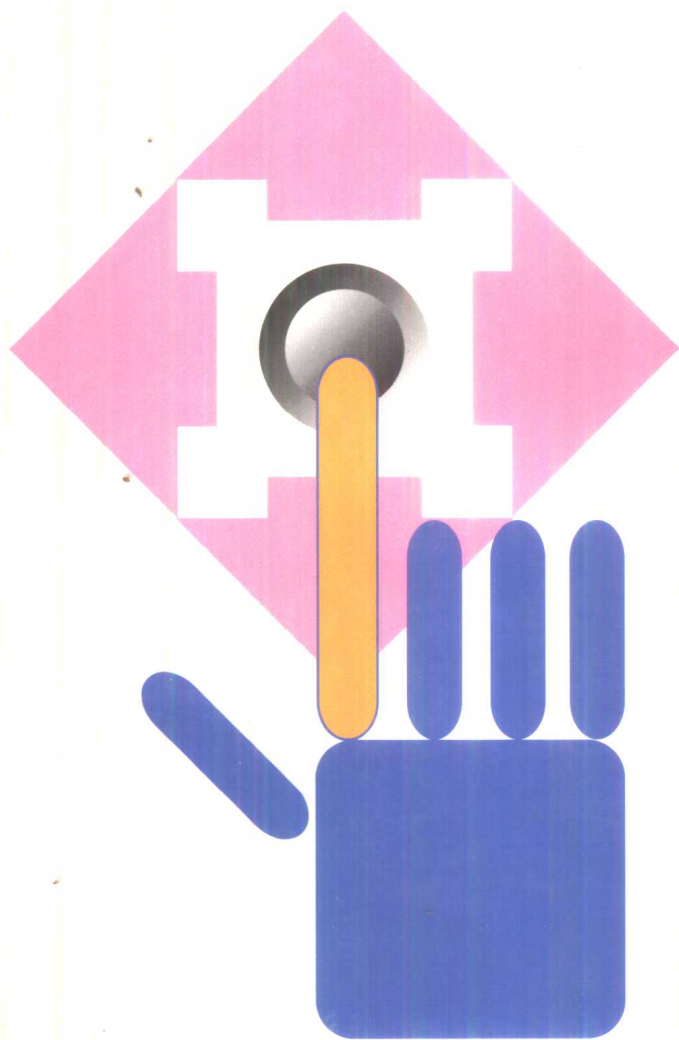


高等院校信息管理与信息系统专业系列教材

# 操作系统原理教程

孟 静 编著



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>



高等院校信息管理与信息系统专业系列教材

# 操作系统原理教程

孟静 编著

清华大学出版社

**(京)新登字 158 号**

## 内 容 简 介

本书全面系统地介绍了操作系统的经典和最新原理内容,并作为完整实例介绍了最新主流操作系统 Linux、Windows NT、Solaris 的工作原理。

本书共 8 章:第 1 章为操作系统概论,第 2 到第 6 章依次讲述 CPU 管理、主存管理、文件系统、设备管理和进程通信的原理,第 7 章介绍现代分布式、并行和网络操作系统,第 8 章介绍操作系统性能评价和设计技术。

相对于现诸操作系统原理教材而言,本书有以下鲜明特点:①选取最新主流操作系统 Linux、Solaris 和 Windows NT 作为完整实例来介绍,并在全面反映最新原理内容的同时处处注意理论联系实际;②本书提出了“硬件相关、应用无关”的操作系统本质思路,并围绕该本质思路统一组织各章节和各问题的讲述。全书内容整体感和逻辑感强,从而切实解决了操作系统课程教学的“散杂”问题;③全书每处内容的讲述层次都经过了认真组织并精益求精,紧紧抓住读者思路,启发式而深入本质地讲述,以激励读者去思维,适合自学。

作者讲授大学“操作系统原理”课程多年。本书作为讲义和教材在实际教学中已使用多年并经过了反复雕琢,可作为大学计算机专业、计算机应用专业和信息管理与信息系统专业的本科教材,同时也很适合作为自学参考用书和考试复习用书。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

书 名: 操作系统原理教程

作 者: 孟 静

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 中国科学院印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 24 字数: 613 千字

版 次: 2000 年 4 月第 1 版 2000 年 10 月第 2 次印刷

书 号: ISBN 7-302-01123-0/TP·309

印 数: 8001~13000

定 价: 25.00 元

# 出版说明

20世纪三四十年代,长期搜索前进的古老的计算技术与刚走向成熟的电子技术结合。这一结合,不仅孕育了新一代计算工具——电子计算机,还产生了当时谁也没有料到的巨大效应。电子计算机——这种当初为计算而开发出来的工具,很快就超出计算的范畴,成为“信息处理机”的代名词;人类开始能够高效率地开发并利用信息;信息对人类社会的作用得以有效地发挥,并逐步超过材料和能源成为人类社会的重要支柱;信息产业急剧增长,信息经济高速发展,社会生产力达到了新的高度;人们的信息化意识不断加强,人类在信息资源方面开始更加激烈的竞争,社会发展走上信息化轨道。

科学技术是第一生产力,教育是基础。为了加速社会信息化的过程,以培养信息资源开发人才为目标的信息管理与信息系统专业应运而生。

从与信息有关的学科纵向来看,信息管理处于信息学、信息技术、信息管理、信息经济、信息社会学这个层次的中间,它下以信息学和信息技术为基础,上与信息经济和 Information 社会学相关联。从其涉及的学科横向来看,它处在管理学、信息科学与技术、系统科学等有关学科领域的交叉点上。它对技术有极高的要求,又要求对组织的深刻理解和对行为的合理组织,反映了科学与人本融合的特点。这种交叉和融合正是信息管理与信息系统专业的最重要的特征,是别的学科或专业难以取代和涵盖的。

