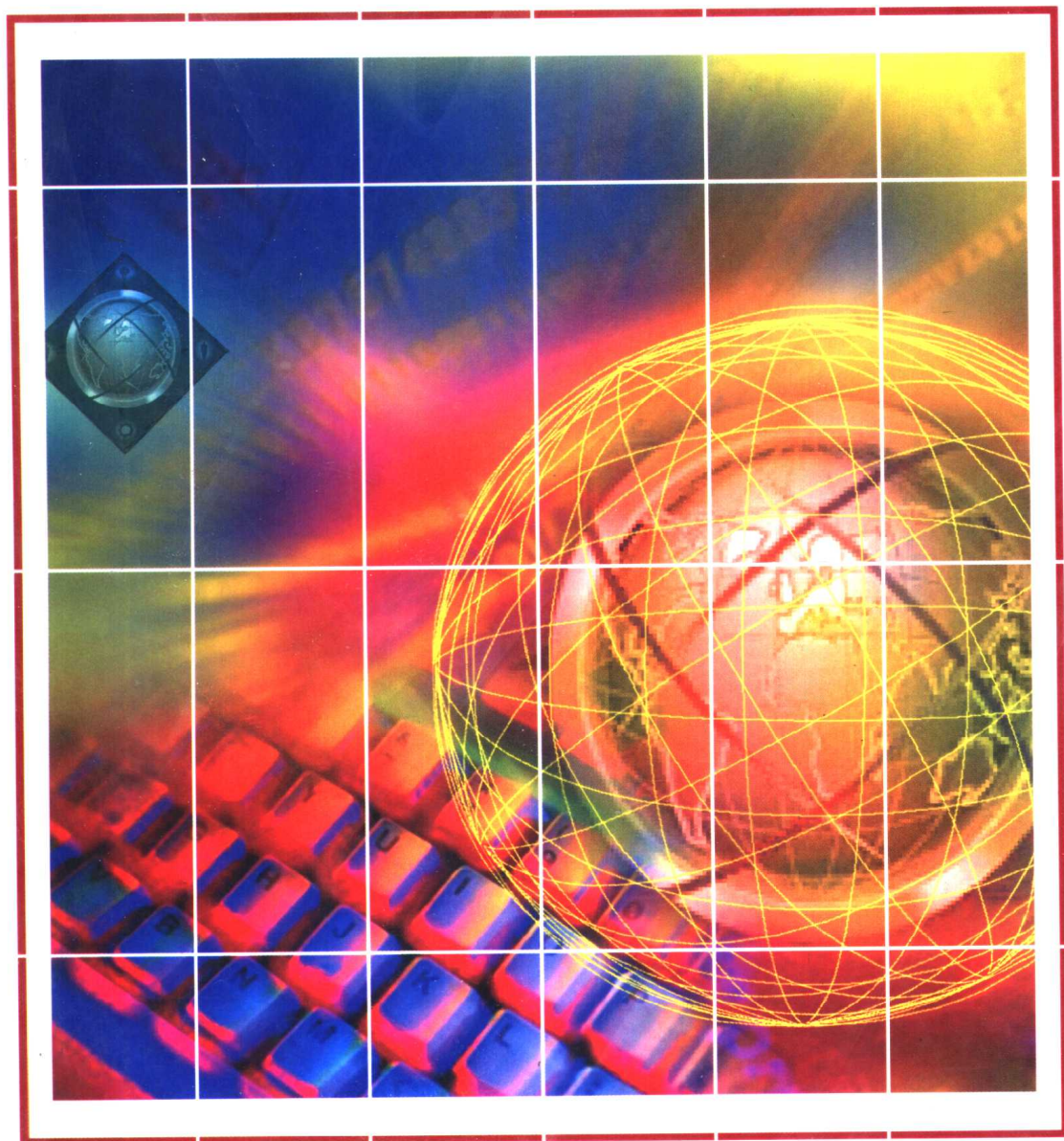


新世纪计算机类本科系列教材



计算机操作系统教程

徐甲同 陆丽娜 谷建华 编著



西安电子科技大学出版社
<http://www.xduph.com>

□ 新世纪计算机类本科系列教材

计算机操作系统教程

徐甲同 陆丽娜 谷建华 编著

西安电子科技大学出版社

2001

内 容 简 介

本书是由西安电子科技大学、西安交通大学和西北工业大学三校教师联合编写的“操作系统”课程的教科书。全书共八章，第一章为操作系统概论，第二章为操作系统与用户的界面；第三章至第六章分别为进程管理、存储管理、文件系统和设备管理；第七章为 Linux 系统；第八章介绍网络和操作系统环境下的编程及应用。

本书以先进性、简明性和实用性为编写的指导原则。全书体系合理、内容充实、结构清晰，便于教学。

本书适合作为高等学校计算机专业本科教材，也可作为非计算机专业本科、研究生的教学参考书，还可供工程技术人员学习操作系统时参考。

图书在版编目(CIP)数据

计算机操作系统教程/徐甲同等编著. —西安:西安电子科技大学出版社, 2001.2

新世纪计算机类本科系列教材

ISBN 7-5606-0990-2

I. 计… I. 徐… III. 操作系统(软件)—高等学校—教材 N. TP316

中国版本图书馆 CIP 数据核字(2001)第 00461 号

责任编辑 陈宇光 夏大平

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)8227828

邮编: 710071

<http://www.xduph.com>

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 空军西安印刷厂

版 次 2001 年 2 月第 1 版 2001 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 17.25

