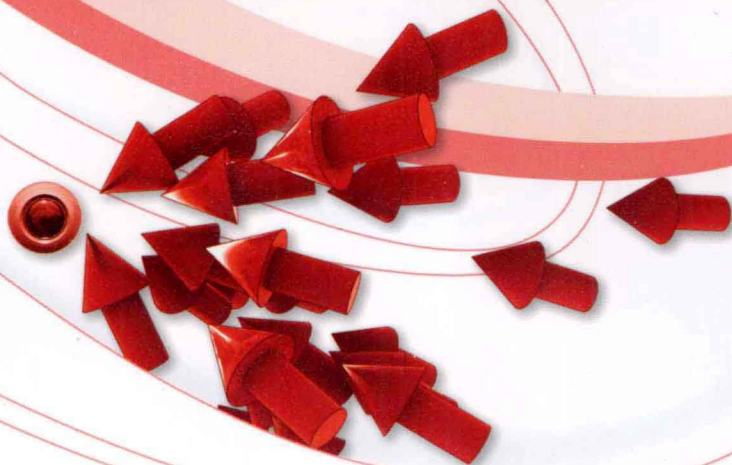




北京高等教育精品教材

BEIJING GAODENG JIAOYU JINGPIN JIAOCAI



高等学校计算机规划教材

操作系统原理教程

第3版)

张丽芬 刘美华 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

北京高等教育精品教材
高等学校计算机规划教材

操作系统原理教程

(第3版)

张丽芬 刘美华 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要介绍操作系统的基本概念、结构、基本功能和实现原理,以及当前世界上最流行的两大操作系统派系——UNIX/Linux 和 Windows 的特点和实现技术。

本书共分为三篇 18 章。第一篇分为 6 章,主要介绍操作系统的基本概念、三个基本操作系统类型(批处理、分时和实时)和特点、操作系统基本功能(处理机管理、存储器管理、文件管理和设备管理)和操作系统的进一步发展;第二篇分为 7 章,以 Linux 操作系统为例,介绍类 UNIX 系统设计和各部分功能的具体实现技术;第三篇分为 5 章,介绍以面向对象方法设计的特例 Windows 2000/XP 操作系统的实现技术。

本书注意吸收国内外较新的操作系统理论和实现技术,以反映现代操作系统发展的新动向。以操作系统的基本原理与实现技术为主要内容,同时注意到实际的应用。

本书可作为高等学校计算机科学与技术、软件工程,以及电子信息和自动控制类专业的教材,也可以作为计算机工程和应用人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

操作系统原理教程 / 张丽芬, 刘美华编著. —3 版. —北京: 电子工业出版社, 2013.8

高等学校计算机规划教材

ISBN 978-7-121-21025-9

I. ①操… II. ①张… ②刘… III. ①操作系统—高等学校—教材 IV. ①TP316

中国版本图书馆 CIP 数据核字(2013)第 162557 号

责任编辑: 韩同平 特约编辑: 李佩乾

印 刷: 北京季蜂印刷有限公司

装 订: 北京季蜂印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 22.5 字数: 576 千字

印 次: 2013 年 8 月第 1 次印刷

印 数: 3 000 册 定价: 45.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

本书为北京市高等教育精品教材。

操作系统是计算机系统中不可缺少的基本系统软件，主要用来管理和控制计算机系统的软硬件资源，提高资源利用率，并为用户提供一个方便、灵活、安全和可靠地使用计算机的工作环境。

计算机操作系统不仅是计算机专业的核心骨干课程，也是从事计算机系统和应用开发人员的必修知识。

本书作者从 1979 年以来一直从事操作系统的教学和科研工作，对多种机器的操作系统，如国产的实时操作系统 RTOS、200 系列中型机的操作系统、微机上的 CP/M、MS-DOS、UNIX 操作系统等进行了研究，特别对类 UNIX 各种操作系统 UNIX 系统 V、Solaris 和 Linux 以及微软的 Windows 系统等，进行了较系统的、全面的源代码的剖析和专题研究，查阅了近几年来大量的国内外有关资料，并结合科研工作，积累了大量的知识素材和实践经验。

本书特点如下：

(1) 注意吸收国内外较新的操作系统理论和实现技术，以反映现代操作系统发展的新动向。以操作系统的基本原理与实现技术为主要内容，同时注意到实际的应用。力求做到理论联系实际、由浅入深、循序渐进，有利于学生的学习。

