

国家精品课程 / 国家精品资源共享课程配套教材



“十二五”普通高等教育本科国家级规划教材

计算机 操作系统

第3版 | 微课版

Computer Operating System (3rd Edition)

庞丽萍 阳富民 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



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

国家精品课程 / 国家精品资源共享课程配套教材



“十二五”普通高等教育本科国家级规划教材

计算机 操作系统

第3版 | 微课版

Computer Operating System (3rd Edition)

庞丽萍 阳富民 编著



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：微课版 / 庞丽萍, 阳富民编著
: — 3版. — 北京: 人民邮电出版社, 2018. 1(2018. 6重印)
ISBN 978-7-115-46069-1

I. ①计… II. ①庞… ②阳… III. ①操作系统—教材 IV. ①TP316

中国版本图书馆CIP数据核字(2017)第139838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。书中讨论了现代操作系统采用的并行处理技术和虚拟技术, 且以 UNIX 和 Linux 系统为实例, 剖析了其特点和具体的实现技术。

为了便于读者的学习和理解, 本书针对重难点内容录制了微课视频, 读者可通过扫描二维码进行观看。

本书既可作为高等院校计算机和信息类相关专业教材, 也可供从事计算机科学、工程和应用等方面工作的科技人员参考。

-
- ◆ 编 著 庞丽萍 阳富民
责任编辑 邹文波
责任印制 陈 犇
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 21.5 2018年1月第3版
字数: 549千字 2018年6月北京第2次印刷
-

定价: 49.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号



人民邮电

第3版前言 PREFACE

自《计算机操作系统（第2版）》一书出版以来，我们又经历了几轮教学实践，对操作系统的教学内容、实践环节不断地进行研究和探讨。我们考虑在《计算机操作系统（第2版）》的基础上再增加实例操作系统的内容和若干习题，针对教学中的重点和难点内容录制微课视频，以促进教师的教学与学生自学，期望收到更好的教学效果。为此，有必要对《计算机操作系统（第2版）》进行修订再版。

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。操作系统具有如下特点。

- (1) 内容庞杂、涉及面广。操作系统是计算机系统的核心管理软件，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。所有的计算机都必须配置操作系统，各种类型的操作系统都在运转，为用户提供良好的服务。
- (4) 技术发展快。操作系统的实现技术和方法在不断地进步与完善。

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、进程控制与进程调度功能）及方法。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。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。

我们认为，在操作系统原理教学中应让学生更多地了解实际操作系统的实现技术，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。本书既保留了

当前流行的实例操作系统 Linux 的内容,剖析其特点和实现技术,又增加了 UNIX 系统的相关内容。因为 UNIX 系统是一个经典的操作系统,其设计思路有创新,技术实现高效,代码简洁清晰。UNIX 系统为后来的多个操作系统树立了典范。本书增加的内容包括 UNIX 系统结构,UNIX 系统功能调用,UNIX 系统的进程管理、进程调度、存储管理、设备管理和文件系统。

此次再版,在第 2 版教材的基础上做了如下调整。

- (1) 针对操作系统教学中的重点与难点内容,录制了微课视频。
- (2) 增加了 UNIX 系统的内容,并将第 2 版第 9 章 Linux 系统的内容分解到对应的各个章节,这样每章最后两节是 UNIX 系统和 Linux 系统的相关内容(第 5 章资源分配与调度除外),作为理论学习后的案例剖析,便于加深读者对理论知识的认识与理解。
- (3) 增加第 6 章处理机调度。将第 2 版教材第 4 章的“进程调度”一节移至第 6 章,另外增加处理机的多级调度的有关内容。
- (4) 部分章增加了若干习题。

本书仍然保持深入浅出、通俗易懂的特点,使读者便于阅读和理解。

在本书的编写过程中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 1~9 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,应选择 1~9 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。

我们在教学和编写教材过程中,学习、参考了有关操作系统、UNIX、Linux 系统方面的优秀教材,这些书都给了我们很大的帮助,不断地学习使我们加深了对操作系统的理解。在本书出版之际,我们要感谢指导、帮助过我们的专家、作者、老师和朋友,和他们的讨论、交流使我们受益匪浅。