字 数 406 千字

印 数 1~6000 册

定 价 18.00 元

ISBN 7-5606-0990-2/TP·0890

* * * 如有印装问题可调换 * * *

本书封面贴有西安电子科技大学出版社的激光防伪标志, 无标志者不得销售。

前 言

本书是由西安电子科技大学、西安交通大学和西北工业大学三校教师联合编写的，本书集中了三个学校操作系统课程教学和实践的经验，从三个学校关于操作系统教学实际需要出发，吸取了国内外操作系统教材的成功经验，经过我们多次反复研究、讨论，最后形成了本书。

本书编写本着简明性、先进性和实用性的原则，即原理叙述力求简明，方法技术反映先进，习题例题体现实用。为做到这一点，首先，我们在原理部分引进了线程、微内核和客户/服务器等新概念；其次，我们把 Windows NT 所采用的新技术分散到各章作为实例来讲述，而把当前最流行的操作系统 Linux 专门单列一章，其目的在于帮助读者建立起操作系统的整体与动态概念；最后，为强调操作系统的实用性，在本书的最后一章给出了操作系统环境下用户编程和具体应用的实例。

全书共八章。第一章为操作系统概论，介绍了操作系统中的基本概念，描述了操作系统的概貌。第二章为操作系统与用户之间的界面，介绍了用户如何使用操作系统的方法。第三章至第六章依次讲述操作系统的进程/线程管理、存储管理、文件系统和输入输出系统。第七章为 Linux 系统简介。第八章是操作系统环境下用户编程及应用。

本书的第一、三、四章由徐甲同教授编写，第二、七章由陆丽娜教授编写，第五、六章由谷建华副教授编写，第八章由冯锐老师编写。最后由徐甲同教授统一修改、编排并定稿。

本书适合作为高等学校计算机相关专业本科教材，参考教学时数为50~60学时，若时数不够，则第七、八两章可以不讲，第七章可在讲课教师的指导下组织学生自学，第八章可作为学生上机实习时参考。

本书在编写过程中曾得到西安市各高校计算机操作系统授课老师的大力支持，借此机会，我们向在编写和出版过程中给予帮助的同志表示衷心感谢。在编写本书的过程中给予热心帮助和大力支持的有：西安电子科技大学的王长山、王亚平、方敏，西安交通大学的杨麦顺、张晓亮等老师，他们对本书的编写提出了不少宝贵的意见和建议。在本书出版过程中得到了西安电子科技大学出版社，尤其是李荣才、陈宇光等同志的热心帮助和大力支持，才使本书尽快与读者见面。

由于时间仓促，加之编写水平有限，书中难免存在一些错误和缺点，殷切希望广大读者批评指正。

编者

2000年12月

目 录

第一章 操作系统概论		第二章 用户与操作系统的接口	
1.1	什么是操作系统	2.1	作业控制级接口
1.1.1	操作系统作为最基本的系统软件	2.1.1	作业、作业步、作业流
1.1.2	操作系统作为资源管理器	2.1.2	作业的类型
1.1.3	操作系统作为虚拟机	2.1.3	作业控制命令与键盘操作命令
1.2	多道程序设计的概念	2.2	程序级接口
1.2.1	多道程序设计的硬件支持	2.2.1	管态与算态
1.2.2	多道程序设计原理	2.2.2	特权指令与访管指令
1.2.3	多道程序设计的实现	2.2.3	系统调用
1.3	操作系统的功能和主要特征	2.2.4	系统调用类型
1.3.1	操作系统的功能	2.2.5	系统调用的使用和执行过程
1.3.2	操作系统的主要特征	2.3	UNIX 的用户界面 Shell 简介
1.4	操作系统的结构	2.3.1	Shell 命令语言
1.4.1	操作系统的内核	2.3.2	Shell 过程
1.4.2	传统的操作系统结构设计模式	2.4	小结
1.4.3	现代的操作系统的结构设计模式	习题	
1.5	操作系统的分类		
1.5.1	单用户(微机)操作系统		
1.5.2	批处理系统		
1.5.3	分时系统		
1.5.4	实时系统		
1.5.5	网络操作系统		
1.5.6	分布式操作系统		
1.5.7	多处理机操作系统		
1.6	小结		
习题			

第三章 进程管理

3.1	为什么要引入进程的概念
3.1.1	从顺序程序设计谈起
3.1.2	程序的并发执行和资源共享
3.1.3	程序并发执行的特性
3.1.4	进程概念的引入
3.2	进程的表示和调度状态
3.2.1	进程的表示
3.2.2	进程的调度状态
3.3	进程的控制
3.3.1	进程的控制机构

3.3.2	进程控制原语	53
3.4	进程调度	54
3.4.1	交通控制程序和进程调度程序	54
3.4.2	进程调度算法的设计	55
3.4.3	常用的进程调度算法	57
3.4.4	作业、进程和程序之间的区别和联系	59
3.5	线程及其管理	60
3.5.1	线程概念的引入	60
3.5.2	什么是线程	60
3.5.3	Windows NT 中的进程和线程	61
3.6	进程通讯	63
3.6.1	进程间的同步和互斥	63
3.6.2	信号量和 P、V 操作	66
3.6.3	高级通讯原语	73
3.7	死锁	77
3.7.1	死锁的起因及产生死锁的必要条件	78
3.7.2	死锁举例	79
3.7.3	对死锁采取的对策	83
3.7.4	死锁的预防	83
3.7.5	死锁的避免	84
3.7.6	系统模型	87
3.7.7	死锁的检测	88
3.7.8	死锁的解除	90
3.8	小结	90
	习题	91

