

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

操作系统原理

CAOZUO XITONG YUANLI

(第二版)

庞丽萍 编

华中理工大学出版社



操作系统原理

(第二版)

庞丽萍编

华中理工大学出版社

图书再版编目(CIP)数据

操作系统原理/庞丽萍 编.-2版
武汉:华中理工大学出版社,1994年9月
ISBN 7-5609-0196-4

- I. 操…
- II. 庞…
- III. 计算机软件-操作系统-教材
- IV. TP31

操作系统原理(第二版)

庞丽萍 编
责任编辑 唐元瑜

华中理工大学出版社出版发行
(武汉市 邮编:430074)
新华书店湖北发行所经销
湖北省安陆市印刷厂印刷

开本:787×1092 1/16 印张:19.25 字数:464 000
1994年9月第2版 1999年9月第12次印刷
印数:75 001-77 000
ISBN 7-5609-0196-4/TP·18
定价:18.50元
(本书若有印装质量问题,请向出版社发行科调换)

JS100/14 内 容 简 介

本书全面系统地阐述了计算机操作系统的基本原理、主要功能及实现技术，给出了操作系统的逻辑结构和用户界面，并以 UNIX 系统为例介绍了操作系统的具体的层次结构、资源管理和系统调用方法。全书共分十章，其主要内容包括：描述并发活动的基本概念——进程，进程控制及进程的交互作用；计算机系统各类资源的分配、调度策略及资源管理功能的具体实施方法等。此外，对分布式系统也作了适当的介绍。

本书可供高等院校计算机类本科、专科各专业和其他有关专业作为教材，亦可供从事计算机科学、工程和应用等方面工作的科技人员自学参考。

再版前言

《操作系统原理》自1988年出版以来已有五年多了，在这期间我又经过了多次教学实践，对操作系统课程的教学改革进行了深入的研究，同时也得到了同行们的许多宝贵意见、建议和支持。

操作系统由于它在计算机系统所处的地位决定了它的重要性。学好了操作系统不仅能掌握操作系统的基本理论，而且能具备开发系统软件的技能，能建立并程序设计的思想方法，且为以后学习数据库系统、计算机网络、分布式系统等课程打下基础。这里强调的是真正学好操作系统，不是只会背书上的条文，而是对概念的真正理解，对操作系统实现技术的深切体会。学习操作系统，不能只读一本有关操作系统原理方面的教材，而应将操作系统的理论与操作系统的实际有机地结合起来。

操作系统有如下特点：

(1) 内容庞杂，涉及面广。由于操作系统承担的任务决定了它的内容十分丰富。它要管理系统中所有的软、硬件资源，控制计算机的工作流程。它在计算机系统中处于裸机与应用层之间，对下直接与硬件接口，对上要提供简单、方便的用户界面。针对这一特点，我在《操作系统原理》的内容选择和组织上作了一些探索，力图把操作系统原理的内容组织成一个逻辑清晰的整体。

(2) 实践性强。操作系统来自客观需要，现正运行在各种不同类型的机器上。操作系统原理中的概念在实际操作系统中体现，操作系统的实现技术在实际操作系统中实施。操作系统并不是空洞、抽象的理论。问题的关键是如何将操作系统原理与操作系统实际有机地结合起来。这种结合有两个层次。首先是，操作系统的基本概念、原理、实现技术与操作系统实例相结合，通过剖析实例操作系统中某一概念的定义、某一算法的实现，加深对原理的理解。其次是，通过实验手段实现操作系统某一功能，进一步还可以设计并实现一个多任务（进程）的操作系统。通过操作系统原理的讲授和实践环节的实施可以使学生真正掌握操作系统的理论，并培养其分析问题、解决问题的能力，实际动手能力和系统软件开发能力。

操作系统课程的教学应包括三个环节：操作系统原理的讲授，操作系统原理实验和操作系统课程设计。经过了几年的努力，我们研制了与操作系统原理讲授相配套的一组实验，开发了操作系统课程设计——设计并实现一个多任务（多进程）的操作系统，并提出了一套卓有成效的实施方案。这些实验和课程设计内容在三届学生的教学中实践，取得了明显的效果。《操作系统原理实验与课程设计》一书及其与之配套的软件将由华中理工大学出版社另行出版，而《操作系统原理》一书则此次进行再版。

