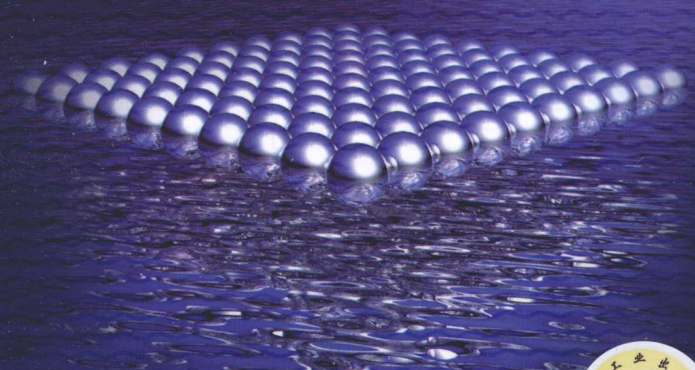
免费提供
电子教案

高等院校规划教材
计算机科学与技术系列

操作系统基础

邓胜兰　编著



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

操作系统基础

邓胜兰　编著

机械工业出版社

无论是计算机软件系统的开发者、管理者，还是使用者，都需要了解和掌握操作系统的相关知识。借助于操作系统的知识和功能，开发者可以编写出运行速度更快的应用程序；管理者可以合理配置出性能更优的应用系统；使用者可以更好地理解系统功能，从而高效、安全地使用计算机系统。本书根据上述需求，以通俗易懂的语言，阐述了操作系统的基本原理知识，内容包括总体结构、中断/异常处理、进程管理、存储管理、并发与通信、设备管理、文件系统和系统安全，同时以 Linux 操作系统的相关内容作为辅助实例。

本书可以作为高等院校计算机专业及相关专业本科学生的教材或参考书，也可作为从事计算机相关工作的专业技术人员以及计算机爱好者的自学读物。

图书在版编目（CIP）数据

操作系统基础/邓胜兰编著. —北京：机械工业出版社，2009.5
（高等院校规划教材·计算机科学与技术系列）
ISBN 978-7-111-27042-3

Ⅰ.操… Ⅱ.邓… Ⅲ.操作系统-高等学校-教材 Ⅳ.TP316

中国版本图书馆 CIP 数据核字（2009）第 070685 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：赵 慧
责任编辑：陈 皓 王 师
责任印制：李 妍
北京汇林印务有限公司印刷

2009 年 6 月第 1 版·第 1 次印刷
184mm×260mm·18.75 印张·463 千字
0001—3000 册
标准书号：ISBN 978-7-111-27042-3
定价：32.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294 68993821
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739

出 版 说 明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。另外，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

在信息时代的今天，人们的日常生活与计算机密切相关。当人们旅行时，需要通过计算机订购电子机票和预定旅馆房间；当坐在电影院里看电影时，那些令人惊叹的宏伟大片都使用了计算机特效技术。在办公室，可以使用自动化办公系统，轻松完成日常工作；回到家里，还可以上网聊天，放松自己。计算机为人们带来越来越多的便利，每天都有包含计算机技术的新产品和新功能出现。所有这些不仅依赖于功能强大的计算机硬件，也依赖于同样功能强大的操作系统。

操作系统是计算机系统中最基本的核心软件。无论是计算机软件系统的开发者、管理者，还是使用者，都需要了解和掌握操作系统的相关知识，以便更好地理解计算机系统的行为和能力。借助于操作系统的知识和功能，开发者可以编写出运行速度更快的应用程序；管理者可以配置出性能更优的应用系统；使用者可以更好地理解系统功能，从而高效、安全地使用计算机系统。

然而，操作系统又是计算机系统中最复杂的软件。它包含众多的、相互关联的功能模块和复杂的资源管理机制，为用户提供了方便快捷的使用界面，使普通用户远离计算机硬件、设备及其繁琐的操作过程。因此，对于普通用户来说，学习操作系统不是一件容易的事情。因为操作系统的功能机制深藏于“用户界面”之下，人们几乎不与它们直接交互，所以对于操作系统中的概念和机制，许多人感觉比较抽象和不易理解。

