

面向 **21** 世纪

高等学校计算机类专业系列教材

操作系统教程

——Linux实例分析

A Course in Operating Systems

孟庆昌 编著

西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校计算机、信息工程类专业系列教材

操作系统教程

——Linux 实例分析

A Course in Operating Systems

孟庆昌 编著

西安电子科技大学出版社

2004

内 容 简 介

本书以 Linux 操作系统为实例,全面、系统地介绍了操作系统的概念、理论、技术和实现方法。

全书共分 11 章,分别介绍操作系统概述,进程管理,处理机调度,存储器管理,文件系统,设备管理,中断、陷入和信号机构,死锁,多处理器系统,分布式系统,性能评价及安全性。

本书可作为计算机科学和工程类专业的教材,以及从事计算机工作的广大科技人员学习操作系统知识和 Linux 技术的参考书。

图书在版编目(CIP)数据

操作系统教程——Linux 实例分析=A Course in Operating Systems/孟庆昌编著.

—西安:西安电子科技大学出版社,2004.7

(面向 21 世纪高等学校信息工程类专业系列教材)

ISBN 7-5606-1385-3

I. 操… II. 孟… III. Linux 操作系统—高等学校—教材 IV. TP316.89

中国版本图书馆 CIP 数据核字(2004)第 031402 号

责任编辑 潘恩祥 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2004 年 月第 1 版 2004 年 7 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 19.75

字 数 464 千字

印 数 1~4000 册

定 价 21.00 元

ISBN 7-5606-1385-3/TP·0736(课)

XDUP 1656001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

序

第三次全国教育工作会议以来,我国高等教育得到空前规模的发展。经过高校布局和结构的调整,各个学校的新专业均有所增加,招生规模也迅速扩大。为了适应社会对“大专业、宽口径”人才的需求,各学校对专业进行了调整和合并,拓宽专业面,相应的教学计划、大纲也都有了较大的变化。特别是进入 21 世纪以来,信息产业发展迅速,技术更新加快。面对这样的发展形势,原有的计算机、信息工程两个专业的传统教材已很难适应高等教育的需要,作为教学改革的重要组成部分,教材的更新和建设迫在眉睫。为此,西安电子科技大学出版社聘请南京邮电学院、西安邮电学院、重庆邮电学院、吉林大学、杭州电子工业学院、桂林电子工业学院、北京信息工程学院、深圳大学、解放军电子工程学院等 10 余所国内电子信息类专业知名院校长期在教学科研第一线工作的专家教授,组成了高等学校计算机、信息工程类专业系列教材编审专家委员会,并且面向全国进行系列教材编写招标。该委员会依据教育部有关文件及规定对这两大专业的教学计划和课程大纲,对目前本科教育的发展变化和相应系列教材应具有的特色和定位以及如何适应各类院校的教学需求等进行了反复研究、充分讨论,并对投标教材进行了认真评审,筛选并确定了高等学校计算机、信息工程类专业系列教材的作者及审稿人。这套教材预计在 2004 年春季全部出齐。

审定并组织出版这套教材的基本指导思想是力求精品、力求创新、好中选优、以质取胜。教材内容要反映 21 世纪信息科学技术的发展,体现专业课内容更新快的要求;编写上要具有一定的弹性和可调性,以适合多数学校使用;体系上要有所创新,突出工程技术型人才培养的特点,面向国民经济对工程技术人才的需求,强调培养学生较系统地掌握本学科专业必需的基础知识和基本理论,有较强的本专业的基本技能、方法和相关知识,培养学生具有从事实际工程的研发能力。在作者的遴选上,强调作者应在教学、科研第一线长期工作,有较高的学术水平和丰富的教材编写经验;教材在体系和篇幅上符合各学校的教学计划要求。

