

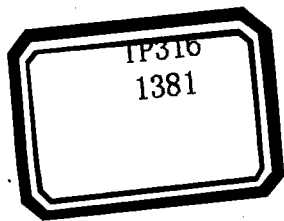
JISUANJICAOZUOXITONGJIAOCHENG

# 计算机操作系统 教程

王万森 汪国安 涂相华 编著



河南大学出版社



计算机教材

# 计算机操作系统教程

王万森 汪国安 涂相华

计算机操作系统教程

王万森 汪国安 涂相华

河南大学出版社

计算机操作系统教程

王万森 汪国安 涂相华

河南大学出版社

计算机操作系统教程

王万森 汪国安 涂相华

河南大学出版社

(豫)新登字 09 号

内 容 简 介

本教材是根据 1993 年全国高等学校计算机教育研究会推荐的计算机本科教学计划操作系统教学大纲,结合作者多年教学经验编写而成。本教材注重吸取了国内外较新的操作系统理论和技术,舍弃了现有操作系统教材中的一些陈旧概念和方法,反映了当代操作系统发展的新动向、新水平。

全书共分为四部分:第一部分从第一章到第九章,依据资源观点讨论操作系统的基本原理和技术;第二部分包括第十、十五两章,讨论操作系统的结构设计;第三部分包括第十一、十二两章,讨论操作系统的最新发展;第四部分包括第十三、十四章,分别介绍当代最流行的 UNIX 操作系统和 90 年代最新推出的 WINDOWS NT 操作系统的内部结构和用户界面,使读者能对操作系统建立起动态、整体的概念。

本教材内容叙述由浅入深,难点分散,体系合理,便于教学,适合作为高等院校计算机有关专业本科教材,经剪裁(删去带“\*”的章节),也可作为计算机专科教材,还可供科技、工程技术人员学习操作系统时阅读参考。

计算机操作系统教程

王万森 汪国安 涂相华 编著

责任编辑 汪远征

河南大学出版社出版

(开封市明伦街 85 号)

河南省新华书店发行

河南大学出版社电脑排版

河南新郑印刷厂印刷

开本:787×1092 毫米 1/16 印张 19.5 字数:450 千字

1995 年 8 月第 1 版 1995 年 8 月第 1 次印刷

印数:1—3500

定价:14.30 元

ISBN7-81041-205-1/T·5

## 前 言

操作系统是现代计算机系统中必不可少的重要系统软件。操作系统课程是研究操作系统基本原理和实现方法的一门计算机专业主干课程。

我们结合多年的操作系统教学实践,并根据 1993 年高等学校计算机教育研究会推荐的计算机本科教学计划操作系统教学大纲,编写了这本计算机操作系统教程。

全书共分为以下四大部分:第一部分从第一章到第九章,依据资源观点讨论操作系统的基本原理和技术;第二部分包括第十、十五两章,讨论操作系统的结构设计;第三部分包括第十一、十二两章,讨论操作系统的最新发展;第四部分包括第十三、十四两章,讨论两个操作系统实例。全书各章的安排分别为:第一章是操作系统概述,其目的是让读者先对操作系统有一个概括的了解,并初步建立起操作系统的整体概念;第二章是作业管理;第三章是进程及处理机管理;第四章是实存管理;第五章是虚存管理;第六章是设备管理;第七章是文件管理;第八章是并发进程的协调;第九章是死锁;第十章是操作系统结构设计;第十一章是多处理机操作系统;第十二章是网络操作系统与分布式操作系统;第十三章是 UNIX 操作系统,以 System V 为背景;第十四章是 Windows NT 操作系统;第十五章是面向对象的操作系统。为使本书能尽量反映操作系统的新进展,书中适当地充实了一些国内外较新操作系统的概念和方法。

本书是作为计算机专业本科教材编写的。同时,也考虑到专科教学的需要,对个别章节加了星号(\*)。去掉加星号部分(不影响内容的连续性),便是一本完整的专科教材。全书各章之后都附有一定数量的习题,这些习题可以帮助学生加深对所学内容的理解。

书中各章的编写分工是:王万森编写了第一、三、八、十、十一、十二、十三、十四、十五章;汪国安编写了第四、五、六、七章;涂相华编写了第二、九章。各自的书稿完成后,我们又一起讨论并通审了全书。

在本书编写之前,我们都主讲过不少版本的操作系统教材,这些教材不管对以往的教学,还是对本书的编写,都有很大的帮助。在此,谨向这些教材的作者深表谢意。

本书的编写曾得到河南大学计算机系、河南师范大学计算机系、信阳师范学院计算机系领导的积极支持,谨向他们表示衷心感谢。

在本书的编辑、出版过程中,河南大学出版社的领导和职工同志们付出了辛勤的劳动;责任编辑汪远征同志做了认真细致的编辑工作,并对书稿提出了许多宝贵意见;激光照排中心的刘健同志绘制了书中的不少图表,在此,亦一并向他们深致谢意。

由于作者水平有限,错误和不妥之处在所难免,恳请广大同行和读者不吝赐教。

编者 1995. 1. 20.