我国的信息管理与信息系统专业创建于20世纪70年代末。在不到20年的时间里,已发展到150多个点,成为培养信息化人才的主要摇篮。其发展速度之快、影响之深远,已令世人和学术界刮目相看。

然而,作为一个新的学科,这个专业的课程体系、教学内容以及教学方法都需要经历一个逐步完善、逐步成熟的过程。特别是教材的建设更需要经过长期的实践和探索。没有这样一个过程,具有专业特点、符合中国实际的教材是不可能产生的。近20年来大家一直在课程体系的完善和建设并具有自己专业特点的教材方面不断进行探讨。1991年全国10所财经类院校的经济信息管理专业负责人汇聚在太原召开第一次教学研讨会。以后,1993年在大连、1995年在武汉、1997年在烟台,又有更多的院校参加了这一研讨。在讨论中,各校的同仁一致认为,教材建设是当务之急,它不仅直接体现和落实培养目标,同时也是学科建设的根本所在。目前一些课程缺乏专业特点,简单搬用其他专业教材的状况亟待改变。在武汉会议上,这一共识得到了与会的国家教委有关部门负责同志的赞许,清华大学出版社也对此表示了热情的支持。会议确定了首批计划编写八九本教材,由张基温教授主持实施,由清华大学出版社出版。在实施过程中,还聘请了魏晴宇、陈禹两位教授作为顾问。

经过两年多的工作,在全国许多高等院校的同仁共同努力下,其中7本已完成初稿。我们希望这批教材的问世,能够起到抛砖引玉的作用,对各校信息管理与信息系统专业的建设和发展有所裨益。

近20年来的实践使我们对信息管理与信息系统专业的重要性和困难有了切身的体会。一方面,席卷全球的信息化大潮把信息管理推到了时代发展的前沿,信息、信息管理、信息系统已经成为全社会关注的热点。这为信息管理与信息系统专业的建设创造了良好的外部条件,提供

了难得的机遇。另一方面,信息技术的迅速发展与普及,多种社会经济因素的互相渗透和影响,前所未有的许多新问题、新情况的出现,又给这个专业的发展带来了很大的困难。我们深感责任之重大和任务之艰巨。在这套教材问世之时,我们再次表示这样一个心愿:希望与全国的同行共勉,为祖国信息化建设的宏伟事业多添一块砖,多加一块瓦,多出一份力,培养出更多的优秀人才。

由于上述种种原因,这套教材当然不会是完整的,也不会是完美的。它必然要不断补充、不断修改、不断完善。因此,对于它的任何修改意见,都是我们非常盼望的。希望能够在这套教材出版后,收到更多的意见和建议,使之逐步走向成熟。

全国高等院校计算机基础教育研究会  
财经信息管理专业委员会  
信息管理与信息系统专业教材编委会  
1997.9

# 前 言

本书全面系统地介绍了操作系统的经典和最新原理,包括操作系统内部工作过程与结构及相关概念、技术和理论,并作为完整实例介绍了最新主流操作系统 Linux、Windows NT、Solaris 的工作原理。本书共有 8 章:第 1 章为操作系统概论,第 2 到第 6 章依次讲述 CPU 管理、主存管理、文件系统、设备管理和进程通信的原理,第 7 章介绍现代分布式、并行和网络操作系统内容,第 8 章介绍操作系统性能评价和设计技术。

本书是大学计算机专业、计算机应用专业和信息专业的本科教材,参考学时为 70~80 学时,要求先修课为汇编语言、数据结构、计算机组成原理和 C 语言(其中有些课程可与本课程同学期开设)。本书也很适合作为自学参考用书和考试复习用书。

本书作者在大学讲授本科操作系统原理课程多年,在备课和写作过程中参阅了大量英文教材和实际系统剖析文献,该书作为讲义和教材已使用多年并经过反复雕琢。相对于现诸操作系统原理教材,本书有以下鲜明特点:

(1) 选取**最新主流操作系统 Linux、Solaris 和 Windows NT** 作为完整实例来介绍,并在讲述通用原理内容时处处注意理论联系实际。

(2) 本书提出了“**硬件相关、应用无关**”的操作系统本质思路,并围绕该本质思路统一组织各章节和各问题的讲述。全书内容整体感和逻辑感强,时刻使学生感受到的是由该本质思路主导和贯穿着一个整体,操作系统的所有功能和接口都由该本质思路决定,从而切实解决了操作系统课程教学的“散杂”问题。

(3) 本书以操作系统内部工作**过程**(及相应结构)作为讲述的核心和重点,并相应设计了大量图表和例子,从而使学生切实掌握操作系统内部工作原理。

(4) 书中内容全面反映操作系统**最新**技术发展,例如写时复制、内存映像文件及现代分布式、并行、网络操作系统技术等。

(5) 全书每处内容的讲述层次都经过了精心组织并精益求精,紧紧抓住读者思路,启发式而深入本质的讲述,以激励读者**思维**,适合自学。

(6) 与本书配套出版的教材还有《**操作系统原理教程题解与上机实验指导**》、另有**讲课胶片光盘和操作系统内部工作过程动态演示软件光盘**。联系电子信箱:meng jing @ mail. ruc. edu. cn

本书的出版首先要感谢丛书主编张基温教授,他不仅是个慧眼的伯乐,也是温和而严格的导师,我永远忘不了他对这本书的理解和建议和我个人能力的理解和建议。本套丛书的顾问魏晴宇教授最早推荐了我这本书,在此表示衷心的感谢。本书制图、排版和校对过程中得到了许多学生的帮助,他(她)们都是些可爱和可敬的孩子,在此向他(她)们表示感谢。在本书备课和写作过程中参考了大量英文和中文书籍与文献,都列在本书末的参考文献中,在此向这些文献的作者一并表示感谢。中国科学院张尤腊研究员在百忙之中审阅了本书,并提出了中肯的修改意见,在此表示深深的谢意。