相信这套精心策划、精心编审、精心出版的系列教材会成为精品教材,得到各院校的认可,对于新世纪高等学校教学改革和教材建设起到积极的推动作用。

系列教材编委会
2002 年 8 月

高等学校计算机、信息工程类专业

系列教材编审专家委员会

主任：杨震（南京邮电学院副院长、教授）

副主任：张德民（重庆邮电学院通信与信息工程学院院长、教授）

韩俊刚（西安邮电学院计算机系主任、教授）

李荣才（西安电子科技大学出版社总编辑、教授）

计算机组

组长：韩俊刚（兼）

成员：（按姓氏笔画排列）

王小民（深圳大学信息工程学院计算机系主任、副教授）

王小华（杭州电子工业学院计算机分院副院长、副教授）

孙力娟（南京邮电学院计算机系副主任、副教授）

李秉智（重庆邮电学院计算机学院院长、教授）

孟庆昌（北京信息工程学院教授）

周娅（桂林电子工业学院计算机系副主任、副教授）

张长海（吉林大学计算机科学与技术学院副院长、教授）

信息工程组

组长：张德民（兼）

成员：（按姓氏笔画排列）

方强（西安邮电学院电信系主任、教授）

王晖（深圳大学信息工程学院电子工程系主任、副教授）

胡建萍（杭州电子工业学院电子信息分院副院长、副教授）

徐祎（解放军电子工程学院电子技术教研室主任、副教授）

唐宁（桂林电子工业学院通信与信息工程系副主任、副教授）

章坚武（杭州电子工业学院通信工程分院副院长、教授）

康健（吉林大学通信工程学院副院长、教授）

蒋国平（南京邮电学院电子工程系副主任、副教授）

总策划：梁家新

策划：马乐惠 云立实 马武装 马晓娟

电子教案：马武装

前 言

本书是在《操作系统教程——UNIX 实例分析(第二版)》的基础上大幅度修改而成的。首先,以 Linux 系统为实例。众所周知, Linux 是当代最著名的自由软件,在发展、应用速度上首屈一指。它的开放性、安全性、与 UNIX 的兼容性、网络应用性等诸多方面都是其他系统无法比拟的。目前,学习、应用、改造 Linux 系统已成为一股热潮。可以说,除微软以外所有的软件公司都对 Linux 的发展予以支持。在这种潮流下,以 Linux 作为实例讲述操作系统的理论和技术是大势所趋。

全书共分 11 章。第 1 章是操作系统概述;第 2 章介绍进程管理,其中添加了线程概念;第 3 章介绍处理机调度,还讲述了 Linux 系统初启的过程;第 4 章介绍存储器管理,前面是不带虚存的各种技术,后面是提供虚存的存储管理技术;第 5 章介绍文件系统,加强了对文件保护的讲述;第 6 章介绍设备管理,增加了磁盘管理的内容;第 7 章介绍中断、陷入和信号机构;第 8 章介绍死锁,增加了银行家算法应用示例,加强了对死锁恢复和“饥饿”的讲解;第 9 章介绍多处理器系统,内容改动较大;第 10 章介绍分布式系统,修改地方也很多;第 11 章是性能评价及安全性,添加了对计算机病毒知识及对策的介绍。

本书在编写时注意难度适中,篇幅不宜过大,知识要新,读者定位在大多数普通高校的学生。同时,也兼顾到考研的需求。

参加本书编写、校对等工作的人员还有刘振英、孟平、唐伟杰、孟欣和张震等。在此,谨向诸位表示衷心感谢。

我们总期望把事情办得更好点。然而,由于水平有限,时间紧迫,加之新理论、新技术、新产品不断涌现,因此书中仍难免存在不当之处,恳请广大读者批评指正。

编 者

2004 年 2 月于北京

目 录

第 1 章 操作系统概述	1	2.1.4 进程概念的引入和描述	35
1.1 计算机发展简史	1	2.1.5 进程的状态及其变迁	36
1.2 计算机系统组成	1	2.1.6 进程的组成	37
1.2.1 计算机系统结构	2	2.1.7 进程控制块	37
1.2.2 计算机软件分类	3	2.1.8 PCB 的组织方式	38
1.3 操作系统的概念	4	2.2 线程	40
1.4 操作系统的主要功能	5	2.2.1 线程概念	40
1.5 操作系统的发展历程	9	2.2.2 线程的实现方式	42
1.5.1 手工操作阶段	9	2.3 进程管理	43
1.5.2 早期批处理阶段	10	2.3.1 创建进程	43
1.5.3 执行系统阶段	11	2.3.2 终止进程	44
1.5.4 多道程序系统阶段	12	2.3.3 更换进程映像	44
1.6 操作系统的类型	13	2.3.4 阻塞进程	44
1.6.1 多道成批系统	13	2.3.5 唤醒进程	45
1.6.2 分时系统	14	2.4 进程间通信	45
1.6.3 实时系统	15	2.4.1 进程间的关系	45
1.6.4 个人机系统	17	2.4.2 竞争条件和临界区	46
1.6.5 网络操作系统	17	2.4.3 用锁操作原语实现互斥	48
1.6.6 分布式操作系统	18	2.4.4 信号量上的 P、V 操作原语	49
1.7 操作系统的特征	19	2.4.5 用 P、V 原语实现互斥	50
1.8 操作系统的结构	19	2.4.6 用 P、V 原语实现简单同步	51
1.8.1 单块结构	20	2.4.7 生产者—消费者问题	52
1.8.2 层次结构	20	2.5 经典进程同步问题	54
1.8.3 虚拟机	22	2.5.1 读者—写者问题	54
1.8.4 客户/服务器模型	22	2.5.2 哲学家进餐问题	55
1.9 UNIX/Linux 系统的特点和结构	23	2.5.3 困睡的理发师问题	57
1.9.1 UNIX 系统的发展和特点	23	2.6 管程	58
1.9.2 Linux 系统的历史和特点	26	2.7 进程通信	61
1.10 小结	29	2.7.1 消息缓冲通信	61
习题	29	2.7.2 信箱通信	64
第 2 章 进程管理	31	2.8 Linux 进程管理	64
2.1 进程概念	31	2.8.1 进程和线程的概念	65
2.1.1 程序的顺序执行	31	2.8.2 进程的结构	66
2.1.2 程序的并发执行和资源共享	32	2.8.3 对进程的操作	68
2.1.3 程序并发执行的特性	33	2.8.4 进程同步和通信	70
		2.9 小结	71