# 目 录

<b>第一章 操作系统概述</b> .....	( 1 )
§ 1.1 什么是操作系统 .....	( 1 )
§ 1.2 操作系统的形成过程 .....	( 3 )
§ 1.3 操作系统的基本类型 .....	( 6 )
§ 1.4 操作系统的特征和功能 .....	(11)
§ 1.5 操作系统的进一步发展 .....	(14)
§ 1.6 历史上重要的操作系统简介 .....	(17)
§ 1.7 中断在操作系统中的地位 .....	(19)
§ 1.8 操作系统的装入和初启 .....	(19)
习题一 .....	(20)
<b>第二章 作业管理</b> .....	(21)
§ 2.1 作业管理概述 .....	(21)
§ 2.2 作业管理的功能 .....	(22)
§ 2.3 作业控制 .....	(25)
§ 2.4 作业调度 .....	(31)
习题二 .....	(42)
<b>第三章 进程管理</b> .....	(43)
§ 3.1 进程的基本概念 .....	(43)
§ 3.2 进程状态及其转换 .....	(48)
§ 3.3 进程控制块和进程队列 .....	(49)
§ 3.4 进程控制 .....	(52)
§ 3.5 进程调度 .....	(56)
习题三 .....	(67)
<b>第四章 实存管理</b> .....	(69)
§ 4.1 引言 .....	(69)
§ 4.2 单一连续区存储管理 .....	(74)
§ 4.3 分区式存储管理 .....	(75)
§ 4.4 覆盖和交换 .....	(85)
§ 4.5 分页存储管理 .....	(87)
§ 4.6 分段存储管理 .....	(93)
习题四 .....	(97)
<b>第五章 虚存管理</b> .....	(98)
§ 5.1 虚拟存储系统的基本概念 .....	(98)
§ 5.2 请求分页存储管理 .....	(99)

§ 5.3	动态分段存储管理	(112)
§ 5.4	段页式存储管理	(114)
习题五		(118)
<b>目 录</b>		
第六章	设备管理	(120)
§ 6.1	设备管理概述	(120)
(1) § 6.2	输入输出系统结构	(121)
(1) § 6.3	外设的启动和I/O中断处理	(124)
(1) § 6.4	驱动调度	(128)
(1) § 6.5	虚拟设备	(132)
(1) § 6.6	设备分配程序	(134)
(1) 习题六		(140)
第七章	文件管理系统	(142)
(1) § 7.1	文件管理系统概述	(142)
(1) § 7.2	文件和文件系统	(142)
(1) § 7.3	文件的结构和存取方法	(145)
(1) § 7.4	文件目录	(151)
(1) § 7.5	文件存储空间的管理	(156)
(1) § 7.6	文件的存取控制	(158)
(1) § 7.7	文件系统和用户的接口	(161)
(1) 习题七		(163)
第八章	并发进程的协调	(164)
(1) § 8.1	并发的条件和描述	(164)
(1) § 8.2	进程的互斥和同步	(166)
(1) § 8.3	经典的互斥和同步问题	(174)
(1) § 8.4	信号量集机制	(178)
(1) § 8.5	管程	(179)
(1) § 8.6	进程通信	(183)
(1) 习题八		(188)
第九章	死锁	(190)
(1) § 9.1	死锁概述	(190)
(1) § 9.2	产生死锁的原因和必要条件	(191)
(1) § 9.3	死锁的预防	(195)
(1) § 9.4	死锁的避免	(197)
(1) § 9.5	死锁的检测	(201)
(1) § 9.6	死锁的解除	(204)
(1) 习题九		(205)
第十章	操作系统结构设计	(207)
(1) § 10.1	引言	(207)
(1) § 10.2	常用的操作系统结构设计方法	(209)

(§§ 10.3) 管程设计方法 .....	(212)
(习题十) .....	(214)
<b>第十一章 多处理机操作系统</b> .....	<b>(215)</b>
§ 11.1 多处理机操作系统的概念 .....	(215)
*§ 11.2 多处理机调度 .....	(218)
*§ 11.3 多处理机系统的进程同步 .....	(222)
*§ 11.4 多处理机进程通信 .....	(225)
习题十一 .....	(227)
<b>*第十二章 网络操作系统和分步式操作系统</b> .....	<b>(228)</b>
§ 12.1 计算机网络概述 .....	(228)
§ 12.2 网络操作系统概述 .....	(232)
§ 12.3 分布式系统概述 .....	(235)
§ 12.4 分布式资源共享 .....	(236)
§ 12.5 分布式事件定序 .....	(237)
§ 12.6 分布式同步 .....	(239)
§ 12.7 分布式系统的死锁 .....	(241)
§ 12.8 分布式文件系统 .....	(242)
§ 12.9 健壮性 .....	(243)
习题十二 .....	(244)
<b>第十三章 UNIX 操作系统</b> .....	<b>(246)</b>
§ 13.1 UNIX 操作系统的简史和基本结构 .....	(246)
§ 13.2 UNIX 的用户接口 .....	(247)
§ 13.3 UNIX 进程管理 .....	(249)
§ 13.4 UNIX 存储管理 .....	(259)
§ 13.5 UNIX 设备管理 .....	(263)
§ 13.6 UNIX 文件管理 .....	(268)
§ 13.7 UNIX 进程通信 .....	(274)
习题十三 .....	(280)
<b>第十四章 Windows NT 操作系统</b> .....	<b>(281)</b>
§ 14.1 Windows NT 的设计目标和系统结构模型 .....	(281)
§ 14.2 Windows NT 的进程和线程 .....	(284)
§ 14.3 虚拟内存管理 .....	(286)
§ 14.4 Windows NT 输入/输出系统 .....	(288)
§ 14.5 Windows NT 网络 .....	(292)
习题十四 .....	(292)
<b>*第十五章 面向对象的操作系统</b> .....	<b>(293)</b>
§ 15.1 面向对象概述 .....	(293)
§ 15.2 面向对象和操作系统 .....	(295)
§ 15.3 面向对象操作系统的分类 .....	(298)

§ 15.4 Windows NT 的面向对象机制 ..... (299)

习题十五 ..... (302)

