控制或实时信息处理，应用程序需要实时响应，这时操作系统必须提供实时和具有可预测能力的服务，即提供实时操作系统。所以，在操作系统的设计和采用的实现技术上都要考虑自己的定位，要充分考虑用户和上层软件的需求。

操作系统自诞生之日起就明确了自己的宗旨，那就是提高计算机的使用效率，方便用户的使用。在操作系统发展的初期，由于硬件价格的昂贵，将提高计算机的使用效率放在了第一位。随着计算机硬件技术、微电子技术的快速发展，计算机应用的普及和应用水平的日益提高，方便用户的使用、提高服务质量（QoS）越来越重要。在操作系统的功能实现上必须考虑这一因素和变化。

1.1.3 操作系统与计算机体系结构的关系

计算机系统的硬件基础是冯·诺依曼计算机，而构成计算机系统的一个重要的系统软件是操作系统。操作系统是运行在计算机上的第一层系统软件，必然受到冯·诺依曼计算机结构特点的制约和影响。

操作系统的宗旨是提高计算机的使用效率，方便用户的使用。早期的计算机上配置的操作系统是单用户操作系统。这样的操作系统只允许一个用户使用计算机，用户独占计算机系统的各种资源，整个系统为该用户的程序运行提供服务。在这种情况下，除了 CPU 和外部设备有可能提供并行操作外，其余的活动都是顺序操作。这种单用户操作系统也是顺序计算模型，容易实现。但存在的问题是，昂贵的计算机硬件没有得到充分利用，计算机的性能特别是资源利用率不能充分地发挥。

为了提高资源利用率，操作系统必须能支持多个用户共用一个计算机系统，必须解决多个应

用程序共享计算机系统资源的问题,也需要解决这些应用程序共同执行时的协调问题。为此,人们研究、实现了一系列新的软件技术,如多道程序设计技术、分时技术,多任务控制和协调,解决资源分配和调度的策略和方法。这些技术已经载入操作系统发展的光荣史册,并被人们誉为 20 世纪 60 年代至 20 世纪 70 年代计算机科学的奇迹,在近代又得到不断的完善和进一步的发展,这些技术的应用取得了可观的经济效益。人们所做出的努力实际上是采用了并行处理技术,将单处理机系统改造成了逻辑上的多计算机系统(现代操作系统大都是多用户、多任务操作系统)。多用户、多任务操作系统的计算模型是并行计算模型。

由于计算机系统的计算模型是顺序计算模型,其特点是集中顺序过程控制,而操作系统需要支持的是多用户、多任务的同时执行,是并行计算模型,这就产生了一对矛盾,即硬件结构的顺序计算模型和操作系统的并行计算模型的矛盾。为了解决这一矛盾,单处理机的操作系统的实现技术变得非常复杂、不易理解,最终使操作系统成为一个庞然大物,且效果并不一定很理想。

在单 CPU 计算机上配置的操作系统越来越复杂的情况下,人们研究与并行计算模型一致的计算机系统结构,出现了多处理机系统、消息传递型多计算机和计算机网络等具有并行处理能力的计算机系统结构。

1. 多处理机系统

多处理机系统具有多个处理器,所有处理器共享一个公共主存,共享 I/O 通道、控制器和外部设备。它的特点是通过共享存储器实现多个处理机(结点)之间的互相通信,由于高度的资源共享,被称为紧耦合系统。多处理机系统存在瓶颈、可扩展性差的问题。

2. 消息传递型多计算机

消息传递型多计算机由两台以上的计算机组成,每台计算机有自己的控制部件、本地存储器(处理机/存储器对)或 I/O 设备,按 MIMD(多指令流多数据流)模式执行程序,采用消息通信机制实现通信。消息传递型多计算机的一般结构如图 1.3 所示。