习题	72	4.4.2 分页系统中的地址映射	113
第3章 处理机调度	75	4.4.3 快表和页表构造	113
3.1 调度级别	75	4.4.4 页的共享和保护	117
3.2 作业调度	76	4.5 分段技术	118
3.2.1 作业状态	76	4.5.1 分段存储管理的基本概念	118
3.2.2 作业调度	77	4.5.2 地址转换	120
3.3 进程调度	78	4.5.3 段的共享和保护	120
3.3.1 进程调度的功能和时机	78	4.6 虚拟存储器	122
3.3.2 两级调度模型	79	4.6.1 虚拟存储器概念	122
3.3.3 三级调度模型	80	4.6.2 虚拟存储器特征	123
3.4 性能评价标准	80	4.7 请求分页技术	124
3.4.1 调度策略的选择	80	4.7.1 请求分页的基本思想	124
3.4.2 性能评价标准	81	4.7.2 硬件支持及缺页处理	124
3.5 常用调度算法	82	4.7.3 请求分页的优缺点	126
3.5.1 先来先服务(FCFS)	82	4.7.4 请求分页的性能	126
3.5.2 短作业优先(SJF)	83	4.7.5 页面置换	127
3.5.3 优先级(Priority)	84	4.8 页面置换算法	128
3.5.4 抢占式和非抢占式算法	85	4.8.1 先入先出法(FIFO)	129
3.5.5 轮转法(RR)	87	4.8.2 最优置换算法(OPT)	130
3.5.6 多级队列法(MQ)	88	4.8.3 最久未使用算法(LRU)	130
3.5.7 多级反馈队列法(MFQ)	89	4.8.4 第二次机会算法(SCR)	132
3.5.8 多级调度综合示例	90	4.9 内存块分配算法和抖动问题	133
3.6 Linux 系统中的进程调度	91	4.9.1 内存块分配算法	133
3.6.1 进程调度	91	4.9.2 抖动(Thrashing)问题	135
3.6.2 shell 基本工作原理	93	4.9.3 工作集	136
3.6.3 系统初启	94	4.10 段式虚拟存储器	137
3.7 小结	96	4.10.1 基本工作过程	137
习题	97	4.10.2 连接中断处理	138
第4章 存储器管理	99	4.10.3 段式虚拟存储的优点和缺点	139
4.1 引言	99	4.11 段页式结合系统	139
4.1.1 用户程序的主要处理阶段	99	4.12 Linux 系统的存储管理	140
4.1.2 重定位	101	4.12.1 Linux 的多级页表	140
4.2 基本的内存管理技术	104	4.12.2 内存页的分配与释放	141
4.2.1 单一连续分配法	104	4.12.3 内存交换	142
4.2.2 分区法	104	4.13 小结	143
4.2.3 可重定位分区分配	108	习题	145
4.3 对换技术	110	第5章 文件系统	148
4.3.1 早期对换技术	110	5.1 文件系统概述	148
4.3.2 多道程序环境下的对换	110	5.1.1 文件及其分类	148
4.4 分页技术	111	5.1.2 文件系统的功能	151
4.4.1 分页存储管理的基本概念	111	5.1.3 文件系统的结构	152
		5.1.4 用户对文件的存取方法	152