本书虽经多年修改,但作者水平有限,错误在所难免,敬请读者多多批评指教。

编者

1999. 10

# 目 录

|  |    |
|--|----|
| <b>第 1 章 操作系统概述</b> .....  | 1  |
| 1.1 操作系统是什么 .....  | 1  |
| 1.1.1 你所用过的操作系统 .....  | 1  |
| 1.1.2 操作系统是什么与做什么 .....  | 3  |
| 1.1.3 操作系统的规模、数量与重要性 .....   | 6  |
| 1.2 操作系统大致如何工作 .....   | 7  |
| 1.2.1 操作系统的第一个工作:负责所有用户程序的启动和结束<br>——程序的 5 种启动方式和两种结束方式 .....            | 7  |
| 1.2.2 操作系统的第二个工作:用户程序中对操作系统的调用<br>——系统调用和中断 .....                        | 14 |
| 1.2.3 操作系统的第三个工作:为常用的基本操作提供现成的实用程序 .....                                 | 22 |
| 1.2.4 操作系统的第四个工作:解决效率和安全问题——并发技术等 .....                                  | 22 |
| 1.3 从各种角度看操作系统 .....   | 27 |
| 1.3.1 操作系统的结构 .....  | 27 |
| 1.3.2 操作系统的接口(使用方式) .....  | 28 |
| 1.3.3 操作系统的工作过程 .....  | 29 |
| 1.3.4 操作系统的特点 .....  | 30 |
| 1.3.5 操作系统的类型 .....  | 30 |
| 1.3.6 操作系统的各种别名、比方和观点 .....  | 31 |
| 1.4 操作系统发展简史 .....   | 31 |
| 1.4.1 操作系统出现以前的计算机使用方式 .....   | 32 |
| 1.4.2 操作系统的第一个发展阶段:单任务自动批处理 .....  | 36 |
| 1.4.3 操作系统的第二个发展阶段:多任务和多方式 .....   | 39 |
| 1.4.4 操作系统的第三个发展阶段:软件工程和小型化 .....  | 41 |
| 1.4.5 操作系统的第四个发展阶段:开放系统和并行分布 .....                                       | 42 |
| 1.4.6 操作系统发展现状和展望 .....  | 44 |
| 1.4.7 操作系统发展历史小结 .....   | 45 |
| 1.5 目前最常用的操作系统 .....   | 46 |
| 1.5.1 Microsoft 公司的操作系统产品:Windows 系列及 MS DOS .....                       | 46 |
| 1.5.2 UNIX 大家庭:SVR4,Solaris,HP-UX,SCO UNIX,BSD,AIX 等 .....               | 51 |
| 1.5.3 自由软件中的操作系统:Linux,freeBSD,BeOS... .....                             | 55 |
| 1.5.4 IBM 公司的操作系统产品:AIX,OS/390,OS/2,OS/400,<br>PCDOS7 和 PCDOS 2000 ..... | 62 |
| 1.6 本章小结.....  | 65 |
| 习题 .....   | 65 |

|  |     |
|--|-----|
| <b>第 2 章 处理机管理</b> .....                         | 67  |
| 2.1 处理机管理概述 .....                                | 67  |
| 2.2 进程模型 .....                                   | 69  |
| 2.2.1 进程三态的转换过程分析 .....                          | 69  |
| 2.2.2 进程模型实现 .....                               | 71  |
| 2.2.3 专题:CPU 调度算法 .....                          | 73  |
| 2.3 CPU 管理实例分析(1): UNIX 早期版本的 CPU 管理(进程模型) ..... | 74  |
| 2.3.1 UNIX 下如何建立进程和在一个程序中启动另一个程序 .....           | 74  |
| 2.3.2 UNIX 的进程层次和初启过程 .....                      | 75  |
| 2.3.3 UNIX 进程模型的基本结构和工作过程 .....                  | 77  |
| 2.4 CPU 管理实例分析(2): Linux CPU 管理(进程模型) .....      | 78  |
| 2.5 线程模型 .....                                   | 87  |
| 2.5.1 线程概念 .....                                 | 87  |
| 2.5.2 线程的作用和必要性 .....                            | 88  |
| 2.5.3 线程的实现:线程包内容、用户态线程、核心态线程 .....              | 90  |
| 2.6 处理机管理实例分析(3): Solaris (UNIX)的进程模型和线程模型 ..... | 96  |
| 2.6.1 用户态线程、LWP、核心线程的区别和关系以及为什么同时需要它们 .....      | 96  |
| 2.6.2 Solaris 线程模型的设计目标和实现机制总概 .....             | 98  |
| 2.6.3 SVR4 和 SVR4.2MP 的进程和线程模型概述 .....           | 99  |
| 2.7 处理机管理实例分析(4): Windows NT 的进程和线程模型 .....      | 99  |
| 习题 .....   | 102 |
| <b>第 3 章 主存管理</b> .....                          | 103 |
| 3.1 主存管理概述 .....                                 | 103 |
| 3.1.1 内存的定义、作用和计算机存储层次 .....                     | 103 |
| 3.1.2 主存硬件接口使用特性:微观角度(指令级)和宏观角度(程序级) .....       | 105 |
| 3.1.3 用户程序对内存的使用要求 .....                         | 108 |
| 3.1.4 内存管理的功能和任务 .....                           | 109 |
| 3.2 连续模式(1):操作系统出现前的内存无管理模式 .....                | 113 |
| 3.2.1 概述 .....                                   | 113 |
| 3.2.2 覆盖技术 .....                                 | 113 |
| 3.2.3 动态装入 .....                                 | 114 |
| 3.3 连续模式(2):单一分区模式 .....                         | 115 |
| 3.3.1 概述和基本实现机制 .....                            | 115 |
| 3.3.2 交换技术的作用和实现 .....                           | 116 |
| 3.3.3 单一分区模式的总结和评价 .....                         | 119 |
| 3.4 连续模式(3):固定分区模式 .....                         | 119 |
| 3.5 连续模式(4):可变分区模式 .....                         | 122 |
| 3.5.1 本模式的内外基本工作过程 .....                         | 122 |
| 3.5.2 策略选择(算法与数据结构) .....                        | 125 |
| 3.5.3 本模式评价改善与实际系统采用情况 .....                     | 126 |