- (18) ..... 进程控制块 ..... 299
- (19) ..... 进程的创建 ..... 299
- (20) ..... 进程的调度 ..... 300
- (21) ..... 进程的同步 ..... 300
- (22) ..... 进程的通信 ..... 300
- (23) ..... 进程的终止 ..... 300
- (24) ..... 进程的优先级 ..... 300
- (25) ..... 进程的共享 ..... 300
- (26) ..... 进程的互斥 ..... 300
- (27) ..... 进程的临界区 ..... 300
- (28) ..... 进程的自举 ..... 300
- (29) ..... 进程的初始化 ..... 300
- (30) ..... 进程的清理 ..... 300
- (31) ..... 进程的回收 ..... 300
- (32) ..... 进程的统计 ..... 300
- (33) ..... 进程的跟踪 ..... 300
- (34) ..... 进程的调试 ..... 300
- (35) ..... 进程的测试 ..... 300
- (36) ..... 进程的验证 ..... 300
- (37) ..... 进程的部署 ..... 300
- (38) ..... 进程的更新 ..... 300
- (39) ..... 进程的备份 ..... 300
- (40) ..... 进程的恢复 ..... 300
- (41) ..... 进程的迁移 ..... 300
- (42) ..... 进程的复制 ..... 300
- (43) ..... 进程的克隆 ..... 300
- (44) ..... 进程的继承 ..... 300
- (45) ..... 进程的封装 ..... 300
- (46) ..... 进程的抽象 ..... 300
- (47) ..... 进程的接口 ..... 300
- (48) ..... 进程的兼容性 ..... 300
- (49) ..... 进程的兼容性 ..... 300
- (50) ..... 进程的兼容性 ..... 300
- (51) ..... 进程的兼容性 ..... 300
- (52) ..... 进程的兼容性 ..... 300
- (53) ..... 进程的兼容性 ..... 300
- (54) ..... 进程的兼容性 ..... 300
- (55) ..... 进程的兼容性 ..... 300
- (56) ..... 进程的兼容性 ..... 300
- (57) ..... 进程的兼容性 ..... 300
- (58) ..... 进程的兼容性 ..... 300
- (59) ..... 进程的兼容性 ..... 300
- (60) ..... 进程的兼容性 ..... 300
- (61) ..... 进程的兼容性 ..... 300
- (62) ..... 进程的兼容性 ..... 300
- (63) ..... 进程的兼容性 ..... 300
- (64) ..... 进程的兼容性 ..... 300
- (65) ..... 进程的兼容性 ..... 300
- (66) ..... 进程的兼容性 ..... 300
- (67) ..... 进程的兼容性 ..... 300
- (68) ..... 进程的兼容性 ..... 300
- (69) ..... 进程的兼容性 ..... 300
- (70) ..... 进程的兼容性 ..... 300
- (71) ..... 进程的兼容性 ..... 300
- (72) ..... 进程的兼容性 ..... 300
- (73) ..... 进程的兼容性 ..... 300
- (74) ..... 进程的兼容性 ..... 300
- (75) ..... 进程的兼容性 ..... 300
- (76) ..... 进程的兼容性 ..... 300
- (77) ..... 进程的兼容性 ..... 300
- (78) ..... 进程的兼容性 ..... 300
- (79) ..... 进程的兼容性 ..... 300
- (80) ..... 进程的兼容性 ..... 300
- (81) ..... 进程的兼容性 ..... 300
- (82) ..... 进程的兼容性 ..... 300
- (83) ..... 进程的兼容性 ..... 300
- (84) ..... 进程的兼容性 ..... 300
- (85) ..... 进程的兼容性 ..... 300
- (86) ..... 进程的兼容性 ..... 300
- (87) ..... 进程的兼容性 ..... 300
- (88) ..... 进程的兼容性 ..... 300
- (89) ..... 进程的兼容性 ..... 300
- (90) ..... 进程的兼容性 ..... 300
- (91) ..... 进程的兼容性 ..... 300
- (92) ..... 进程的兼容性 ..... 300
- (93) ..... 进程的兼容性 ..... 300
- (94) ..... 进程的兼容性 ..... 300
- (95) ..... 进程的兼容性 ..... 300
- (96) ..... 进程的兼容性 ..... 300
- (97) ..... 进程的兼容性 ..... 300
- (98) ..... 进程的兼容性 ..... 300
- (99) ..... 进程的兼容性 ..... 300
- (100) ..... 进程的兼容性 ..... 300



# 第一章 操作系统概述

今天,无论是巨、大、中、小、微型计算机系统,还是网络、多机、分布式计算机系统,都配置有相应的操作系统。操作系统已成为现代计算机系统的重要组成部分,它提供计算机用户与计算机硬件之间的接口,有效地控制和管理计算机系统的全部资源,合理地组织计算机系统的工作流程。

学习和掌握操作系统的基本原理和方法,是计算机科学技术发展的需要,它可以帮助我们宏观上理解计算机系统的动态工作过程。为了使读者尽快地对操作系统的整体概貌有一个初步了解,本章先从不同角度对操作系统进行较粗略的全面概述。这些内容同时也是以后各章学习的基础。

## § 1.1 什么是操作系统

每一个使用过计算机的人,几乎都知道和使用过操作系统,都得到过操作系统的服务与帮助。然而,要给操作系统下一个精确定义却并非那么容易。几十年来,人们从不同角度对操作系统有过许多种不同的解释,但至今仍无一个统一的定义。为了说明什么叫操作系统,下面是我们从诸多不同解释中选出的最具有代表性的几种观点。