5.2 文件的逻辑组织和物理组织.....	154	6.3.2 SPOOLing 系统	202
5.2.1 文件的逻辑组织	154	6.4 I/O 软件构造原则	203
5.2.2 文件的物理组织	155	6.4.1 I/O 软件目标	203
5.3 目录结构	159	6.4.2 中断处理程序	204
5.3.1 文件控制块和文件目录	159	6.4.3 设备驱动程序	204
5.3.2 目录结构的类型	160	6.4.4 与设备无关的 I/O 软件	209
5.4 文件存储空间的管理	164	6.4.5 用户空间 I/O 软件	210
5.4.1 空闲空间表法	164	6.5 磁盘调度和管理	210
5.4.2 空闲块链接法	165	6.5.1 物理特性	211
5.4.3 位示图(Bit Map)法	166	6.5.2 磁盘调度算法	212
5.4.4 空闲块成组链接法	166	6.5.3 磁盘调度算法的选择	215
5.5 对文件的主要操作	167	6.5.4 磁盘管理	215
5.5.1 创建和删除文件	168	6.6 Linux 系统的设备管理	217
5.5.2 打开与关闭文件	168	6.6.1 设备管理概述	217
5.5.3 读文件与写文件	169	6.6.2 设备驱动程序和内核之间 的接口	218
5.5.4 链接文件与解除链接	170	6.7 小结	221
5.6 文件系统的可靠性	171	习题	222
5.6.1 坏块管理	171		
5.6.2 后备	171	第 7 章 中断、陷入和信号机构	223
5.6.3 文件系统的一致性	172	7.1 概述	223
5.7 文件保护	173	7.1.1 中断及其一般处理过程	223
5.7.1 一般文件的保护机制	174	7.1.2 中断优先级和多重中断	227
5.7.2 保护域	176	7.1.3 中断屏蔽	228
5.7.3 存取控制表	178	7.2 UNIX 系统对中断和陷入的处理	228
5.7.4 权力	179	7.2.1 中断处理	228
5.8 Linux 文件系统	180	7.2.2 陷入处理	230
5.8.1 EXT2 文件系统	181	7.3 系统调用的实施举例	232
5.8.2 虚拟文件系统	185	7.4 UNIX 信号机构	234
5.8.3 管道文件(Pipe)	189	7.4.1 信号分类	235
5.9 小结	190	7.4.2 信号处理方式	236
习题	192	7.4.3 信号的检测和处理	236
		7.5 小结	237
第 6 章 设备管理	194	习题	238
6.1 概述	194		
6.1.1 设备分类	194	第 8 章 死锁	240
6.1.2 设备管理的功能	195	8.1 概述	240
6.1.3 通道技术	196	8.1.1 死锁的概念	240
6.1.4 设备控制器	198	8.1.2 资源概念	241
6.2 缓冲技术(Buffering)	198	8.1.3 进程推进顺序与死锁	243
6.2.1 缓冲技术的引入	198	8.2 产生死锁的条件	245
6.2.2 缓冲区的设置	199	8.2.1 产生死锁的必要条件	245
6.3 设备分配技术	200	8.2.2 资源分配图	246
6.3.1 设备分配技术	200		

8.2.3 处理死锁的方法	247	10.2.2 分布式操作系统的设计因素 ...	280
8.3 死锁的预防	247	10.3 分布式系统的通信	282
8.4 死锁的避免	250	10.3.1 ISO OSI 参考模型	282
8.4.1 安全状态	250	10.3.2 客户——服务器模型	283
8.4.2 银行家算法	251	10.3.3 远程过程调用	284
8.4.3 对单体资源类的简化算法	253	10.3.4 组通信	284
8.4.4 银行家算法应用示例	253	10.4 分布式进程管理	285
8.5 死锁的检测与恢复	255	10.4.1 进程迁移	285
8.5.1 对多体资源类的死锁检测	255	10.4.2 分布式系统的进程管理	286
8.5.2 对单体资源类的死锁检测	256	10.4.3 处理器分配	287
8.5.3 从死锁中恢复	256	10.5 分布式系统中的同步	289
8.5.4 “饥饿”状态	258	10.5.1 事件排序	289
8.6 处理死锁的综合方式	258	10.5.2 互斥	290
8.7 小结	259	10.6 分布式系统中的死锁	291
习题	260	10.6.1 死锁的检测	291
第 9 章 多处理器系统	263	10.6.2 死锁的预防	292
9.1 概述	263	10.7 分布式文件系统	292
9.2 多处理器结构	264	10.7.1 文件服务接口	292
9.2.1 UMA 多处理器结构	265	10.7.2 目录服务器接口	293
9.2.2 NUMA 多处理器结构	268	10.7.3 文件共享语义	295
9.3 多处理器操作系统类型	269	10.8 小结	296
9.3.1 多处理器系统的问题	269	习题	297
9.3.2 多处理器操作系统类型	270	第 11 章 性能评价及安全性	298
9.4 多处理器调度和同步	271	11.1 性能评价	298
9.4.1 多处理器调度	271	11.1.1 性能评价的目的	298
9.4.2 多处理器同步	275	11.1.2 性能评价技术	299
9.5 小结	276	11.2 操作系统的安全性	300
习题	276	11.2.1 安全性问题	300
第 10 章 分布式系统	277	11.2.2 常见的安全性攻击	301
10.1 概述	277	11.2.3 对付病毒的常用方法	302
10.1.1 分布式系统特征	277	11.3 安全对策	303
10.1.2 分布式系统优点	278	11.4 小结	303
10.1.3 三种多机操作系统的比较	279	习题	303
10.2 分布式操作系统	279	参考文献	305
10.2.1 分布式操作系统的功能	280		

第 1 章 操作系统概述

当今世界上发展最快的科学技术当属计算机技术。在 50 多年里,计算机技术经历了五个时代,现在其应用已遍及社会生活的各个角落。与硬件技术发展相适应的是软件技术正日新月异地前进。

操作系统是计算机系统中的基础软件,负责对整个系统资源的控制、调度和管理,提供用户使用计算机的界面,并为用户提供各种服务。

本章介绍什么是操作系统,它有什么功能,有哪些主要类型,有什么基本特征,以及 Linux 系统的特点。

1.1 计算机发展简史

从 1946 年世界上第一台电子数字计算机(以下简称计算机)问世以来,至今 50 多年。在此期间,它的性能得到飞速的提高,其应用领域遍及社会生活的各个角落,相应地计算机技术和理论都得到迅猛发展。在当今信息时代,计算机技术已处于举足轻重的地位。

