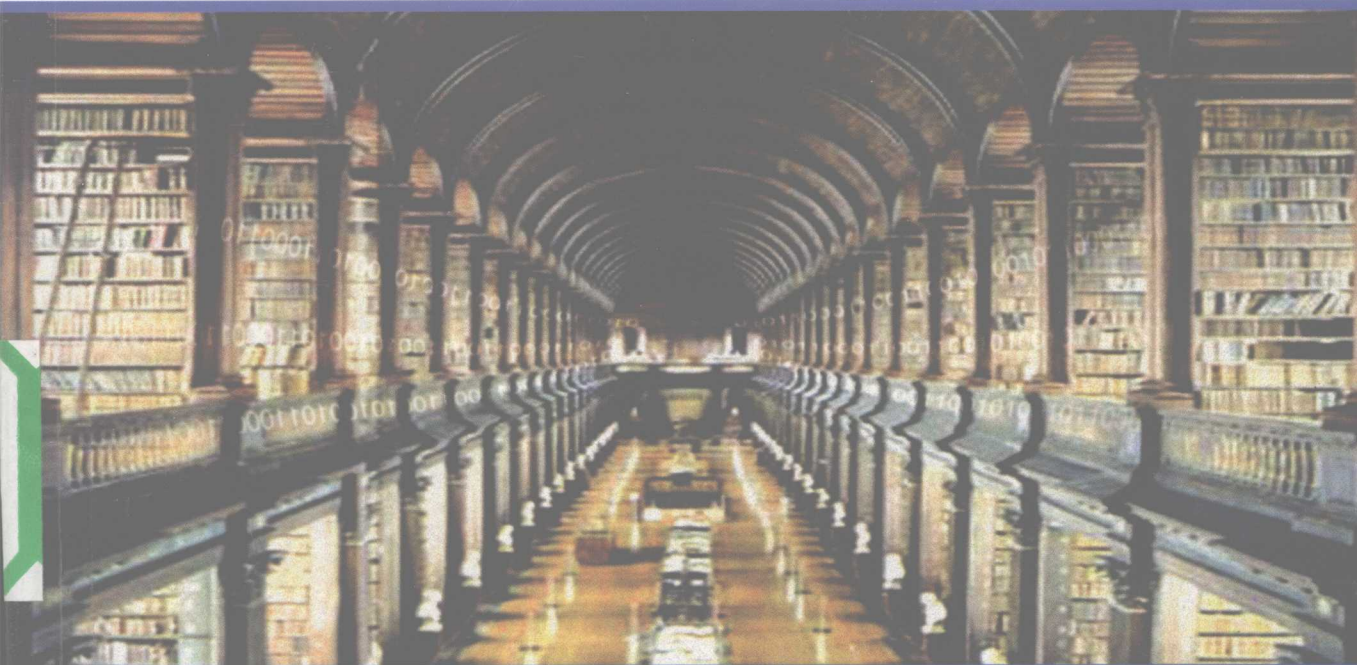


操作系统 原理与设计

Operating System

张红光 李福才 编著

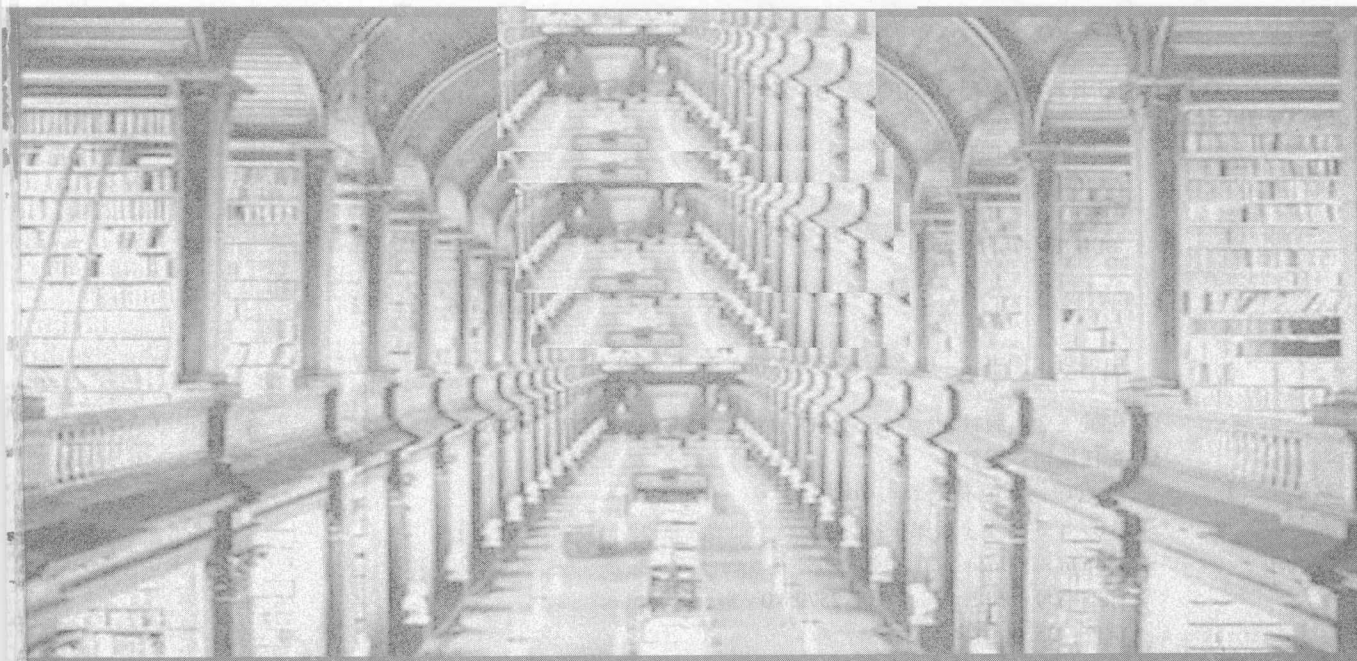


机械工业出版社
China Machine Press

高等院校计算机教材系列

操作系统 原理与设计 Operating System

张红光 李福才 编著



机械工业出版社
China Machine Press

本书以计算机专业 21 世纪教学改革为导向,以操作系统理论为依据,以当今主流操作系统实现技术为内容,全面介绍操作系统的基本理论和内核实现技术。全书共 10 章,主要介绍了计算机系统知识、操作系统基本理论、并行处理技术、存储管理技术、I/O 管理技术、操作系统安全知识等内容。每章后面都有本章小结及难度适宜的习题,便于读者自学或巩固所学知识。