(2) 重点讨论了传统操作系统的基本概念、基本方法、基本功能和实现原理，通过本课程的学习，能够对操作系统有一个完整和清楚的了解。

(3) 在操作系统基本原理讲解的基础上，以当代世界上最流行、最具代表性的两大操作系统派系——UNIX/Linux 和 Windows 为实例，较详细地讲解了它们的特点和实现技术，使学生通过实例的学习，充分理解和掌握操作系统的原理和技术。

(4) 通过理论与实例的学习，掌握操作系统的设计方法和实现技术，从而培养分析问题和解决问题的能力，以满足学生今后从事科研和就业的需要。

本书第一篇，即前 6 章讲解的是操作系统的基本概念、理论和实现原理，是本书的基本的和必修的内容。关于第二篇和第三篇，建议授课教师抽取其中具有特色的一部分进行讲解，如 Linux 的文件系统、Windows 的进程和存储器管理等，其余部分留给学生自学。本书适合课堂教学的学时数为 48~72。

本书第 7、8、11 章由刘美华编写，其余部分由张丽芬编写。在本书编写过程中，课程

组的刘昕和王全玉参与了有关 Linux 部分的编写工作，刘昕和刘利雄参与了有关 Windows 部分的编写工作。张丽芬对全书进行了统稿。

限于作者的水平，书中编写不当或疏漏之处，恳请读者批评指正。

作者联系方式：lmh123@bit.edu.cn

编著者
于北京理工大学

目 录

第一篇 操作系统的基本原理

第 1 章 操作系统概论	(1)
1.1 操作系统的定义	(1)
1.2 操作系统的形成与发展	(2)
1.2.1 顺序处理 (手工操作阶段)	(2)
1.2.2 简单的批处理系统	(3)
1.2.3 多道成批处理系统	(5)
1.2.4 分时系统	(7)
1.2.5 实时系统	(9)
1.2.6 嵌入式系统	(10)
1.3 操作系统的功能、服务和特性	(10)
1.4 操作系统的进一步发展	(12)
1.5 用户与操作系统的接口	(15)
1.5.1 用户与操作系统的操作接口	(15)
1.5.2 系统调用接口	(17)
1.6 操作系统的运行环境	(19)
1.7 操作系统的设计规范和结构设计	(19)
1.8 小结	(22)
习题	(23)
第 2 章 进程管理	(24)
2.1 进程的引入和概念	(24)
2.2 进程的描述	(27)
2.3 进程的控制	(29)
2.4 处理机的调度	(32)
2.5 线程的引入	(37)
2.6 小结	(39)
习题	(39)
第 3 章 进程之间的并发控制和死锁	(41)
3.1 并发进程的特点	(41)
3.2 进程之间的低级通信	(42)
3.2.1 进程之间的互斥	(42)
3.2.2 进程之间的同步	(45)
3.2.3 信号量和 P、V 操作	(45)
3.2.4 利用信号量解决计算机中的经典问题	(47)
3.3 管程	(49)
3.4 进程的高级通信	(53)
3.4.1 消息缓冲通信	(53)
3.4.2 其他通信机制	(56)
3.5 死锁	(57)
3.5.1 死锁的定义和死锁产生的必要条件	(57)
3.5.2 解决死锁的方法	(58)

3.6	小结	(66)
	习题	(66)
第4章	存储器管理	(70)
4.1	概述	(70)
4.2	单用户单道程序的存储器分配	(72)
4.3	多用户多道程序的存储器分配——分区分配	(73)
4.3.1	固定式分区	(73)
4.3.2	可变式分区	(74)
4.3.3	分区管理的地址重定位和存储器保护	(77)
4.3.4	分区管理的优缺点	(77)
4.4	覆盖与交换技术	(78)
4.5	页式存储器管理	(79)
4.6	段式存储器管理	(84)
4.7	虚拟存储器管理	(86)
4.7.1	虚拟存储器	(86)
4.7.2	页式虚拟存储器管理	(87)
4.7.3	页式管理设计中应考虑的问题	(91)
4.7.4	段式虚拟存储器管理	(94)
4.7.5	段页式虚拟存储器管理	(96)
4.8	小结	(98)
	习题	(98)
第5章	文件系统	(101)
5.1	文件和文件系统	(101)
5.2	文件目录结构	(103)
5.3	文件的逻辑结构和存取方法	(105)
5.4	文件的物理结构和存储介质	(107)
5.4.1	文件的物理结构	(107)
5.4.2	文件的存储介质	(110)
5.5	文件记录的组块与分解	(112)
5.6	文件存储器存储空间的管理	(113)
5.7	文件的共享与保护	(114)
5.8	文件的操作命令	(118)
5.9	文件系统的组织结构	(119)
5.10	存储器映射文件	(120)
5.11	小结	(121)
	习题	(122)
第6章	设备管理	(124)
6.1	I/O 硬件组成	(124)
6.1.1	I/O 设备分类	(124)
6.1.2	设备控制器	(124)
6.1.3	I/O 数据传输的控制方式	(126)
6.1.4	通道	(128)
6.2	I/O 软件的组成	(130)
6.2.1	I/O 软件的设计目标	(130)
6.2.2	I/O 软件的功能	(131)
6.2.3	同步 I/O 和异步 I/O	(135)