第四章 存储管理

4.1	存储管理的基本概念	95
4.1.1	存储管理研究的课题	95
4.1.2	地址再定位	96
4.1.3	虚拟存储器概念的引入	97
4.2	早期的存储管理	98
4.2.1	单一连续分配	98
4.2.2	分区分配	99
4.3	分页存储管理	107
4.3.1	分页原理	107
4.3.2	地址变换机构	108
4.3.3	分页存储管理算法	111
4.3.4	分页存储管理方案的评价	112
4.4	请求分页存储管理	113
4.4.1	请求分页原理	113
4.4.2	页面置换算法	116
4.4.3	性能分析	118
4.4.4	请求分页存储管理方案的评价	121
4.5	分段存储管理	122
4.5.1	分段原理	122
4.5.2	段变换表	123
4.5.3	分段存储管理方案的评价	125
4.6	段页式存储管理	127
4.6.1	段页式存储管理的实现	127
4.6.2	段页式存储管理的评价	129
4.7	Windows NT 虚拟内存管理	129
4.7.1	进程的虚拟地址空间	129
4.7.2	虚拟存储的实现	130
4.8	小结	132
	习题	133

第五章 文件系统

5.1	文件系统概述	135
5.1.1	文件和文件系统	135
5.1.2	文件的类型	136
5.1.3	文件系统的基本功能	137
5.2	文件的结构和存取法	138
5.2.1	文件的逻辑结构	138
5.2.2	文件的物理结构	138
5.2.3	文件的存取方法	142
5.2.4	文件结构、文件存储设备和存取法的关系	143
5.3	文件目录	144
5.3.1	简单的文件目录	144
5.3.2	二级目录	145
5.3.3	多级目录	146
5.3.4	文件目录项的组织	147
5.4	文件存储空间的管理	149
5.4.1	空白文件目录	149
5.4.2	空白块链	150
5.4.3	位示图(Bit Map)	150
5.4.4	MS-DOS 的盘空间的管理	150
5.4.5	UNIX 文件存储空间的管理	151
5.5	文件的共享	152
5.5.1	目录结构中的共享	152
5.5.2	打开文件结构中的共享	154

5.5.3 管道文件(Pipe)	155
5.6 文件的存取控制	158
5.6.1 文件存取控制法	158
5.6.2 文件系统的安全性	161
5.7 文件系统和用户间的接口	162
5.7.1 文件的创建和删除	162
5.7.2 文件的打开和关闭	163
5.7.3 文件的读写	164
5.8 小结	164
习题	165

第六章 输入/输出系统

6.1 I/O 系统的硬件结构	166
6.1.1 I/O 设备类型	166
6.1.2 I/O 设备的物理特性	167
6.1.3 I/O 系统的硬件组织	173
6.2 采用通道模型的 I/O 系统	176
6.2.1 通道类型	176
6.2.2 多通路 I/O 系统	176
6.2.3 通道命令和通道程序	177
6.2.4 CPU 和通道间的通讯	179
6.3 I/O 系统的软件组织	180
6.3.1 I/O 软件设计的目标	180
6.3.2 中断处理程序	181
6.3.3 设备驱动程序	181
6.3.4 与设备无关的 I/O 软件	182
6.3.5 用户空间的 I/O 软件	182
6.3.6 I/O 系统软件的层次结构	183
6.4 缓冲技术	184
6.4.1 单缓冲	184
6.4.2 双缓冲	185
6.4.3 缓冲池	185
6.4.4 预先读与延迟读	187
6.5 磁盘的驱动调度	187
6.6 设备分配程序	189
6.6.1 设备分配的数据结构	189
6.6.2 I/O 调度程序	191
6.6.3 设备分配的实施	192
6.7 Windows NT 的 I/O 系统	194
6.7.1 I/O 系统的结构	194
6.7.2 驱动程序模型	195
6.7.3 异步 I/O 操作	196

6.8 小结	198
习题	198

第七章 Linux 操作系统简介

7.1 Linux 简介	200
7.1.1 Linux 的发展过程	200
7.1.2 Linux 内核结构	202
7.1.3 Linux 运行模式、地址空间与 上下文	205
7.2 Linux 进程管理	206
7.2.1 Linux 进程控制块结构	206
7.2.2 进程调度算法和调度策略	208
7.2.3 进程使用的文件	209
7.2.4 进程使用的虚拟内存	210
7.2.5 系统调用	211
7.2.6 进程的创建与终止	212
7.3 进程间的通讯机制	212
7.3.1 信号(Signal)	213
7.3.2 管道(Pipe)	213
7.3.3 消息队列	214
7.3.4 信号量	215
7.3.5 共享内存	216
7.4 Linux 存储管理	216
7.4.1 虚拟内存的实现机理	216
7.4.2 80386 体系结构的存储管理 功能	216
7.4.3 Linux 分页管理机制	217
7.4.4 空闲物理内存空间管理	218
7.4.5 虚拟段的组织	219
7.4.6 共享内存	220
7.4.7 请求换页与页面换入	221
7.4.8 交换空间	221
7.4.9 换出与丢弃页面	222
7.4.10 存储管理系统的高速缓冲 机制	222
7.5 Linux 文件系统	223
7.5.1 Linux 文件系统的特点	223
7.5.2 EXT2 文件系统物理结构	224
7.5.3 EXT2 索引节点	224
7.5.4 EXT2 超级块	225
7.5.5 EXT2 组描述符	226
7.5.6 EXT2 目录	226