本书适合作为高等院校计算机专业或相关专业操作系统课程的教材,也可以作为从事操作系统设计与系统内核开发的技术人员的参考书籍。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP)

操作系统原理与设计/张红光等编著. —北京:机械工业出版社,2009.4
(高等院校计算机教材系列)

ISBN 978-7-111-25795-0

I. 操… II. 张… III. 操作系统-高等学校-教材 IV. TP316

中国版本图书馆 CIP 数据核字 (2008) 第 197459 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:王春华

北京瑞德印刷有限公司印刷

2009 年 5 月第 1 版第 1 次印刷

184mm×260mm·21 印张

标准书号:ISBN 978-7-111-25795-0

定价:35.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010)68326294

前 言

近年来,随着计算机技术的高速发展,操作系统无论从内涵还是外部界面都发生了巨大的变化。这些变化正朝着两个不同的方向发展,一个是以微软等大型系统软件公司为代表的通用操作系统平台,其系统的用户界面更加友好,功能更加强大。这样带来的结果是操作系统更加繁复和庞大,系统内部结构更加复杂,人们对它的实现技术很难探索清楚。而另一个方面是随着手机等便携嵌入式系统的蓬勃发展,操作系统技术又朝着可剪裁、浓缩化和小型化发展。特别是开放式操作系统 Linux 技术的出现,给沉闷的操作系统技术研究和开发注入了新的活力,使更多的人有机会、有环境、有能力来学习、研究操作系统的核心与管理技术的精髓。

对操作系统这门课程的学习与研究大至可分为两个层面:面向一般用户的应用层面和面向系统设计人员的核心层面。由于不同层面关注问题的角度不同,所涉及的知识深度也不同。应用层面需要了解的是操作系统的各种执行特点以及应用知识,学习并掌握将操作系统提供的基本功能和特性融汇到实际应用的编程技术和设计方法。而核心层面是针对操作系统内核构造、并行调度管理机制、资源管理及 I/O 管理方法的原理和实现技术的描述。因此,对于操作系统的学习和研究,读者可以根据自己的实际情况有重点地学习有关知识和设计技术。

长期以来,关于操作系统方面的国内外教材和资料有很多,它们从不同的角度讲述了操作系统的原理、设计思想、实现技术以及应用方面的知识和技术。但是由于这门技术近些年来发展速度较快,人们在总结和消化新技术、新方法时总会有一定的延迟,使得操作系统中许多新的实用技术和设计思想在现在的本科教学中体现不足。这就促使我们思索和筹划将自己最新的研究成果和教学心得进行系统地总结,写一本易教易学、适合大学本科操作系统课程的教科书。经过努力,今天这个愿望终于得以实现。

我们采用较通俗的语言将我们长期从事嵌入式系统内核设计与研究及多年操作系统方面的教学经验和心得融入本书,佐以大量的编程分析案例帮助读者理解那些貌似抽象的概念和令人望而生畏的神秘内核技术。本书基本涵盖了操作系统设计原理的知识,主要包括计算机系统知识、操作系统基本理论、并行处理技术、存储管理技术、I/O 管理技术、操作系统安全知识等内容。全书共分 10 章,每章最后都附有本章小结、适量的习题等,便于读者自学和巩固所学知识。本书最后的附录详细介绍两个重要实际操作系统内核实现技术,即 Solaris 系统的内存管理技术分析和 Linux 系统的驱动程序开发技术,这些对了解实际操作系统设计方法和技术有很好的参考价值。

本书的主要特点是以操作系统理论为依据,以当今主流操作系统实现技术为内容,全面介绍操作系统的基本理论和内核实现技术。在写作上,我们尽量采用理论与实践相结合的方式,将枯燥的操作系统理论用实际系统的实现技术进行解释和分析,用大量编程范例帮助读者理解操作系统的理论和实现技术。本书重点阐述现代操作系统中使用的技术和知识,减略了传统操作系统原理教材中大量的基本原理介绍和算法推导过程,将重点放在理解现行操作系统所面临的问题、操作系统结构的变革以及如何实现 I/O 接口和系统资源的管理、如何实现多道并发环境中的管理机制等问题上。

本书适合作为各高等院校计算机及相关专业的本科教材或参考教材,也可以作为从事操作系统设计与系统内核开发的人员的参考书籍。阅读本书的读者最好已经掌握了计算机原理和 C

语言编程的基础知识，这样学习书中的实验例题会感到比较轻松。另外，由于本书中大部分例子都是以 UNIX 和 Windows 环境为例说明的，读者应该对 Windows 2000/XP 及 Linux 系统的使用环境有所了解，还应该阅读一些有关这两种系统的设计策略和实现技术方面的资料。

本书的编写得到了南开大学教材改革计划的支持和资助。李福才老师编写了本书的第 1 章、第 2 章和第 7 章，张红光老师编写了其他章节，并对全书做了编辑和统稿。直接参与本书资料收集、实例编写、验证及文本编写、编辑工作的有潘岳、李海丰、闫国光、齐国梁等；此外，南开大学学习操作系统课程的本科生以自己学习操作系统的心得、体会为本书提出了许多好的建议，其中有许多被我们采纳，在此一并对他们表示感谢。还要感谢机械工业出版社华章分社的编辑们，他们为本书的出版付出了辛勤的劳动并提出了宝贵的建议，在此也对他们表示诚挚的谢意。

由于种种原因，本书直至今日才得以脱稿。由于书中大量涉及自己的体会和理解，可能与此类教材的传统写作风格相悖，因此难免会有不到和谬误之处。诚恳希望专家和读者给予指正，我们将不胜感谢并安排在今后的版本中加以修改。

作 者

2008 年 12 月于南开园

教学建议

本书涵盖了操作系统设计原理和实现技术的大部分知识点,按照大类划分,包含了计算机系统知识、操作系统基本理论、并行处理技术、存储管理技术、I/O 管理技术、文件管理技术、分布式系统原理和操作系统安全知识等内容。