|            |   |            |
|------------|---|------------|
| 3.5.4      | 多对基寄存器 .....                                  | 128        |
| 3.6        | 不连续模式(1): 页模式(paging) .....                   | 128        |
| 3.6.1      | 实存页模式的基本工作过程与结构 .....                         | 129        |
| 3.6.2      | 虚存页模式的基本工作过程与结构 .....                         | 133        |
| 3.6.3      | 虚存概念和作用 .....                                 | 134        |
| 3.6.4      | 页模式实现策略专题讨论(1): 进程页表的实现——快表、<br>页表页和页目录 ..... | 134        |
| 3.6.5      | 页模式实现策略专题讨论(2): 大而稀疏内存使用 .....                | 137        |
| 3.6.6      | 页模式实现策略专题讨论(3): 页分配策略、写时复制和页长确定 .....         | 137        |
| 3.6.7      | 页模式实现策略专题讨论(4): 页淘汰策略、工作集理论和颠簸 .....          | 139        |
| 3.6.8      | 专题讨论(5): 动态连接共享库 .....                        | 140        |
| 3.6.9      | 页模式评价和实际系统采用情况 .....                          | 141        |
| 3.7        | 不连续模式(2): 段模式 .....                           | 142        |
| 3.7.1      | 引言: 用户内存观点和段的划分模式 .....                       | 143        |
| 3.7.2      | 模式定义、基本工作过程与结构 .....                          | 144        |
| 3.7.3      | 实现中的一些策略 .....                                | 147        |
| 3.7.4      | 段模式的评价与实际系统采用情况 .....                         | 148        |
| 3.8        | 不连续模式(3): 段页模式 .....                          | 150        |
| 3.9        | 主存管理实例分析(1): Windows NT 的主存管理 .....           | 151        |
| 3.9.1      | 内部实现 .....                                    | 152        |
| 3.9.2      | 用户界面 .....                                    | 159        |
| 3.10       | 主存管理实例分析(2): Linux 内存管理 .....                 | 163        |
| 3.11       | 主存管理概念模型: 四空间概念 .....                         | 165        |
|            | 习题 .....                                      | 166        |
| <b>第4章</b> | <b>外存管理和文件系统</b> .....                        | <b>169</b> |
| 4.1        | 外存管理和文件系统概述 .....                             | 170        |
| 4.1.1      | 外存硬件接口使用特性 .....                              | 170        |
| 4.1.2      | 用户对外存的使用要求 .....                              | 177        |
| 4.1.3      | 文件系统的界面高度和工作任务 .....                          | 179        |
| 4.2        | 文件系统用户界面 .....                                | 183        |
| 4.2.1      | 文件 .....                                      | 183        |
| 4.2.2      | 目录(树)和链接 .....                                | 185        |
| 4.2.3      | 文件子系统级的用户界面 .....                             | 189        |
| 4.3        | 文件的实现 .....                                   | 195        |
| 4.3.1      | 连续分配背景下的讨论 .....                              | 196        |
| 4.3.2      | 不连续分配背景下的讨论 .....                             | 205        |
| 4.4        | 目录的实现 .....                                   | 216        |
| 4.4.1      | 用目录文件方法来实现目录树结构 .....                         | 216        |
| 4.4.2      | 设备目录与文件目录的分离 .....                            | 219        |
| 4.4.3      | 链接的内部实现 .....                                 | 223        |