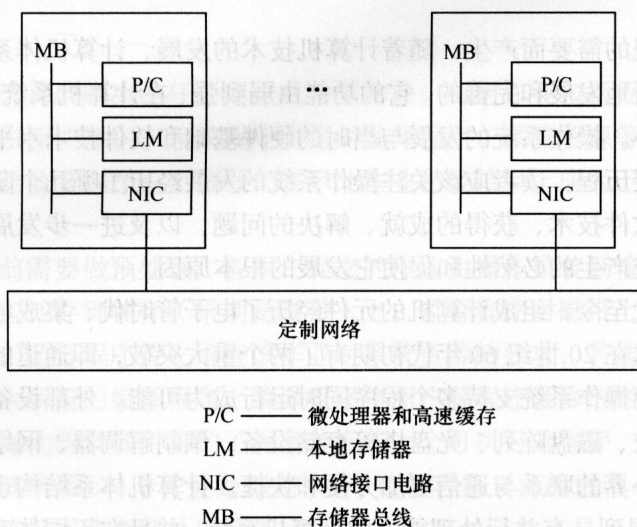


图 1.3 消息传递型多计算机的一般结构

消息传递型多计算机又可称为大规模并行计算机(massively parallel processor, MPP),其中定制网络的结构可以是网格、环、超立方体、带环立方体等。

消息传递型多计算机的结构特点是,① 多个处理机/存储器对;② 分布存储,无共享资源;③ 消息传递网络,由硬件直连,传递速率高;④ 可扩展性好。这种结构的并行计算机是具有分

布存储的多计算机系统。

3. 计算机网络

计算机网络是通过通信线将独立自主的计算机互连而成的集合体。互连是指两台计算机之间彼此交换信息，可以通过导线、激光、微波、卫星等方式进行互连。独立自主指的是网络中的每一台计算机都是独立自主的，没有主从关系。

计算机网络的特点是：① 具有多个处理部件；② 无公共主存；③ 有消息通信机制。组成计算机网络的各计算机可以分布在不同的地方，甚至可以相距很远。这一点与消息传递型多计算机不同。消息传递型多计算机的各结点由定制网络连接，它们互相分离但距离不大。

伴随着计算机体系结构的变化，计算机的应用需求也在不断地提高，操作系统随之不断地发展变化，出现了不同类型的操作系统。其类型有批量操作系统、分时操作系统、实时操作系统、个人计算机操作系统，它们采用了分时技术，多任务并发活动处理、资源分配与调度等技术。现代操作系统的代表有 Windows、UNIX、Linux 系统等。在具有并行处理能力的计算机系统上配置的操作系统类型依具体的结构不同而不同，有网络操作系统、多处理机操作系统、集群操作系统、分布式操作系统等。

目前在市场上销售的计算机，大部分仍然采用冯·诺依曼式计算机的结构，预计将来也仍然如此。因此，我们一方面要学好当前计算机系统上配置的操作系统，另一方面要关心计算机系统结构发展的新趋势。从计算机体系结构的角度出发去分析操作系统，就容易理解操作系统的功能和实现技术。通过这样的分析，不但可以学到对当前有用的知识，而且可以鉴别哪些是合理的，哪些是需要改造的，哪些是将来仍然是有用的。

1.2 操作系统的形成和发展

操作系统是由客观的需要而产生，随着计算机技术的发展、计算机体系结构的变化和计算机应用的日益广泛而不断地发展和完善的。它的功能由弱到强，在计算机系统中的地位也不断提高，以至于成为系统的核心。操作系统的发展与当时的硬件基础和软件技术水平有着密切的关系。本章讨论操作系统的发展历程，读者应该关注操作系统的发展经历了哪几个阶段，每个阶段具备的硬件支持、所采用的软件技术、获得的成就、解决的问题，以及进一步发展出现的新问题，这样我们就能了解操作系统产生的必然性和促使它发展的根本原因。