本书在授课时可安排 40~60 学时的课堂教学和 20~30 学时的实验课程。实验的内容可参照每章后面的练习题或者另行布置具有多方面知识要求的综合实验作业,若将实验建立在 Linux 环境下进行,推荐使用张红光、蒋跃军编写的由机械工业出版社出版的《UNIX 操作系统实验教程》作为实验参考教程。

本书的第 1 章和第 2 章是两个概述,第 1 章介绍了计算机系统的基本知识,第 2 章介绍了操作系统的基本概念。在教学中可将这两章内容作为操作系统知识学习的基本入门,每章至少需要 3 学时教学时间,重点讲述计算机系统的组成结构、处理器及各种硬件资源的特征、各类计算机软件的功能与作用、操作系统的基本分类以及计算机硬件与系统软件之间的关联关系。为提高学生的学习兴趣,还可以结合实际使用系统中的硬件环境和操作系统情况做具体说明,使学生对计算机硬件平台和操作系统设计建立起一个比较全面的认知体系。

第 3 章主要是介绍进程的概念。授课时应该从进程的基本概念谈起,然后介绍进程的派生机制、进程的状态、进程描述和进程控制等问题。通过这一章的内容主要使学生能够理解:操作系统是如何以进程为基础建立并行管理机制,如何将进程作为资源分配和处理器调度单位。这一章中有几个知识难点:一个是对原语和临界区的理解,另一个是对系统中并发进程调度的理解。为了提高教学成效,授课时除了进行课堂教学外,还应该让学生完成适量的进程生成和使用练习,使同学们可以亲身感受到进程的存在,并深入地理解系统对进程的调度管理机制。

第 4 章主要包含了两部分的内容:进程通信和处理器调度。授课时应从进程的同步与互斥机制中引出进程通信的现实意义,重点介绍基本进程通信机制和典型的 IPC 问题。在处理器调度管理问题中,重点介绍处理器分级调度的思想和常用调度算法的各种适应性,以及调度算法的评价指标等问题。

第 5 章主要介绍存储管理技术,这部分内容比较多,应本着由简到繁的思想展开教学。首先使学生理解存储管理中的重定位意义和主要的工作内容,然后重点说明操作系统中经常采用的基本存储管理技术,包括分区、分页、分段存储管理的实现方法以及它们的主要优缺点。最后要重点阐述虚拟存储技术,描述如何利用动态分配思想和主、辅存之间的交换技术来实现虚拟存储管理。教学中要考虑让学生建立起对存储管理的整体认识,把握各种分配与释放算法的特点,同时,对于置换算法中存在的弊端也要有清醒的认识。

第 6 章应重点讲授操作系统是如何通过进程和线程这两种并行单元来实现系统并发管理的,使学生认识到在有线程管理的系统中,进程模型已经发生了变化,因为其中要包含对线程的描述。另外,应说明线程与进程类似,也是一个可调度的实体。在教学中应结合实际系统,说明多线程管理机制可以解决许多当代计算机应用中的问题,而这些问题也是操作系统技术本身所要面临的新问题。

第 7 章是关于 I/O 管理的问题,教学中应结合 I/O 管理技术的一些特点,从概念和实践两方面展开教学。在概念方面应阐明 I/O 管理的基本概念,重点说明操作系统为了能够管理复杂

的I/O设备,会采用各种分类方法来简化I/O管理问题,同时还会尽量用统一的方式实现I/O控制。在实践方面,应该鼓励学生针对一种常见操作系统平台中的I/O管理模块进行解剖分析,对一到两个典型的I/O管理问题进行一次有益的设计尝试,使教学与实际问题形成较好的结合。

第8章是对文件管理技术的阐述,这一章中包含的知识点也比较多。建议从文件的使用方法和特点入手,展开对文件管理技术的深入描述。这里文件的概念、文件的存储结构、目录管理等问题会比较好理解,介绍时除了一般概念外,还可以结合一些系统实例进行描述,使教学内容更加生动。关于磁盘文件系统的建立和文件并发访问控制问题是本章中的技术难点,可根据学生掌握情况进行适量的扩展或删减。对于文件管理的实现方法,可以结合UNIX的有关系统调用的使用和实现分析加以阐述。

在第9章中,应在介绍网络技术的基础上阐述分布式系统概念。分布式系统概念是本章教学的重点内容,而关于分布式操作系统的实现技术可根据 ([具体情况]进行适当描述。

第10章是关于操作系统安全知识的介绍,在教学中可以采用一些比较灵活的方法,让大家认识到操作系统安全的重要性,并能够有意识地主动防范各种不安全因素的侵扰。

归纳起来,在本书中除了前两章的概述内容以外,后面的8章内容是对操作系统原理知识的分题讲述,这里进程概念及进程通信、存储管理、I/O技术、文件管理等是需要重点介绍的核心内容。第10章操作系统安全知识,在现代操作系统中也比较重要,应该作为一个3学时的 ([教学内容] ([进行,关于线程技术和分布式系统两章中的内容,由于其中需要的知识层次有一定的深度和难度,教学时可根据学生掌握的情况,对内容做适量的删减或扩展,以增加教学效果。

目 录

前 言
教学建议

第1章 计算机概述	1
1.1 计算机硬件	1
1.1.1 处理器	1
1.1.2 存储器	5
1.1.3 I/O 设备	8
1.1.4 时钟部件	8
1.1.5 计算机总线	10
1.1.6 各功能部件组织结构	11
1.2 计算机软件	12
1.2.1 固化软件	12
1.2.2 系统软件	13
1.2.3 工具软件	13
1.2.4 应用软件	14
1.3 机器指令与程序执行	14
1.3.1 指令集	14
1.3.2 指令执行与指令周期	14
1.4 中断机制	15
1.4.1 中断的作用	16
1.4.2 中断查询机制	16
1.4.3 中断管理程序	17
1.4.4 中断类型及中断处理	18
1.4.5 多中断处理技术	19
1.5 高速缓存技术	20
1.5.1 高速缓存在系统中的作用	20
1.5.2 具有高速缓存的主存储器访问机制	21
1.6 I/O 访问方式	21
1.7 本章小结	22
习题	23