|              |  |            |
|--------------|--|------------|
| 4.5          | 文件子系统的实现 .....                                       | 226        |
| 4.5.1        | 文件子系统的实现机制总述 .....                                   | 227        |
| 4.5.2        | UNIX ss 的文件子系统的实现机制(单类型<br>文件子系统) .....              | 228        |
| 4.5.3        | Linux(和 SVR4、Solaris)的文件子系统的实现机制<br>(多类型文件子系统) ..... | 229        |
| 4.5.4        | 本节小结 .....   | 229        |
| 4.6          | 文件系统性能实现机制 .....                                     | 230        |
| 4.6.1        | 物理地址与存取单位的优化 .....                                   | 232        |
| 4.6.2        | 文件打开与关闭技术 .....                                      | 232        |
| 4.6.3        | 文件共享 .....   | 232        |
| 4.6.4        | 当前目录结构和名字快速缓存 .....                                  | 235        |
| 4.6.5        | 内存缓冲区与缓冲池 .....                                      | 236        |
| 4.6.6        | 磁臂调度技术 .....   | 237        |
| 4.6.7        | 其他技术概述 .....   | 240        |
| 4.6.8        | 文件系统的安全性和可靠性 .....                                   | 242        |
| 4.7          | 文件系统实例分析 .....                                       | 243        |
| 4.7.1        | DOS 文件系统——FAT 文件系统 .....                             | 243        |
| 4.7.2        | UNIX s5 文件系统 .....                                   | 246        |
| 4.7.3        | UNIX BSD 文件系统——FFS 和 UFS .....                       | 246        |
| 4.7.4        | Linux 文件系统 .....                                     | 249        |
| 4.7.5        | Windows NT 文件系统 .....                                | 250        |
| 4.8          | 本章总结和有关文件系统模型 .....                                  | 252        |
|              | 习题 .....   | 254        |
| <b>第 5 章</b> | <b>设备管理</b> .....                                    | <b>256</b> |
| 5.1          | 设备管理概述 .....   | 257        |
| 5.1.1        | 计算机外部设备的定义与分类 .....                                  | 257        |
| 5.1.2        | 外部设备硬件接口特性 .....                                     | 260        |
| 5.1.3        | 用户对设备的使用要求 .....                                     | 268        |
| 5.1.4        | 操作系统设备管理功能的任务 .....                                  | 270        |
| 5.2          | UNIX 设备 I/O 用户界面和内部实现 .....                          | 271        |
| 5.2.1        | 用户界面 .....   | 271        |
| 5.2.2        | 内部实现过程与结构 .....                                      | 273        |
| 5.3          | DOS I/O 实例分析 .....                                   | 278        |
| 5.4          | 操作系统设备管理功能界面和实现原理综述 .....                            | 279        |
| 5.4.1        | 用户界面综述 .....   | 279        |
| 5.4.2        | 内部结构与过程综述 .....                                      | 280        |
| 5.4.3        | 速度匹配技术专题讨论(1): 设备完成技术、同步和异步 I/O 操作 .....             | 285        |
| 5.4.4        | 速度匹配技术专题讨论(2): 缓冲技术 .....                            | 288        |
| 5.4.5        | 设备分配与共享技术专题讨论: 独占、共享和虚拟设备 .....                      | 289        |

|              |                                    |            |
|--------------|------------------------------------|------------|
| 5.4.6        | 速度匹配技术专题讨论(3): 联机、脱机和假脱机操作 .....   | 291        |
| 5.4.7        | 非编程 I/O 技术专题讨论: DMA、通道等 .....      | 292        |
| 5.4.8        | 操作系统设备管理功能与其他功能间的关系 .....          | 297        |
| 5.5          | 设备驱动程序 .....                       | 298        |
| 5.5.1        | 设备驱动程序及其与核心间的接口(标准) .....          | 298        |
| 5.5.2        | 驱动程序例析: 显示器、键盘、打印机驱动程序 .....       | 299        |
| 5.6          | Windows NT 设备管理实例分析 .....          | 299        |
| 5.7          | Linux 设备管理实例分析 .....               | 301        |
| 5.8          | 本章小结 .....                         | 302        |
|              | 习题 .....                           | 302        |
| <b>第 6 章</b> | <b>进程通信</b> .....                  | <b>304</b> |
| 6.1          | 背景 .....                           | 304        |
| 6.2          | 临界区问题 .....                        | 306        |
| 6.2.1        | 两进程的解决方法 .....                     | 307        |
| 6.2.2        | 多个进程的解决方法 .....                    | 309        |
| 6.3          | 同步硬件 .....                         | 311        |
| 6.4          | 信号量 .....                          | 312        |
| 6.4.1        | 用法 .....                           | 313        |
| 6.4.2        | 实现 .....                           | 313        |
| 6.4.3        | 死锁和饥饿 .....                        | 314        |
| 6.5          | 经典的进程协同问题 .....                    | 315        |
| 6.5.1        | 有界缓冲区问题 .....                      | 315        |
| 6.5.2        | 读写问题 .....                         | 316        |
| 6.5.3        | 哲学家用餐问题 .....                      | 317        |
| 6.6          | 进程间的通信 .....                       | 318        |
| 6.6.1        | 命名方式 .....                         | 319        |
| 6.6.2        | 缓冲 .....                           | 321        |
| 6.6.3        | 异常条件 .....                         | 322        |
| 6.7          | 小结 .....                           | 323        |
|              | 习题 .....                           | 323        |
| <b>第 7 章</b> | <b>现代操作系统: 网络、分布式和并行操作系统</b> ..... | <b>325</b> |
| 7.1          | 目标 .....                           | 325        |
| 7.1.1        | 分布式系统与集中式系统相比所具有的优点 .....          | 325        |
| 7.1.2        | 分布式系统与许多独立的 PC 相比所具有的优点 .....      | 326        |
| 7.1.3        | 分布式系统的缺点 .....                     | 327        |
| 7.2          | 硬件概念 .....                         | 327        |
| 7.2.1        | 分类 .....                           | 327        |
| 7.2.2        | 总线型多处理机 .....                      | 329        |
| 7.2.3        | 开关型多处理机 .....                      | 330        |
| 7.2.4        | 总线型多计算机 .....                      | 331        |