6.3	磁盘管理	(136)
6.3.1	磁盘调度	(136)
6.3.2	磁盘的错误处理	(139)
6.3.3	提高磁盘性能和容错性的机制——独立磁盘的冗余阵列	(139)
6.4	小结	(142)
	习题	(143)

第二篇 Linux 操作系统

第 7 章	Linux 进程管理	(145)
7.1	Linux 进程的组成	(146)
7.1.1	进程的定义	(146)
7.1.2	进程的状态	(149)
7.2	Linux 进程链表	(149)
7.3	Linux 进程控制	(152)
7.3.1	进程创建	(152)
7.3.2	进程撤销	(155)
7.4	Linux 进程切换	(155)
7.5	Linux 进程调度	(156)
7.6	内核同步	(159)
7.7	小结	(161)
	习题	(162)
第 8 章	Linux 存储器管理	(163)
8.1	进程地址空间的管理	(163)
8.1.1	Linux 中的分段	(163)
8.1.2	虚拟内存区域	(165)
8.1.3	虚拟内存描述符	(168)
8.1.4	创建进程的地址空间	(170)
8.1.5	堆的管理	(170)
8.2	物理内存管理	(170)
8.3	slab 管理	(176)
8.3.1	slab 分配器	(176)
8.3.2	slab 着色	(178)
8.4	高端内存区管理	(179)
8.5	地址转换	(180)
8.6	请求调页与缺页异常处理	(181)
8.7	盘交换区空间管理	(182)
8.8	小结	(183)
	习题	(184)
第 9 章	Linux 文件系统	(185)
9.1	Ext2 的磁盘涉及的数据结构	(185)
9.1.1	Linux 文件卷的布局	(185)
9.1.2	超级块	(186)
9.1.3	块组描述符	(187)
9.1.4	文件目录与索引节点结构	(188)
9.1.5	访问控制表 ACL	(191)
9.2	Ext2 的主存数据结构	(191)

9.2.1	超级块和索引节点对象	(192)
9.2.2	位图高速缓存	(193)
9.3	Ext2 磁盘空间管理	(194)
9.3.1	磁盘索引节点的管理	(195)
9.3.2	空闲磁盘块的分配与回收	(196)
9.4	Ext2 提供的文件操作	(197)
9.5	小结	(198)
	习题	(198)
第 10 章	Linux 虚拟文件系统	(199)
10.1	虚拟文件系统涉及的数据结构	(199)
10.1.1	超级块对象	(200)
10.1.2	索引节点对象	(202)
10.1.3	文件对象	(205)
10.1.4	目录项对象	(206)
10.1.5	与进程打开文件相关的数据结构	(208)
10.2	文件系统的注册与安装	(210)
10.2.1	文件系统注册	(210)
10.2.2	文件系统安装	(211)
10.3	VFS 系统调用的实现	(213)
10.3.1	文件的打开与关闭	(213)
10.3.2	文件的读写	(214)
10.4	小结	(216)
	习题	(216)
第 11 章	Linux I/O 系统	(217)
11.1	设备驱动模型	(217)
11.1.1	sysfs 文件系统	(217)
11.1.2	设备驱动模型的组件	(219)
11.2	设备文件	(221)
11.3	设备驱动程序	(222)
11.3.1	块设备驱动程序	(223)
11.3.2	字符设备驱动程序	(228)
11.4	高速缓存	(228)
11.4.1	页高速缓存	(229)
11.4.2	把块存放在页高速缓存中	(231)
11.5	小结	(232)
	习题	(233)
第 12 章	中断、异常和信号处理	(234)
12.1	中断和异常处理的硬件基础	(234)
12.2	中断和异常处理	(236)
12.2.1	硬件完成的处理	(236)
12.2.2	软件处理	(237)
12.2.3	如何处理中断	(237)
12.3	信号处理机制	(239)
12.3.1	信号概述	(239)
12.3.2	信号的发送与安装	(240)
12.3.3	信号集	(241)