市面上有不少关于 Windows 或 Linux 的书，它们只介绍操作系统的命令和使用方法，不讨论操作系统的基本概念和实现原理，这类书能够帮助读者快速地掌握操作系统的命令和使用方法，可作为特定操作系统的入门指导和使用手册。但是，作者在这里建议读者至少选读一本操作系统原理的书，这样才能做到知其然知其所以然。

本书以普通大众作为主要读者对象，以通俗易懂的语言，阐述了操作系统的基本原理知识，可以作为高等院校计算机专业及相关专业的教材和参考书，也可作为从事计算机相关工作的专业技术人员以及计算机爱好者的自学读物。本书内容包括操作系统总体结构、中断/异常处理、进程管理、存储管理、并发与通信、设备管理、文件系统和系统安全，同时以 Linux 操作系统的相关内容作为讨论背景和辅助实例。

本书是作者总结多年从事操作系统设计开发和操作系统教学工作的经验，参考近几年来国内外出版的教材和文献，精心编著而成的。书中既包含了操作系统的基本概念和理论，也介绍了与操作系统功能相关的命令和应用实例，还插入了许多说明图，力求使抽象的操作系统原理变得通俗易懂。另外，每章之后配有习题和上机实验，可以帮助读者加深理解。

本书各章的主要内容如下。

第 1 章介绍操作系统的发展历史，分为两个阶段：以早期操作系统和现代操作系统为例，介绍操作系统的任务和地位、主要功能和体系结构等基本概念；以 Linux 和 Windows XP 的体系结构作为实例，让读者建立关于操作系统的整体观念。

第 2 章以 Linux 为主要背景介绍操作系统的用户界面，包括用户管理、命令解释程序 Shell 及其编程、常用的 Linux 命令、Linux 联机手册的使用、用户上机过程（编辑、编译和

运行程序）、Linux 实用软件工具及应用实例。通过本章的学习，读者可以掌握 Linux 的基本使用方法，为后面章节中安排的上机实验做好准备。

第 3 章介绍操作系统内核的基本概念和基本机制，使读者掌握操作系统内核的工作机理和运行流程。本章讨论了核心态与用户态的作用和区别，介绍了中断/异常处理机制和系统调用机制的工作原理，以 Linux 系统调用机制作为实例让读者了解系统调用的实现过程。

第 4 章介绍进程和线程。本章从进程的定义、表示和映像等方面介绍了进程的基本概念，围绕进程的生命周期和运行状态讨论了进程管理功能，介绍了进程调度的过程、目标和主要调度算法。从讨论进程的局限性出发介绍线程概念及其优势，讨论多处理机系统中的线程调度方法。最后，介绍 Linux 中进程和轻权进程的管理和调度策略，以程序实例说明相关系统调用和命令的使用方法。

第 5 章介绍存储管理。首先介绍计算机系统中的存储体系结构，讨论存储管理系统需要实现的 4 个基本任务。然后，从存储管理的 4 个基本任务方面介绍了固定分区存储管理、可变分区存储管理和页式存储管理。本章的重点放在页式存储管理和页式虚存管理，介绍了它们的工作原理、实现机制和相关算法。最后介绍了 Linux 的进程地址空间管理、磁盘交换区管理和存储映射文件的应用编程。

第 6 章介绍进程并发与通信。本章从讨论同步互斥问题的现象和原因出发，介绍了临界段、原语和信号量等基本概念，以解决生产者和消费者问题为背景，讨论信号量的应用及特点，在此基础上介绍多元信号量概念。以 Linux 为背景介绍进程间通信（IPC）机制及其应用程序例子。最后介绍了进程死锁的基本概念和基本处理方法。

第 7 章介绍设备管理。首先介绍 I/O 设备的基本概念、CPU 对 I/O 设备的控制方式和设备管理的任务，按照 I/O 系统的层次结构介绍 I/O 功能的实现原理以及从用户角度看到的输入/输出全过程。然后，讨论缓冲区管理和磁盘请求调度等 I/O 优化技术。最后介绍 Linux 的设备驱动程序接口及其动态加载机制和一个虚拟盘的驱动程序实例。

