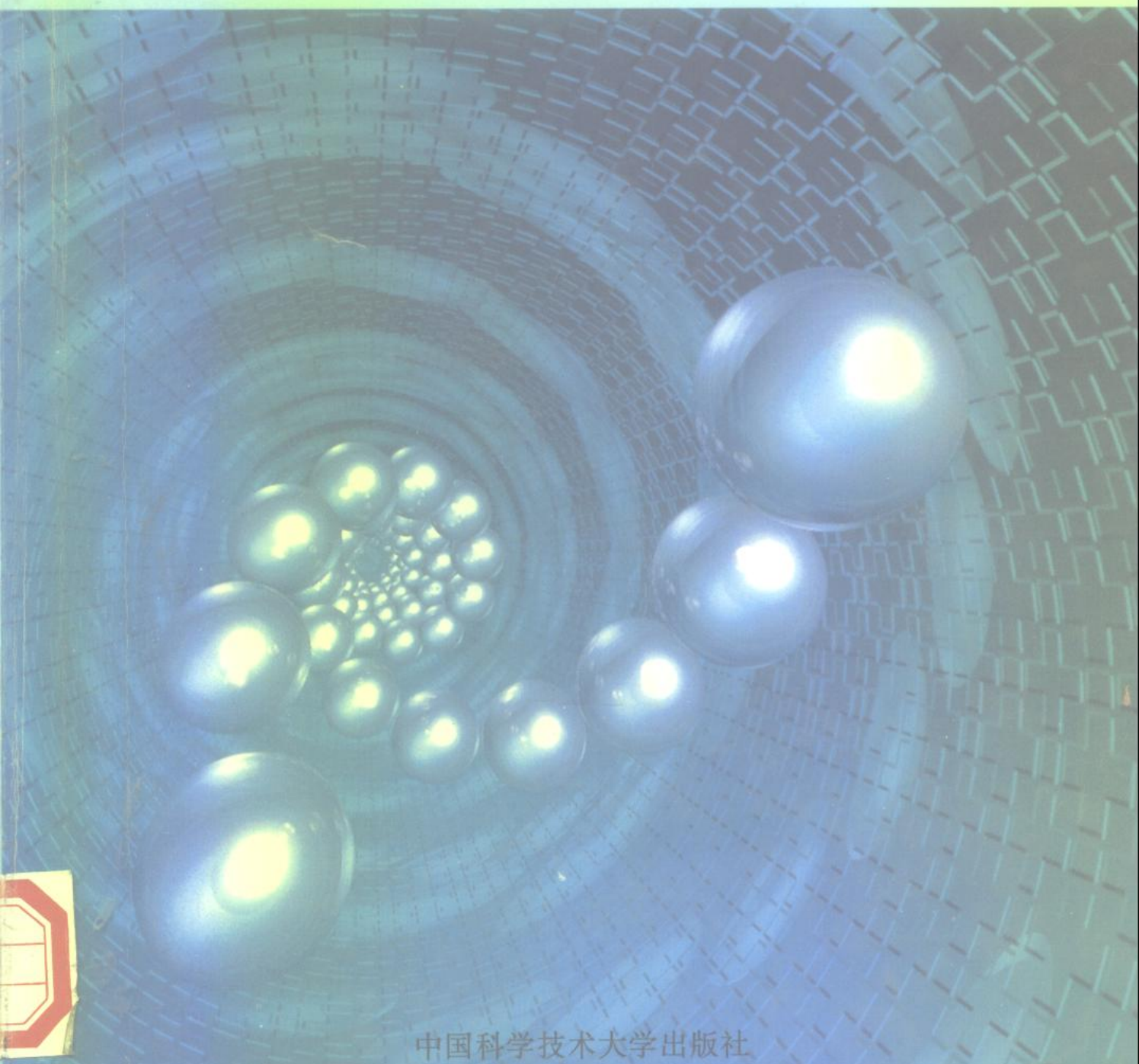
操作系统原理与实现技术

史杏荣　杨寿宝　编著



中国科学技术大学出版社

操作系统原理与实现技术

史杏荣　杨寿保　编著

中国科学技术大学出版社

1997·合肥

图书在版编目（CIP）数据

操作系统原理与实现技术／史杏荣　杨寿保　编著

—合肥：中国科学技术大学出版社，1997 年 10 月

ISBN　7-312-00932-8

Ⅰ　操作系统…

Ⅱ　史杏荣　杨寿保

Ⅲ　操作系统　资源管理　实现技术

Ⅳ　TP

中国科学技术大学出版社出版发行

（安徽省合肥市金寨路 96 号，邮编：230026）

中国科学技术大学印刷厂印刷

全国新华书店经销

开本：787× 1092／16　印张：20　字数：480 千字

1997 年 10 月第 1 版　　1997 年 10 月第 1 次印刷

印数：1—6000 册

ISBN　7-312-00932-8／TP· 191　　定价：25.00 元

内 容 简 介

本书系统地介绍了操作系统的基本概念、总体结构、用户接口、进程管理、作业管理、存储管理、文件管理和设备管理。在材料的组织和叙述上，结合了目前广泛使用的 MS-DOS 和 UNIX V 操作系统，既注重介绍了操作系统的基本原理与构造技术，又详细阐述了 MS-DOS 和 UNIX 操作系统的结构、基本原理、数据结构和实现技术。在第二章进程管理中，还介绍了多道程序技术和系统地讨论了 UNIX 系统的进程通讯机制，并列举了应用编程实例。

本书注意吸收了国内外较新的操作系统理论和技术，以反映现代操作系统发展的新动向、新水平；以操作系统的基本原理与实现技术为主要内容，同时注意到实际应用。可用作大专院校计算机系和电子类学科的本科生教材；对内容适当取舍后，也可作为计算机及其应用专业大专生的《操作系统》课程的教材；对于学习和从事计算机应用和开发的科技人员，本书也是一本内容翔实而易懂的自学教材和参考书。

前　言