7.5.7	EXT2 文件查找	226	7.6.4	设备驱动程序的结构	242
7.5.8	EXT2 文件扩展策略	227	7.7	小结	245
7.5.9	虚拟文件系统(VFS)概述	227	习题	246
7.5.10	VFS 超级块	228	第八章 操作系统环境下的编程及应用		
7.5.11	VFS 索引节点	228	8.1	应用程序编程接口概述	247
7.5.12	文件系统的注册	229	8.2	进程间通讯实现方法与实例	248
7.5.13	文件系统的挂接与卸装	229	8.2.1	管道	248
7.5.14	文件系统管理的缓存机制	230	8.2.2	System V IPC 机制	252
7.5.15	bdflush 内核守护程序	232	8.2.3	套接字	260
7.6	Linux 设备管理	232	8.3	线程编程及实现方法	266
7.6.1	设备驱动程序环境	233			
7.6.2	设备文件	238			
7.6.3	设备驱动程序概述	240			

第一章 操作系统概论

计算机科学与技术发展到今天,从个人计算机到巨型计算机,乃至超级计算机,无一例外地都配置了一种或多种操作系统。如果要让用户去使用一台没有操作系统的计算机,那将是不堪想像的。那么什么是操作系统呢,操作系统在计算机系统中的地位以及操作系统的结构、功能和主要特性,将在本章作一简要叙述。首先,我们从什么是操作系统讲起。

1.1 什么是操作系统

关于操作系统,至今尚无严格统一的定义。对操作系统的定义有各种说法,人们从不同的角度,用不同的观点去揭示操作系统的本质特征。为了更好地理解操作系统,我们来分析一下,操作系统“是什么”,它“能做什么”。

1.1.1 操作系统作为最基本的系统软件

1. 计算机系统资源

一个计算机系统由两部分构成:系统硬件和系统软件。系统硬件是指构成计算机系统所必须配置的全部设备。现代计算机系统一般都包含一个或多个处理器、内存、磁盘、光盘、打印机、时钟、鼠标、键盘、显示器、网络接口以及其它输入输出设备。计算机系统硬件构成了计算机本身和用户作业赖以活动的物质基础。通常,我们把计算机系统中所配置的硬件称为硬件资源。只有系统硬件而无系统软件的计算机称为裸机。用户直接使用裸机不仅不方便,而且将严重降低系统效率。系统软件是一个计算机系统必须配置的程序和数据的集合。它们是专门为计算机系统所配置的,例如操作系统、各种语言处理程序(汇编程序、编译程序等)、编辑程序、连接装配程序、系统实用程序以及系统维护程序等,这些都可算是系统软件,但是最为重要、最为基本的系统软件便是我们所要讲的操作系统了。

由此可见,操作系统本身是计算机系统软件中的重要组成部分,而其物质基础是系统硬件。系统硬件和系统软件统称为计算机系统资源。

2. 计算机系统的层次结构

一个计算机系统可以看成是由硬件和软件按层次结构组成的系统,如图 1.1 所示。

从图 1.1 可以看出,整个计算机系统由四层构成,其中每一层代表一组功能并提供相

应的接口。所谓接口，就是掩盖该层内部功能的实现细节，向外部提供一些使用的约定。

硬件层(即裸机)包含一个具有一组指令系统的处理器、可供访问的存储器以及系统的 I/O 结构，它是操作系统设计和工作的基础。

操作系统层是硬件层的第一次扩充，在这一层上实现了操作系统的全部功能，并提供了相应的接口。语言处理程序层包括各种程序设计语言的编译程序以及诸如文本编辑程序、动态调试程序等实用性程序。语言处理程序层是操作系统层的扩充，而应用程序层是语言处理程序层的进一步扩充。用户可以使用各种程序设计语言，在操作系统的支持下，编写并运行满足用户需要的各种应用程序。

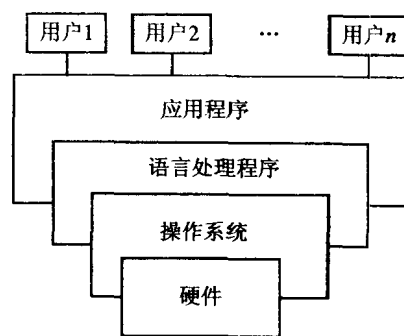


图 1.1 计算机系统的层次结构

1.1.2 操作系统作为资源管理器

上面我们谈到，一个计算机系统的资源分为硬件资源和软件资源。操作系统的任务就是如何管理这些资源，使系统资源得到充分有效的利用，并且在相互竞争的作业或程序之间有序地控制系统资源的分配，从而实现对计算机系统工作流程的控制。

操作系统作为计算机系统资源管理器的观点是目前人们对操作系统认识的一种主要观点。在这种观点之下，操作系统的首要任务是要跟踪资源的使用状况，满足用户程序对资源的请求，提高系统资源利用率，协调各程序对资源的使用冲突。具体地说，资源管理器的作用是：

(1) 跟踪资源状态。时刻保持系统资源分配的全局信息，掌握系统资源的种类和数量、已分配和未分配的情况。

(2) 分配资源。处理对资源的使用请求，协调请求中的冲突，确定资源分配算法。当有多个用户争用某个资源时，进行裁决。同时，根据资源分配的条件、原则和环境决定是立即分配还是暂缓分配。