第 8 章介绍文件系统。本章介绍文件及文件系统的基本概念，讨论设备文件系统（即文件卷）的组织结构和实现原理，介绍文件管理系统的层次结构、主要功能和工作原理，以 Linux 为背景介绍文件的基本操作和应用实例，最后介绍 Linux 的特殊文件系统——proc 文件系统。

第 9 章介绍操作系统的保护与安全机制。本章从讨论计算机系统面临的安全威胁出发，介绍操作系统中各种安全机制的作用和工作原理。然后，介绍 Linux 中的安全机制和操作系统的安全评测标准。

附录 A ~ D 包含了 Linux 常用的命令、系统调用函数、C 函数和上机实验指导，作为参考资料提供给读者，帮助读者完成每章后面的上机实验。

由于作者才学有限，书中可能存在不足之处，恳请同行、专家及读者批评指正。

编　者

目　录

第1章 概 述

操作系统位于硬件与应用程序之间，是计算机系统中最底层的核心软件。它负责管理所有硬件资源，如处理机、内存、I/O 设备等，为应用程序提供使用这些硬件资源的功能函数。除此之外，操作系统还提供许多实用的软件工具，如命令解释程序、图形用户界面、即插即用功能等，使用户能够方便地使用计算机系统。本章回顾了操作系统的发展历程，阐述了操作系统的任务、地位及总体结构。

1.1 早期操作系统

1.1.1 操作系统的产生

1946 年，世界上第一台计算机 ENIAC 在美国宾夕法尼亚大学诞生，这是一台专门用于数字积分运算的十进制计算机。ENIAC 包含 18 000 个电子管，重 30 t。采用十进制，每秒能完成5 000 次加法，通过人工插接线路设定计算程序。1948 年，第一台二进制的冯·诺依曼结构计算机在英国的曼彻斯特大学诞生，它包含一个只有 1024 位的电子管存储器，首次使用了索引（Index）和基址（Base）寄存器，能对存储器单元进行寻址访问。从此，开始了辉煌的计算机发展历史。

早期的计算机几乎没有任何输入、输出设备，人们只能通过拔插接线插头或拨动控制面板上的开关，给计算机输入简单的程序和数据，并通过控制面板上的指示灯读出计算结果。后来，出现了纸带、卡片、打印机和磁带等简单的输入、输出设备。那时，人们在计算机上运行程序的过程如图 1-1 所示。

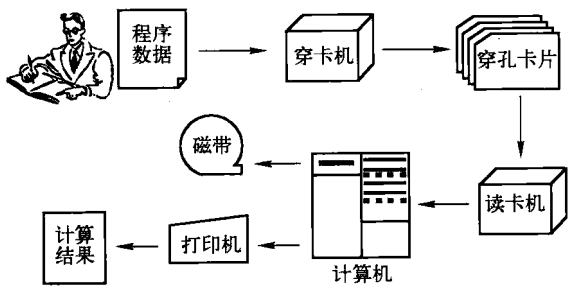


图 1-1 早期计算机上运行程序的过程

程序员编写好程序后，首先需要使用穿孔机，将程序和数据制作成纸带或卡片（穿孔的位置和数目表示不同的数字或字母）。然后将制作好的纸带或卡片放在与计算机相连的读卡设备上，启动计算机。计算机从读卡机上将程序和数据输入到内存中，并开始运行程序，最后将计算结果存储到磁带或通过打印机打印出来。

每运行一次程序，用户都需要独占计算机，重复上述的操作步骤。这意味着，程序员不

仅要掌握编程语言，还要掌握所有设备的操作过程，每次运行程序需要做许多额外的工作。另外，每个应用程序都需要包含驱动读卡机和打印机的代码，不仅加大了程序员的工作量，也增加了程序出错的机率。为此，人们想出一种解决办法，就是在计算机中预存一部分程序。这部分程序能够自动地完成从输入、编译、运行到输出的整个作业流程，被称为监控程序（Moniter）。这就是操作系统的雏形。