操作系统是计算机系统配置的基本系统软件之一，它在计算机系统软件中占有中心的地位。操作系统控制和管理计算机软、硬件资源，组织计算机的工作流程，充分、合理、有效地使用计算机的所有资源。它是协调计算机各部分的关系、人—机关系以及方便用户的大型软件系统。它为用户创造了一个方便、有效、安全、可靠地使用计算机的工作环境。所以，了解和掌握现代计算机操作系统的基本原理及其功能，对计算机用户和开发者来说，都是极其重要的。

本教材结合目前广为流行的 UNIX 系统 V 和 MS-DOS 操作系统，系统地阐述了操作系统的基本原理和实现技术。第一章概述了操作系统的形成与发展史、操作系统的类型及其功能；从四种不同的观点描述了操作系统的基本概念和系统设计；并以 UNIX 和 MS-DOS 为例介绍了系统的总体结构。第二章系统地阐述了进程的概念、进程调度算法、进程间的同步与通讯以及死锁等重要论题。第三章至第六章介绍了作业管理、存储管理、设备管理和文件管理。在材料的选取上，注意吸收了国内外较新的操作系统理论和技术，舍弃了一些陈旧的概念和方法。我们既重点介绍操作系统的基本原理和实现技术，又结合目前广泛使用的 UNIX 和 MS-DOS 操作系统，介绍它们的结构、基本原理、数据结构和使用的主要技术。在阐述存储管理时，结合 Intel 80386 / 80486 CPU 的存储管理部件，详细描述了段页式存储管理技术。在本教材中，还介绍了多道程序技术和系统地讨论了 UNIX 系统的进程通讯机制，并列举了应用编程实例，在此基础上可开发计算机网络通讯软件。

在材料的组织和叙述上，力求由浅入深、循序渐进，便于自学。各章后面均附有习题和思考题，供练习使用。本教材是在给本科生和大专生多年讲授《操作系统》课程讲稿的基础上编写的，适用于计算机系和电子类学科本科生《操作系统》和《软件技术基础》（操作系统部分）课程的教材。本书以操作系统基本原理与实现技术为主要内容介绍操作系统，同时注意到实际应用，可作为大学高年级的学生、研究生和科技人员学习和掌握操作系统的参考书，也可作为计算机应用的培训教材。