### 1.1.1 操作系统是硬件系统功能的首次扩充

这是一种虚拟机观点,该观点认为操作系统是添加在硬件机器上的第一层软件,是对硬件系统功能的直接延伸。

众所周知,现代计算机系统由硬件和软件两大部分组成。软件部分是指用来管理计算机本身操作的系统软件和用来提供给用户解题的应用软件,硬件部分是指未配置任何软件的裸露的硬件机器(简称裸机)。裸机又可分为物理器件、微程序与机器语言三个层次。在这三个层次中,把物理器件称为硬件无可非议,但把微程序、机器语言划为硬件只是一种习惯。微程序是直接控制物理器件并为下一层软件提供清晰接口的原始软件,其本质上是一个解释机器语言的程序,它把机器语言分解为一系列的操作步。例如,执行一条 ADD 命令,微程序须首先确定加数的位置,然后取出并相加,最后再把所得结果存储到某个位置。由微程序解释的这一套机器指令规定了机器语言。机器语言实际上根本不是硬件机器的组成部分,只不过由于它总被放入机器手册的硬件部分进行描述,致使人们认为它是硬件机器的一部分。

如果计算机系统中没有配置操作系统,让用户直接使用裸机,那么就必須把对中央处理机、主存储器、时钟、终端、磁盘和其它输入输出(I/O)设备等管理和使用的细节全部交给用户去完成,这将使程序设计和计算机应用变得非常复杂和困难。正是为了避免这种实现细节上的复杂性给程序设计和计算机应用所带来的极大麻烦,人们才不断在裸机上添加软件,延伸裸机功能,构造基于裸机,但又比裸机更易于用户理解和编程的虚拟计算机(简称虚拟机)。这一层添加在裸机上的软件就是操作系统。可见,操作系统是对硬件系统

功能的首次扩充。

除操作系统以外的其它系统软件,如命令解释程序、编译程序、编辑程序等都位于操作系统之上。尽管它们也都属于系统软件,但操作系统是在核心态(或称管态)下运行的那一部分系统软件,硬件保护它免受用户程序的破坏;而命令解释程序、编译程序、编辑程序等则是在用户态(或称目态)下运行的那一部分软件,它们不受硬件保护。例如,用户可以自由修改某些编译程序,但无权去编一个磁盘中断处理程序,因为磁盘中断处理程序是操作系统的一部分,是受硬件保护的,用户不能随意修改它。至于应用程序,则处在所有系统软件之上。计算机系统的这种抽象结构如图1-1所示。

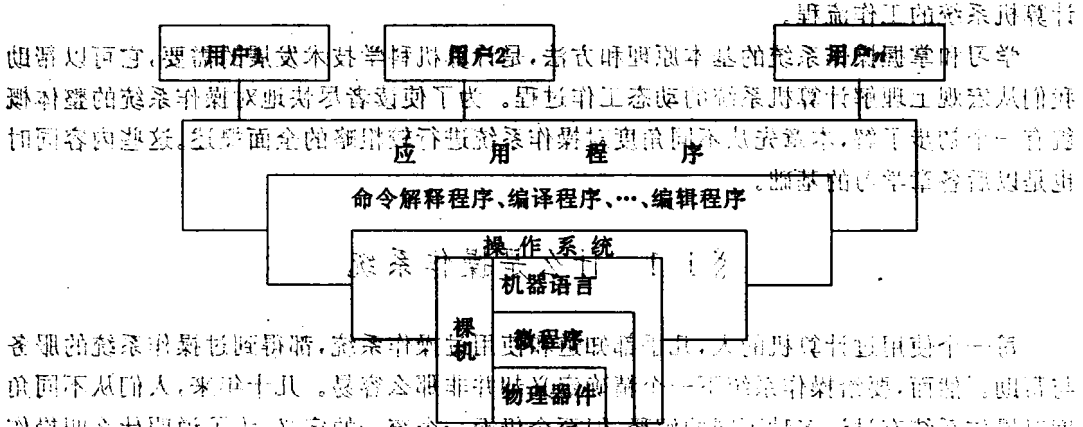


图 1-1 计算机系统的抽象层次结构

按照这种观点,其它系统软件和应用软件都全部建立在操作系统的基础上,操作系统实际上是向它们隐藏了硬件和硬件操作细节的真相,把它们对裸机的使用转化为对虚拟机的使用。

### 1.1.2 操作系统是整个系统资源的管理者

这是一种资源管理观点。该观点认为操作系统是管理计算机系统资源的软件,它负责控制和管理计算机系统的全部资源,确保资源能被高效合理的使用,系统得以有条不紊地运行。

现代计算机系统的系统资源可大致分为硬件资源和软件资源两大类。硬件资源是计算机系统硬件的总和,它包括中央处理机、主存储器、输入输出设备、辅助存储器等。软件资源也称信息资源,是系统中各种程序和数据的总和,这些程序和文件均以文件的形式保存在系统中。从资源管理观点出发,根据操作系统所管理的资源类型,可把整个操作系统分为处理机管理、存储管理、设备管理、作业管理和文件管理五大部分。它们分别负责 CPU 的运行和分配,主存储器的分配、回收、保护与扩充,输入输出设备的分配、回收与控制,作业的控制与调度,以及文件存储空间和文件信息的管理。所有这些管理,构成了操作系统的资源管理功能。

### 1.1.3 操作系统是用户与计算机之间的接口

这是一种用户服务观点。该观点认为操作系统是一个为用户服务的大型程序。从用户角度看,当计算机配置了操作系统后,用户不再直接使用系统硬件,而是利用操作系统所提供的命令和服务去使用计算机,这就把操作系统看成了用户与计算机之间的一个接

用户利用这个接口使用计算机的过程中所需要的公共服务主要包括：**① 处理好资源分配**（随着程序的装入与换程，各种程序的用户程序都必须被装入内存，然后才能在 CPU 上执行。