当时的监控程序包含以下功能：

（1）加载和启动作业

作业是指由计算机完成的一组相关任务，每个任务称为一个作业步。在早期计算机系统中，作业通常包含输入程序和数据、运行编译程序、运行目标程序和输出数据等作业步。每个作业包含一组穿孔卡片，多个作业的卡片组按照执行顺序放在读卡机上。监控程序依照次序读入作业的卡片组，将程序加载到内存，从指定地址启动执行程序。当作业结束时，监控程序清理处理机内部状态和初始化硬件设备，然后从读卡机读入下一个作业。

（2）解释执行 JCL，控制作业运行

作业的流程用作业控制语言（Job Control Language，JCL）来描述。这些 JCL 语句也被制作成穿孔卡片插在程序和数据卡片的中间，如图 1-2 所示。作业卡片组的第一张是包含 JOB 语句的 JCL 卡片，它说明了作业的相关信息，如作业名称、作业需要的内存大小、设备和执行时间等。作业卡片组的最后一张是包含 END 语句的 JCL 卡片，它是作业卡片组的结束标志。所有 JCL 语句（卡片）组成“作业说明书”。

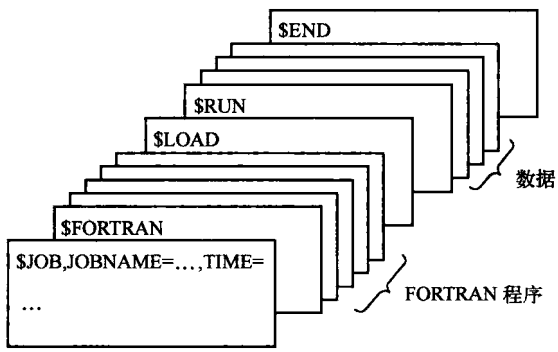


图 1-2　作业卡片组示例

监控程序在读入作业卡片时，会解释执行每张 JCL 卡片中的语句。例如，在读入图 1-2 中的作业卡片时，监控程序从第一张卡片中获得作业的名称和资源需求，为作业分配内存和设备；第二张卡片是 JCL 语句 $ FORTRAN，监控程序将 FORTRAN 编译程序装入内存，开始执行编译程序；FORTRAN 编译程序读入后面的包含 FORTRAN 程序的卡片，将其编译成目标代码存放到磁带上，然后返回监控程序；监控程序继续读入卡片，发现卡片是 JCL 语句 LOAD，于是从磁带上将目标代码装入内存；接下来读入的卡片是 RUN 语句，监控程序开始执行用户程序；用户程序则读入后面的数据卡片，进行处理、计算和输出结果，在运行结束时返回监控程序；监控程序继续读入卡片，下张卡片上的 END 语句告诉监控程序此作业结束，监控程序开始清理处理机内部状态和初始化硬件设备，准备执行下一个作业。

（3）设备驱动程序和公用子程序

监控程序中还包含了磁带、打印机、读卡机等设备的操作程序和常用的数学子程序。所有用户程序都可以调用这些程序来完成相应的功能。这样做的好处是不仅减轻了程序员的负担，使程序员有更多的精力去解决应用中的核心问题；而且，也能提高程序的运行效率，因为这些子程序通常是由经验丰富的程序员编写，其正确性也经过了时间的考验。

监控程序在实现上述功能的过程中，需要解决几个关键问题。而这些问题的解决需要依靠硬件的特殊支持，并由此产生了操作系统中重要的基本概念和机制。

第一个需要解决的问题是如何保证监控程序不被用户程序所破坏。监控程序和用户程序同处在内存中，如果用户程序能够访问监控程序所在的内存区域，那么就有可能破坏监控程序的数据或代码，造成整个计算机系统无法正常运行。因此，必须禁止用户程序访问监控程序所在的内存区域。