第2章 操作系统概述	25
2.1 操作系统的作用与功能	25
2.1.1 用户对操作系统的需求	25

2.1.2 操作系统的作用	26
2.1.3 操作系统的功能	27
2.2 操作系统的发展历程	28
2.3 操作系统分类	29
2.3.1 批处理操作系统	29
2.3.2 分时操作系统	32
2.3.3 实时操作系统	32
2.3.4 多处理器的操作系统	33
2.3.5 网络操作系统	34
2.3.6 分布式操作系统	35
2.3.7 个人计算机操作系统	36
2.3.8 嵌入式操作系统	36
2.4 操作系统设计	36
2.4.1 操作系统设计难点	37
2.4.2 软件工程思想的应用	38
2.5 操作系统中的核心技术	38
2.5.1 并行管理技术	38
2.5.2 存储管理技术	39
2.5.3 文件与 I/O 管理技术	40
2.5.4 调度算法与信息安全控制	40
2.6 操作系统体系结构	41
2.6.1 无结构系统	41
2.6.2 层次结构	42
2.6.3 虚拟机结构	42
2.6.4 微内核结构	43
2.7 典型操作系统	44
2.7.1 MS-DOS	44
2.7.2 Microsoft Windows	45
2.7.3 UNIX 操作系统	46
2.8 本章小结	48
习题	49

第3章 进程与进程管理	50
3.1 什么是进程	50
3.1.1 多道环境中的程序执行	50
3.1.2 进程的定义	52

3.1.3 进程的特性	52
3.1.4 进程与程序的区别	53
3.2 进程的派生机制	53
3.3 进程的状态	54
3.3.1 进程执行	54
3.3.2 两状态进程	56
3.3.3 5 状态进程	56
3.3.4 挂起进程模型	59
3.4 进程描述	60
3.4.1 进程描述内容	60
3.4.2 进程控制块	61
3.4.3 进程映像	62
3.5 进程控制	63
3.5.1 进程管理	63
3.5.2 进程控制操作	64
3.6 本章小结	66
习题	67

第4章 进程通信及处理器调度..... 69

4.1 程序顺序执行与进程并发	69
4.1.1 程序顺序执行	69
4.1.2 进程并发条件	70
4.1.3 构造进程并发环境	71
4.2 进程的同步与互斥	72
4.2.1 进程交互方式	73
4.2.2 进程互斥的实现	73
4.2.3 进程同步问题	79
4.2.4 用信号量管理进程的同步与 互斥	82
4.2.5 管程	83
4.3 进程通信机制	85
4.3.1 消息传递通信	85
4.3.2 消息通信应用	87
4.3.3 经典 IPC 问题: 读者-写者问题 ...	88
4.3.4 经典 IPC 问题: 哲学家就餐问题 ...	90
4.4 处理器调度	92
4.4.1 调度中应关注的问题	92
4.4.2 调度方式与操作系统分类	93
4.4.3 处理器分级调度	93
4.4.4 调度的衡量标准	94
4.4.5 处理器调度算法	96
4.4.6 调度算法的性能分析	101

4.5 UNIX 进程调度分析	102
4.5.1 调度时机安排	102
4.5.2 调度标志设置	102
4.5.3 进程调度策略及优先数的 计算	102
4.5.4 进程调度实现	103
4.6 本章小结	103
习题	104

第5章 存储管理..... 106

5.1 计算机存储结构	106
5.1.1 存储器配置方式	106
5.1.2 常见 PC 机存储结构	106
5.2 地址重定位及内存访问保护	107
5.2.1 地址空间	107
5.2.2 地址重定位	107
5.2.3 地址重定位及存储信息保护	108
5.3 存储管理中的分配技术	109
5.3.1 进程交换的意义	109
5.3.2 用位示图法控制存储分配	110
5.3.3 用链表法实现内存分配管理	111
5.4 分区存储管理技术	111
5.4.1 单一分区内存管理	111
5.4.2 固定大小的多分区管理	111
5.4.3 动态分区管理	113
5.5 分区分配算法	115
5.5.1 分区分配算法描述	115
5.5.2 分配算法使用特性	116
5.6 页式管理	117
5.6.1 分页的基本思想	117
5.6.2 静态页式管理	117
5.6.3 动态页式管理	120
5.7 段式管理	121
5.7.1 段式管理基本原理	121
5.7.2 段式管理的地址变换机制	122
5.8 段页式存储管理	123
5.8.1 分页与分段管理的特点	123
5.8.2 段页式管理方式	124
5.9 虚拟存储技术	124
5.9.1 局部性原理	124

5.9.2 虚拟存储的基础	125	7.2.2 程序控制 I/O 方式	164
5.9.3 用分页管理实现虚拟存储	126	7.2.3 中断驱动 I/O 方式	165
5.9.4 虚拟存储页面置换算法	127	7.2.4 直接存储器 I/O 访问方式	166
5.10 本章小结	131	7.2.5 I/O 软件变迁过程	166
习题	131	7.3 I/O 管理子系统设计	167
第6章 线程管理	133	7.3.1 I/O 管理子系统设计特点	167
6.1 线程基本概念	133	7.3.2 I/O 管理子系统结构	168
6.1.1 线程定义	133	7.3.3 中断处理程序设计	168
6.1.2 进程与线程的分工	134	7.3.4 设备驱动程序设计	170
6.2 基于线程并发的优势	135	7.3.5 I/O 管理中的设备无关性 设计	171
6.2.1 线程切换优势	135	7.3.6 I/O 缓冲技术	173
6.2.2 操作系统对线程支持的 综合优势	136	7.4 用户层 I/O 软件设计	175
6.3 包含线程的进程描述模型	137	7.4.1 建立 I/O 访问库	176
6.4 线程管理实现机制	138	7.4.2 虚拟设备管理	176
6.4.1 线程执行状态	138	7.5 磁盘管理技术	177
6.4.2 用户级线程管理模式	139	7.5.1 磁盘性能参数	177
6.4.3 核心级线程管理模式	140	7.5.2 磁盘调度策略	178
6.4.4 混合型线程管理模式	140	7.5.3 RAID 技术	179
6.5 适合多线程的应用	141	7.5.4 磁盘格式化问题	182
6.5.1 用多线程解决实际问题	141	7.6 时钟管理	184
6.5.2 线程控制语句	143	7.6.1 时钟硬件的组成	184
6.5.3 关于多线程标准库	145	7.6.2 时钟软件功能	185
6.5.4 多线程编程规则	146	7.7 字符终端管理	185
6.6 多线程程序设计实例	148	7.7.1 字符终端接口	185
6.6.1 用多线程提高程序执行 效率	148	7.7.2 字符设备输入软件	186
6.6.2 用线程实现数据库查询	150	7.7.3 字符设备输出软件	187
6.6.3 用多线程完成文件复制	151	7.8 I/O 子系统设计实例	187
6.7 本章小结	154	7.8.1 Windows 2000/XP 的 I/O 子系统	187
习题	155	7.8.2 Linux 的 I/O 子系统	190
第7章 I/O 技术与设备管理	156	7.9 本章小结	193
7.1 I/O 设备的硬件	156	习题	194
7.1.1 I/O 设备分类	157	第8章 文件管理	196
7.1.2 设备控制器	157	8.1 文件管理概述	196
7.1.3 I/O 端口的描述与访问	158	8.1.1 信息描述单元	196
7.1.4 DMA 访问机制	160	8.1.2 文件系统的作用	197
7.1.5 I/O 中断	161	8.1.3 文件命名规则	197
7.2 I/O 软件设计	163	8.1.4 文件分类	198
7.2.1 I/O 软件功能	163	8.2 文件存储结构	198
		8.3 目录管理	200