从 20 世纪 40 年代至今，组成计算机的元件经历了电子管时代、集成电路时代、超大规模集成电路时代。硬件技术在 20 世纪 60 年代初期有了两个重大突破，即通道的引入和中断技术的出现。硬件技术的成果使操作系统支持多个程序同时运行成为可能。外部设备从纸带输入机、磁带机，发展到磁盘、磁鼓、磁盘阵列、光盘塔等存储设备，调制解调器、网络接口等通信设备，这些设备使操作系统与外界的联系与通信更加方便和快捷。计算机体系结构也在不断地变化，由单 CPU 的计算机系统发展到具有并行处理能力的计算机系统，并且在不同体系结构的计算机系统上配置了不同类型的操作系统。

操作系统的发展过程经历了操作系统发展的初级阶段、操作系统的形成和进一步发展三个大的阶段。操作系统发展的初级阶段又可分为早期批处理、脱机批处理和执行系统三个过程。操作系统形成的标志性特征是采用了多道程序设计技术和分时技术，这一阶段出现了批量操作系统、分时操作系统和实时操作系统。20 世纪 80 年代以来，操作系统得到了进一步发展。出现了功能

更强、使用更为方便的各种不同类型的操作系统。

1.2.1 操作系统发展的初级阶段

1946 年至 20 世纪 50 年代后期，计算机的发展处于电子管时代，构成计算机的主要元件是电子管，其运算速度很慢（只有几千次/秒）。早期计算机由主机、输入设备（如纸带输入机、卡片阅读机）、输出设备（如打印机）和控制台组成。人们在早期计算机上解题采用的是手工操作方式，即用户用手工方式安装或拆卸，控制数据的输入或输出，通过设置物理地址启动程序运行，这些手工操作称为“人工干预”。在早期计算机中，由于计算机的速度慢，这种人工干预的影响还不算太大。

在 20 世纪 50 年代后期，计算机进入晶体管时代，计算机的速度、容量、外设的品种和数量等方面和电子管时代相比都有了很大的提高，这时手工操作的低速度和计算机运算的高速度之间形成了人机矛盾。表 1.1 所示为人工操作时间与机器有效运行时间的关系，由表 1.1 可见人机矛盾的严重性。

表 1.1 人工操作时间与机器有效运行时间的关系

机 器 速 度	作业在机器上 计算所需时间	人工操作时间	操作时间与机器 有效运行时间之比
1 万次/秒	1 小时	3 分钟	1 : 20
60 万次/秒	1 分钟	3 分钟	3 : 1

说明：通常，将计算机完成用户算题任务所需进行的各项工作称为一道作业。

为了解决人机矛盾，必须去掉人工干预，实现作业的自动过渡。人们编制了一个小的核心代码，称为监督程序。监督程序实现了作业的自动过渡，它常驻主存。这个监督程序就是操作系统的萌芽。

1. 早期批处理

每个用户将需要计算机解决的任务组织成一道作业。每道作业包括程序、数据和一个作业说明书。作业说明书提供用户标识、用户程序所需的编译程序、系统资源等信息。每道作业的最后是一个终止信息，它给监督程序一个信号，表示此作业已经结束，应为下一个用户作业的服务做好准备。

在控制作业运行前需要做预处理。各用户提交的作业由操作员装到输入设备上，然后由监督程序控制将这一批作业转存到辅存（早期是磁带）上。监督程序对磁带上的一批作业依次调度，使它们进入主存运行。监督程序首先审查该作业对系统资源的要求，若能满足，则将该作业调入主存，并在磁带上输入所需的编译程序。编译程序将用户源程序翻译成目标代码，然后由连接装配程序把编译后的目标代码及其所需的子程序装配成一个可执行的程序，接着启动执行。计算完成后输出该作业的计算结果。一个作业处理完毕后，监督程序又自动地调入下一个作业进行处理。重复上述过程，直到该批作业全部处理完毕。

监督程序实现了作业的成批处理，I/O 工作由 CPU 直接控制，这样的系统称为（早期）联机批处理系统。注意，这里说的联机指的是中央处理器对 I/O 的控制方式，若 CPU 直接控制 I/O 操作，则系统采用的是联机操作方式。