限于编者水平，书中疏漏及不当之处在所难免，敬请读者批评指正。

编者

1997 年 4 月于中国科学技术大学

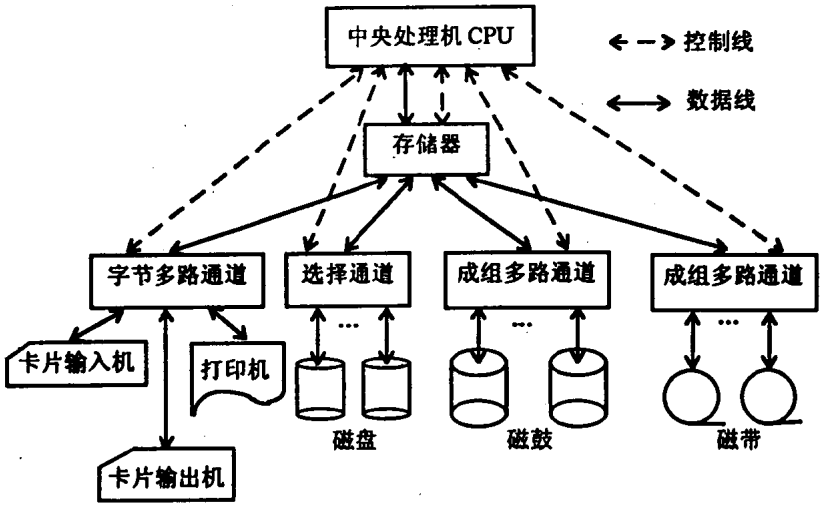
目　　录

第一章 引 论

现代计算机系统，不论是大型机、小型机或微型机系统，都由计算机硬件、软件两部分组成。小型到大型计算机大多由中央处理机 CPU、输入输出处理机（又称通道）、存储器和输入输出设备组成，图 1.1（a）示例了中型机 IBM 370 系统的硬件组织。通常这类计算机都是非总线结构，它们是以存储器为中心组织系统硬件。CPU 从存储器取指令，并访问存储在存储器的数据。当 CPU 需要和 I／O 设备交换信息时，它在存储器组织通道程序，并命令 I／O 处理机进行输入输出的管理和控制。I／O 处理机负责存储器和外部设备之间的信息传送。CPU 则为输出操作在存储器组织数据或从存储器读取输入数据。在 CPU 和 I／O 设备之间不存在数据通路，所以，在 CPU 和外设之间不能直接进行数据传送。

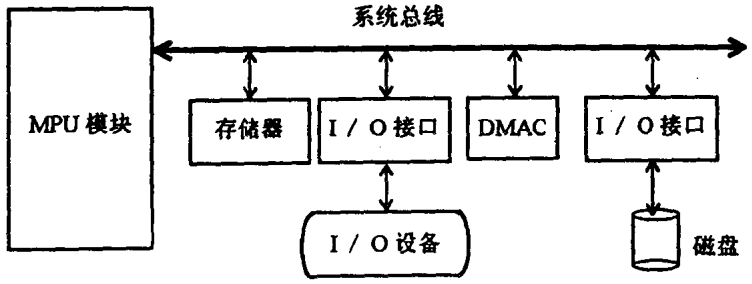
大多数微机系统则采用单总线结构，以总线为纽带组成微机系统，如图 1.1（b）所示。微处理器、存储器和 I／O 设备（通常 I／O 接口电路和总线连接）之间都通过总线进行信息交换。在单总线结构的微机系统中，CPU 可以执行 I／O 指令以在 CPU 和外设（实际上是 I／O 端口）之间直接进行数据传送；或以 DMA 方式在存储器和 I／O 设备之间进行成组数据传送。在多处理器微型机系统中，通常有一个专用的 I／O 处理机芯片，如图 1.1（c）中的 IOP 8089，由它专门负责对输入输出的控制和管理。当 CPU 需要和外设进行数据传送时，CPU 在存储器为其准备通道程序，然后向 IOP 8089 发通道注意信号。IOP 8089 在收到通道注意信号后，执行通道程序，完成输入输出操作。