《操作系统原理》再版时主要作了以下几个方面的修改：

(1) 操作系统实例改用 UNIX 操作系统，而原第十章 PC-DOS 简介删去（PC-DOS 作为操作系统原理实验和课程设计的实验环境）。UNIX 系统以其强有力的生命力风靡世界，它是一个分时操作系统，具有进程控制子系统和文件子系统，还具有其他许多特色，因此可以作为与原理课相匹配的最好的实例。它融汇在本书中，除第五章资源分配与调度和第十章分布式

系统之外的各章中都有对 UNIX 相关部分的剖析。

(2) 与 UNIX 系统相配套, 本书中所有算法改为类 C 的伪码来描述。这种语言与原 PDL 语言十分相似, 它含有更多的自然语言, 这样使读者更容易掌握算法的功能。

(3) 在内容的组织上, 原书的第九章变为本书的第二章。这一章给出了操作系统虚拟机和操作系统逻辑结构, 加强了基本硬件结构的内容, 并增加了 UNIX 操作系统结构部分。

(4) 增加了分布式系统的内容作为第十章。这一章讨论了分布式系统的特征以及分布式操作系统设计中的主要问题。

本书主要是为满足目前高校计算机类专业教学工作的实际需要而修订再版的, 同时也应广大教师的要求, 较好地顾及了计算机专科和其它有关专业教学工作的实际需要。即: 本书用于高校计算机本科教学时, 原则上应讲授其全部内容, 其授课时数建议按 70~80 学时安排; 本书用于高校计算机专科教学时, 书中带“*”号和涉及 UNIX 系统方面的内容以及第十章可以不讲授(但可作为自学拓宽知识面方面的内容), 其授课时数建议按 50~60 学时安排; 本书用于高校其它有关专业本科或研究生教学时, 其讲授内容和学时数, 则可由任课教师根据具体情况确定。

在这几年中又出版了不少有关操作系统方面的好书, 不断地学习使我加深了对操作系统的理解。本书再版时我又参考了〔美〕莫里斯·贝奇著、陈葆珏等译的《UNIX 操作系统设计》, 黄铠、F. A. 布里格斯著的《计算机结构与并行处理》, 郑衍衡、徐良贤主编的《操作系统高等教程》, 刘键著的《分布式系统》, 还有尤晋元主编的《UNIX 操作系统教程》, 刘日升、孙玉芳编著的《UNIX 操作系统原理与应用》等书, 这些书都给了我很大的帮助。

在此书再版之际, 我要感谢华中理工大学教务处和计算机系的领导, 因为他们对操作系统课程建设十分关心, 特别是操作系统教学环节的全面改革得到了教务处的大力支持。对此书的再版我还要感谢华中理工大学出版社的领导及有关同志, 因为他们对此书的再版和发行十分关心并做了大量的工作。另外, 此书再版时刘键教授审阅了第十章的内容并提出了宝贵的意见, 在此也表示衷心的感谢。

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

作 者

1994年3月于武汉

目 录

第一章 绪 论	(1)
1.1 存贮程序式计算机	(1)
1.1.1 存贮程序式计算机的结构和特点	(1)
1.1.2 计算机系统结构与操作系统的关系	(2)
1.2 操作系统的形成和发展	(3)
1.2.1 早期计算机	(3)
1.2.2 批处理	(4)
1.2.3 多道程序设计 with 多道成批系统	(7)
1.2.4 分时系统	(9)
1.2.5 实时系统	(10)
1.2.6 计算机网络	(11)
1.2.7 分布式系统	(13)
1.3 操作系统的基本概念	(14)
1.3.1 操作系统的定义及其在计算机系统中的地位	(14)
1.3.2 操作系统的功能	(16)
1.4 操作系统的特性及其应解决的基本问题	(18)
1.4.1 操作系统的特性	(18)
1.4.2 操作系统的性能指标	(18)
1.4.3 操作系统应解决的基本问题	(19)
1.5 UNIX 操作系统	(20)
1.5.1 UNIX 操作系统的发展	(20)
1.5.2 UNIX 操作系统的主要特点	(21)
习题一	(23)
第二章 操作系统的逻辑结构	(24)
2.1 概 述	(24)
2.1.1 操作系统虚拟机	(24)
2.1.2 操作系统逻辑结构	(25)
2.2 基本硬件结构	(26)
2.2.1 处理机的状态及特权指令	(26)
2.2.2 存贮器	(27)
2.2.3 实时时钟	(28)
2.2.4 中断与俘获	(29)
2.3 系统核	(36)
2.4 核外各层	(37)