12.3.4	信号应用示例	(242)
12.4	小结	(243)
	习题	(243)
第 13 章	UNIX 系统进程之间的通信	(244)
13.1	管道通信	(244)
13.1.1	创建无名管道	(244)
13.1.2	管道涉及的数据结构	(246)
13.1.3	创建一个有名管道	(247)
13.2	UNIX 系统 V 的交互进程通信	(249)
13.3	信号量机制	(250)
13.3.1	信号量机制使用的数据结构	(251)
13.3.2	信号量机制的系统调用	(253)
13.4	消息缓冲机制	(254)
13.4.1	消息缓冲使用的数据结构	(255)
13.4.2	消息缓冲的系统调用	(256)
13.4.3	消息缓冲的通信示例	(258)
13.5	共享内存区机制	(259)
13.6	小结	(262)
	习题	(263)

第三篇 Windows 2000/XP 操作系统研究

第 14 章	Windows 2000/XP 操作系统模型	(264)
14.1	Windows 2000/XP 的体系结构	(264)
14.1.1	用户态进程	(264)
14.1.2	子系统动态链接库	(266)
14.1.3	核心态的系统组件	(266)
14.2	Windows 2000/XP 操作系统的特点	(268)
14.3	Windows 2000/XP 的系统机制	(269)
14.3.1	陷阱处理程序	(269)
14.3.2	中断调度	(270)
14.3.3	异常调度	(273)
14.3.4	系统服务调度	(274)
14.4	对象管理器	(275)
14.4.1	对象结构	(275)
14.4.2	管理对象	(276)
14.5	对象之间的同步	(279)
14.6	小结	(283)
	习题	(283)
第 15 章	Windows 2000/XP 进程和线程管理	(284)
15.1	Windows 2000/XP 进程和线程	(284)
15.1.1	进程对象	(284)
15.1.2	线程对象	(285)
15.2	线程调度	(287)
15.3	对称多处理机系统上的线程调度	(291)
15.3.1	几个与调度有关的概念	(291)
15.3.2	线程调度程序的数据结构	(291)

15.3.3	多处理机的线程调度算法	(292)
15.3.4	空闲线程的调度	(294)
15.4	线程优先级提升	(294)
15.5	Windows 2000/XP 的线程同步	(296)
15.5.1	同步对象	(296)
15.5.2	同步对象的应用示例	(296)
15.6	小结	(300)
	习题	(300)
第 16 章	Windows 2000/XP 的存储器管理	(302)
16.1	存储器管理的基本概念	(302)
16.1.1	进程地址空间的布局	(302)
16.1.2	进程私有空间的分配	(303)
16.2	Windows 2000/XP 地址转换	(307)
16.2.1	地址转换所涉及的数据结构	(307)
16.2.2	页错误处理	(311)
16.3	页调度策略	(313)
16.4	小结	(315)
	习题	(316)
第 17 章	Windows 2000/XP 的文件系统	(317)
17.1	文件系统概述	(317)
17.2	主控文件表	(318)
17.2.1	主控文件表的结构	(318)
17.2.2	主控文件表的记录结构	(319)
17.3	NTFS 文件的引用和索引	(321)
17.4	Windows 2000/XP 文件系统模型	(322)
17.4.1	文件系统分层模型	(322)
17.4.2	Windows 2000/XP 文件系统驱动程序的体系结构	(323)
17.5	NTFS 可恢复性支持	(326)
17.5.1	文件系统优化技术	(326)
17.5.2	日志文件服务的实现	(327)
17.5.3	NTFS 坏簇恢复的支持	(331)
17.6	小结	(332)
	习题	(332)
第 18 章	Windows 2000/XP 的设备管理	(333)
18.1	Windows 2000/XP 的 I/O 系统结构	(333)
18.2	I/O 管理系统所涉及的关键数据结构	(334)
18.3	Windows 2000/XP 的 I/O 处理	(337)
18.3.1	对单层驱动程序的 I/O 请求	(337)
18.3.2	设备 I/O 的中断处理	(338)
18.3.3	I/O 请求的完成处理	(339)
18.3.4	对多层驱动程序的 I/O 请求	(340)
18.4	多处理 I/O 中的同步问题	(345)
18.5	快速 I/O	(345)
18.6	即插即用 PnP 管理器	(346)
18.7	小结	(347)
	习题	(348)
	参考文献	(349)