通常把未配置任何软件的计算机称为裸机，用户直接使用裸机来编程和运行程序是相当困难的，甚至是不可能的。操作系统（Operating System）是对裸机的首次扩充，以建立用户和计算机间的接口。所以，用户只能通过操作系统使用计算机。设置操作系统的目的有:

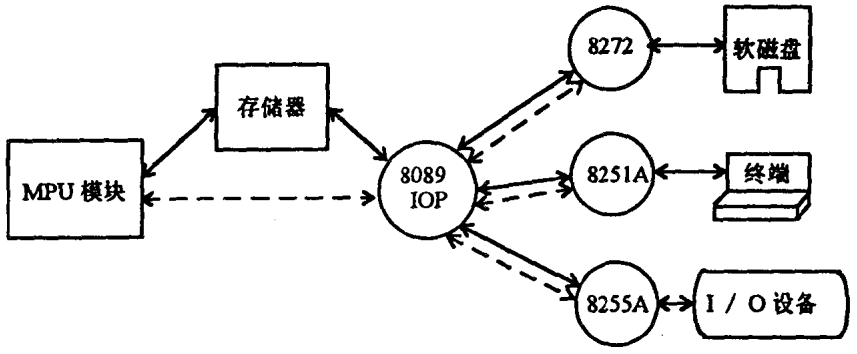


(a) IBM 370 系统结构

(a) IBM 370 系统结构



(b) 微型机系统结构



(c) 多处理器微型机系统结构

图 1.1　计算机系统结构

i) 将裸机改造成一台功能更强、服务质量更高、用户使用方便灵活、安全可靠的虚拟机。

ii) 有效地控制和管理计算机系统中各种软硬件资源，提高系统资源的使用率。

iii) 合理地组织计算机系统的工作流程，以改善系统性能。

一个计算机系统的软件通常可以分为两大类：系统软件和应用软件。系统软件用于计算机的管理、维护、控制和运行，以及对各种语言的源程序进行翻译、连接装配和加载等服务工作。系统软件又可分为：操作系统、语言处理程序和常用的例行服务程序。语言处理程序包括各种高级程序设计语言的编译程序、解释程序和汇编程序。服务程序的种类繁多，通常包括数据库管理程序、编辑程序、连接装配程序、诊断调试程序等。应用软件是指那些为了某一类的应用而研制的程序、或用户为解决某一特定问题而编制的程序或程序系统，例如汉字排版系统就是一例。

计算机系统中的硬件和软件以及软件各部分之间的关系是一种层次结构关系，如图 1.2 所示。软件是在硬件基础之上对硬件的性能加以扩充和完善，使之成为新的更强功能的计算机，通常称之为虚拟机。与此类似，软件之间也是一种层次关系，一部分软件的运行要以另一部分软件的存在并为其提供一定的服务为基础，而新增添的软件可以看作是对原来那部分软件的扩充和完善，以形成新的更强功能的虚拟机。例如，操作系统在裸机上运行，配置操作系统的裸机则成为一台能接收用户命令、具有 CPU、存储器、I／O 设备和文件管理的虚拟机，并能完成实际的 I／O 操作。各种语言的处理程序在操作系统之上运行，并通过操作系统提供的系统调用使用计算机的资源。这样，配置高级语言编译程序

的计算机就成为一台能识别高级语言的虚拟机。

从计算机的系统管理员的角度来看操作系统，那么它是计算机系统工作流程的组织者，计算机软、硬件资源的协调管理者。它减轻了系统管理员的负担，提高了计算机系统的经济效益。