8.3.1 文件控制块	200	9.2 网络操作系统	243
8.3.2 目录文件	201	9.2.1 网络操作系统模型	243
8.3.3 文件目录结构	203	9.2.2 操作系统对 Internet 的支持 ...	244
8.4 磁盘文件系统	206	9.3 分布式系统概念	246
8.4.1 磁盘分区与格式化	206	9.3.1 分布式基础	246
8.4.2 文件存盘管理	208	9.3.2 分布式操作系统	247
8.4.3 目录存储策略	211	9.3.3 分布式程序设计语言	248
8.4.4 多级目录中文件共享	212	9.3.4 分布式文件和数据库系统	248
8.4.5 文件系统构造	215	9.4 分布式操作系统技术	249
8.5 用于文件管理的系统调用	216	9.4.1 分布式进程通信	249
8.5.1 系统调用功能	216	9.4.2 分布式资源管理	252
8.5.2 UNIX 文件描述符	216	9.4.3 分布式文件系统	254
8.5.3 文件创建和文件链接	217	9.5 本章小结	258
8.5.4 文件打开及文件关闭	219	习题	258
8.5.5 对文件的读/写操作	220		
8.5.6 改变文件访问指针位置	221	第 10 章 操作系统的安全性	260
8.5.7 捕获文件当前指针位置	223	10.1 系统安全基本知识	260
8.6 文件并发访问控制	224	10.1.1 对系统造成威胁的人	260
8.6.1 文件并发访问机制	224	10.1.2 安全管理的目标	261
8.6.2 文件记录加锁技术	224	10.1.3 操作系统安全原则	262
8.6.3 UNIX 文件记录加锁方法	225	10.1.4 系统级安全措施功能	263
8.7 文件系统实例: UNIX 文件系统 ...	226	10.2 数字加密技术	264
8.7.1 UNIX 的文件与目录	226	10.2.1 数字加密	264
8.7.2 安装与卸载文件系统	227	10.2.2 数字签名	266
8.7.3 UNIX 文件系统组成	228	10.3 用户身份验证	267
8.7.4 UNIX 文件存储策略	228	10.3.1 用户口令验证	267
8.7.5 文件访问动态管理机制	229	10.3.2 持有物信息验证	269
8.7.6 UNIX 虚拟文件系统	230	10.3.3 人体生物识别	269
8.8 文件系统实例: Windows 文件 系统	232	10.4 计算机病毒攻击与防范	270
8.8.1 磁盘访问特点	232	10.4.1 植入系统内部的危害	270
8.8.2 建立磁盘主引导区	232	10.4.2 来自系统外部的危害	274
8.8.3 磁盘分区	233	10.4.3 病毒引发分析	274
8.8.4 NTFS 文件系统	234	10.4.4 病毒传播	276
8.9 本章小结	236	10.4.5 系统安全与反入侵技术	276
习题	237	10.5 本章小结	279
		习题	280
第 9 章 网络与分布式操作系统	239	附录 A Solaris 内存分页管理技术 分析	281
9.1 计算机网络	239		
9.1.1 网络分类	239	附录 B Linux 设备驱动程序 设计	306
9.1.2 网络协议	240		
9.1.3 网络结构	240		

第1章 计算机概述

操作系统知识是计算机系统最重要的基础知识之一，要学习操作系统的理论与实现技术首先要了解计算机的相关知识。传统计算机技术分类概括地将计算机系统分为两大部分，即计算机硬件和计算机软件。操作系统属于计算机软件部分中的系统软件，它的主要任务之一是通过计算机对计算机系统拥有的硬件和软件资源的管理和调度，为计算机用户提供一整套便捷、易用的系统服务程序。由此可见，操作系统与计算机的硬件和软件都有着不可分隔的联系。在学习操作系统理论与知识之前，首先了解一些计算机组成原理和系统结构方面的知识是非常必要和十分有益的。

由于篇幅和侧重点不同，本章不详细阐述计算机组成原理和计算机系统结构的概念，只对与操作系统有关的计算机系统硬件知识和计算机基本系统结构进行介绍。如果需要详细了解和深入学习完整的计算机组成原理和计算机系统结构的有关知识，还需要学习计算机原理的有关教程。

1.1 计算机硬件

在一个计算机系统中包含着硬件、固化软件、系统软件、应用软件等项内容，而计算机硬件是系统的基础。概括地讲，计算机硬件由五大部分组成：运算器、控制器、存储器、输入及输出设备。计算机硬件部分搭建并保证了软件运行和数据存储的平台。要了解计算机硬件的组成，首先需要学习一系列的计算机系统概念和知识，这些概念包括：处理器、存储器、地址空间、指令结构、程序控制、指令周期，以及寄存器使用方法、I/O 访问技术等。下面我们介绍一些主要的计算机硬件知识。