|              |                                 |            |
|--------------|---------------------------------|------------|
| 7.2.5        | 开关型多计算机 .....                   | 331        |
| 7.3          | 软件概念 .....                      | 332        |
| 7.3.1        | 网络操作系统和 NFS .....               | 332        |
| 7.3.2        | 真正的分布式系统 .....                  | 338        |
| 7.3.3        | 多处理机分时系统 .....                  | 338        |
| 7.4          | 实现策略 .....                      | 340        |
| 7.4.1        | 透明性 .....                       | 340        |
| 7.4.2        | 灵活性 .....                       | 342        |
| 7.4.3        | 可靠性 .....                       | 343        |
| 7.4.4        | 性能 .....                        | 344        |
| 7.4.5        | 伸缩性 .....                       | 344        |
| 7.5          | 分布式文件系统 .....                   | 346        |
| 7.5.1        | 文件服务接口 .....                    | 346        |
| 7.5.2        | 目录服务接口 .....                    | 347        |
| 7.5.3        | 文件共享的语义 .....                   | 350        |
| 7.6          | Mach 概述 .....                   | 351        |
| 7.6.1        | Mach 的历史 .....                  | 351        |
| 7.6.2        | Mach 的目标 .....                  | 352        |
| 7.6.3        | Mach 的微内核 .....                 | 352        |
| 7.6.4        | Mach 上的 BSD UNIX 服务器 .....      | 353        |
| 7.7          | 小结 .....                        | 354        |
|              | 习题 .....                        | 355        |
| <b>第 8 章</b> | <b>操作系统的性能与设计</b> .....         | <b>356</b> |
| 8.1          | 性能 .....                        | 356        |
| 8.1.1        | 性能和性能指标总述 .....                 | 356        |
| 8.1.2        | 可扩充性、可移植性和兼容性 .....             | 357        |
| 8.1.3        | 安全性 .....                       | 359        |
| 8.1.4        | 可靠性和 RAS 技术 .....               | 360        |
| 8.2          | 操作系统结构设计 .....                  | 362        |
| 8.2.1        | 单体模型 .....                      | 362        |
| 8.2.2        | 层次模型 .....                      | 363        |
| 8.2.3        | 客户/服务器模型与微核结构 .....             | 364        |
| 8.2.4        | 策略与机制的分离 .....                  | 364        |
| 8.2.5        | 面向对象方法和模型 .....                 | 365        |
| 8.2.6        | 面向对象技术的实例分析:NT 的面向对象模型的实现 ..... | 365        |
| 8.3          | 操作系统的用户界面设计 .....               | 369        |
|              | 习题 .....                        | 369        |
|              | 参考文献 .....                      | 371        |

# 第1章 操作系统概述

你肯定是带着许多问题开始读这本书的,这些问题可以概括为下面6个:

1. What——操作系统是什么?做什么?
2. Why——为什么需要操作系统(其作用和重要性)?
3. How——操作系统如何工作?如何使用操作系统?
4. What——本书学习内容?
5. Why——学了有何用处?
6. How——如何学?

本书第1章就来解答这6个问题。当然,对其中的第3个问题只作大致的解答,其系统深入的答案构成全书的主要内容和目标。其余5个问题的答案也需要在全书的学习过程中逐步加深理解。实际上,在学任何一门课程之前,都有类似的这6个问题。对任何一门课程的学习也都是对这6个问题的解答,当然主要内容是第3个问题。

本章共分5节:1.1节讲述操作系统是什么、做什么,为什么需要操作系统(即其作用和重要性),操作系统是一个多大规模的软件;1.2节介绍操作系统大致是如何工作的;1.3节从1.2节中总结出操作系统的接口、过程和结构;并从各种角度看操作系统,包括其特点、类型、各种别名、比方和观点;1.4节简介操作系统的发展历史;1.5节介绍目前最常用的几个实际的操作系统。最后为本章小结与习题。

## 1.1 操作系统是什么

操作系统是什么?做什么?

也许你说不出操作系统是什么,但你肯定用过操作系统,至少一个,如果你用过计算机的话。那么我们现在先把“操作系统是什么”这个问题放一放,先来看一看你用过的那(几)个操作系统。

### 1.1.1 你所用过的操作系统

我们已经学过一些计算机课程,用过或听说过一些程序或软件。下面这些软件中哪些是操作系统呢?如 TETRIS(搭积木)、DOS、Windows、Turbo-C、WORD、FoxPro、UNIX、你自己编的一个C语言源程序、Turbo-ASM、VI和Linux。

虽然你不能概括地说出操作系统的定义,但你知道上述软件中只有DOS、Windows、Linux和UNIX是操作系统,其余的软件都是用户程序和其他系统软件(表1.1)。实际存在的操作系统有许多种(如果包括过去的,有几百种),其中目前最常用的就是这4种,其他常用的还有NetWare(Novell公司的网络操作系统)、OS/2(IBM为其高档个人计算机设计的操作系统)、OS/400(IBM小型机上的操作系统)及Macintosh(苹果计算机上的操作系统),等等。

表 1.1 下述软件中哪些是操作系统

| 软件名             | 说明   |
|-----------------|--|
| TETRIS(搭积木)     | 游戏软件。属于应用软件。                                 |
| DOS             | 操作系统。是 80 年代 IBM 个人计算机系列上最常用的操作系统。           |
| Windows         | 操作系统。是 IBM 个人计算机系列上最常用的窗口多任务操作系统。            |
| Turbo-C         | 编译程序。属于系统软件。                                 |
| Word            | 编辑排版软件。是 Windows 下的编辑排版软件。                   |
| FoxPro          | 数据库管理软件。                                     |
| UNIX            | 操作系统。是多用户计算机上最常用的操作系统。                       |
| 你自己编的一个 C 语言源程序 | 严格说,这不算程序,只能算数据。这个源程序经编译连接后产生的才是程序——可执行目标程序。 |
| Turbo-Assembler | 汇编程序。属于系统软件。                                 |
| VI              | 编辑软件。是 UNIX 下的编辑软件。                          |
| Linux           | 操作系统。其界面克隆自 Unix。是目前最常用的 OS 之一。              |

根据你的使用经验(虽然可能是有限的),能说一一说你所用过的 DOS、Windows 和 UNIX 这些操作系统能做些什么事呢?