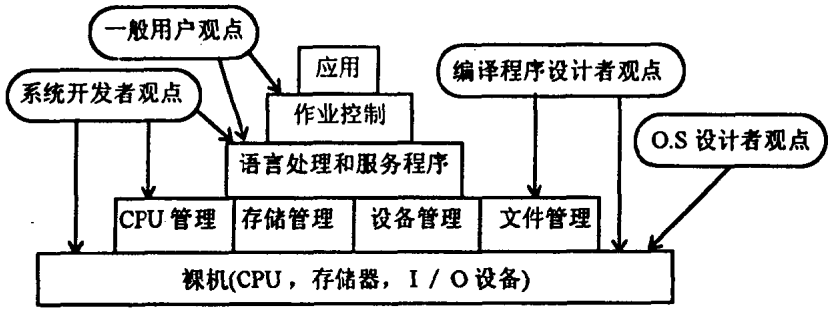


图 1.2　计算机系统层次结构

如果从一般用户的角度来看操作系统，它为用户提供了一台比裸机功能更强、服务质量更高、使用方便灵活且安全可靠的虚拟机，它是用户和计算机系统之间的一个接口界面，用户通过操作系统使用计算机。

1.1　操作系统发展史

操作系统是在不断地改善计算机系统性能和提高资源利用率的过程中，且伴随着计算机体系结构的发展，逐步地形成和发展起来的。

1.1.1　手工操作方式

40 年代中至 50 年代初，那时计算机问世不久，还没有操作系统。用户使用机器语言和汇编语言的手编程序编程，操作员用“拨开关”方式控制计算机的运行，以“看氖灯”为途径，了解系统运行情况。用户事先把要运行的程序和需要的数据通过穿孔机产生纸带或卡片。然后由操作员将纸带 / 卡片装入纸带 / 卡片输入机，以把程序和数据输入存储器，并启动计算机运行。当程序运行结束，由用户取走纸带或打印结果，下一个作业再重复上述过程。

这种手工操作方式存在下述缺点：

i) 独占全机资源。一台计算机为一个用户独占，虽然一个用户可以很方便地使用全机资源，但资源利用率极低。

ii) 手工联机操作，人工干预输入输出操作，辅助时间长。

手工操作方式严重地影响了系统资源的利用率，产生了人–机矛盾。对早期的计算机来说，人–机矛盾并不突出，因为系统资源不多，且计算机速度也低。例如，为了运行一作业程序，纸带 / 卡片装入等手工操作的时间为 3 分钟，但 CPU 运行时需 1 小时，显然，这是可以接受的。如果 CPU 的运行速度提高 60 倍，则运行作业程序的时间仅为 1 分钟，这样，缩短手工操作时间就成为迫切的现实问题了。

1.1.2 批处理系统

为了减少手工操作时间，提高价格昂贵的计算机资源利用率，人们首先想到的是从运行一个作业程序过渡到另一个作业程序时，力求摆脱人工干预，使其自动进行。

1. 早期的批处理系统

为了缩短作业的建立时间，人们首先研制了监控程序（Monitor）。把用户作业分组集中在一起，成批地进行处理，构成了作业自动执行序列，仅当一个作业处理结束后，监控程序自动地装入下一个作业，并且执行该作业。所谓作业是指用户程序及其数据和操作命令的集合。

操作员把用户提交的若干作业集中成为一批作业，并且把各作业卡片叠放在卡片输入机上，由监控程序把这一批作业读入并且写到磁带上。当该批作业输入完成后，再由监控程序自动地从磁带上依次逐个地读入作业到内存，并且对该用户程序进行汇编或编译。然后由装配程序把汇编或编译生成的目标程序装入内存，再启动执行之。当计算机完成一个作业的全部处理后，监控程序自动地调入第二个作业，并进行处理。在一批作业全部处理完毕后，监控程序又从卡片输入机上输入另一批作业到磁带上，并重复上述步骤进行处理。这样，监控程序不停地处理各个作业，因而实现了作业到作业的自动转换，缩短了作业建立时间和手工操作时间。