1.1.1 处理器

处理器通常称为 CPU (Central Processing Unit, 中央处理器)。这种叫法暗示该部件是整个计算机系统的中央处理机，计算机所完成的主要操作都需要 CPU 进行控制、调度和处理。因为在传统计算机中只有一个 CPU，它是计算机系统的核心。但随着现代计算机技术的发展，计算机组成结构已发生了改变，特别是在提倡更高、更快的计算处理技术的今天，并行处理、超前处理、流水线技术等已经打破了单一中央处理器完成所有计算机操作的局面。今天的计算机中可能包含多个处理器，它们之间有的存在主从关系，有的完全是平等的关系。今天的计算机所完成的操作并不一定是仅靠一个处理器完成指令要求的数据控制和处理，而可能是由多个处理器分工合作而完成。例如，指令的预处理、数据的后处理、I/O 操作的专门处理可能是由不同的处理器分别来完成，它们各自完成的操作和它们之间的相关操作在时间上完全可以通过严格而精细的管理实现并行操作。因此，严格地说，我们称其为计算机“处理器”会更合适一些。

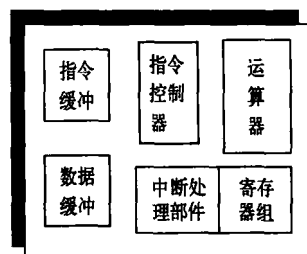


图 1-1 处理器中的主要部件

在学习计算机原理时我们已经知道，处理器的主要功能是进行指令要求的数字运算、逻辑运算和各种数据处理，在其内部设立的功能部件又可以分为控制单元、算术逻辑单元和存储单元等几大部分。一个处理器中所包含的主要部件如图 1-1 所示。

处理器按照其可处理信息的字长分为 8 位处理器、16 位处理器、32 位处理器以及 64 位处理器等；也可以按照其完成功能分为运算处理器、图像/图形处理器、数据管理处理器、I/O 处理器、网络管理处理器等。

1. 运算器

运算器是处理器中的一个主要部件，通常叫做“算术逻辑单元(ALU)”，它的主要任务是完成计算机软件程序指令要求的各种运算操作。由于在程序指令运行时，大多数的操作都需要进行各种各样的数据操作和运算，有些应用甚至是非常独特的，因此，市面上存在多种类型的处理器。考量运算器的最重要的指标是它的运算速度，其次是它的单位时间指令处理能力。平时我们所说的 300MHz CPU、500MHz CPU 等，其含义是指 CPU 中运算器的速度和单位时间内指令完成效率。运算器的基本结构如图 1-2 所示。

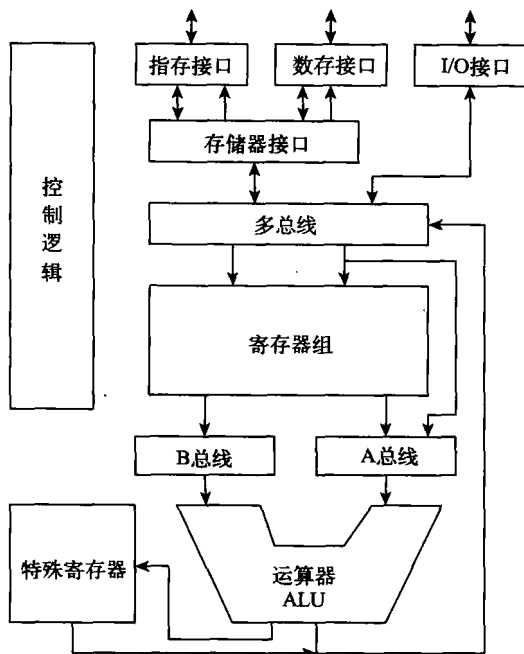


图 1-2 基本运算器结构图

从图 1-2 中我们可以看出，基本运算器是由运算命令控制逻辑、算术运算单元、运算数据传输通道、数据寄存器组、特殊寄存器、各种外部存储部件接口（如图中的指存接口、数存接口）及 I/O 接口组成。这些功能部件组成了一个功能完善、可高效完成各种运算操作的指令运算及数据处理部件——ALU。

2. 控制器

传统概念上控制器就是处理器中的控制单元，是处理器的控制中枢部件。该部件的主要作用是协调和管理处理器中各大部件间的工作过程，它就像公司的总经理或军队的指挥官，接收来自上级的各种命令，指挥各下属高效、协调地完成任务。在 CPU 中，软件与硬件的交互是由控制器完成的。控制器接受来自软件程序的指令（命令），对这些指令进行分解、译码并进行一系列的命令排序，将其下发给 CPU 的各个执行部件，指挥和命令它们完成此条指令要求的各个命令和动作。

例如，控制器接收到一条要求两个寄存器内容相加的指令“ADD DR, SR;”其中 ADD 是算术加法指令的助记符，DR 是指目的寄存器编号，SR 是指数源寄存器编号。指令要求的操作是

将 DR 中存的数取到运算器，再将 SR 中存的数也取到运算器，对这两个数做算数加法，结果存到 DR 中，SR 内容不变。运算结果改变系统状态字 CF(进位标志)、ZF(结果为全“0”标志)、SF(结果为负数标志)、OVF(结果为有溢出标志)。因此，控制器的基本结构如图 1-3 所示。

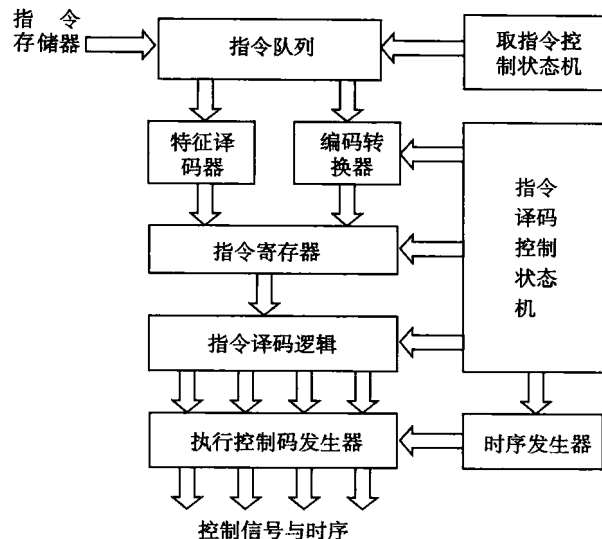


图 1-3 基本指令控制器结构图

下面我们结合图 1-3 来看看控制器是如何执行上面那条“ADD DR, SR”指令的。

- 1) 首先取指令控制状态机发出取指命令，从指令队列中取出这条“ADD DR, SR”指令。
- 2) 指令译码控制状态机发出命令，将这条指令的不同字段分别打入特征译码器和指令编码转换器中。
- 3) 指令译码逻辑器分析和译出该指令的意义。
- 4) 指令译码状态机发出一系列的命令与时序，与译出的指令编码组合成一系列的控制命令，并将这些命令发送到运算器。
- 5) 命令运算器首先从 DR 寄存器中取出目的操作数(我们常称为操作数 AX)，从 SR 寄存器中取出数源操作数(我们常称为操作数 BX)，将两个操作数送到运算器的数据输入端。
- 6) 打开运算器算数运算通道，进行加法运算。
- 7) 再将运算结果送到 DR 中，同时根据运算结果，设置状态字 FLAG。