第一篇 操作系统的基本原理

第 1 章 操作系统概论

1.1 操作系统的定义

1. 计算机系统的组成

随着计算机技术的迅速发展，计算机系统的硬件和软件资源越来越丰富。从功能上，可把整个计算机系统划分为四个层次：硬件、操作系统、实用程序和应用程序，如图 1.1 所示。这四层的关系表现为一种单向调用关系，即外层的软件必须以事先约定的方式调用内层软件或硬件提供的服务。通常把这种约定称为界面或接口（interface）。下面简单介绍一下各个层次的特点。

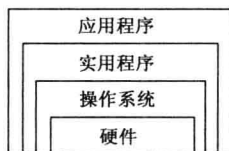


图 1.1 计算机系统的组成

(1) 硬件

硬件层由计算机的硬件资源组成，它包括中央处理机（简称 CPU）、存储器和输入/输出设备。其中，存储器包括主存和辅助存储器（又叫外存，包括磁盘、磁带、光盘等）；输入设备包括卡片或纸带输入机、键盘、鼠标、图形扫描仪等；输出设备有卡片或纸带输出机、显示器、打印机、绘图机等。这种不加任何扩充的计算机称为裸机。一个裸机的功能即使很强，用户也很难使用，因为这些裸机的部件是通过执行机器指令来实现计算和输入/输出功能的。

(2) 操作系统

操作系统是整个计算机系统的管理和控制中心。一个计算机如果没有操作系统，整个系统将无法工作。

(3) 实用程序

实用程序层包括下面一些程序，且它们通常是驻留在磁盘上的。

- ① 各种语言编译程序。语言编译程序包括高级语言和汇编语言的编译和汇编程序。
- ② 文本编辑程序。文本编辑程序是用来建立和修改用户的源程序或其他文本数据的，这类程序种类繁多，功能各异，有面向行的和面向全屏幕的。
- ③ 调试排错程序。它用来帮助用户调试程序，从而方便地找出程序中的逻辑错误。
- ④ 连接装配和装入程序。连接装配程序把用户独立编译好的各目标程序连接装配成一个可执行的程序。而装入程序，则负责将一个可执行程序装入到主存运行。

还有一些实用程序，如标准过程和函数、系统诊断程序、文件加密/解密程序以及用户连网使用的 Internet 浏览器等。总之，这些实用程序为用户提供了一个良好的使用计算机系统环境。

(4) 应用程序

应用程序是计算机系统的最外层软件，它主要负责解决用户的实际问题。这些程序通常由用户或专门的软件公司编制。这类软件比较丰富，如各种数据库管理软件、计算机辅助设计软件、各种事务处理软件（如制表软件、Web 浏览器）等。

2. 操作系统的定义和设计目标

为了深入、全面地理解操作系统的含义，从以下两个方面来阐述：

① 从计算机系统设计者的角度看，操作系统是由一系列程序模块组成的一个大的系统管理程序，它依据设计者设计的各种管理和调度策略，对计算机的硬件和软件（程序和数据）资源进行管理和调度，合理地组织计算机的工作流程，从而提高资源的利用效率。由此可认为，操作系统是计算机软硬件资源的管理和控制程序。

② 从用户角度看，配上操作系统的计算机是一台比裸机功能更强、使用更方便简单的虚拟机。也即，它是用户与计算机系统之间的一个接口界面，用户通过它来使用计算机。它向用户及其程序提供了一个良好的使用计算机的环境。它使系统变得容易维护、安全可靠、容错能力强和更加有效。

从上述两个角度可总结出，设计操作系统的目标有两个：一个是使用户方便、简单地使用计算机系统，另一个就是使计算机系统能高效可靠地运转。故操作系统是现代计算机系统不可缺少的关键部件。计算机系统越复杂，操作系统越显得重要。为此，有必要了解操作系统的组成和功能，以便更好地利用计算机进行系统和应用开发。这正是本书要讨论的主题。

1.2 操作系统的形成与发展

1.2.1 顺序处理（手工操作阶段）

早期的计算机（20 世纪的 40 年代~50 年代中期），程序员直接与计算机硬件打交道，没有操作系统。计算机由 CPU、主存、某种类型的输入设备（卡片输入机）、一台打印机，以及装有显示灯、乒乓开关的操作控制台组成。用机器代码或汇编或高级语言编写的程序通过输入设备装入计算机，再由程序员或操作员从控制台上通过设置乒乓开关启动汇编或编译或装入程序运行等。程序处理过程中，若出现错误，则通过控制台上的指示灯指示错误产生的条件并停止运行。经修改后可再次运行。如果程序正常完成，将输出结果送打印机打印。然后，将计算机转让给另一个程序员使用。