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

庞丽萍 阳富民

2017 年 5 月于武汉

计算机操作系统 微课索引

第 1 章 绪论	
认识操作系统 (1.1 节)	
操作系统采用的技术 (1.5 节)	
第 2 章 操作系统的结构和硬件支持	
处理机的态 (2.3 节)	
中断及其处理 (2.4 节)	
第 3 章 操作系统的用户接口	
操作系统的 用户接口及分类 (3.2 节)	
系统功能调用及其 实现技术 (3.3 节)	
Linux 系统功能调用 (3.5 节)	
第 4 章 进程及进程管理	
并发进程及其特点 (4.1 节)	
进程定义 (4.2.1 节)	
进程的状态及变迁 (4.2.2 节)	

进程控制块及 进程队列 (4.2.3 节)	
进程控制 (4.3 节)	
进程互斥的概念 (4.4.2 节)	
进程同步的概念 (4.4.3 节)	
进程互斥的实现 (4.6.2 节)	
合作进程的 执行次序 (4.6.3 节)	
共享缓冲区的合作 进程的同步 (4.6.3 节)	
生产者—消费者问题 (4.6.4 节)	
第 5 章 资源分配与调度	
虚拟资源与 资源分配策略 (5.1 节)	
死锁 1 (5.3.1 节)	
死锁 2 (5.3.3 节)	

第6章 处理机调度	
处理机的多级调度 (6.1节)	
作业调度 (6.2节)	
进程调度 (6.3节)	
第7章 主存管理	
主存空间和程序的 地址空间 (7.1节)	
地址映射和虚拟存储器 (7.2节)	
分区分配的放置策略 (7.3节)	
页式地址变换 (7.4.1节)	
页面请调机制 (7.4.3节)	
页面淘汰机制与 淘汰策略 (7.4.4节)	
页面置换算法 (7.4.5节)	
段式系统与段页式系统 (7.5节)	
Linux 系统的段页式 地址变换 (7.7.2节)	

第8章 设备管理	
设备独立性与 设备控制块 (8.1节)	
缓冲技术 (8.2节)	
输入/输出控制 (8.4节)	
UNIX 系统缓冲区管理 (8.5.3节)	
第9章 文件系统	
文件系统概念和 文件结构 (9.1节)	
文件的物理结构 (9.3节)	
文件目录概念及树形 文件目录 (9.5节)	
当前目录和链接技术 (9.6.3节)	
UNIX 文件系统及文件 索引结构 (9.8.1节)	
UNIX 文件存储器 空闲块的管理 (9.8.5节)	
Linux 系统的索引结构 (9.9.7节)	

目 录 CONTENTS

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	18
1.4.1 批量操作系统	18
1.4.2 分时操作系统	18
1.4.3 实时操作系统	19
1.4.4 个人计算机操作系统	20
1.4.5 网络操作系统	21
1.4.6 分布式操作系统	22
1.5 操作系统采用的关键技术	25
1.5.1 操作系统采用的并行处理技术	25
1.5.2 操作系统采用的虚拟技术	26
1.6 UNIX、Linux 操作系统概述	26
1.6.1 UNIX 操作系统的发展	26
1.6.2 UNIX 操作系统的类型及特点	27
1.6.3 Linux 系统及其特点	29
习题 1	33

第2章 操作系统的结构和硬件支持	35
2.1 操作系统虚拟机	35
2.2 操作系统的组织结构	36
2.2.1 操作系统的结构	36
2.2.2 运行时的组织结构	39
2.2.3 操作系统与计算机系统各层次的接口	40
2.3 处理机的特权级	41
2.3.1 处理机的态及分类	41
2.3.2 特权指令	41
2.4 中断及其处理	42
2.4.1 中断概念及类型	42
2.4.2 向量中断和探询中断	44
2.4.3 中断进入	44
2.4.4 软件中断处理过程	46
2.5 UNIX、Linux 系统结构	48
2.5.1 UNIX 系统的体系结构	48
2.5.2 UNIX 系统的核心结构	49
2.5.3 Linux 系统的内核结构	50
2.6 Linux 系统的特权级与中断处理	50
2.6.1 Linux 系统的特权级	50
2.6.2 中断处理的上半部和下半部	51
2.6.3 中断处理下半部的实现机制	52
习题 2	53
第3章 操作系统的用户接口 ...	54
3.1 用户工作环境	54
3.1.1 操作系统提供的环境	54
3.1.2 操作系统的生成和系统初启	55
3.1.3 应用程序的处理	58
3.2 用户接口	60
3.2.1 用户接口的定义	60