(3) 回收资源。用户程序在资源使用完了之后要释放资源。此时，资源管理器应及时回收资源，以便下次重新分配。

(4) 保护资源。资源管理器负责对资源进行保护，防止资源被有意或无意地破坏。

系统资源的使用方法和策略决定了操作系统的规模、类型、功能与实现，基于这一点，就可以把操作系统看成是由一组资源管理器(即资源管理程序)所组成的。

计算机系统软、硬件资源数量庞大、种类繁多，因此有必要对资源进行抽象研究，找出各种资源的共性，用“资源”这一统一的概念研究资源的使用方法和策略，以便寻求一种资源管理的普遍原则和系统方法。

采用资源管理器的观点，人们将计算机系统资源划分为四大类，处理机、存储器、I/O 设备和信息(程序和数据)。针对这四大类资源，可以为操作系统建立相应的四类管理器：处理机管理、存储管理、设备管理和信息管理(通常指文件系统)。目前，多数操作系统教材的内容也是按这四个方面来组织的。

1.1.3 操作系统作为虚拟机

在一个计算机系统的四个层次组成中，最低层是硬件。对多数计算机而言，在机器语言一级的体系结构(包括指令系统、存储组织、I/O 和总线结构)上编程是相当困难的，尤其是输入输出操作。

为了让用户和程序员在使用计算机时不涉及硬件细节，使硬件细节和程序员隔离开来，需要建立一种简单的高度抽象。这种抽象就是为用户提供一台等价的扩展计算机，这样的计算机称为虚拟计算机，简称虚拟机。

操作系统作为虚拟机为用户使用计算机提供了方便，用户可不必了解计算机硬件工作的细节。用户通过操作系统来使用计算机，操作系统就成了用户和计算机之间的接口。

用户如何通过操作系统使用计算机呢？操作系统为用户提供了两级接口：命令接口和编程接口。有了这两级接口，用户就可以不涉及硬件的实现细节，方便而有效地取得操作系统为用户所提供的各种服务，合理地组织计算机工作流程。

现在我们回到操作系统定义的讨论上来，它“是什么”，它是计算机系统中最重要最基本的软件。也就是说，它是程序和数据的集合。它“能做什么”，它作为计算机系统的资源管理器，控制、管理计算机系统中的所有软、硬件资源；它作为虚拟机，为用户使用计算机提供了方便。于是，可以这样给操作系统下一定义：操作系统是控制和管理计算机系统的硬件和软件资源，合理地组织计算机工作流程及方便用户使用的程序和数据的集合。

在计算机系统中设置操作系统的目的在于提高计算机系统的效率，增强系统的处理能力，充分发挥系统资源的利用率，方便用户的使用。至于操作系统“如何做”，这正是本书其余各章所要讲述的内容。

1.2 多道程序设计的概念

现代计算机系统(包括部分个人计算机)一般都采用基于多道程序设计的技术。通常多道程序设计是指在主存中同时存放多道用户作业，使它们都处于执行的开始点和结束点之间。

1.2.1 多道程序设计的硬件支持

多道程序设计技术的出现，得到了来自计算机硬件的两方面支持：中断系统和通道技术。

1. 中断系统

什么是中断？从系统操作的观点来看，中断的定义如下：

中断指的是：① 对异步或例外事件的一种响应；② 这一响应自动地保存 CPU 状态以便将来重新启动；③ 自动转入中断处理程序。

中断的引进，最初是为了实现外部设备和 CPU 的并行工作，但这一概念到后来被大大地扩展了。现在，在系统中发生的需要处理机暂停正在执行的程序转而进行所要干预的所有事件，都要通过中断机构进行处理。这些事件(称为中断源)的类型与计算机的系统结构以及操作系统的设计关系极大。在计算机系统中，一般将中断分为如下几种类型：

- (1) I/O 中断。这是外部设备完成了预定 I/O 操作或在 I/O 操作中出错所引起的中断。
- (2) 程序中断。这是由程序中的错误引起的中断。
- (3) 硬件故障中断，或称机器检验中断。
- (4) 外中断。外中断来自外部信号，这些信号可能来自其它机器。外中断还包括时钟中断，以及来自键盘的中断。
- (5) 访管中断。由机器中的访管指令引起的中断。

用户程序在执行过程中，当中断发生时，硬件的中断机构首先判断发生的中断类型，保护当前的处理机状态，以便将来进行恢复。其次，根据中断类型自动转入预先规定好的中断处理程序。最后，当中断处理程序结束时，恢复现场并返回被中断的程序。中断的处理过程如图 1.2 所示。图中的 PSW 为程序状态字，用来控制指令的执行顺序、保存和指示与当前正在执行的程序有关的系统状态，即所说的处理机现场。每一类型的中断都有与之相关的两个程序状态字，即旧 PSW 和新 PSW。中断前的程序状态字称为旧 PSW，中断后的程序状态字称为新 PSW。