早期的这种操作方式，操作员或程序员要花费大量时间操作计算机，导致系统昂贵资源的无效使用，这主要由下面两方面问题造成。

1. 人工负责计算机的调度

计算机各资源的使用是通过一张纸登记的。一般的做法是，各用户说明他大约使用计算机多长时间，由机房负责人为他安排上机时间。如果一个用户预约 1 小时，结果用 45 分钟完成了，剩余的 15 分钟被浪费了。另一方面，若用户程序运行过程中出现问题，由于要检查错误和解决问题，被迫暂停，等待下一次进行预约后才能运行。

2. 人工负责编排作业的运行顺序

程序，又叫作业。在操作系统中，通常把用户在一次算题过程中要求计算机所做工作的集合叫做一个作业。以现在执行一个作业为例，用户在计算机上进行算题时，通常要经历以下几步：

① 采用某种语言按算法编写源程序，将源程序通过某种手段（如卡片输入机等）送入计算机；

② 调用某语言的汇编或编译程序，对源程序进行汇编或编译，产生目标代码程序；

③ 调用链接装配程序，将目标代码及调用的各种库代码连接装配成一个可执行程序；

④ 装入可执行程序 and 运行时所需数据，运行该程序并产生计算结果。

由此可见，一个算题任务的完成通常要经历建立、编译、连接装配和运行，才能得到计算结果。把这些相对独立的每一步骤叫做作业步。一个作业的各作业步之间总是相互联系的，在逻辑上是顺序执行的。下一作业步能否执行，完全取决于上一作业步是否成功完成。比如，若汇编或编译失败，则不可能进行连接装配。

一次运行可能只是编译一个程序。为此要将编译程序和一个用高级语言编写的源程序装入主存，保存被编译好的目标程序。然后连接和装入目标程序和公共库函数。其中，每一步都涉及安装和卸下磁带或卡片叠等。如果在一次处理中出现错误，用户不得不返回到编排序列的开始重新运行，因此，大量时间用在编排程序运行上了，导致系统效率极低。这种操作方式叫顺序处理。

1.2.2 简单的批处理系统

早期计算机造价昂贵，上述的那种人工调度和编排作业的方式浪费了大量的计算机时间。简单的批处理系统（Simple Batch System）正是在 20 世纪 50 年代后期到 60 年代中期，伴随第二代计算机的出现而研制成功的。

简单的批处理模式的中心思想是使用一个监控程序软件，各个用户将各自作业的卡片叠或纸带交给机房的操作员，再由操作员将这些作业的卡片或纸带按序成批地放在一个输入设备上，由监控程序自动控制输入设备将各个作业读入到磁带上。之后，监控程序按照顺序自动地把一个个作业装入内存进行处理。

监控程序常驻主存，工作非常简单，就是将 CPU 的控制权自动地从一个作业转换到另一个作业。很清楚，手工操作阶段的两个问题已得到圆满解决：一方面由监控程序处理调度问题，各作业以尽可能快的速度执行，从而不存在空闲的机器时间；另一方面由监控程序处理作业编排问题。为了使监控程序能代替用户完成对作业的编排控制，系统向用户提供了一套作业控制命令。每当用户提交作业时，将对作业的控制意图用作业控制卡或作业说明书的形式提交给监控程序。下面是监控程序使用的几个典型的控制卡。控制卡以“\$”开头，以区别其他一般的卡片。

控制卡	功能
\$JOB	启动一个新程序
\$FORTRAN	调用 FORTRAN 编译程序
\$LOAD	调用装入程序，装入要执行的程序
\$RUN	运行程序
\$END	作业结束卡

图 1.2 给出了以卡片形式提交的一个用 FORTRAN 语言编写的程序及程序所用数据，以及控制作业执行的控制指令卡。

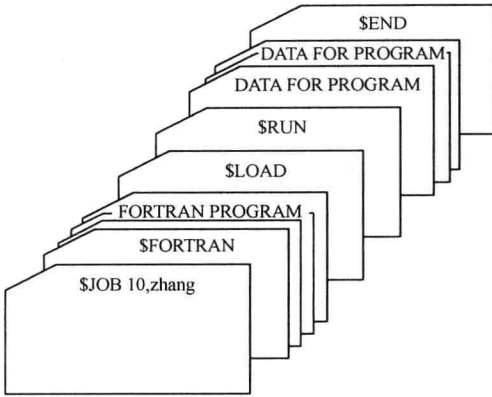


图 1.2 一个典型的输入作业的结构

一个用户作业的成功或错误完成，都引起监控程序扫描输入卡，直到遇到下一个作业的控制卡 \$JOB 为止。

当计算机运行中发生错误或意外时，监控程序通过控制台打字机输出信息向操作员报告。这种输出信息不仅比亮灯显示所表达的更为丰富，而且便于操作员理解。总之，用这种半自动方式控制计算机不仅提高了效率，而且方便了使用。这种简单的批处理在硬件结构上有两种不同的控制方式。