3.2.2	操作系统提供的用户接口	60
3.3	系统功能调用	62
3.3.1	系统功能调用的定义	62
3.3.2	系统功能调用的实现	63
3.3.3	应用程序的编程接口	64
3.4	UNIX 系统功能调用	65
3.4.1	UNIX 系统调用的分类	65
3.4.2	UNIX 系统调用的实现	66
3.5	Linux 系统功能调用	68
3.5.1	Linux 系统功能调用的过程	69
3.5.2	Linux 系统功能调用的实现机制	70
3.5.3	增加一个新的系统调用的方法	71
3.5.4	从用户空间访问新的系统调用	72
习题 3		73
第 4 章	进程及进程管理	74
4.1	进程引入	74
4.1.1	顺序程序及特点	74
4.1.2	并发程序及特点	76
4.1.3	与时间有关的错误	78
4.2	进程概念	79
4.2.1	进程的定义	79
4.2.2	进程的状态及变迁	80
4.2.3	进程控制块	81
4.3	进程控制	83
4.3.1	进程控制的概念	83
4.3.2	进程创建与撤销	84
4.3.3	进程等待与唤醒	85
4.4	进程之间的约束关系	86
4.4.1	进程竞争与合作	86
4.4.2	进程互斥的概念	87
4.4.3	进程同步的概念	89
4.5	同步机构	90
4.5.1	锁和上锁、开锁操作	91
4.5.2	信号灯和 P、V 操作	92
4.6	进程互斥与同步的实现	94

4.6.1	上锁原语和开锁原语实现进程互斥	94
4.6.2	信号灯实现进程互斥	94
4.6.3	进程同步的实现	95
4.6.4	生产者—消费者问题	99
4.7	进程通信	100
4.7.1	进程通信的概念	100
4.7.2	进程通信方式	100
4.8	线程概念及特点	102
4.8.1	线程的概念	102
4.8.2	线程的特点与状态	103
4.9	操作系统的并发机制实例	105
4.9.1	创建进程及应用实例	105
4.9.2	创建线程及应用实例	107
4.9.3	等待进程、线程的终止及其应用	107
4.9.4	信号量与使用方法	109
4.9.5	共享主存及应用实例	110
4.10	UNIX 系统的进程管理	112
4.10.1	UNIX 系统的进程及映像	112
4.10.2	UNIX 进程的状态及变迁	115
4.10.3	UNIX 进程的创建	118
4.10.4	UNIX 进程的终止与等待	120
4.10.5	UNIX 进程的睡眠与唤醒	122
4.11	Linux 系统的进程管理	123
4.11.1	Linux 系统的进程与线程	123
4.11.2	进程描述符及其主要内容	123
4.11.3	进程描述符的获得	126
4.11.4	Linux 系统的进程状态变迁	127
4.11.5	Linux 系统的进程创建和终止	127
4.11.6	Linux 系统的进程等待与唤醒	129
4.11.7	Linux 系统中线程的实现	130
习题 4		130
第 5 章	资源分配与调度	135
5.1	资源管理概述	135