(2) 输入/输出操作 任何一个用户程序都需要通过输入设备输入数据，程序运行或运行结束时所得出的运算结果也都需从输出设备输出。此外，用户程序应该能够把自己需要保存的信息以某种形式存放在辅助存储器上，系统要为用户的这种要求提供相应的支持和服务，既要使用户能方便地建立、删除、存取和使用自己的文件，又要防止文件的泄密、被盗或丢失。

(4) 作业运行控制 用户希望对其作业的运行过程和测试过程实施某些控制，系统应该提供有效的手段，使用户能按照自己的意图控制其作业的运行。

(5) 错误检测和处理 用户程序在运行过程中有可能出现各种错误，如算术运算溢出、程序中出现非法指令访问地址越界等，操作系统应有能力对这些错误进行处理。此外，操作系统还应能及时地检测系统本身有可能出现的各种软硬件故障，以便修复和保证用户程序的顺利运行。上述服务功能是任何一个用户都需要的，是公共的和基本的，实现这些功能的程序就是操作系统。当然，操作系统不仅为用户程序的运行提供基本服务，也为其它系统程序（如编辑程序、编译程序等）的运行提供同样的服务。

## § 1.2 操作系统的形成过程

操作系统同其它任何事物一样，也有一个产生和发展的过程，为了帮助大家进一步理解操作系统，有必要对操作系统的形成过程作一简单回顾。在 50 年代以前的第一代计算机中没有操作系统，计算机只是由控制台控制的一个庞大的物理机器。当时，人们使用计算机的过程大致为：首先由程序员将其编好的程序从纸带或卡片上装入内存，然后通过控制台上的按钮或开关启动程序执行，最后当程序运行完毕时，取下纸带和运算结果，开始下一个用户程序。依此重复上述过程。这种手工操作方式有以下两个主要特点：**(1) 用户独占全机**，当第一个用户开始操作后，计算机中的全部资源都归他所有，直到运行结束时才把这些资源转交给下一个用户。这样，用户可以较方便地使用各种资源，不会出现因所需资源被其它用户占用而等待的现象，但是资源利用率却非常低。**(2) 操作联机**和输入输出联机，所谓操作联机是指当操作员在进行装卸卡、带等手工操作时 CPU 处于空闲状态，所谓输入输出联机是指当程序在进行输入或输出时，CPU 处于空闲状态。操作联机和输入输出联机使得 CPU 的利用率非常低。手工操作方式对早期的计算机来说矛盾尚不突出，原因是那时计算机本身所拥有的

资源量较少,计算速度也较慢。一般程序所需要的计算时间比其它时间(装卸带卡, I/O等)要长。但是,随着 CPU 速度的大幅度提高,存储容量的迅速增加以及外部设备的不断增多,上述矛盾就变得越来越尖锐,使得这种方式到了非改不可的地步。

## 1.2.2 批量处理阶段

为了解决上述问题,人们自然会首先想到摆脱从一个用户程序过渡到另一个用户程序时的手工干预,使其转换能自动进行。这就产生了由计算机对一批用户程序进行自动处理的所谓批量处理技术。在批量处理中,一个用户程序及其所需要的数据和命令被统称为一个作业。批量处理技术出现于 50 年代末的第二代计算机中,它又可分为以下两个阶段:

### 1. 早期批量处理

早期批量处理方式是把若干个用户作业集中起来组成一批作业,并在内存中放置一个监督程序,由监督程序来负责实现对这批作业的处理和从一个作业过渡到另一个作业时的自动转换。

在这种处理方式下,先由操作员把一批用户作业的卡片叠放在读卡机上,再由监督程序把这一批作业输入到磁带上。当该批作业的输入工作全部完成后,监督程序便开始对它们进行自动处理。其处理过程为:监督程序自动地把磁带上该批作业中的第一个调入内存,对它进行汇编或编译,并把汇编或编译结果装入内存、启动执行,运算结束时输出其计算结果。当第一个作业全部完成后,监督程序自动地将该批作业中的第二个作业调入内存,并重复以上过程,直到该批作业全部完成为止;在监督程序处理第一批作业的同时,操作员可将第二批作业的卡片叠放到读卡机上。当第一批作业全部处理完后,监督程序便可自动地再从读卡机上输入第二批作业,并重复上述过程,依次处理每一个作业。这样,监督程序不停地对一批批、一个个的作业进行处理,从而实现了作业之间的自动转换,解决了手工操作时期所存在的操作联机问题。

### 2. 脱机批量处理

在早期批量处理中,尽管解决了操作联机问题,但作业的输入输出仍然是联机的。也就是说,作业信息由卡片送到磁带,再由磁带调入内存,以及计算结果在打印机上输出,都是由 CPU 来处理的。这种 CPU 和 I/O 设备之间的串行工作方式大大降低了程序的执行速度。为解决这一问题,在批量处理中引入了脱机输入输出技术,形成了脱机批量处理方式。这种处理方式如图 1-2 所示,它是在早期批量处理的基础上新增加了一台功能较差的处理机。原来的那台处理机称为主处理机(简称主机),专门负责计算工作,不直接与输入输出设备交互作用;新增的这台处理机称为卫星机,专门负责输入输出工作。

在图 1-2 中,卫星机负责把读卡机上的作业输入到输入带上和把输出带上的计算结果从输出带上输出,主机直接从输入带上调入作业执行,并把计算结果送到输出带上。这样,作业的输入输出工作完全与主机脱离,并且卫星机的工作和主机的执行同时进行。因此,这种批量处理系统比早期的批量处理系统提高了处理能力。但是,它仍存在着一些根本问题没有解决。例如,监督程序、系统程序和用户程序之间是通过相互调用的办法进行转移的。这样,当目标程序企图执行一条非法指令时,整个系统就会停顿下来;若陷入死循环,则整个系统也无法继续下去。更为严重的是它无法防止用户程序冲掉一部分监督程序,潜伏着搞乱系统的危险。