人们通常用“代”这个术语来表示计算机技术发展过程的不同阶段。从组成计算机系统的主要电子器件的角度来看,计算机的发展历史大致可分为以下几个阶段:

第一代:1946 年~1959 年,以美国建造的 ENIAC 为代表,主要电子器件是电子管。

第二代:1960 年~1964 年,主要特征是以晶体管为主要电子器件,如 IBM 7090 系列。

第三代:1965 年~1973 年,以集成电路作为计算机的主要器件,如 IBM 360 机种。

第四代:从 1974 年至今,大规模集成电路(LSI)、超大规模集成电路(VLSI)用于计算机,从巨型机到微型机、个人计算机,其类型层出不穷。与之相应,计算机科学和技术也得到了迅速发展。

当然,对于计算机“代”的划分至今尚无统一定义。有人认为从 1990 年超大规模集成电路得到应用后,就进入第五代计算机时期。对此,我们就不做进一步讨论了。

1.2 计算机系统组成

一个完整的计算机系统是由硬件和软件两大部分组成的。通常硬件是指计算机物理装置本身,如 CPU、内存、各种 I/O 设备、Modem 等。软件是相对硬件而言的,它是与数据处理系统的操作有关的计算机程序、过程、规则和相关文档资料的总称。如大家所熟悉的 Windows 2000、Linux、UNIX、Word、IE(Internet Explorer)等都属于软件范畴。简单地说,软件就是计算机可以执行的程序。

1.2.1 计算机系统结构

现代通用计算机系统由 CPU、内存和多种 I/O 设备组成，通过系统总线连接在一起，实现彼此通信。图 1-1 示出多数微型机采用的系统结构。

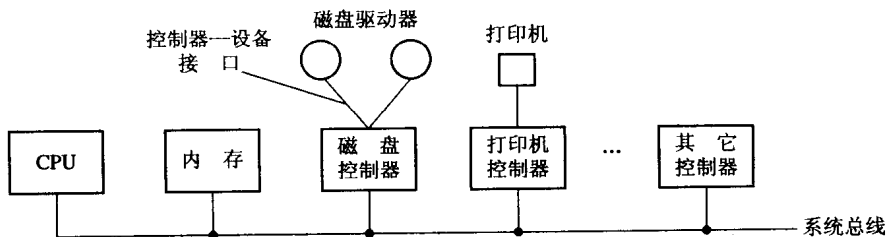


图 1-1 利用总线连接 CPU、内存、控制器和 I/O 设备

1. CPU

CPU 是计算机系统中的“大脑”。它从内存(或高速缓存)中取出指令并执行它们。其基本工作顺序是：提取指令，译码分析，执行指令。

CPU 内部包含若干个寄存器。其中，一类是通用寄存器，用来存放关键变量和中间结果；另一类是专用寄存器，如程序计数器(PC)、栈指针寄存器和程序状态字(PSW)。PC 中保存下面要提取的指令的内存地址。栈指针寄存器指向当前内存栈的顶端，栈中保存有关函数(过程)调用时的现场信息。PSW 中包括条件码位、CPU 优先级、程序执行模式等控制信息。

通常，系统提供两种处理机执行模式：核心模式和用户模式。这是为了保护操作系统程序(特别是其内核部分)免受用户程序的损害。当执行操作系统的程序时，CPU 处于核心态(也称为系统态、管理态)，具有较高的特权，可以执行机器指令集中的全部指令，包括一般用户程序中不能使用的特权指令，从而能访问所有寄存器和内存单元中的内容、启动 I/O 操作等。当 CPU 执行用户程序时，它处于用户态(也称为目态)，其权限较低，只能执行非特权指令。

2. 存储器

在现代计算机中，可以存放信息的部件很多，但它们在存取速度、容量和成本等方面有很大差别。一个存储器系统往往由各种部件构成一个层次结构，图 1-2 示出了三级存储器的结构。图中，左部文字说明存储器的性能、成本依箭头所示方向相对变化。

为了执行或者存取程序和数据，需要把它们放在内存中。但是在具体系统中，内存容量是有限的，而且其价格较昂贵。为了支持多道程序运行和大量的

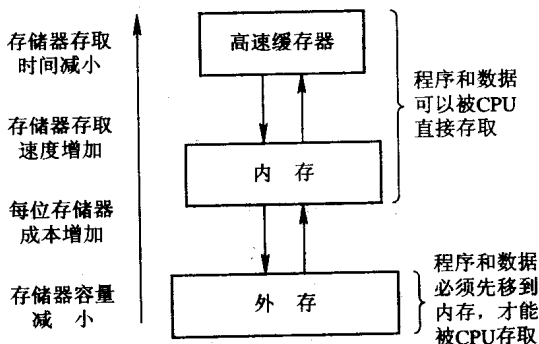


图 1-2 三级存储器结构

数据处理,就利用磁盘、磁带等容量巨大、价格(指每位存储)便宜的存储器作为对内存的后备支持。CPU 不能直接存取外存上的信息,但是内存和外存之间可以相互传递信息。也就是说,在外存上的程序和数据必须先移到内存中才能被 CPU 访问。内存中的程序和数据也可换到外存中,以腾出更多空间供众多进程使用。很显然,由于物理机制的限制,外存的存取速度低于内存的存取速度。