5.1.1 资源管理的目的和任务	135	6.6.2 可变优先级	174
5.1.2 虚拟资源	136	6.6.3 可变时间片	175
5.2 资源管理的机制和策略	137	6.6.4 进程调度用的数据结构	177
5.2.1 资源分配机制	137	6.6.5 Linux 系统的进程调度算法	178
5.2.2 资源分配策略	139	习题 6	179
5.3 死锁	143	第 7 章 主存管理	181
5.3.1 死锁的定义与例子	143	7.1 主存管理概述	181
5.3.2 产生死锁的原因和必要条件	145	7.1.1 主存分片共享	181
5.3.3 系统模型和死锁的处理	146	7.1.2 程序的逻辑组织	182
5.3.4 解决死锁问题的策略	150	7.2 主存管理的功能	182
5.3.5 死锁的预防	151	7.2.1 虚拟存储器	182
5.3.6 死锁的避免	152	7.2.2 地址映射	184
5.3.7 死锁的检测与忽略	153	7.2.3 主存分配	186
习题 5	154	7.2.4 存储保护	187
第 6 章 处理机调度	157	7.3 分区存储管理及存在的问题	188
6.1 处理机的多级调度	157	7.3.1 动态分区存储管理技术	188
6.2 作业调度	158	7.3.2 分区分配机构	189
6.2.1 作业的状态	158	7.3.3 分区分配与放置策略	190
6.2.2 作业调度的功能	158	7.3.4 碎片问题及拼接技术	192
6.2.3 作业控制块	159	7.4 页式存储管理	193
6.2.4 调度算法性能的衡量	160	7.4.1 页式系统应解决的问题	193
6.2.5 作业调度算法	161	7.4.2 页式地址变换	194
6.3 进程调度	163	7.4.3 请调页面的机制	197
6.3.1 调度/分派结构	163	7.4.4 淘汰机制与策略	199
6.3.2 进程调度的功能	164	7.4.5 几种置换算法	201
6.3.3 调度方式	165	7.5 段式和段页式存储管理	204
6.3.4 进程优先数调度算法	165	7.5.1 段式地址结构	204
6.3.5 循环轮转调度	167	7.5.2 段式地址变换	204
6.3.6 多级反馈队列调度	168	7.5.3 扩充段表功能	205
6.3.7 调度用的进程状态变迁图	169	7.5.4 段页式存储管理	206
6.4 线程调度	170	7.6 UNIX 系统的存储管理	207
6.5 UNIX 系统的进程调度	171	7.6.1 概述	207
6.5.1 UNIX 系统的进程调度算法	171	7.6.2 请求调页的数据结构	208
6.5.2 进程切换调度程序 switch	172	7.6.3 UNIX 系统的地址变换	209
6.6 Linux 系统的进程调度	173	7.6.4 页面错	210
6.6.1 进程调度程序的设计目标和特点	173	7.7 Linux 系统的存储管理	211
		7.7.1 主存寻址	211

7.7.2	Linux 系统段页式地址变换	214
7.7.3	Linux 系统动态内核管理	215
7.7.4	Linux 系统的进程地址空间	218
习题 7		220

第 8 章 设备管理 223

8.1	设备管理概述	223
8.1.1	设备管理的功能	223
8.1.2	设备独立性	224
8.1.3	设备控制块	226
8.2	缓冲技术	226
8.2.1	缓冲概述	226
8.2.2	常用的缓冲技术	228
8.3	设备分配	229
8.3.1	设备分配概述	229
8.3.2	独享分配	230
8.3.3	共享分配	230
8.3.4	虚拟分配	231
8.4	输入/输出控制	233
8.4.1	输入/输出硬件	233
8.4.2	输入/输出控制方式	234
8.4.3	输入/输出子系统	237
8.4.4	输入/输出控制的例子	239
8.5	UNIX 系统的设备管理	241
8.5.1	UNIX 系统设备管理的特点	241
8.5.2	UNIX 系统的设备驱动程序接口	242
8.5.3	UNIX 系统的缓冲区管理	244
8.5.4	UNIX 系统的设备 I/O 控制	251
8.6	Linux 系统的设备驱动	253
8.6.1	Linux 系统设备的分类	253
8.6.2	设备文件及其标识	254
8.6.3	Linux 块设备的处理	256
8.6.4	用于块设备处理的数据结构	257
8.6.5	输入/输出调度程序	260
8.6.6	策略例程	261
习题 8		262