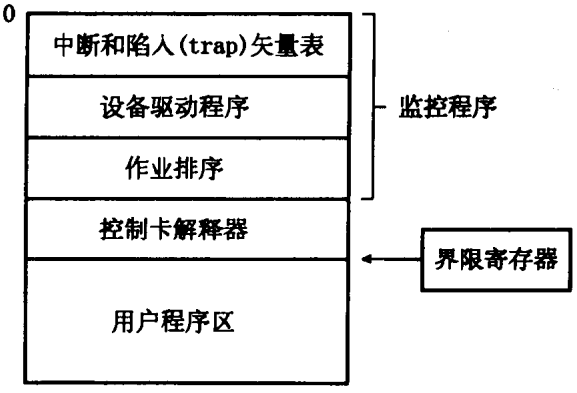


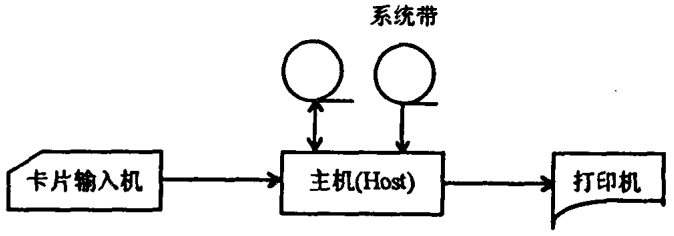
图 1.3 驻留监控程序系统的内存布局

图 1.3 显示了引入驻留监控程序系统的内存布局。与此相适应，计算机系统有两种工作模式：核心态和用户态，此外系统还设有一个界限或隔离（Fence）寄存器。仅当在核心态下运行的监控程序才能执行控制设备输入输出操作的 I／O 指令和修改界限寄存器，并能访问整个存储空间。在用户态下运行的用户程序只能访问用户程序区，并通过陷入(trap)监控程序才能使用计算机资源。

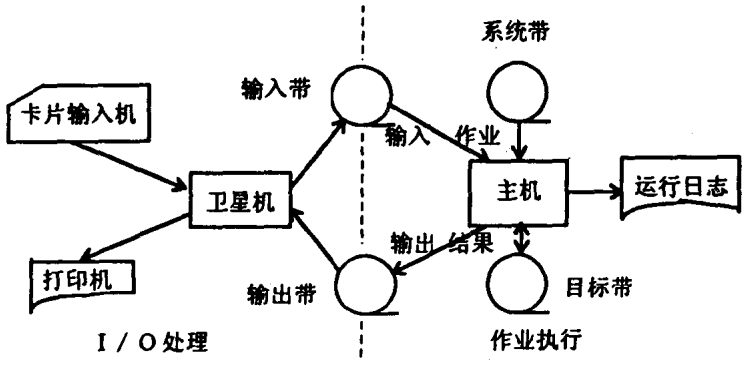
2. 脱机批处理系统

在早期的批处理系统中，作业的输入输出都是联机的（On-line）。作业信息由卡片输入机读入到磁带，再从磁带调入内存，由计算机计算结果，并且将计算结果在打印机上输出。这些都是由 CPU 来处理的，联机的低速输入输出设备的操作严重影响了计算机系统

的性能。所以在作业批处理系统中，引进了脱机（Off-line）输入输出技术，以解决人-机矛盾和CPU与I/O设备速度不匹配的矛盾。图1.4显示了联机和脱机输入输出批处理系统的结构模型。



(a) 联机批处理系统模型



(b) 脱机输入输出系统模型

图1.4 作业批处理系统

脱机I/O操作的系统有两种组织方式，一种是使用专用设备，实现从卡片输入机到磁带和从磁带到行式打印机的信息传输任务。另一种方式是使用专用的卫星机，由它负责作业的输入和作业计算结果的输出操作，如图1.4（b）所示。卫星机从卡片输入机上读入作业并写到磁带上；而主机则从磁带上把作业调入内存，进行处理和执行作业，并把计算结果输出到磁带上。然后，由卫星机把磁带上的信息在打印机上输出。这样，卫星机负责低速设备的输入输出操作，且能和主机并行工作，有效地提高了计算机系统的性能。