为了进一步提高运算速度和增强处理能力,可采用高速缓存器(cache)来存放程序和数据。高速缓存器是具有特殊目的的缓冲存储器,它比内存的容量小,但速度更快。它用来存放从内存中取出的指令和数据,这些信息很可能被 CPU 在以后直接使用。但是高速缓存器的成本远远高于内存的成本。

3. I/O 设备

I/O 设备是人机交互的工具,通常由控制器和设备本身组成。控制器是插板上的一块芯片或一组芯片,它接收来自操作系统的命令,然后执行它们,从而实际地控制设备的动作。在很多情况下,对设备的实际控制工作很琐碎繁杂,利用控制器就简化了操作系统与设备的接口。每个设备控制器只负责某类设备的控制,多台同类设备(如 10 台终端)可连到一个控制器上。所以,控制器是 I/O 设备的电子部分,它与操作系统交互作用,而设备本身并不与操作系统直接作用。

对磁盘控制器来说,其工作是把一系列二进制位流转换成字节块,并执行必要的错误校正。先在控制器内部的缓冲区中收集一个字节块,校验无错后再拷贝到主存中。

对 CRT 终端来说,控制器从内存中读取要显示的字符,然后产生信号,调节 CRT 的电子束,从而把字符写到屏幕上。控制器也调节电子束的水平回扫和垂直回扫。

每个控制器都有少量的寄存器,用于和 CPU 通信。有些计算机中,这些寄存器是普通存储地址空间的一部分,这种模式称为存储映像 I/O(Memory-mapped I/O),如 680X0 系列就采用这种方式。而另外的计算机使用专用地址空间表示 I/O,每个控制器分配确定的一部分,如 IBM PC 机中,各 I/O 控制器有其 I/O 地址和中断向量。

4. 总线

总线部件的功能是负责 CPU、存储器和设备控制器彼此间的信息或数据的传送的,是计算机内部的“公交车”。在现代计算机系统中都采用多总线结构,每条总线上传送专门的信息。主要有三类总线:数据总线——各部件之间传送数据的专用通道;地址总线——从 CPU 送来地址的地址线;控制总线——传送各个模块之间在传送数据时所需的全部控制信号。

1.2.2 计算机软件分类

按照所起的作用和需要的运行环境,软件通常可分为三大类,即应用软件、支撑软件 and 系统软件。应用软件是为解决某一类应用需求或某个特定问题而设计的程序,如图形软件、财务软件、软件包等等,这是范围很广的一类软件。支撑软件是辅助软件技术人员从事软件开发工作的软件,如各种开发工具、测试工具等,所以又称为工具软件,借以提高软件生产率,改善软件产品质量。系统软件包括操作系统、编译程序、汇编程序、连接装配程序、数据库管理系统、网络软件等,这种软件对计算机系统的资源进行控制、管理,并为

用户使用和其他程序的运行提供服务。

计算机系统中硬件和软件以及各类软件之间是按层次结构组织的，如图 1-3 所示。

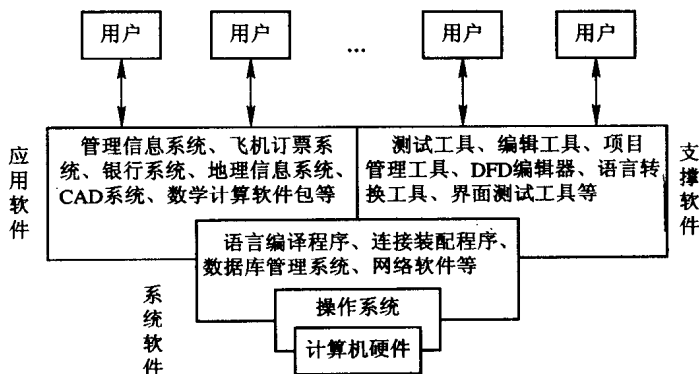


图 1-3 计算机系统的层次关系

由图 1-3 中可以看出，操作系统是裸机之上的第一层软件，与硬件关系尤为密切。它不仅对硬件资源直接实施控制、管理，而且其很多功能的完成是与硬件动作配合实现的，如中断系统。操作系统的运行需要有良好的硬件环境。这种硬件配置环境往往称做硬件平台。

操作系统是整个计算机系统的控制管理中心，其他所有软件都建立在操作系统之上。操作系统对它们既具有支配权力，又为其运行建造必备环境。因此，在裸机之上每加一层软件后，用户看到的就是一台功能更强的机器，通常把经过软件扩充功能后的机器称为“虚拟机”。在裸机上安装了操作系统后，就为其他软件 and 用户提供了工作环境。往往把这种工作环境称做软件平台。