至此完成此条指令要求的所有操作。从上述“ADD DR, SR”指令的操作中可以看出，指令的解译和指令的执行由两大部件组成。对于指令的解译，通常叫做指令控制器，也就是传统概念的计算机的基本控制器。

3. 处理器运行状态

随着系统软件设计复杂度的增加，对处理器设计本身也提出了一些新要求，而为处理器运行设定不同的状态(核心态与用户态)就可以为操作系统管理提供有利的基础保障。

对微处理器系统来说，自 Intel 386 以后处理器中就实现了 4 个特权级模式，其中特权级 0(Ring 0)是留给操作系统、设备驱动程序使用的。由于在操作系统和驱动程序的代码设计中要使用一些系统内部的指令子集，而这些指令子集的操作对系统的影响非常大，因此不宜允许一般用户访问，而只允许特殊需要的程序执行，这样就将它们定义为只允许在系统核心态下运行的程序。而最低特权级 3(Ring 3)是留给普通用户程序使用的，规定用户程序只允许在这个级别上运行。这种用户程序只允许在用户态下运行的机制，可以有效地保护系统程序正常运行，提

高系统的可靠性和安全性。处理器特权级模式在系统中如何应用,是由操作系统决定的,例如,Windows 系统就只用到了两级特权模式,其他特权模式并没有实际意义。

在有特权级别管理的处理器中,运行于核心态的代码基本上不会受到系统的任何限制,它们被允许访问任何有效地址、对端口直接读写。而运行于用户态的代码则需要受到处理器内部的多项检查,例如,系统规定用户态程序只能访问映射到其地址空间的指定地址内容,而且对于端口的访问也非常有限,如只能对任务状态段(TSS)中 I/O 许可位图(I/O Permission Bitmap)中规定的可访问端口进行直接访问。这些规定的实现可以通过将处理器状态和控制标志寄存器 EFLAGS 中的 IOPL 位标记为 0 来表示,有这些标志时,说明当前可以进行直接 I/O 访问的最低特权级别是 Ring0。

上述这些功能都只限于保护模式的操作系统,对于实模式的操作系统(如 DOS 系统)无效,因为在实模式操作系统中没有实现这些系统保护机制,系统中的所有代码都被看做运行在一种状态(即核心态)下。处理器模式转换(如从 Ring3 向 Ring0 的切换)机制需要在操作系统中给出规定,通常是在控制权转移时进行。例如,可以规定在以下两种情况下发生模式转换:当调用长转移指令 CALL 时或者碰到了中断指令或自陷指令 INT 时,就进行处理器的模式转换;也可以规定只在遇到中断指令时转换处理器运行模式,而在执行自陷指令时不进行模式转换等。由于涉及复杂的安全保护检查和堆栈切换机制,具体的处理器模式转换实现细节在此不赘述,有兴趣的读者请参阅相关资料学习。

在现代操作系统中,通常使用中断指令来提供系统服务,通过执行一条陷入指令来完成模式切换,这种指令在 Intel X86 上是 INT,在 Win9X 下是 INT30(保护模式回调),在 Linux 下是 INT80,在 WinNT/2000 下是 INT2E。用户模式的服务程序(如系统 DLL)通过执行一个 INTXX 来请求系统服务,然后处理器模式将切换到核心态,工作于核心态的对应系统代码将完成此次请求服务,并将执行结果传递给用户程序。

4. 处理器中的寄存器

寄存器是处理器内部的存储功能部件,不同的处理器提供的寄存器个数会有很大差异。寄存器主要用来暂存指令在处理器执行过程中需要的临时数据、访问地址以及指令信息本身。在系统程序设计中有可能需要访问寄存器,通过访问寄存器可以达到两种效果:

1) 提高指令执行的速度和效率。通过对用户可见寄存器的合理使用,可以减少指令对主存储器的访问次数,这样就可以加快指令的执行速度。在编译程序的优化代码部分包含许多分配寄存器的技术,在有些高级语言(如 C 语言)中,还允许程序员对寄存器的使用提出自己需要的方案。

2) 控制处理器的操作和状态。系统中大部分寄存器是归处理器使用的,它们由控制器自动传递、自动判别,来实现对处理器的控制。同时有些寄存器也可以通过特权指令访问,例如,通过操作系统中的某些程序可以控制用户程序的执行过程。寄存器按照完成的功能不同,可划分出多种类型,计算机系统中常见的寄存器类型有:

- 数据寄存器(data register)。这类寄存器可以由程序员指派其中内容。
- 地址寄存器(address register)。这类寄存器可以是指向数据或指令的地址,也可以是指向一个地址的指针,例如,变址寄存器(index register)、段指针(segment pointer)、栈指针(stack pointer)。
- 条件码或标志位寄存器。这类寄存器一般允许程序访问,但不允许程序对其修改,它所包含的内容通常是根椐操作结果由处理器硬件完成设置的。
- 程序计数器(Program Counter, PC)。它用来存储处理器下一条将要执行指令的地址。
- 指令寄存器(Instruction Register, IR)。它用来存储处理器将要执行的下一条指令的内容。

- 程序状态字(Program Status Word, PSW)。它用来存储指令执行结果表现的各种状态。
- 中断寄存器(Interrupt Register)。它用来保存与中断管理有关的信息和状态字。
- 用于存储管理的寄存器。它是指处理器与存储器交互中用到的寄存器。
- 用于 I/O 控制的寄存器。它是指处理器与 I/O 控制模块交互中使用到的寄存器。