作业批处理系统在50年代末和60年代初应用于第二代计算机上，它的出现促进了其它软件的发展。标准I/O程序、高级语言编译程序、连接装配程序和标准程序库就是在这个时代产生的。

1.1.3 多道程序系统

在50年代末和60年代初期，在计算机体系结构方面取得了两项突破性的进展，一是通道技术的引进，二是中断技术的出现，使得通道具有中断主机工作的能力。这样，操作系统进入了多道程序设计系统阶段。

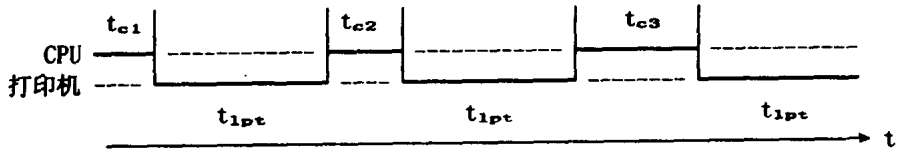
所谓通道是专门用来控制输入输出设备的处理机（I/O处理机）。通道和主机相比，一般速度较慢，价格也便宜得多；它可以和中央处理机（CPU）并行工作。这样，当需要传输数据时，CPU为输入输出操作建立通道程序，并且向通道发命令，通知通道执行

通道程序。在通道完成传输工作后，通过中断机构向 CPU 报告完成情况。由于通道技术的引入，使得原来由 CPU 直接控制的输入输出转移给了通道，价格昂贵且高性能的 CPU 则主要用于数据处理工作。

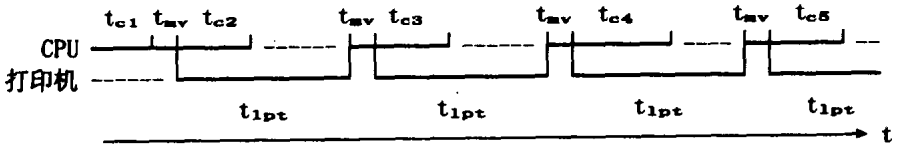
为了使通道的数据传输能与主机并行工作，并且减少主机的等待时间，在程序设计中引进了缓冲技术。下面通过一个简单例子说明缓冲技术对程序运行的影响。设在一段时间内，系统中有一个用户程序正在运行且使用行式打印机输出计算结果。其工作过程是:

i) 计算并产生一行打印的信息，打印信息存放在字符数组 line[MAXLEN]中，用 t_c 表示这段时间。

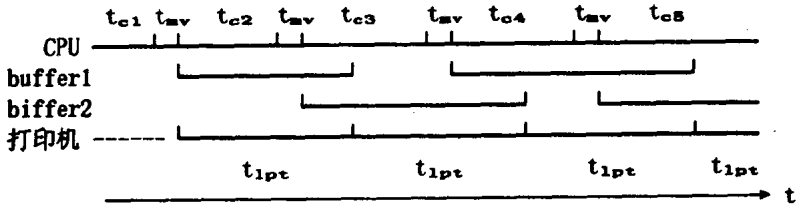
ii) 系统将 line 字符数组中的信息在行式打印机上输出，用 t_{lpt} 表示输出一行的时间。



(a) 无缓冲存储区



(b) 单缓冲区



(c) 双缓冲区

图 1.5 缓冲技术示例

上述过程可以重复多次，如图 1.5（a）所示。在各次循环中，t_c 可能不同，分别表示为 t_{c1}，t_{c2} 等；打印机输出一行的时间大致相等，都用 t_{lpt} 表示。图中实线表示程序正在 CPU 上运行或打印机正在进行打印操作；虚线部分则表示 CPU 处于等待或打印机处于空闲状态。由图可见，CPU 和打印机都有很多时间处于空闲状态，浪费了系统资源。