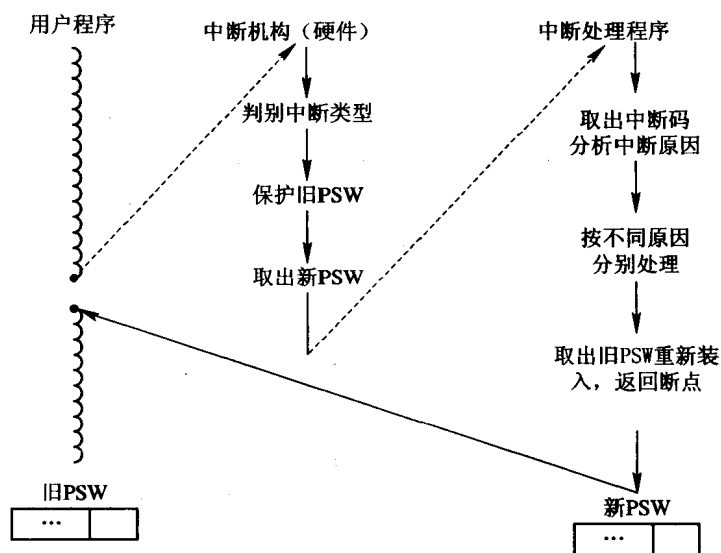


图 1.2 中断的处理过程

2. 通道技术

为了获得 CPU 与外设之间更高的并行能力，也为了让种类繁多、物理特性各异的外设能以标准的接口方式连到系统中，计算机系统中引入了各自独立体系的通道结构。

通道又称 I/O 处理机，它能完成主存和外设之间的信息传输，并与中央处理器并行操作。采用通道技术实现了 I/O 操作的独立性和各部件工作的并行性。通道把 CPU 从繁琐的输入输出操作中解放出来。采用通道技术后，不仅能实现 CPU 与通道的并行操作，而且通道与通道之间也能实现并行操作，各通道上的外设也能实现并行操作，从而可达到提高计算机系统效率的根本目的。

在具有通道结构的计算机系统中，主存、通道、控制器和设备之间采用四级连接，实现三级控制，如图 1.3 所示。

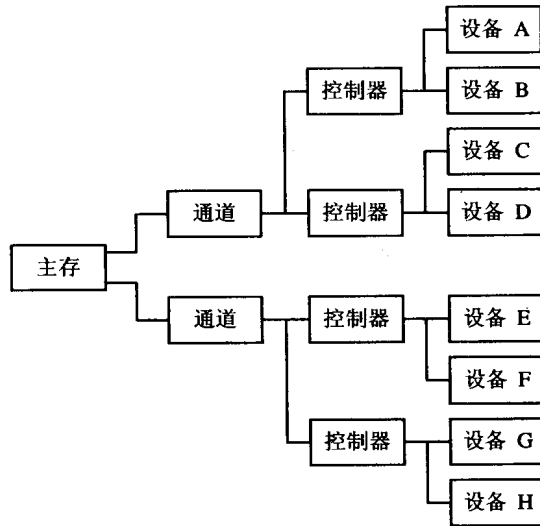


图 1.3 单通路 I/O 系统

通常，一个 CPU 的主存可以连接若干通道，一个通道可以连接若干个控制器，一个控制器又可连接若干台设备，即所谓四级连接。三级控制是指 CPU 执行 I/O 指令实施对通道的控制，通道执行通道命令对控制器实施控制，控制器控制设备执行相应的输入输出操作。

3. CPU 和通道的通信

CPU 与通道之间的关系是主从关系。CPU 是主设备，通道是从设备。采用通道方式实现数据传输的过程如下：

- (1) 当运行的程序要求数据传输时，CPU 向通道发 I/O 指令，命令通道工作；
- (2) 通道接收到 CPU 的 I/O 指令后，从内存中取出相应的通道程序，通过执行通道程序完成 I/O 操作；
- (3) 当 I/O 操作完成(或出错)时，通道以中断方式中断 CPU 正在执行的程序，请求 CPU 的处理。

1.2.2 多道程序设计原理

现在让我们考察一个例子。设有两道作业程序 A 和 B，要在同一台处理机上运行。假设程序 A 在其执行中要进行 I/O 操作(或者要求从外设读入数据到主存缓冲区，或者将主存缓冲区中的数据写入外设)，后继程序要求等到 I/O 操作完成后才能执行。图 1.4 给出了作业程序 A、B 的执行过程如下：

- ① 处理机从这里开始执行程序 A 中的指令；
- ② 程序 A 在这里提出 I/O 请求；
- ③ 程序 A 提出 I/O 请求后，启动通道开始数据传输；
- ④ 在处理机启动通道后，通道开始 I/O 操作，而处理机却转去执行程序 B；
- ⑤ 通道数据传输结束，产生 I/O 中断；
- ⑥ 停止程序 B 的执行；
- ⑦ 返回到程序 A，从这里开始继续执行程序 A 的后继指令。