第 9 章 文件系统 264

9.1	文件系统概述	264
9.1.1	文件	264
9.1.2	文件系统	266
9.1.3	文件的组织	267
9.2	文件的逻辑结构和存取方法	268
9.2.1	文件的逻辑结构	268
9.2.2	文件的存取方法	269
9.3	文件的物理结构	269
9.3.1	连续文件	270
9.3.2	串联文件	271
9.3.3	索引文件	272
9.3.4	文件物理结构比较	274
9.4	文件存储空间的管理	275
9.4.1	空闲文件目录	275
9.4.2	空闲块链	275
9.4.3	位示图	275
9.4.4	分配策略	276
9.5	文件目录	277
9.5.1	文件目录及其内容	277
9.5.2	一级文件目录及缺点	278
9.5.3	多级文件目录	278
9.6	共享与安全	279
9.6.1	文件共享与安全性的关系	279
9.6.2	存取权限的类型及其验证	280
9.6.3	用文件路径名加快文件的查找	282
9.7	文件操作与文件备份	284
9.7.1	文件操作	284
9.7.2	文件备份	285
9.8	UNIX 文件系统的主要结构及实现	286
9.8.1	UNIX 文件系统的特点	286
9.8.2	UNIX 系统的索引文件结构	287
9.8.3	UNIX 系统文件目录结构	291
9.8.4	UNIX 系统的打开文件机构	292
9.8.5	文件存储器空闲块的管理	296

9.8.6 UNIX 文件系统调用.....	298	9.9.8 Ext2 主存数据结构.....	316
9.9 Linux 文件系统.....	303	习题 9.....	317
9.9.1 虚拟文件系统 VFS (Virtual File System) 概述.....	304	模拟试题 1.....	320
9.9.2 VFS 通用文件系统模型与 VFS 对象.....	304	模拟试题 2.....	322
9.9.3 与进程相关的数据结构.....	307	模拟试题 1 答案.....	324
9.9.4 VFS 系统调用的实现.....	308	模拟试题 2 答案.....	328
9.9.5 Ext2 文件系统概述.....	310	参考文献.....	332
9.9.6 Ext2 磁盘数据结构.....	312		
9.9.7 Ext2 磁盘空间的管理.....	314		

随着科学技术的飞速发展,人类生活质量的不断提高,生产实践和科学活动的水平不断地提高,计算机已广泛应用于国民经济和日常生活中,如科学计算、数据处理、国防、航天、电信、信息管理等,都有大量的问题需要计算机来解决。

许多问题的求解都需要给出其形式化定义和求解方法的形式化描述。对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,通常将一种算法的实现称作一次计算,而对问题的求解还必须要有实现算法的工具或设施。实现算法的工具或设施从最初算筹,大量使用的计算器,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不变的,算盘和计算器都可以进行加、减、乘、除运算。人们为解决某一问题,只将问题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代数算法,这种计算称为手工计算方式,算盘或计算器是手工计算的一种工具。在手工计算方式下,人们按照预先确定的一系列计算方案,先输入原始数据,然后按顺序进行第一步计算,记下中间结果,再当第二步计算,直到算出最终结果,并把结果记于纸上。在这一过程中,输入原始数据,执行运算操作,中间结果而存储和最终结果的计算都是依靠人来操作和完成的,所以,这一计算过程是手工操作过程。

现代计算机和运算器还是进行四则运算,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他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,并基于此构建了计算机体系结构,实现了计算的自动化。计算机要进行自动计算,必须有预先编好的程序存放在计算机内,计算机还必须能“识别”程序语言的含义并顺序地执行指定的操作,能及时取回或控制中间数据,能够自动地输出结果。所以,按冯·诺依曼计算机原理,有一个存储器用来存放程序和数;有一个控制器来执行指定的操作;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。



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,计算机应用随之广泛深入。在计算机应用中,如科学计算、数据处理、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。对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作一次计算。而对问题的求解还必须要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,大量使用的计算器,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,算盘和计算器都可以进行加、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来操作完成的,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,必须有计算方案或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,能够自动地输出结果。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;有一个运算器用以执行指定的操作;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。



认识操作系统

存储程序式计算机的结构包括中央处理器（CPU）、存储器和输入/输出设备。所有的单元都通过总线连接，总线分为地址总线 and 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 结构的计算机而言，仍然是存储程序式计算机的体系结构。