1. 早期的联机批处理

早期的联机批处理的硬件控制方式是：作业的输入、计算和输出都是在 CPU 直接控制下进行的。这样，在输入或输出过程中，主机的速度降低为慢速的输入或输出设备的速度。图 1.3 给出了联机批处理的模型。

2. 早期的脱机批处理

为了提高 CPU 的利用率，使 CPU 从慢速的外设控制中解脱出来，引入了早期的脱机批处理方式。该方式下，系统增设一台小型卫星机专门用来控制外部设备的输入/输出。其模型如图 1.4 所示。

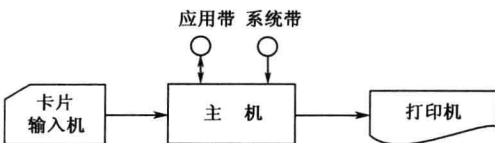


图 1.3 早期的联机批处理模型

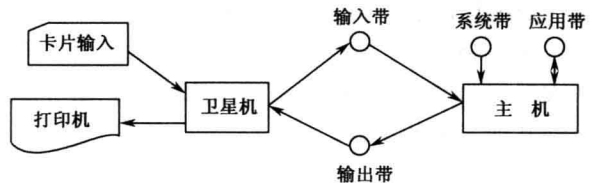


图 1.4 早期的脱机批处理模型

在这种模型中，小型卫星机的作用如下。

- ① 把卡片输入机上的作业逐个地记到输入磁带上，以便主机执行。
- ② 控制打印机输出由主机送入输出磁带上的作业的执行结果。

由此可见，采用这种脱机技术后，主机的所有输入/输出都是通过磁带进行的，而且主机与慢速外部设备可以并行，从而提高了主机运行效率。从 20 世纪 50 年代后期到 60 年代中期，脱机批处理运行得相当成功。脱机技术的实质是用快速的输入/输出设备代替慢速的设备。

1.2.3 多道成批处理系统

1. 多道程序设计 (Multiprogrammed System)

进入 20 世纪 60 年代中后期，计算机的硬件有了突飞猛进的发展，产生了硬件通道、中断和缓冲技术，从而使得计算机在组织结构上发生了重大变革。原先以 CPU 为中心的体系结构，转变为以主存为中心，其结构模型如图 1.5 所示。

所谓通道，实际是一种比 CPU 速度较慢、价格较便宜的硬件。它是比小型卫星机更经济的、独立于 CPU 的、专门用于控制输入/输出设备的 I/O 处理机。通道连接着主存和外设，具有与主存直接交换数据的能力。当需要输入/输出时，CPU 只要向通道发一个命令，通道就独立地控制相应的外部设备完成指定的传输任务。通道通过中断机构向 CPU 报告其完成情况。这样，利用缓冲技术，通道控制输入/输出传输任务，使 CPU 与外部设备的操作可以更充分地并行执行，提高了 CPU 的利用率。有关通道、中断和缓冲技术，将在本书后面章节介绍。

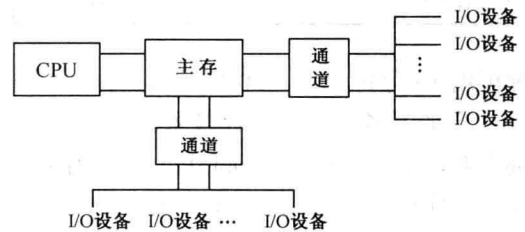


图 1.5 多道程序系统的计算机模型

简单的批处理系统提供了作业自动定序处理。通道和中断的出现，使 CPU 摆脱了对慢速设备的控制，从而提高了系统的处理效率。但由于主存只存放一个用户作业，当作业请求输入/输出时，CPU 空闲等待输入/输出完成。如果能在主存同时放多个用户作业，当一个作业等待数据传输时，CPU 转去执行其他作业，从而保证 CPU 与系统中的输入/输出设备并行操作。

这种在主存中同时存放多个作业，使之同时处于运行状态的程序设计方法，叫做多道程序设计。对于一个单处理机的系统来说，“多作业同时处于运行状态”是从宏观上理解的，其含义是指每个作业都已开始运行，但都尚未完成。而从微观上看，各个作业是串行执行的。在任何特定时刻，只有一个作业在处理机上运行。