2.5 UNIX 系统结构	(39)
2.5.1 UNIX 系统的体系结构	(39)
2.5.2 UNIX 核心结构	(39)
习题二	(40)

第三章 用户界面 (41)

3.1 用户工作环境	(41)
3.1.1 用户环境	(41)
3.1.2 系统生成和系统启动	(42)
3.1.3 运行一个用户程序的过程	(43)
3.2 用户与操作系统的接口	(44)
3.2.1 操作系统提供的两个接口	(45)
3.2.2 作业控制语言和键盘命令	(45)
3.3 系统服务请求	(47)
3.3.1 系统调用命令	(47)
3.3.2 系统调用的实现	(48)
3.3.3 UNIX 系统调用的实现	(49)
3.3.4 UNIX 系统调用的分类	(51)
习题三	(52)

第四章 并发处理 (53)

4.1 并发活动——进程的引入	(53)
4.1.1 程序的顺序执行	(53)
4.1.2 程序的并发执行	(55)
4.1.3 并发执行实例——誊抄	(56)
4.1.4 与时间有关的错误	(59)
4.1.5 并发程序的特点	(61)
4.2 进程概念	(63)
4.2.1 进程的定义	(63)
4.2.2 进程的类型	(64)
4.2.3 进程的状态	(65)
4.2.4 进程的描述——进程控制块	(66)
4.3 进程控制	(68)
4.3.1 进程控制的概念	(68)
4.3.2 进程创建	(68)
4.3.3 进程撤消	(70)
4.3.4 进程挂起	(71)
4.3.5 进程唤醒	(71)
4.3.6 进程延迟	(72)
4.4 进程的相互制约关系	(74)

4.4.1	资源共享	(74)
4.4.2	进程合作	(74)
4.5	进程互斥	(75)
4.5.1	互斥的概念	(75)
4.5.2	锁和上锁、开锁操作	(77)
4.5.3	用上锁原语和开锁原语实现进程互斥	(79)
4.6	信号灯和 p、v 操作	(80)
4.6.1	信号灯的概念	(80)
4.6.2	p、v 操作	(81)
4.6.3	用信号灯实现进程互斥	(82)
4.7	进程同步	(83)
4.7.1	同步的概念	(83)
4.7.2	同步的例子	(84)
4.7.3	用信号灯实现进程同步	(84)
4.7.4	生产者-消费者问题	(87)
4.8	进程通信	(90)
4.8.1	进程通信的概念	(90)
4.8.2	消息缓冲通信	(90)
4.9	UNIX 系统的进程管理	(93)
4.9.1	UNIX 系统的进程及映像	(93)
4.9.2	UNIX 进程的状态及变迁	(96)
4.9.3	进程的创建	(99)
4.9.4	进程终止与等待	(102)
4.9.5	进程的睡眠与唤醒	(103)
	习题四	(106)

第五章	资源分配与调度	(109)
5.1	资源管理概述	(109)
5.1.1	资源管理的目的和任务	(109)
5.1.2	资源的几种分类方法	(110)
5.1.3	资源管理的机构和策略	(111)
5.2	资源分配机构	(112)
5.2.1	资源描述器	(112)
5.2.2	资源信息块	(113)
5.3	资源分配策略	(113)
5.3.1	概述	(113)
5.3.2	先请求先服务	(114)
5.3.3	优先调度	(114)
5.3.4	适应调度	(115)
5.3.5	均衡调度	(116)
5.3.6	针对设备特性的调度	(116)