1.3 操作系统的概念

我们所使用的计算机都安装了操作系统。大家所熟悉的操作系统有 Windows 98、Windows 2000、Windows NT、UNIX、Linux、OS/2、DOS 等。虽然操作系统已存在并且应用了很多年，但它的定义还没有一个统一的表述。按照多数专家的意见，可以给出如下定义：

操作系统是控制和管理计算机系统内各种硬件和软件资源、有效地组织多道程序运行的系统软件（或程序集合），是用户与计算机之间的接口。

为了深入理解操作系统的定义，我们应注意以下几点：

(1) 操作系统是系统软件，而且是裸机之上的第一层软件。操作系统由一整套程序组成。如 Linux 系统是用 C 语言编写的一个很大的程序，由上千个模块组成。在这些模块中有的负责 CPU 调度，有的负责内存管理，有的负责文件管理，等等。另外，操作系统是直接安装在裸机之上的，而其他所有的软件，无论是用户软件，还是系统工具、应用程序等，都必须建立在某个操作系统的环境之上。

(2) 操作系统的基本职能是控制和管理系统内的各种资源，有效地组织多道程序的运

行。操作系统本来就是资源管理程序，由它管理计算机系统各种资源，控制、协调各用户对这些资源的利用，尽可能地发挥资源的作用。作为“管理者”，操作系统主要负责如下事情：

- ① 监视各种资源并随时记录它们的状态；
- ② 实施某种策略以决定谁获得资源，何时获得，获得多少；
- ③ 分配资源供需求者使用；
- ④ 回收资源，以便再分配。

(3) 设置操作系统的另一个目的是扩充机器功能以方便用户使用。计算机系统的基本资源包括硬件(如处理机、内存、各种设备等)、软件(系统软件和应用软件)和数据。操作系统要保证整个系统在运行时各个用户对这些资源的需求，提供一台比裸机功能更强、且易于用户使用的机器。也就是说，用户面前所使用的计算机是经操作系统进行功能扩充之后的虚拟机，其特性不同于作为其运行基础的物理机器。设想一下：如果你所用的机器上没有装入操作系统，那么你无法使用命令，不能利用应用程序，连设备、内存等都需要你亲自管理，那将是一种多么“可怕”的局面呀！从而也看出操作系统的重要存在价值。

看待操作系统的观点有多种，主要是用户观点和系统观点。

从计算机用户的角度来看，操作系统是计算机裸机的扩充。它处于用户与计算机硬件系统之间，为用户提供了功能更强、使用更方便的“虚拟机”。

以系统内部实现的角度来看，操作系统是一种层次化、模块化结构的程序集合，对系统内的所有资源实施统一的控制、调度和管理。它详细记录资源的使用情况，按一定的策略进行调度分配，尽量提高资源的利用率，同时保证系统工作的安全、可靠。

1.4 操作系统的主要功能

通过理解操作系统可以“干什么”就更容易明白操作系统“是什么”。综合众多因素，可以看出，现代操作系统的主要功能有五个方面：存储器管理、处理机管理、设备管理、文件管理和用户接口。

1. 存储器管理功能

存储器管理的主要功能包括：内存分配、地址映射、内存保护和内存扩充。

1) 内存分配

内存分配的主要任务是为每道程序分配一定的内存空间。为此，操作系统必须记录整个内存的使用情况，处理用户提出的申请，按照某种策略实施分配，接收系统或用户释放的内存空间。

由于内存是宝贵的系统资源，并且往往出现这种情况：用户程序和数据对内存需求量的总和大于实际内存可提供的使用空间。为此，在制订分配策略时应考虑到提高内存的利用率，减少内存浪费。

2) 地址映射

大家都有这种经历：我们在编写程序时并不考虑程序和数据要放在内存的什么位置，程序中设置变量、数组和函数等只是为了实现这个程序所要完成的任务。源程序经过编译之后，会形成若干个目标程序，各自的起始地址都是“0”(但它并不是实际内存的开头地

址!），各程序中用到的其他地址都分别相对起始地址来计算。这样一来，在多道程序环境下，用户程序中所涉及的相对地址与装入内存后实际占用的物理地址就不一样了。CPU 执行用户程序时，要从内存中取出指令或数据，为此就必须把所用的相对地址（或称逻辑地址）转换成内存的物理地址。这就是操作系统的地址映射功能（需要有硬件支持）。

3) 内存保护

不同用户的程序都放在一个内存中，必须保证它们在各自的内存空间中活动，不能相互干扰，更不能侵犯操作系统的空间。为此，就必须建立内存保护机制。例如，设置两个界限寄存器，分别存放正在执行的程序在内存中的上界地址值和下界地址值。当程序运行时，所产生的每个访问内存的地址都要作合法性检查，就是说该地址必须大于或等于下界寄存器的值，并且小于上界寄存器的值。如果地址不在此范围内，则属于地址越界，将发生中断并进行相应处理。