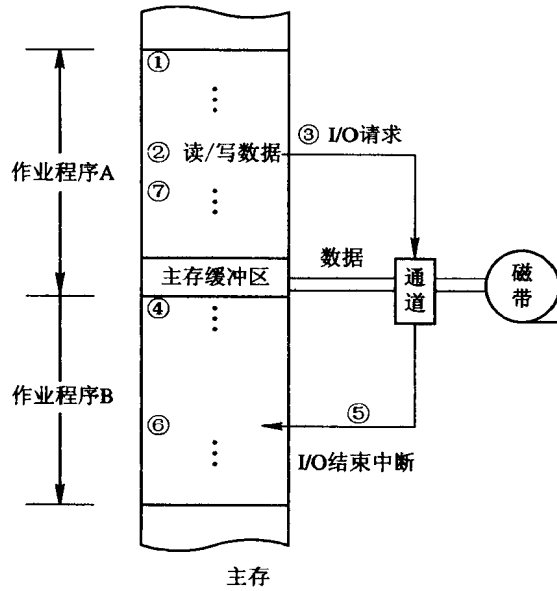


图 1.4 多道程序运行概念图

上述执行过程可用时间图来描述，见图 1.5。

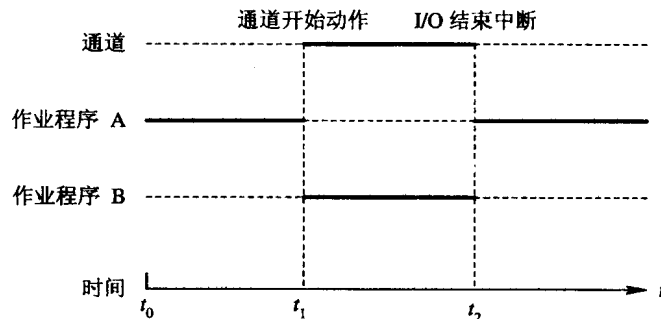


图 1.5 多道程序执行过程的时间图

从图 1.5 可以看出，在通道进行数据传输期间，CPU 并没有空等，为了提高系统的效率，CPU 执行主存中另一道准备好了的程序 B。CPU 在执行程序 B 期间，如果 I/O 操作完成，便接到从通道来的 I/O 结束中断，使 CPU 停止程序 B 的执行，返回程序 A，继续执行 A 的后继指令。

多道程序设计的主要目的是充分利用系统的所有资源且尽可能地让它们并行操作。这种技术可把硬件的代价交叉地分布在大量并行用户之间而使计算机系统的代价极小化。

1.2.3 多道程序设计的实现

分析上面的例子，为实现多道程序设计，必须妥善解决以下三个问题：

(1) 存储保护和地址重定位。在多道程序设计环境下，在主存中的几道程序它们共享同一主存，硬件必须提供必要的手段，防止各道程序相互侵犯，特别是当某道程序出错时，

不致影响其它程序。

由于多道程序同时存放在同一主存中，并且某一程序运行结束，新的程序又要进入主存，因此程序员无法知道他的程序在主存的确切位置，这就要求程序或部分程序能随机地从一个主存区域移到另一主存区域，而不影响程序的执行结果。这就是所谓的“地址重定位”或“程序浮动”问题。

(2) 处理机管理和调度。多道程序共享同一处理机，就有处理机分配的问题。在同一时刻只能由一道程序占有处理机，则说它处于“运行”状态；当一个程序等待某个事件发生时，则说它处于“阻塞”状态；当一个程序可以运行但尚未得到处理机时，则说它处于“就绪状态”。主存中的每道程序在任一时刻必定处于上述三种状态之一。

在多道程序设计环境下，可能有几道程序都处于就绪状态，处理机调度程序按照某种规则，譬如按程序的优先级调度，它选择一个优先级最高的处于就绪状态的程序投入运行，该程序占有处理机后，由就绪状态转入运行状态。处于运行状态的程序当等待某一事件发生时，便进入阻塞状态；处于阻塞状态的程序当所等待的事件发生后，又进入了就绪状态，以便由处理机调度程序重新调度。

图 1.6 给出一个例子，说明了三个程序按优先级调度时，各程序的执行情况及程序状态的转换。这里假定程序 A 的优先级最高，程序 B 次之，程序 C 最低。

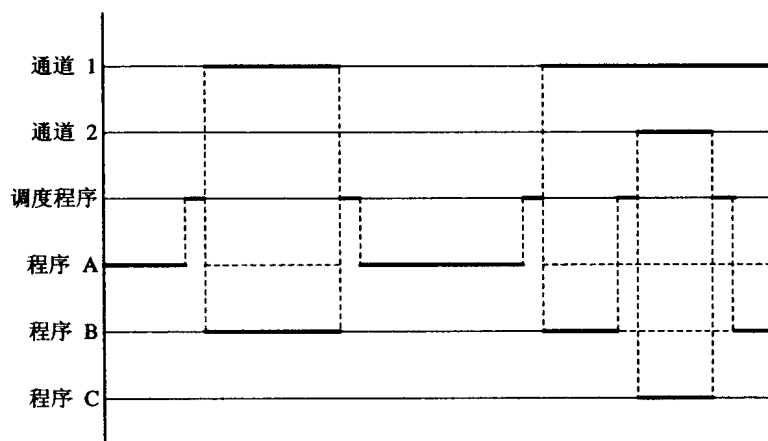


图 1.6 多道程序设计环境下各程序的执行和状态的转换

(3) 资源的管理和分配。在多道程序设计环境下，系统资源为多道程序所共享。为此，必须对系统中的资源实施有效的管理和分配，有关管理和分配的细节将在本书后续章节中讲述。

综上所述，可归纳多道程序设计的特点如下：

(1) 多道，即主存中有两道或两道以上的程序，它们都处于执行的开始点和结束点之间，也就是说，它们在任一时刻必处于就绪、运行、阻塞三种状态之一。

(2) 宏观上并行。从宏观上看，它们在同时执行。

(3) 微观上串行。从微观上看，它们在交替、穿插地执行，因为在任一时刻，在一台处理机上只能执行一道程序的一条指令。

采用多道程序设计后，减少了 CPU 时间的浪费。对计算型作业，由于 I/O 操作较少，

CPU 浪费的时间很少；然而对于 I/O 型作业，例如商业数据处理，I/O 时间通常占到 80%~90%，采用多道程序设计将会得到明显的效果。如果主存中存放足够多的作业，可使 CPU 的利用率接近 100%。当然程序的道数也不是越多越好。