5.4	死 锁	(117)
5.4.1	死锁的概念	(117)
5.4.2	死锁的起因	(119)
5.4.3	解决死锁问题的策略	(121)
5.4.4	死锁的预防	(123)
5.4.5	死锁的避免	(124)
5.4.6	死锁的检测与恢复	(126)
	习题五	(126)
第六章	处理机调度	(128)
6.1	处理机的二级调度	(128)
6.2	作业调度	(128)
6.2.1	作业调度的功能	(128)
6.2.2	作业控制块	(129)
6.2.3	调度性能的衡量	(130)
6.2.4	先来先服务调度算法	(132)
6.2.5	短作业优先调度算法	(132)
6.2.6	响应比高者优先调度算法	(133)
6.2.7	优先数调度算法	(134)
6.2.8	均衡调度算法	(134)
6.3	进程调度	(134)
6.3.1	调度/分派结构	(134)
6.3.2	进程调度的功能	(135)
6.3.3	调度方式	(136)
6.3.4	调度用的进程状态变迁图	(137)
6.3.5	进程优先数调度算法	(138)
6.3.6	循环轮转调度	(139)
6.3.7	多重时间片循环调度	(140)
6.4	UNIX 系统的进程调度	(140)
6.4.1	UNIX 的进程调度算法	(140)
6.4.2	进程切换调度程序 switch	(141)
	习题六	(142)
第七章	存贮管理	(144)
7.1	主存共享特征——空间分片	(144)
7.2	主存管理的功能	(144)
7.2.1	主存映射	(145)
7.2.2	程序的逻辑组织	(147)
7.2.3	主存分配	(147)
7.2.4	提供虚存	(148)

7.2.5 存贮保护	(149)
7.3 分区存贮管理	(150)
7.3.1 概 述	(150)
7.3.2 用基址寄存器实现动态地址映射	(152)
7.3.3 分区分配机构	(152)
7.3.4 分区的分配与回收	(153)
7.3.5 几种基本的放置策略	(157)
7.3.6 碎片问题及拼接技术	(159)
7.4 页式存贮管理	(160)
7.4.1 页式系统应解决的问题	(160)
7.4.2 页式地址变换	(161)
7.4.3 请调策略	(164)
7.4.4 淘汰策略	(166)
7.4.5 几种置换算法	(167)
7.4.6 页式系统的存贮分配	(170)
7.4.7 放置策略	(172)
7.4.8 页式系统中的共享	(172)
7.4.9 页式系统的保护措施	(173)
7.4.10 工作集模型	(173)
7.5 段式系统	(175)
7.5.1 段式系统的特点	(175)
7.5.2 段式地址变换	(175)
7.5.3 扩充段表功能	(176)
7.6 UNIX 系统存贮管理策略	(177)
7.6.1 概 述	(177)
7.6.2 对换空间的管理	(177)
7.6.3 对换进程	(179)
7.6.4 请求调页的数据结构	(181)
7.6.5 UNIX 系统的地址映射	(182)
7.6.6 页面错	(183)
习题七	(184)

第八章 输入输出管理

8.1 I/O 管理概念	(187)
8.1.1 引 言	(187)
8.1.2 设计目标	(188)
8.1.3 I/O 管理功能	(189)
8.1.4 设备独立性	(189)
8.1.5 设备控制块	(191)
8.2 缓冲技术	(192)

8.2.1 缓冲概述	(192)
8.2.2 双缓冲	(193)
8.2.3 环形缓冲	(193)
8.2.4 缓冲池	(194)
8.3 设备分配	(196)
8.3.1 设备分配原则	(196)
8.3.2 独享分配	(198)
8.3.3 共享分配	(198)
8.3.4 虚拟设备技术	(198)
8.3.5 SPOOLING 系统	(199)
8.3.6 慢速字符设备的分配	(201)
8.4 输入输出控制	(203)
8.4.1 输入输出控制方式	(203)
8.4.2 I/O 控制功能	(206)
8.4.3 I/O 控制的接口程序	(207)
8.4.4 设备处理进程	(208)
8.5 UNIX 系统的设备管理	(209)
8.5.1 UNIX 系统设备管理的特点	(209)
8.5.2 UNIX 设备驱动程序的接口	(210)
8.5.3 UNIX 缓冲区的管理	(213)
8.5.4 UNIX 的设备 I/O 控制	(221)
习题八	(224)