1.1.2 存储器

存储器(memory)是计算机的记忆设备,主要用来存放程序和数据。能够在计算机中使用的所有信息(包括原始数据、计算机程序、中间运行结果和最终运行结果、图形图像等)都被保存在存储器中。关于存储器,我们从以下几个方面进行介绍。

1. 存储器的分类

计算机中的存储器有多种,对存储器的分类方法也有很多,下面按常见的分类方法对存储器进行分类。

1)按存储介质划分,存储器分为:

- 半导体存储器——用半导体器件组成的存储器。
- 磁介质存储器——用磁性材料做成的存储器。
- 光存储器——例如,常见的 VCD、DVD 等。

2)按存储方式划分,存储器分为:

- 随机存储器——指存储器中的任何存储单元的内容都可以被随机存取,而且存取时间与存储单元的物理位置无关。
- 顺序存储器——只能按某种顺序访问存储器,存取时间与存储单元的物理位置有关。

3)按存储器的读写功能划分,存储器分为:

- 只读存储器(ROM)——指存储器中的存储内容是固定不变的,对存储器只能进行读操作而不能进行写操作的半导体存储器。
- 随机读写存储器(RAM)——指那些既能进行读操作又能进行写操作的半导体存储器。

4)按信息的可保存性划分,存储器分为:

- 非永久记忆的存储器——指断电后信息即消失的存储器。
- 永久记忆性存储器——指断电后仍能保存信息的存储器。

5)按存储器的功能划分,存储器分为:

- 主存储器——是指可以实现以电子速度运行的高速内存,主存储器是以半导体单元构成的存储介质,因此习惯上也将其称为“半导体存储器”。主存储器中的每个存储单元可以存储一个二进制信息,计算机中所有需要运行的程序和指令都必须首先装入主存储器中,然后才可以运行。主存储器中的各个单元内容很少被单独地读取或直接用来编辑,它们一般是按照固定大小的单元组进行操作,因此系统需要对内存单元组进行约定和定义。目前流行的单元组定义单位就是我们平时所说的“字”或“字节”。内存中的每个字都被分配了一个特定地址与之对应,通过该地址可以访问到内存该地址中保存的内容。另外,存储数据字的位数被称为计算机的字长。字长随机器的类型不同而不同,典型的字长包括:8位、16位、32位、64位。一个计算机主存储器的大小是衡量计算机存储规模的一个重要指标,例如,可以是 * * * K、* * * M 或 * G 等级别,这些级别分别称为 K 级、兆级、G 级。
- 辅助存储器——是指除了主存储器以外的多种存储器,它们通常由磁性介质或光材料构成,其主要特点是可以存储大批量的程序和数据信息。虽然主存储器的访问效率很高,但造价也非常昂贵,同时由于主存储器在掉电后其中的信息将会丢失,因此在系统中只

有主存储器是不够的。存放在计算机系统中的大批量信息(包括数据和程序)并不是总被处理器访问着(实际上系统运行中正在使用的信息非常有限),这些信息没有必要占用主存储器,而是被存放在比较便宜的辅助存储介质中。目前使用的辅助存储器种类很多,例如,磁盘、磁带、CD-ROM 和 U 盘等。

- 高速缓存器——也称为 Cache,这种存储器是由特殊硬件电路实现的。这些电路主要完成小容量、高速存储器的功能,用来匹配寄存器与主存储器间速度差异比较大的问题,可以实现主存储器与寄存器之间的分级数据移动操作。有关高速缓存的作用和管理特性在本章后面有专门描述。
- 寄存器——是处理器单元中包含的存储部件,它们也是由特殊电路构成的存储单元,与处理器的速度相匹配,可以实现计算机系统中最快的数据读写操作。

第 5 种分类方式比较常见,而且用户接触最多的是主存储器和辅助存储器,其他两种是建立在系统内部机制中的,一般用户不直接对其进行访问。

通常保存在主存储器中的信息可以依据控制器指定的位置完成写入和读出操作,而在辅助存储器中的信息通常按照文件名进行访问。存储器是计算机系统中必不可少的部件,而对存储器的管理也是操作系统的一项重要工作,但是在专业术语中将主存储器的分配与回收管理称为存储器管理,而对辅助存储器的管理被划归为文件或 I/O 设备的管理之中。

2. 不同介质的存储器特点

构成主存储器和辅助存储器的介质主要以半导体器件和磁性材料为主。存储器中最小的存储单位就是一个双稳态半导体电路或者一个 CMOS 晶体管或磁性材料的存储元,最小存储单位可以存储一个二进制代码。由若干个存储元组成一个存储单元(每个存储单元可以存放一个字节),然后再由许多存储单元组成一个存储器。

为了便于访问,存储器中的每个存储单元的位置都有一个编号,这个编号就是存储地址,一般用十六进制码表示。一个存储器中所有存储单元可存放数据的总和称为该存储器的**存储容量**。例如,一个存储器的地址码由 20 位二进制数(即 5 位十六进制数)组成,那么它可以表示 1M(1 048 576)个存储单元地址;如果一个存储单元中存放一个字节,则该存储器的容量就是 1M 字节的容量。

不同介质的存储器会表现出不同的存储特征,无论是在存储容量上还是在访问速度上都存在着很大差异。计算机系统一般都需要多种类型的存储器来构成一个合理的存储体系结构。这里我们来看看不同介质的存储器所表现出来的如下特征:

1) 一般来讲,存储速度越快的存储器,其每位(bit)的价格就越高。例如,寄存器是系统中存储速度最快的存储器,它的价格是比较昂贵的,所以寄存器的配置过多的话,就会使系统硬件成本增加。

2) 存储容量比较大的存储器与存储容量较小的存储器相比,其每位(bit)的价格会逐级降低。也就是说,可保存大批量信息的存储器,其价格都相对比较便宜。例如,寄存器的价格是最高的,高速缓存(Cache)相对便宜些,而内存会比 Cache 更便宜,辅助存储器与前几种相比按位计算是最便宜的。

3) 当存储器中的容量越大时,它的访问速度就会越慢。这是由不同存储器的组成介质决定的。在系统中大家都有明显的感觉,当访问内存时要比访问外存快,而访问寄存器时又一定会比访问内存的速度快许多,等等。

不同介质的存储器具有不同的特性,它们在完成存储时所起的作用也不一样,因此不同的存储器会被安排在不同的存储结构层面上,发挥不同的特长,满足系统存储的需求。