引入多道程序设计技术的根本目的是提高 CPU 的利用率，充分发挥系统设备的并行性。这包括程序之间、CPU 与设备之间、设备与设备之间的并行操作。为了更好地理解引入多道程序的好处，下面用两个例子进行说明。

引入多道程序设计技术的根本目的是提高 CPU 的利用率，充分发挥系统设备的并行性。这包括程序之间、CPU 与设备之间、设备与设备之间的并行操作。为了更好地理解引入多道程序的好处，下面用两个例子进行说明。

(1) 主存只有一道程序。这个程序每处理文件中的一个记录，平均需要执行 100 条指令。假定读一个记录需 0.0015 秒，执行 100 条指令需 0.0001 秒，写一个记录需 0.0015 秒，则

平均处理一个记录的时间： $0.0015 + 0.0001 + 0.0015 = 0.0031$ （秒）

处理一个记录 CPU 的利用率：

处理机执行 100 条指令时间/平均处理一个记录的时间 = $0.0001 / 0.0031 = 3.2\%$

由这个例子可看出，处理一个记录时，CPU 的利用率仅为 3.2%。也就是说，CPU 有 96.8%的时间在等待 I/O 设备进行记录的读和写，这显然是对 CPU 的极大浪费。

(2) 主存同时放三个程序。当主存容量足够大时，为提高 CPU 的利用率，应在主存放足够多的作业，以便当一个作业等待 I/O 时，处理机转去执行其他作业。假设一个计算机系统，它有 256K 主存（不包含操作系统）、一个磁盘、一个终端和一台打印机，主存装有三个作业，分别命名为 JOB1、JOB2、JOB3，它们对资源的使用情况如表 1.1 所示。

表 1.1 三个作业的执行特点

作业编号	JOB1	JOB2	JOB3
类型	计算型	I/O 型	I/O 型
占用主存	50K	100K	80K
需磁盘情况	NO	NO	Yes
需终端情况	NO	Yes	NO
需打印机情况	NO	NO	Yes
运行时间	5 分钟	15 分钟	10 分钟

由表 1.1 可见，作业 2 主要使用终端进行作业的输入，作业 3 主要使用磁盘和打印机。

JOB2 和 JOB3 需要较少的 CPU 时间。

对于简单批处理，这些作业将按顺序执行。作业 1 运行 5 分钟完成，作业 2 等待 5 分钟再用 15 分钟完成。20 分钟后，作业 3 开始执行。30 分钟后三个作业全部完成。

若采用多道程序设计技术，让三个作业并行运行。由于它们运行中几乎不同时使用同类资源，故三个程序可同时运行。在作业 1 进行计算的同时，作业 2 可在终端上进行输入/输出，作业 3 使用磁盘和打印机。这样，作业 1 仍需 5 分钟完成，但在作业 1 结束时，作业 2 已完成三分之一，而作业 3 已完成一半。这样，这三个作业在 15 分钟内将全部完成。显然整个系统处理效率明显提高。

为了更好地理解引入多道程序的好处，下面先引入衡量批处理计算机系统性能指标的几个重要概念：

- 资源利用率：指在给定时间内，系统中某一资源，如 CPU、存储器、外部设备等，实际使用的时间所占的比率。显然，要提高资源利用率，就必须使资源尽可能忙碌。
- 吞吐量 (Throughput)：指单位时间内系统所处理的信息量。它通常以每小时或每天所处理的作业个数来度量（这里以小时作为度量单位）。
- 周转时间 (Turnaround Time)：指从作业进入系统到作业退出系统所经历的时间。即作业在系统中的等待时间加运行时间。而平均周转时间是指系统运行的几个作业周转时间的平均值。

依据上述定义三个概念，分别计算上述三个作业单道和多道运行时，处理机和存储器等资源的利用率：

① 单道运行时，只有 JOB1 占有处理机，三个作业运行完成需要 30 分钟。

处理机的利用率为： $5 / (5 + 15 + 10) = 17\%$

存储器的利用率为： $(50 / 256 + 100 / 256 + 80 / 256) / 3 = 30\%$

② 多道时，15 分钟三个作业都完成，其中处理机使用 5 分钟。

处理机的利用率为： $5 / 15 = 33\%$

三个作业共享主存，主存利用率为： $(50 + 100 + 80) / 256 = 90\%$

三个作业单道运行和多道运行时的各资源利用率见表 1.2 中。

由表可知，多道程序运行，使得系统资源的利用率、吞吐量和作业的平均周转时间大大优于单道程序运行，系统性能的改善是明显的。因此，将多道程序设计技术用于批处理系统就构成了多道成批处理系统。