第九章 文件系统

9.1 文件系统的概念	(225)
9.1.1 引言	(225)
9.1.2 文件	(225)
9.1.3 文件系统	(228)
9.2 文件的逻辑组织与存取方法	(229)
9.2.1 文件的组织	(229)
9.2.2 文件的逻辑结构和存取方法	(230)
9.3 文件的物理结构	(231)
9.3.1 连续文件	(231)
9.3.2 串联文件	(231)
9.3.3 文件映照	(234)
9.3.4 随机文件	(234)
9.3.5 文件物理结构比较	(236)
9.4 存贮技术	(237)
9.4.1 文件存贮器的主要参数	(237)
9.4.2 磁带存贮技术	(238)
9.4.3 磁鼓存贮技术	(239)

9.4.4	磁盘存贮技术	(240)
9.5	文件存贮空间的管理	(241)
9.5.1	空白文件目录	(241)
9.5.2	空白块链	(242)
9.5.3	位示图	(242)
9.5.4	分配策略	(243)
9.6	文件目录	(243)
9.6.1	文件目录及内容	(243)
9.6.2	一级文件目录	(244)
9.6.3	二级文件目录	(245)
9.6.4	多级文件目录	(246)
9.7	共享与安全	(247)
9.7.1	文件共享与安全性的关系	(247)
9.7.2	建立“当前目录”实现文件共享	(247)
9.7.3	采用“链接技术”实现文件共享	(248)
9.7.4	基本文件目录和符号文件目录	(248)
9.7.5	存取权限的类型及其验证方法	(249)
9.8	文件的完整性	(252)
9.9	文件操作	(253)
9.10	UNIX 文件系统的主要结构及实现	(254)
9.10.1	UNIX 文件系统的特点	(254)
9.10.2	索引节点	(255)
9.10.3	文件索引结构	(256)
9.10.4	文件目录结构	(257)
9.10.5	内存打开文件结构	(258)
9.10.6	文件存贮器空闲块的管理	(263)
9.10.7	UNIX 文件系统调用	(265)
习题九	(271)

第十章	分布式系统	(273)
10.1	计算机体系结构	(273)
10.1.1	计算机体系结构的演变	(273)
10.1.2	计算机应用的发展	(273)
10.1.3	并行处理与计算机体系结构	(274)
10.2	多处理机系统	(277)
10.2.1	多处理机系统的分类	(277)
10.2.2	多处理机操作系统	(280)
10.3	网络操作系统	(281)
10.3.1	计算机网络的层次结构	(281)
10.3.2	网络操作系统的功能	(281)
10.4	分布式系统	(283)

10.4.1	分布式系统发展的原因	(283)
10.4.2	什么是分布式系统	(284)
10.4.3	分布式系统的设计目标	(285)
10.4.4	分布式系统与计算机网络的异同	(286)
10.4.5	分布式操作系统的特点	(286)
10.5	分布式操作系统设计中的若干问题	(287)
10.5.1	命名	(287)
10.5.2	资源管理	(289)
10.5.3	分布式系统的同步	(290)
10.5.4	分布式系统的容错技术	(291)
10.5.5	保护	(293)

第一章 绪 论

1.1 存贮程序式计算机

1.1.1 存贮程序式计算机的结构和特点

人们在科学实验、生产斗争和社会实践中有大量问题需要求解，如科学计算、数据处理及各种管理问题等。要解决这些问题，首先需要分析所研究的对象，提出对问题的形式化定义和给出求解方法的形式描述。对问题的形式化定义叫做数学模型，而对问题求解方法的形式描述称为算法。其次是必须具备实现算法的工具或设施。我们将一个算法的实现叫做一次计算。显然，一个计算既与算法有关，也与实现该算法的工具有关。算法和实现算法的工具是密切联系在一起的，二者互相影响、互相促进。

人们在生产活动和商业交易中最早需要解决的问题是算术四则运算问题，开始是凭人们的大脑和手来进行计算，随后使用算盘，再后用计算器，这些计算工具可以进行加、减、乘、除运算。当人们要解决某一问题时，只有将问题的求解方法归结为四则运算问题后，才能使用算盘之类的工具进行计算。由此可见，算法和计算工具是相互影响的。因为算法是四则运算，所以计算工具必须具备加、减、乘、除功能。当遇到一个复杂的算法时，如求解一个微分方程，若计算工具仍然只能进行四则运算，则必须把微分方程的解法转化为数值解法。