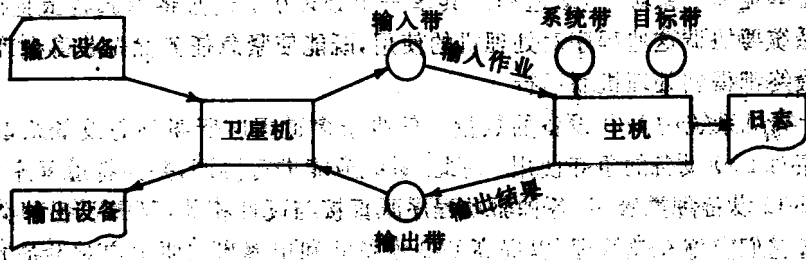


图 1-2 脱机批量处理

### 1.2.3 执行系统阶段

从 50 年代末到 60 年代初,硬件方面获得了两个重要进展:一是通道的引入,二是中断的出现。所谓通道,实质上是一台功能单一、结构简单的 I/O 处理机,它独立于 CPU,并直接控制外部设备与内存进行数据传送。CPU 启动通道后,通道就可独立工作,完成外部设备的启动、关闭以及信息传输等操作。只有在传送完规定的数据后,才向 CPU 发一次中断请求。所谓中断,开始是作为外部设备向中央处理机的“汇报”手段提出来的,即在输入输出结束时,或硬件发生某种故障时,由相应硬件向主机发出信号,主机立即停止正在执行的操作,转去处理中断请求。而在现代计算机系统中,中断概念已被大大扩展。

为了获得主机与外部传输在时间上的重叠,就必须提供中断处理程序和输入输出控制服务,这样就把原来的监督程序扩大到执行系统。执行系统的程序可分为三类: I/O 系统、处理程序、管理程序。执行系统的程序已相当庞大,若把它们全部放到内存,会大大减少用户程序的可用空间。为此,可仅让那些所有程序都将用到的中断处理程序和输入输出控制服务常驻内存,而其它部分放在外存。常驻内存的那部分程序称为执行程序。

执行程序对其它程序拥有控制权。用户程序的输入输出当然也是通过委托执行程序来实现的,系统对错误的 I/O 要求提供自动检查,受托程序完成之后,再用中断信号通知执行程序,以保证系统的安全。此外,用户程序发生死循环也可以通过时钟中断进行检测处理,非法操作也可通过非法操作中断得到及时处理。

### 1.2.4 多道程序设计和操作系统的形成

执行系统较好地解决了 CPU 和 I/O 操作的并行问题,但内存中仍仅能存放一道程序,资源独占问题仍未得到解决,并且当这个程序因等待 I/O 而不能继续执行时,CPU 必须处于空闲状态。为提高 CPU 的利用率,又引入了多道程序设计技术。

所谓多道程序设计,是指同时把几个作业放入内存中,并允许它们交替执行,共享系统中的各种硬、软件资源。当一道程序因 I/O 请求而暂停执行时,CPU 便转去执行另一道程序。这样,不仅使 CPU 能得到充分利用,同时还可以改善 I/O 设备和内存的利用率。允许多道程序运行的系统称为多道程序系统。多道程序系统的出现,标志着操作系统的形成。最早出现的多道程序系统是多道批处理系统,之后又出现了分时系统、实时系统等。

虽然多道程序系统有效地提高了系统资源利用率,但实现多道程序还需要妥善解决下述一系列问题:

(1) 在一个连续的内存空间中同时驻留多道程序,应为每道程序分配自己的内存空间,使它们既不会因相互重叠而丢失信息,又不会因某道程序出现异常而破坏其它程序。

(2) 在单处理机系统中,多道程序共享一个处理机,这必将引起各道程序对处理机的争夺。系统要协调这些程序对处理机的使用,既能使紧急任务优先获得处理机,又能使各道程序最终都得到处理机而运行。

(3) 计算机系统中I/O设备的数量一般少于多道程序所要求的设备总量,这也将引起多道程序对I/O设备的争夺使用。对此,系统应能进行协调,并为各道程序分配I/O设备。另外,I/O设备种类繁多,各种继续程序既直接与硬件相关,又需要各种类型的缓冲,系统应能对它们实施有效管理,以提高I/O设备的利用率和方便用户对设备的使用。

(4) 在一个系统中运行的程序可能具有不同的类型。例如,有的属于计算型,其计算量较大,但输入输出量却较小;有的属于I/O型,其计算量较小,但输入输出量却较大。此外,不同程序所要求的资源量及其紧急程度又各不相同,系统应组织好这些程序的运行,既保证系统有最大的吞吐量和最高的资源利用率,又不延误紧急程序的运行。

(5) 通常,在辅助存储器中以文件形式存放着大量的有用信息。为提高文件存储空间的利用率,加速对信息的检索速度,系统应对它们进行组织和管理。同时,为方便用户使用文件,系统还应提供存储和检索文件信息的手段。

在多道程序系统中,解决上述问题的组合程序就构成了操作系统。

### 1.3 操作系统的基本类型

操作系统诞生以来,一直随着计算机系统的发展而不断发展。目前,已有许多实用操作系统在不同的计算机上运行。如何对它们进行分类,有一些不同的方法。其中最常用的方法是根据操作系统的用户服务方式,把操作系统分为批量处理(操作)系统、分时(操作)系统和实时(操作)系统三种基本类型。