(1) 我们用过 dir、copy、cd、format 等,它们有一个共同的名称: DOS 命令。这些命令都属于 DOS 这个操作系统的功能。同样,类似的 UNIX 命令也属于 UNIX 操作系统的功能。在 Windows 中,类似的功能是通过鼠标来体现的。

(2) 用计算机做任何事,都需先运行某个程序或某条命令。运行程序的最典型的方法是在 DOS 命令提示符后打入程序名。开机后如果不出现命令提示符,你就什么命令也打不了,什么程序也运行不了,也就什么事都做不了,这个命令提示符就是 DOS 显示的。显示命令提示符、接受命令行、启动执行相应的程序,都属于 DOS 的功能。UNIX 中也是这样的。在 Windows 中,类似的功能是通过鼠标体现的。

(3) 在用汇编语言编程序时,都用过 INT 语句。通常使用这些语句的原因都是要做一些 I/O 工作,比如文件读写或打印等。INT 语句实际上是一种特殊的调用语句,它所调用的都是 DOS 的内部功能。INT 语句是 DOS 向所有程序提供的调用接口或调用方式。在 Windows 和 UNIX 中也有类似的内部功能和相应的对外调用接口。

(4) 在 Windows 中,可以同时运行多个程序(即多任务),比如我们在等一个文件打印完成的同时,可以编辑另一个文件,这就提高了用户效率和机器利用率。多任务方式是 Windows 提供的。DOS 中就没有类似的功能,因此我们在用 DOS 打印一个文件时就只能等待。而 UNIX 可以让多个用户同时使用一台计算机,这就更进一步提高了机器的利用率。

.....

至此,你已有了一个印象:操作系统的功能很多、很杂、很散,也很难概括。好像什么事都做。但操作系统又不可能什么事都做,否则还要别的那么多软件干什么。操作系统不做什么事?你能列举出来吗?

- (1) 操作系统不做天气预报,这是由专门的天气预报软件来做的。
  - (2) 操作系统不做房屋设计,这是由专门的建筑 CAD 软件来做的。
  - (3) 操作系统不是编译程序,你用什么语言编的源程序就用相应的什么语言的编译程序。
- .....

总之,操作系统不直接解决最终具体的应用问题,也不负责编译源程序.....

那么操作系统到底做什么,不做什么,它的准确的功能范围是什么,它到底是一个怎样的软件,有没有内在的统一本质呢?

## 1.1.2 操作系统是什么与做什么

操作系统为你完成所有与硬件相关、与应用无关的工作,以给你方便、效率和安全。

操作系统(operating system,简称 OS)是计算机系统中最重要的系统软件,是这样的一组系统程序的集成:这些系统程序在用户对计算机的使用中,即在用户程序中中和用户操作中,负责完成所有与硬件因素相关的(硬件相关)和任何用户共需的(应用无关)基本使用工作,并解决这些基本使用工作中的效率和安全问题。为使用户(操作和上层程序)能方便、高效与安全地使用计算机硬件和其中存储的信息,而从最底层统一提供所有通用的帮助和管理(图 1.1)。

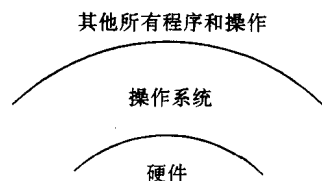


图 1.1 操作系统在计算机系统中的地位

具体地来说,在用户操作和用户程序中,哪些工作内容与硬件相关并与应用无关呢?主要是以下 4 方面工作。也就是说,操作系统是完成以下工作的诸系统程序的集成(表 1.2)。

(1) 负责启动执行每个程序,并负责完成每个程序的结束处理工作。使每个程序可以很方便灵活地开始执行和中止。

(2) 在任何用户程序中,负责完成程序在使用硬件及硬件中存放的信息时的所有与硬件相关并与应用无关的基本使用工作。每当用户程序中涉及这些工作时,就通过调用操作系统来完成(这种调用称为系统调用),从而为程序中对计算机各种硬件和硬件中存放的信息的方便使用提供统一的帮助和管理。

(3) 为用户对硬件及所存放信息的基本操作提供现成的实用程序和相应的管理,以使这些操作能很方便和有效地进行。这里所说的基本操作是指任何应用或开发背景下都通用的和普遍需要且经常发生的(如拷贝文件,删除文件,显示磁盘目录内容,显示文件内容,格式化磁盘等)。

(4) 改善上述 3 方面基本使用中的效率和安全问题,使计算机系统的各部分和整个计算机系统得到高效安全的利用。

以上 4 个工作似乎相互无关,但它们其实都具有这样的共性:硬件相关和应用无关(见表 1.2)。反过来说,计算机使用中的硬件相关和应用无关工作就是这 4 个方面。

何谓硬件相关和应用无关?

在这里,我们说一个工作是硬件相关的,是指这个工作的实现代码中包含内外存及设备的物理地址,包含对设备接口寄存器和设备接口缓冲区的读写,等等。硬件相关的代码必然随硬

表 1.2 操作系统的所有具体工作和它们的共性(详见 1.2 节)