解决上述问题的一种常用技术是系统设置一定数量的缓冲区。例如，系统为行式打印机设置一个缓冲区 char buffer[MAXLEN]；于是，程序使用行式打印机的过程为:

i) 计算并产生一行打印的信息，存放在字符数组 line 中。

ii) 使用系统调用，将 line 数组中的信息交行式打印机输出。

iii) 系统将 line 数组中的信息复制到 buffer 缓冲区，传送时间记为 t_{mv}，然后行式打

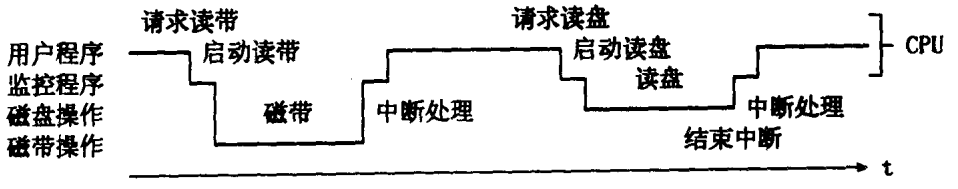
印机输出 buffer 缓冲区中的数据。此后，用户程序重复执行步骤 i)。

图 1.5（b）和（c）显示了使用单缓冲区和双缓冲区的工作情况。显然，缓冲技术提高了 CPU 与外设并行工作的程度。

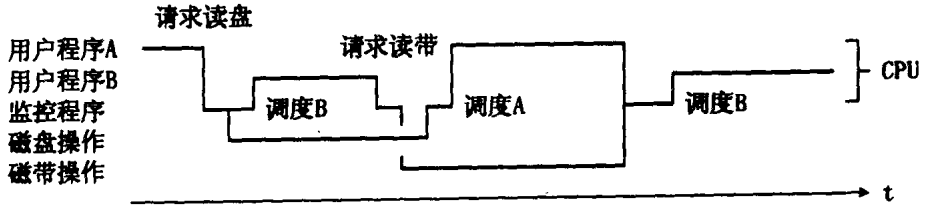
1. 多道程序技术

在采用批处理技术时，内存中仅存放一道运行程序，每当该程序发出 I／O 操作请求后，CPU 便处于等待 I／O 完成状态，致使 CPU 空闲。而当 CPU 运行程序时，I／O 设备又处于空闲状态。为了提高系统资源的使用率，引入了多道程序设计技术，所谓多道程序设计，即在内存中同时驻留多道程序，并且同时处于可运行状态。这些作业程序共享 CPU 的时间和存储器、外设等系统资源。

对于一个单 CPU 系统来说，“作业同时处于运行状态”显然是一个宏观的概念，其含义是指每个作业都已投入运行，但尚未完成。从微观上来说，在任一特定时刻，只有一个作业占用 CPU 运行而处于运行状态，而其它作业则处于等待或阻塞状态。引入多道程序设计技术是为了提高 CPU 和存储器、I／O 设备的利用率，充分发挥并行性，使诸作业之间、CPU 与外设之间、外设与外设之间并行工作。



(a) 单道作业运行情况



(b) 双道作业运行情况

图 1.6　作业运行示例

图 1.6 示例了单道程序和两道程序系统运行情况。在单道程序系统中，CPU 与外设、外设与外设间是串行工作的。据统计，单道程序系统 CPU 的利用率仅为 8%左右。而在多道程序系统中，随着内存中作业道数的增加，可以提高系统中所有设备和 CPU 的并行性和利用率，特别是 CPU 的利用率。

在多道程序系统中，多道程序同时驻留在内存，竞争使用系统资源，并发运行，提高了资源的利用率和系统性能。为了实现多道程序系统，必须妥善解决下述问题：

i) 存储保护与重定位　在多道程序系统中，几道作业程序在内存共享同一内存资源，系统必须采取某种机制以防止一用户作业程序在运行期间有意或无意破坏系统或其它用户存储空间中的信息，即实现用户作业程序间以及与操作系统程序间的隔离与保护。

当一作业程序被调度进入执行状态时，系统为它分配存储空间；作业程序运行结束