上面所谈的计算是一种手工计算方式，而算盘或计算器是手工计算的一种工具。在这种计算方式中，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，最后把结果记录在纸上。这里，一切都是依靠人的操作，即无论是输入原始数据，执行运算操作，还是中间结果的存贮和最终结果的抄录都是依靠人的操作，所以这一计算过程是手工操作过程。

著名数学家 Von Neumann 总结了手工操作的规律以及前人研究计算机的经验教训后，提出了“存贮程序式计算机”方案，从而使计算初步实现了自动化。要使计算机能够自动地计算，必须使机器可以“看到”计算方案即计算机程序，能够“理解”程序语言的含义并顺序执行指定的操作，可以及时取得初始数据和中间数据，能够自动地输出结果。于是，机器必须：有一个存贮器，用来存贮程序和数据；有一个运算器，用以执行指定的操作；要有一个控制部件以便实现自动操作。此外，还要有输入输出部件，以便输入原始数据和输出计算结果。它们就构成了“存贮程序式计算机”或称“Von Neumann 计算机”。

综上所述，存贮程序式计算机由以下五类部件组成：控制器，运算器，存贮器，输入装置，输出装置，人们通常把控制器和运算器做在一起，称为中央处理机或中央处理部件（CPU）。输入装置和输出装置统称为 I/O 设备，如图 1.1 所示。时至今日，人们基本上还是依照这一结构来构造计算机。

CPU 是计算机的“大脑”，能控制指挥各个部件的工作。它是一种能够解释指令、执行指令并控制操作顺序的硬设备。在 CPU 中，控制器负责从主存贮器提取指令、并分析其类型。运算器则完成为实现该指令所需进行的操作。CPU 还包含一个小的快速存贮器，用来存贮一些

暂时的结果和其它控制信息。这个存储器由若干个寄存器组成，每一个都具有某种功能，其中很重要的一个寄存器是程序计数器（PC），它指示下一条应该执行的指令。

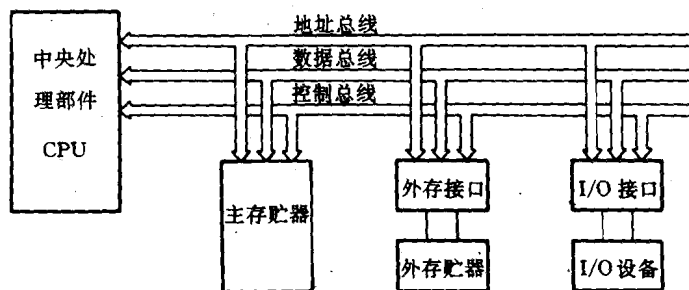


图 1.1 典型的单处理机系统结构

存储器是计算机存储程序和数据的部件。如果没有一个使中央处理机能直接读、写信息的存储器，那就不存在我们所熟悉的可存储程序的数字计算机了。早期的主存储器是由磁芯做成的，价格比较昂贵。因此，大部分计算机还有一个存取速度较慢、价格较便宜、容量大得多的辅助存储器，用于保存大量的数据信息。

输入输出设备则是完成信息传输任务的。当某一个问题需要计算机处理时，必须给定程序和初始数据，它们是通过输入设备进入计算机的。当得出解答后，计算机必须把计算结果通知用户，这是通过输出设备实现的。而控制台则是操作员用来实施控制和发布命令的设备。

Von Neumann 计算机是人类历史上第一次实现自动计算的计算机，可以真正称得上是一架自动机。该机是人类历史上第一次出现的作为人脑延伸的智能工具，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。它的计算模型是顺序过程计算模型，其主要特点是：集中顺序过程控制，即控制部件根据程序对整个计算机的活动实行集中过程控制，并根据程序规定的顺序依次执行每一个操作。计算是过程性的，故这种计算机是模拟人们的手工计算的产物。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，和中间结果一起又执行一个操作，如此计算下去。在遇到有多个可能同时执行的分支时，也是先执行完一个分支，然后再执行第二个分支，直到计算完毕。由于 Von Neumann 计算机的计算模型是顺序过程计算模型，所以它具备的根本特点是集中顺序过程控制。