| 操作系统的所有工作有以下几个方面。                          | 这些工作的具体内容和完成方式举例。  | 这些工作都与硬件相关                             |                                   | 这些工作都与应用无关                             |   |  | 如果没有完成这些工作,操作系统怎么样?       | 由操作系统完成这些工作后,用户和程序会怎么样?                             |   |
|--|--|--|-----------------------------------|--|---|--|---------------------------|---|---|
|  |  | 实现该工作的过程和代码因密切相关(即需要设置物理地址、设备接口、寄存器等。) | 硬件必然复杂,代码量大。                      | 硬件相关的工作,其实现代码不通用,当硬件变化时,需要重新编制或编译相应代码。 | 任何应用(使用)都需要该工作。                           | 在不同应用中,该工作的过程都是相同的。  |                           |   | 与具体应用无直接关系(即与用户所关心的应用目标无直接关系)。  |
| 负责启动每个程序,并负责完成每个程序的工作。                     | 例如:通常启动一个程序只需在命令提示符后输入程序名即可。此外,操作系统还提供了多种启动方式(详见 1.2 节)。 | 启动和结束过程涉及到对 IO 设备、外存、内存的使用,这涉及物理地址及接口。 | 一个程序的启动和结束处理所需的代码量可能比程序本身还大。      | 例如,如果程序从软盘移到磁带上,其代码就需要重新编制。            | 用计算机做任何事都需要相应程序。而每个程序都需要启动和结束处理工作。        | 所有程序都是雷同的(就启动方式而言,详见 1.2 节)。所有程序的结束处理工作是雷同的(就结束原因而言,详见 1.2 节)。 | 例如:一个财务软件启动和结束处理问题没有直接关系。 | 所有程序都必须包含引导程序(详见 1.2 节),所有程序的启动相当于一开机操作——麻烦、费时、不灵活。 | 用户程序中包含已装入的引导代码,不与本程序的功能密切相关。所有程序结束出口处只需一条“返回”指令。不必考虑程序启动与结束问题,不必了解硬件细节知识,不必频繁编制相应代码。 |
| 每当用户到设备、存储器、缓冲区内等时,其硬件中件和相关应用工通过调用操作系统来完成。 | 例如:在计算机上,用户需输入 int 调用系统。                                 | 例如:打字机需要查看打印机接口的物理地址。                  | 用户对 IO 设备的使用涉及程序代码的大多数接口及物理地址等细节。 | 例如:计算机系统换了同型号的打印机,则所涉程序都要重新编制。         | 任何程序执行时,都要使用 IO 设备,就要与设备接口打交道,就要与物理地址打交道。 | 例如:打印字符 A 与字符 E 的过程是一样的,不过处理的数据(A 与 E)不同。                      | 软件中,打印数据与逻辑上无直接关系。        | 所有程序都必须了解硬件知识,当硬件变化时,必须重新编制程序。                      | 用户程序中包含硬件应用代码,即与接口、寄存器、物理地址打交道的那些代码。用户了解硬件知识,编制相应代码。                                  |



| 操作系统所有的工作有以下4方面。                    | 这些工作的具体内容和完成方式举例。  | 这些工作都与硬件相关                                   |  |   | 这些工作都与应用无关                   |                                  |                               | 如果没有操作系统来完成这些工作,用户和用户程序会怎么样? |                  |
|-------------------------------------|--|--|--|---|------------------------------|----------------------------------|-------------------------------|------------------------------|------------------|
|                                     |  | 实现该工程代码和硬件因素密切相关(即需要设置与测试、使用物理地址、设备接口、寄存器等。) | 硬件必然复杂,代码量大。   | 硬件相关的工作,其实现代码不通用,当硬件变化时,需要重新编制或编译相应代码。  | 任何应用(使用)都需要该工作。              | 在不同应用中,该工作的过程都是相同的。              | 与具体应用无直接关系(即与用户关心的应用目标无直接关系)。 |                              |                  |
| 3<br>为用户的基本操作提供程序。这常是基本操作应用普遍需要的操作。 | 例如: DOS 中为文件拷贝提供的程序有 copy、diskcopy、xcopy 等。用户需要只运行这些程序,不行编程。 | 这些操作大量涉及对外 IO 设备的使用。                         | 用户对 IO 设备的使用所涉及的程序占程序主体代码的大部分。这些代码多涉及接口寄存器、物理地址等复杂繁琐的细节。 | 例如: 计算机系统换了一个型号的打印机,则程序中所涉及的那些代码都要重新编制。 | 不管是财务应用还是人口统计,任何应用都需要进行这些工作。 | 拷贝一个总账和一个财务文件清单,其过程都是相同的,只是数据不同。 | 例如: 拷贝的过程与数据无关。               | 用户要常用这些操作自行编程。               | 用户不必为这些常用操作亲自编程。 |
| 4<br>解决工作中的安全效率和全率问题。               | 例如: 操作系统提供多任务提高效率。   | 例如: 多任务的实现需管理 IO 设备。                         | 例如: 提供多种方式供用户操作,不用提供多用户操作,系统要复杂得多。                       | 例如: 计算机系统换了一个型号的打印机,则程序中所涉及的那些代码都要重新编制。 | 每个用户都希望提高效率。每人都希望个人机器都高效率。   |                                  |                               |                              |                  |

件的变化而变化。这样的硬件变化包括内外存物理存储空间大小的变化,程序和数据在内外存物理存储空间中的存放位置的变化,设备数量和类型的变化,等等,但不包括 CPU 指令集的变化,否则所有的程序就都是硬件相关的了。我们说操作系统承担硬件相关工作,使其上程序都是硬件无关的,是指当程序所使用的硬件发生变化(除 CPU 指令集以外)时,程序不必改变,人的操作更不必改变,即使是 CPU 指令集发生改变时,代码的改变也是最小的。

一个工作是应用无关的,是指不管用计算机来做什么,不管在计算机上运行什么程序,只要使用相应硬件或相应信息就要涉及到的工作,是用户共需的,且工作过程都是相同的,是有共性可循,却又与应用本身的问题没有直接关系的。

操作系统为用户操作和用户程序完成所有硬件相关和应用无关的工作,目的和益处是什么呢?硬件相关,必然意味着复杂繁琐、代码量很大(大到经常占代码比例份量的大部分)、代码不通用和变化大,需要用户投入大量的精力来设计实现和维护修改,以及了解相应的大量硬件细节知识,因此有必要统一管理,使用户摆脱负担。应用无关,就意味着更有必要统一管理(因