1.3 操作系统的功能和主要特征

1.3.1 操作系统的功能

操作系统作为计算机系统的资源管理器，其主要任务是对系统中的硬件、软件实施有效的管理，以提高系统资源的利用率。计算机硬件资源主要是指处理机、主存储器和外部设备，软件资源主要是指信息(文件系统)。因此，操作系统的主要功能相应地就有处理机管理、存储管理、设备管理和信息管理。操作系统又可作为一台虚拟机，从虚拟机的观点、操作系统为用户提供了用户接口。下面将这些功能简述如下：

1. 处理机管理

在单道作业或单用户的环境下，处理机为一个作业或一个用户所独占，对处理机的管理十分简单。但在多道程序或多用户的环境下，要组织多个作业同时运行，就要解决处理机的管理问题。在多道程序环境下，处理机的分配和运行都是以进程为基本单位的，因而对处理机的管理可归结为对进程的管理。它包括以下几方面：

(1) 进程控制。进程控制的主要任务是为作业创建进程，撤消已结束的进程以及控制进程在运行过程中的状态转换。

(2) 进程调度。进程调度的任务是从进程的就绪队列中，按照一定的算法选择一个进程，把处理机分配给它，并为它设置运行现场，使之投入运行。

(3) 进程同步。为使系统中的进程有条不紊地运行，系统必须设置进程同步机制，以协调系统中各进程的运行。

(4) 进程通信。系统中的各进程之间有时需要合作，往往要交换信息，为此需要进行通信。

2. 存储管理

存储管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，并提高主存的利用率。存储管理包括以下几个方面：

(1) 地址重定位。在多道程序设计环境下，每个作业是动态装入主存的，作业的逻辑地址必须转换为主存的物理地址，这一转换称为地址重定位。

(2) 存储分配。存储管理的主要任务是为每道程序分配内存空间，在作业结束时要收回它所占用的空间。

(3) 存储保护。保证每道程序都在自己的主存空间运行，各道程序互不侵犯，尤其是不能侵犯操作系统。

(4) 存储扩充。一般来说，主存的容量是有限的。在多道程序设计环境下往往感到主存容量不能满足用户作业的需要。为此操作系统存储管理的任务是要扩充主存容量，但这种扩充是逻辑上的扩充，可以通过建立虚拟存储系统来实现。

3. 设备管理

一个计算机系统的硬件，除了 CPU 和主存，其余几乎都属于外部设备，外部设备种类繁多，物理特性相差甚大。因此，操作系统的设备管理往往很复杂。设备管理主要包括：

(1) 缓冲管理。由于 CPU 和 I/O 设备的速度相差很大，为缓和这一矛盾，通常在设备管理中建立 I/O 缓冲区，而对缓冲区的有效管理便是设备管理的一项任务。

(2) 设备分配。当用户程序提出 I/O 请求后，设备管理程序要依据一定的策略和系统中设备情况，将所需设备分配给它，设备用完后还要及时收回。

(3) 设备处理。设备处理程序又称设备驱动程序，对于未设置通道的计算机系统其基本任务通常是实现 CPU 和设备控制器之间的通信。即由 CPU 向设备控制器发 I/O 指令，要求它完成指定的 I/O 操作，并能接收由设备控制器来的中断请求，给予及时的响应和相应的处理。对于设置了通道的计算机系统，设备处理程序还应能根据用户的 I/O 请求，自动构造通道程序。

(4) 设备独立性和虚拟设备。设备独立性是指应用程序独立于物理设备，使用户编程与实际使用的物理设备无关。

虚拟设备的功能是将低速的独占设备改造为高速的共享设备。

4. 文件管理

处理机管理、存储管理和设备管理都属于硬件资源的管理。软件资源的管理称为信息管理，即文件管理系统。

现代计算机系统中，总是把程序和数据以文件的形式存储在文件存储器中(如磁盘、光盘、磁带等)供所有用户或指定用户使用。为此，操作系统必须配置文件管理机构。文件管理的主要任务是对用户文件和系统文件进行管理，并保证文件的安全性。文件管理包括以下内容：

(1) 目录管理。为方便用户在文件存储器中找到所需文件，通常由系统为每一文件建立一个目录项，包括文件名、属性以及存放位置等，由若干目录项又可构成一个目录文件。目录管理的任务是为每一文件建立其目录项，并对目录项施以有效的组织，以方便用户按名存取。

(2) 文件读、写管理。文件读、写管理是文件管理的最基本功能。文件系统根据用户给出的文件去查找文件目录，从中得到文件在文件存储器上的位置，然后利用文件读、写指针，对文件进行读、写。

(3) 文件存取控制。为了防止系统中的文件被非法窃取或破坏，在文件系统中应建立有效的保护机制，以保证文件系统的安全性。

(4) 文件存储空间的管理。所有的系统文件和用户文件都存放在文件存储器上。文件存储空间管理的任务是为新建文件分配存储空间，在一个文件被删除后应及时释放所占用的空间，文件存储空间管理的目标是提高文件存储空间的利用率，并提高文件系统的工作速度。

5. 用户接口

前面四项都是以资源管理的观点来看操作系统应提供哪些功能。操作系统作为资源管理器提供了处理机管理、存储管理、设备管理和信息管理这四大功能。这里，我们只是简要地介绍了这四项功能，有关详细内容及功能的实现，将在本书的第三章至第六章讲述。