为此，硬件实现了界地址寄存器及地址越界检查部件。将内存的低地址部分（例如，0～k地址）分配给监控程序，而其余部分分配给用户程序。在界地址寄存器中置入分界地址 k（见图 1-3）。当用户程序执行一条访存指令时，地址越界检查部件将访存地址与界地址寄存器中的值 k 进行比较，如果访存地址大于 k，则允许访问，否则不允许访问并报错（置上硬件报错标志）。这样，从访问内存的硬件机制上保证监控程序不被用户程序所破坏。这种保护称为“存储保护”。

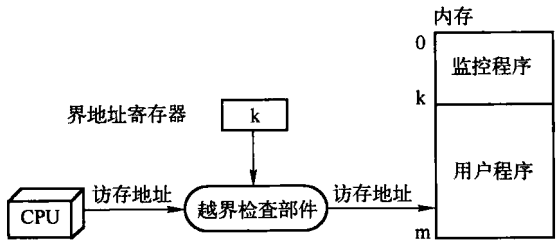


图 1-3　地址越界检查

第二个需要解决的问题是限制用户程序使用特殊寄存器。界地址寄存器就是其中的一个特殊寄存器，如果用户程序能够修改界地址寄存器，将其设置为小于 k 的值，这时，越界检查部件就不能阻止用户程序访问监控程序所在的内存区域。控制输入、输出设备的寄存器也是特殊寄存器，如果允许用户程序操作它们，就可能使这些设备无法正常工作。因此，必须限制用户程序使用特殊寄存器。

为此，硬件实现了核心态和用户态两种处理机执行状态，同时将指令集划分为普通指令和特殊指令两类，后者是专门用于访问特殊寄存器的指令。在处理机状态寄存器中，设置一个标志位表示处理机的当前执行状态。在核心态下，处理机允许执行全体指令集，包括特殊指令；而在用户态下，处理机只允许执行普通指令，不允许执行特殊指令。监控程序在核心态下运行，能够设置或改变处理机的执行状态。监控程序在转去执行用户程序之前，将处理机的执行状态设置为用户态，这样就能保证用户程序不能修改特殊寄存器。

采用硬件的存储保护机制和处理机执行状态机制，保证了用户程序不能访问监控程序所在的内存区域，同时也带来一个问题，就是如何让用户程序调用监控程序提供的功能子程序。这就是监控程序需要解决的第三个问题。

系统调用机制的实现正是为了解决这个问题。该机制依赖于硬件实现的一条特殊的

“陷入”（trap）指令，当用户程序需要调用监控程序的功能子程序时，执行 trap 指令（而不是 call 指令）。Trap 指令的执行结果是使处理机状态从用户态转变成核心态，同时将监控程序的地址置入程序计数器（PC）中。也就是说，执行 trap 指令后，处理机暂停执行用户程序，转去执行监控程序。监控程序根据系统调用散转表（参见 3.3 节），调用指定的功能子程序。然后，监控程序通过修改处理机状态和程序计数器 PC，使处理机恢复执行用户程序。

前面提到的 3 个问题实际上是任何操作系统都必须解决的问题。核心态、用户态、存储保护和系统调用机制是操作系统中的基本概念和机制，它们在现代操作系统中演变得更加复杂和完善，但它们的基本概念和功能没有变。在后面的章节中将对它们进行详细的论述。

监控程序的工作流程描述如下：

1）判断输入设备上是否有作业，没有则停止。

2）从设备上输入一道作业。

3）控制作业运行：

① 读入一张作业卡片，解释执行卡片上的 JCL 语句。

② 如果是“作业终止”语句，删除该作业，转 1）。

③ 根据 JCL 语义，在内存中建立用户程序的运行环境，在用户态下执行该程序。

④ 用户程序结束后，回到核心态下的监控程序，转①。

监控程序可以成批地执行在读卡机上排队的多个作业卡片组，但每次只允许一个作业进入内存中运行，因此，称之为单道批处理（Batch Processing）系统。出现和使用监控程序的时期称为操作系统的单道批处理时代。这个时代的代表产品是 20 世纪 50 年代末安装在 IBM 7090 计算机上的 FORTRAN Monitor System。