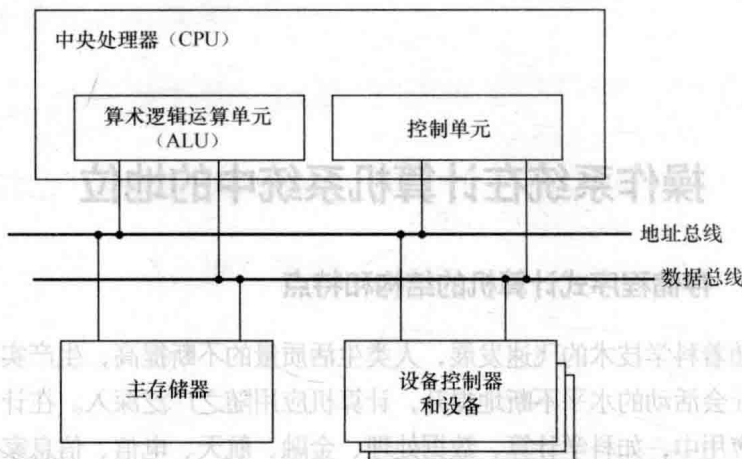


图 1.1 冯·诺依曼计算机体系结构

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逻辑操作的功能单元以及一组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。状态寄存器保存着处理机运行过程中的当前状态。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）三个寄存器组成。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，存储容量大但存取速度较慢的辅存储器组成。

输入/输出设备（I/O 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 and 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。它采

用顺序过程计算模型,具有逻辑判断能力和自动连续运算能力,特点是集中顺序过程控制。其计算是过程性的,完全模拟手工操作过程,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,与中间结果一起执行下一个操作,如此计算下去,直到计算完成。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),计算机根据程序设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。

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将会寸步难行。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,这一切若要求硬件完成,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也将造成极大的障碍。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、数据和在研制过程中形成的各种文档资料组成,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3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 可知,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、应用程序构成。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,这些硬件资源也就没有活力。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、硬资源,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。它在裸机上运行,是所有软件中与硬件相连的第一层软件。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

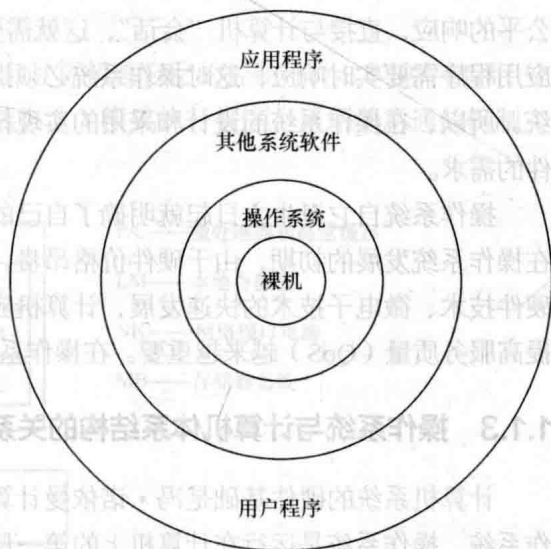


图 1.2 计算机系统的组成

和控制；二是各层对操作系统的影响和制约。

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

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和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，同时还可下载一个文件。那么单 CPU 计算机如何运行多任务呢？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以下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，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。

操作系统自它诞生之日起就明确了自己的宗旨——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，由于硬件价格昂贵，提高计算机的使用效率被放在了第一位，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另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。

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, 昂贵的计算机硬部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, 人们研究并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 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发展, 这些技术的应用取得了可观的经济效益。人们所做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。但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。

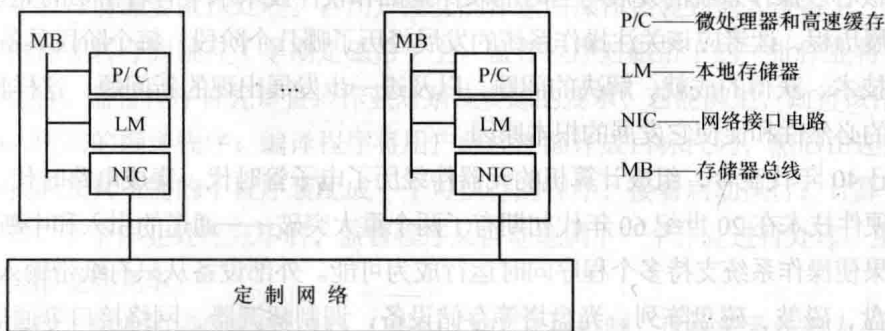


图 1.3 消息传递型多计算机的一般结构

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

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