1.1.2 计算机系统结构与操作系统的关系

计算机系统的硬件基础是 Von Neumann 机的五大部件，而构成计算机系统的另一个重要的系统软件是操作系统，它负责管理计算机系统的硬、软资源和整个计算机的工作流程。顺序过程计算模型决定着 Von Neumann 型计算机的根本特点——集中顺序过程控制，操作系统既是计算机系统中的一个重要组成部分，当然也不能不反映这一特点。最早产生的单用户操作系统正是如此。它只允许一个用户使用计算机，该用户独占计算机系统的各种资源，整个系统为他的程序运行提供服务。在这里，除了 CPU 和外部设备有可能提供并行操作外，其余的都是顺序操作，这种单用户操作系统简单明了，容易实现。但在这样的系统中，昂贵的计

算机硬部件并没有得到充分利用，计算机的性能特别是资源利用率大大低于可能达到的程度。

为了提高资源利用率，人们作了极大的努力把单处理机系统改造成逻辑上的多处理机系统，而且使之能进行并行处理。让多个用户共用一个计算机系统，这就必须解决多个用户的算题任务共享计算机系统资源的问题，也需要解决系统如何控制多个算题任务的共同执行。为此，出现了一系列新的软件技术，如多道程序设计技术、分时技术，以及解决资源分配和调度、进程及进程间的交互作用等问题的技术。这些技术已经载入了操作系统发展的光荣史册，并被人们誉为 60 年代至 70 年代计算机科学的奇迹。在 CPU 和存贮器都十分昂贵的情况下，这些技术的应用取得了可观的经济效益。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，因而导致操作系统必然具备并发、共享的特征。而这种特征正蕴藏着一对矛盾，即硬件结构是顺序过程计算模型，而操作系统却是并行处理（计算）模型，这二者是十分不协调的。这种尖锐的矛盾，使操作系统变得非常复杂、不易理解，成为一个庞然大物，且其效果并不一定很理想。

随着计算机技术的迅猛发展和计算机应用的日益广泛，操作系统出现了多种形式，有批量操作系统、分时操作系统、实时操作系统、单用户磁盘操作系统、计算机网络和正处在研制阶段的分布式操作系统等。其中，用得最广泛的一种是配置在微型机上的磁盘操作系统。这种操作系统的文件系统功能很强，并且提供各种方便用户使用计算机的手段，人们用起来得心应手，很受欢迎。如果某系统想共享其它系统的硬件或软件资源，也可考虑联网使用，这就是现在发展极快的计算机网络。

另一方面，人们也正在研究与并行计算模型一致的计算机系统结构，并使得具有并行处理能力的操作系统具有更强的生命力。在人们研究的多种并行处理结构中，有多指令流单数据流的流水线机，有单指令流多数据流的阵列机，还有多指令流多数据流的多处理机系统。现在有两种结构特别值得注意，一是数据流结构，就是按照数据相关关系用数据来驱动操作的执行，它能获得最大的并行性；另一种是分布式系统，它实行分布式控制并进行并行处理。正在酝酿的第五代计算机，后面这两种结构将是它的主要结构形式。

目前在市场上销售的计算机，大部分仍然采用 Von Neumann 式计算机的结构，在可见的将来也仍然是如此。因此，我们必须学好当前计算机系统上配置的操作系统，另外也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，就比较容易理解操作系统的功能和特点。通过这样的分析，我们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，哪些是将来仍然有用的、哪些是需要改造的。只有深刻地了解过去和现在，才能更好地迎接未来。下面我们首先简单地回顾一下操作系统的形成和发展过程。

1.2 操作系统的形成和发展

1.2.1 早期计算机

操作系统在现代计算机中起着相当重要的作用。它是由于客观的需要而产生，随着计算机技术的发展和计算机应用的日益广泛而逐渐发展和完善的。它的功能由弱到强，在计算机系统中的地位也不断提高，以至成为系统的核心。我们研究操作系统的形成和发展是用一种