另外，还要允许不同用户程序共享一些系统的或用户的程序。

4) 内存扩充

一个系统中内存容量是有限的，不能随意扩充其大小。而且用户程序对内存的需求越来越大，很难完全满足用户的要求。这样就出现各用户对内存“求大于供”的局面，怎么办？物理上扩充内存不妥，就采取逻辑上扩充内存的办法，这就是虚拟存储技术。简单说来，就是把一个程序当前正在使用的部分（不是全体）放在内存，而其余部分放在磁盘上。在这种“程序部分装入内存”的情况下，启动并执行它。以后根据程序执行时的要求和内存当时使用的情况，随机地将所需部分调入内存；必要时还要把已分出去的内存回收，供其他程序使用（即内存置换）。

2. 处理机管理功能

我们都知道，计算机系统最重要的资源是 CPU，对它管理的优劣直接影响整个系统的性能。此外，用户的计算任务称为作业；程序的执行过程称为进程，在传统操作系统中，它是分配和运行处理机的基本单位。因而，处理机管理的功能包括：作业和进程调度、进程控制和进程通信。

(1) 作业和进程调度。一个作业通常要经过两级调度才得以在 CPU 上执行。首先是作业调度，它把选中的一批作业放入内存，并分配其他必要资源，为这些作业建立相应的进程。然后进程调度按一定的算法从就绪进程中选出一个合适进程，使之在 CPU 上运行。

(2) 进程控制。进程是系统中活动的实体。进程控制包括创建进程、撤消进程、封锁进程、唤醒进程等。

(3) 进程通信。多个进程在活动过程中彼此间会发生相互依赖或者相互制约的关系。为保证系统中所有进程都能正常活动，就必须设置进程同步机制，它分为同步方式和互斥方式。

相互合作的进程之间往往需要交换信息，为此系统要提供通信机制。

3. 设备管理功能

设备管理的主要功能包括：缓冲区管理、设备分配、设备驱动和设备无关性。

(1) 缓冲区管理。缓冲区管理的目的是解决 CPU 和外设速度不匹配的矛盾，从而使它们能充分并行工作，提高各自的利用率。

(2) 设备分配。根据用户的 I/O 请求和相应的分配策略,为该用户分配外部设备以及通道、控制器等。

(3) 设备驱动。实现 CPU 与通道和外设之间的通信。由 CPU 向通道发出 I/O 指令,后者驱动相应设备进行 I/O 操作。当 I/O 任务完成后,通道向 CPU 发中断信号,由相应的中断处理程序进行处理。

(4) 设备无关性。又称设备独立性,即用户编写的程序与实际使用的物理设备无关,由操作系统把用户程序中使用的逻辑设备映射到物理设备中。

4. 文件管理功能

文件管理功能应包括:文件存储空间的管理、文件操作的一般管理、目录管理、文件的读写管理和存取控制。

(1) 文件存储空间的管理。系统文件和用户文件都要放在磁盘上。为此,需要由文件系统对所有文件以及文件的存储空间进行统一管理;为新文件分配必要的外存空间,回收释放的文件空间,提高外存的利用率。

(2) 文件操作的一般管理。包括文件的创建、删除、打开、关闭等。

(3) 目录管理。包括目录文件的组织、实现用户对文件的“按名存取”,以及目录的快速查询和文件共享等。

(4) 文件的读写管理和存取控制。根据用户的请求,从外存中读取数据或者将数据写入外存中。为保证文件信息的安全性,防止未授权用户的存取或破坏,对各文件(包括目录文件)进行存取控制。

5. 用户接口

我们上机操作时直接使用操作系统提供的用户接口。操作系统对外提供了多种服务,使得用户可以方便、有效地使用计算机硬件和运行自己的程序。现代操作系统通常向用户提供三种类型的界面:程序界面(系统调用)、命令界面和图形界面。

1) 系统调用

系统调用是操作系统提供给编程人员的接口,因而也称为程序员界面。用户程序要想得到操作系统的服务,必须使用系统调用(或机器提供的特定指令),它们能改变处理机的执行状态:从用户态变为系统态。

系统调用是操作系统内核与用户程序、应用程序之间的接口。在 UNIX/Linux 系统上,系统调用是以 C 函数的形式出现的。所有内核之外的程序都必须经由系统调用才能获得操作系统的服务。系统调用只能在程序中使用,不能作为命令在终端上执行。由于系统调用能直接进入内核操作,所以其执行效率很高。

例如,你要打开数据文件 mydata,且只对它进行读操作,那么在相应的 C 程序中必须写上下面的语句:

```
fd=open("mydata", O_RDONLY);
```

其中,fd 是一个被称为“文件描述字”的小整数,O_RDONLY 是表示打开文件后操作方式的标志参数,它表示只读。

在不同的操作系统上,系统调用的数量、格式是有差别的,有的系统提供几十条系统调用,有的系统提供 100 多条。从操作系统内部实现的层次结构上看,系统调用处于核心