2. 脱机批处理

早期的联机批处理系统实现了作业的自动过渡，同手工操作相比，计算机的使用效率提高了。

但存在的问题是作业从输入机到磁带、由磁带调入主存、结果的输出打印都由中央处理器直接控制。在这种早期的联机操作方式下,随着处理机速度的不断提高,CPU 的高速度和 I/O 设备的低速度之间形成的矛盾不断地加剧。因为在输入或输出时,CPU 是空闲的,使得高速的 CPU 要等待低速的 I/O 设备的工作,从而不能发挥它应有的效率。为了克服这一缺点,在批处理系统中引入了脱机 I/O 技术,从而形成了脱机批处理系统。

脱机批处理系统由主机和卫星机组成,如图 1.4 所示。主机负责计算,卫星机负责 I/O 工作。作业通过卫星机输入到磁带上,然后移到主机上。主机从输入带上调入作业,并予以执行。作业完成后,主机负责把结果记录到输出带上,再由卫星机负责把输出带上的信息打印输出。这样主机摆脱了低速的 I/O 工作,可以较充分地发挥它的高速计算能力。同时,由于主机和卫星机可以并行操作,因此和早期联机批处理系统相比,脱机批处理系统较大程度地提高了系统的处理能力。

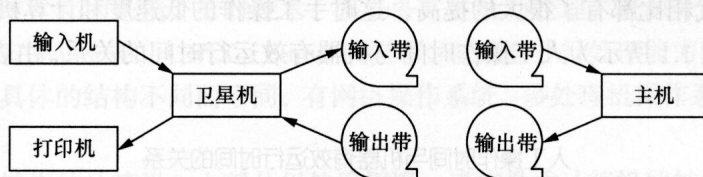


图 1.4 脱机批处理系统

联机批处理系统解决了人机矛盾,而脱机批处理系统进一步解决了 CPU 的高速度和 I/O 设备的低速度这一对矛盾。

3. 执行系统

脱机批处理系统实现了作业的自动过渡,提高了系统的处理能力,但存在着一些缺点。首先是磁带需要人工拆卸,极其不便;其次是系统保护问题越来越突出。一种情况是,由于系统没有任何保护自己的措施,若目标程序执行一条非法的停机指令,机器就会错误地停止运行。另一种情况是,如果一个程序进入死循环,系统就会踏步不前,只有操作员提出终止该作业的请求,删除它并重新启动后,系统才能恢复正常运行。更严重的是无法防止用户程序破坏监督程序和系统程序。

20 世纪 60 年代初期,硬件技术的发展出现了通道和中断,这两项重大成果导致操作系统进入执行系统阶段。通道是一种专用的处理部件,它能控制一台或多台外设的工作,负责外部设备与主存之间的信息传输。它受 CPU 的控制,一旦被启动就能独立于 CPU 运行,这样 CPU 和通道可以并行操作,CPU 和各种外部设备也能并行操作。所谓中断,是指当主机接到某种信号(如 I/O 设备完成信号)时,马上停止原来的工作,转去处理另一事件,当事件处理完毕,主机又回到原来的工作点继续工作。

在通道与中断技术的基础上,I/O 工作可以在主机控制下完成,这与早期联机批处理系统相比较有着本质的区别。因为在通道与中断技术的支持下,不仅实现了联机控制 I/O 工作,而且实现了 CPU 和 I/O 的并行操作。执行系统可以这样描述:借助通道与中断技术,由主机控制 I/O 传输,监督程序不仅负责作业的自动调度,还要负责提供 I/O 控制功能。这个优化后的监督程序常驻主存,称为执行系统。

执行系统节省了卫星机,降低了成本,实现了主机和通道、主机和外设的并行操作,提高了系统的安全性。系统负责用户的 I/O 传输工作,检查用户 I/O 命令的合法性,避免了由于不合法的 I/O 命令造成对系统的威胁。

执行系统(也称为批处理系统)的普及实现了标准文件管理系统和外部设备的自动调节控制