#### 1.1 批量处理系统

批量处理系统是指对用户作业按成批方式进行处理的操作系统。它的主要设计目标是提高系统吞吐率,提高系统的资源利用率,同时也兼顾作业的周转时间。

##### 1. 批量处理系统的类型

批量处理系统又可分为单道批处理系统、多道批处理系统和远程批处理系统。通常所说的批量处理系统多指后两者。

单道批处理系统是早期计算机系统中所配置的一种操作系统类型。它的主要特征有:第一,不区分作业调度和进程调度,只须按作业在辅存上排定的顺序,依次将它们调入内存,并分配处理机,使之运行即可;第二,内存中任何时刻最多只能有一道作业在运行,仅当前一道作业完成,或出现不能处理的异常情况时,才会调度后一道作业进入主存运行;第三,作业完成的顺序由作业进入主存的先后顺序来确定,在无异常情况下先进入主存的作业先完成。

多道批处理系统是在60年代中期以后出现的一种操作系统类型。它的主要特征有:第一,从作业进入系统到它在处理机上执行须经过作业调度和进程调度这两级调度。作业调度主要是决定辅存作业队列中哪个作业能被送入主存,进程调度则主要是决定主存中哪个作业的进程能获得中央处理机;第二,内存中可同时驻留多道作业,对单处理机系统,宏观上有多道程序在同时运行,但微观上任何时刻都最多只能有一道程序在处理机上执

行。第五，作业完成后的先后顺序按作业进入的先后顺序并无严格的对应关系，而是进入内存的排队顺序。第六，作业完成后的先后顺序按作业进入的先后顺序并无严格的对应关系，而是进入内存的排队顺序。第七，作业完成后的先后顺序按作业进入的先后顺序并无严格的对应关系，而是进入内存的排队顺序。第八，作业完成后的先后顺序按作业进入的先后顺序并无严格的对应关系，而是进入内存的排队顺序。第九，作业完成后的先后顺序按作业进入的先后顺序并无严格的对应关系，而是进入内存的排队顺序。第十，作业完成后的先后顺序按作业进入的先后顺序并无严格的对应关系，而是进入内存的排队顺序。

## 2. 批量处理系统的优缺点

批量处理系统虽出现较早，但至今仍是一种主要操作系统类型，并已配置于网络系统中。批处理系统的主要优点是：第一，系统吞吐量大。所谓吞吐量是指系统在单位时间内所完成的作业总量。由于批处理系统能以较多的时间对作业进行处理，操作系统开销较小，故可获得较大的吞吐量。第二，资源利用率高。在批处理系统中采取了一系列措施来提高资源利用率，使系统资源能得到有效的利用。第三，系统结构简单。第四，系统运行稳定。但是，批量处理系统也存在一些严重的缺点：第一，平均周转时间较长。所谓平均周转时间是指一批作业中所有作业周转时间的平均值，而周转时间则是指一个作业从提交给系统开始到完成并取出计算结果为止的这段时间间隔。尽管系统也兼顾作业的周转时间，但由于一个作业一旦运行便将运行到完成，这就必然会使一部分作业的周转时间显著增长，致使一批作业的平均周转时间变得较长。第二，不能提供交互能力。用户一旦把作业提交给系统后，便失去对自己作业的控制能力。虽然用户可以通过作业控制卡或作业说明书把作业运行中可能出现的情况及解决办法告诉系统，但对一些细节和异常情况却很难预料，当然也就无法干预，这将给程序的修改和调试带来许多不便。

## 分时系统

分时系统既是操作系统的一种类型，又是对配置了分时操作系统的计算机系统的一种称谓。在分时环境下，若干个计算机系统联接有若干个本地和远程终端，每个用户都可以在自己所占用的终端上以交互作用方式使用计算机，对系统资源进行时间上的分享。

### 分时和分时系统的实现方法

分时是计算机系统中的一个普遍概念，它可以理解为：两个或两个以上事件按时间轮流使用计算机系统中的某一资源。例如，CPU 和通道同时使用内存，多台设备同时使用通道等。而分时系统中的分时概念则侧重于对中央处理机和内存的分时使用问题。通常，在分时系统中，实现分时的方法有以下几种：

#### (1) 简单分时系统

在简单分时系统中，内存只存放一道用户程序作为现行作业，其它程序都放在外存上。系统中的所有用户作业依次按时间片轮流运行。所谓时间片，是指用户程序分享系统资源的一个基本时间单位。在这种系统中，每个用户作业使用完一个时间片后，便暂停运行，由系统将它调至外存（调出），然后再从外存选择另一个用户作业装入内存（调入），作为下一个时间片的现行作业投入运行。若在较短时间内，例如 3 秒钟内能使所有终端用户作业都轮流运行一个时间片，则基本上可以保证每个终端用户的请求都能得到及时响应。

#### 简单分时系统存在的一个主要问题是：

由于内存中只能存放一道用户程序，故在系统进行用户作业的调进调出时，中央处理机会处于空闲状态。

(2) 具有“前台”和“后台”的分时系统。为了改善系统性能,引入了所谓“前台”和“后台”的概念。它把内存划分为“前台”和“后台”两部分。“前台”部分用来存放按时间片调进/调出的终端型作业,其工作方式同简单分时系统;“后台”部分用来存放批量型作业,仅当“前台”正在调进/调出或无“前台”作业时,才运行“后台”作业,并给它分配更长的时间片。其工作过程如图 1-3 所示。

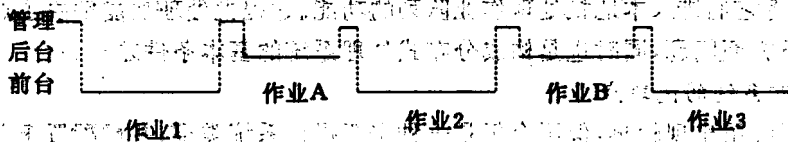


图 1-3 具有“前台”和“后台”的分时系统

(3) 基于多道程序设计的分时系统。为了进一步改善系统性能,在分时系统中引入了多道程序设计技术。即在内存中同时装入多道程序;每道程序无固定位置;程序道数不定,对小作业可多装入几道,对大作业则可少装入几道,系统把所有具备运行条件的作业排成一个队列,使它们依次按时间片轮流运行。若系统中除终端型用户作业外还有批量型作业,则应赋予终端型作业较高的运行优先权,并将它们排成一个高优先权队列,而将所有批量型作业另外排成一个队列。为了保证终端型用户的请求能得到及时响应,应优先运行高优先权队列的作业,仅当该队列为空时,才运行批量型作业队列中的作业。

### 2. 分时系统的特征

综上所述,我们可以概括出分时系统的主要特征如下:

(1) 多路性 系统通过多路转换卡将若干个终端用户连接到一台主机上。从宏观上看,多个终端用户作业在同时运行,共享系统资源。但从微观上看,各终端用户作业却在按时间片依次轮流使用 CPU。多路性提高了系统资源利用率,节省了开支,促进了计算机的广泛应用。

(2) 独立性 每个用户都在自己的终端上独立工作,互不干扰。从用户角度来看,他们并不感到有其它用户的存在,就象整个系统由他自己独占一样。

(3) 及时性 用户提出的请求能在可允许的时间范围内得到响应。这个时间范围通常被规定为 2 到 3 秒,它是衡量分时系统性能的一个重要指标。

(4) 交互性 用户能与系统进行广泛的人机会话,即用户通过键盘输入命令,请求系统服务或控制程序的运行;系统能及时响应用户的请求,并在终端上显示响应结果。交互作用是分时系统的重要特征,因此分时系统也被称为交互作用系统。

### 3. 分时系统的响应时间

所谓响应时间是指从终端用户发出一条命令开始到系统处理完这条命令并作出回答为止所需的这段时间。分时系统的响应时间是衡量分时系统性能的一个重要指标,也是设计分时系统时应注意的一个重要问题。

#### (1) 影响响应时间的若干因素

① 系统开销 在简单分时系统中,系统开销主要指进程的调进/调出(即对换)时间;在多道分时系统中,系统开销主要指进程的调度和对换时间。对换时间可表示为:

对换时间 = 对外存的访问时间 + 信息的传输时间

其中,信息的传输时间为信息对换量与信息对换速度之比,即:



$$\text{信息的传输时间} = \frac{\text{信息对换量}}{\text{信息对换速度}}$$

可见,同样一批信息,内、外存之间的信息对换速度越高,其传输时间就越短,从而会减少响应时间。为此,应选用存取速度高的磁盘作为外部存储器。

② 用户数目 若系统中有  $n$  个同时性用户,时间片为  $q$ ,则每个用户轮转一次所需的时间  $nq$  可近似地看作响应时间。可见,当  $q$  一定时,响应时间与用户数目成正比,这就限制了同时性用户的数目。

③ 时间片 当  $n$  一定时,响应时间正比于  $q$ ,从这种意义上说,减少时间片长度便可改善系统对用户的响应时间。但从另一方面看,同样一个作业,时间片越小,完成它所需的时间片个数和对换次数就会越多,这无疑会增加系统开销,降低系统效率。

④ 信息对换量 根据前面分析,信息的对换时间与信息对换量成正比,在同样对换速度下,信息对换量越少,信息的传输时间将越短,对换时间也越短。反之,对换时间就会越长。

## (2) 改善响应时间的办法

在上述影响响应时间的四个因素中,增加对换速度会受到硬件的限制;减少用户数目将影响系统性能;缩短时间片会降低系统效率;最可行的办法应是减少信息对换量。减少信息对换量有以下两种办法:

① 采用重入码技术 所谓重入码是指可被多个作业共享的代码。用重入码技术编制的文件可供多个终端用户共享,而不必每个用户自带副本,这样可以减少信息对换量。

② 引入虚拟存储技术 虚拟存储技术是一种能自动实现在较小内存中运行较大用户作业的内存管理技术。实现这种技术的主要方法是在外存保留作业的全部副本,每次调入主存的仅是当前时间片所用到的部分。这样就可以大大减少信息对换量。

### 1.3.3 实时系统

实时系统是为了满足实时问题对计算机应用所提出的特殊要求而出现的一种操作系统。所谓实时,意为“立即”、“马上”、“及时”。而实时系统则是指系统对特定输入做出反应的速度足以控制发出实时信号的对象。或者说,计算机能及时响应外部事件的请求,在规定时间内完成对该事件的处理,并控制所有实时设备和实时任务协调一致地运行。

#### 1. 实时系统的分类

实时系统按其使用方式可分为实时控制系统和实时信息处理系统两大类:

(1) 实时控制系统 实时控制系统又称为计算机控制系统,它通常是指以计算机为中心的生产过程控制系统或活动目标控制系统等。对生产过程进行实时控制的例子有:钢铁冶炼、电力生产、化工生产、机械加工、炼油等过程的自动控制;对活动目标进行实时控制的例子有:飞机飞行、导弹发射、火炮射击等活动目标的自动控制。在实时控制系统中,控制计算机通过特定的外围设备从被控制过程实时采集现场信息,如温度、压力、流量、空间位置等参数,并对它们进行及时处理,进而自动地控制相应机构,使这些参数能按预定规律变化或保持不变,以达到保证质量、提高产量、密切跟踪、及时操纵等目的。

(2) 实时信息处理系统 实时信息处理系统通常是指以计算机为中心的服务请求处理系统。例如,飞机订票系统、军事指挥系统等。在这类系统中,计算机及时接受来自终端设备的服务请求(如预订飞机票、检索资料等),根据用户提出的问题,对信息进行快速的