1.1.2 多道批处理系统

计算机硬件的发展使得操作系统由单道批处理时代进入了多道批处理时代。

首先是中断和 I/O 通道硬件机制的出现，使得处理机的程序执行过程与 I/O 设备的输入/输出过程能够同时进行。如图 1-4 所示，当程序需要输入/输出数据时，CPU 只需要向 I/O 通道发出I/O指令，然后继续执行程序；I/O 通道根据 I/O 指令，控制外部设备完成相应的输入/输出过程，然后向 CPU 发送一个中断信号；中断机制使 CPU 暂停执行当前的程序，转去执行一段中断处理程序，处理输入/输出的数据，之后继续执行原来的程序。

其次，硬件存储技术的发展使得计算机的内存容量迅速增大，可以容纳更多的程序和数据。还出现了可以随机读、写的磁盘系统，它造价低、容量大，读/写速度比磁带快，被用作 CPU 的外部存储设备，可以永久性地保存程序和数据。

这些硬件技术的发展使人们看到了提高计算机系统运行效率的新途径。在内存中同时存放多个作业，利用中断机制和 I/O 通道提供的 CPU 与 I/O 设备的并行处理能力，让 CPU 交替地执行这些作业。当一个作业进行数据的输入/输出时，让 CPU 去执行另一个作业。这就是多道批处理系统的基本思想。

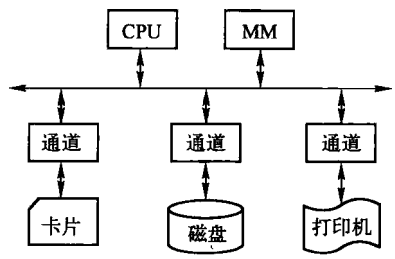


图 1-4　I/O 通道控制 I/O 设备

图 1-5 说明了 3 道作业交替执行的情况。CPU 先执行作业 A，当作业 A 进行输入/输出过程时，CPU 并

没有等待输入/输出过程的结束，而是执行作业 B。当作业 B 进行输入/输出过程时，CPU 执行作业 C，而当作业 C 进行输入/输出过程时，作业 A 的输入/输出过程已经结束，于是 CPU 继续执行作业 A。以这种方式，CPU 交替地执行多道作业，几乎没有任何空闲等待时间，因此提高了系统的作业吞吐量——在单位时间里能够运行更多的作业。

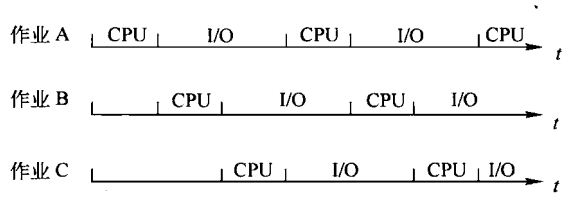


图 1-5　CPU 交替地执行 3 道作业

这种交替地执行多道作业的方式称为作业的并发执行。从宏观上看，多个作业在计算机系统中同时运行。为了实现多道作业的并发执行，必须修改单道批处理系统的监控程序，增加新的功能和机制，解决下面几个关键问题。

（1）作业和资源管理

多个作业在系统中同时运行自己的程序，它们都需要使用内存、CPU、I/O 设备和其他系统资源。操作系统必须对资源的分配和使用情况进行记录和管理，为此引入了“进程”概念。用进程代表“运行中的程序”，将资源分配给进程。进程是操作系统的基本概念，在后面的第 4 章里有详细的论述。

（2）处理机切换和调度

当正在执行的作业进行输入/输出过程时，处理机需要“切换”到另一个作业上执行。在切换过程中，需要保存前一个作业的执行现场，以便在此之后“切换”回该作业时能够恢复它的执行现场。既然系统中有多个可以运行的作业，那么让 CPU 去执行哪个作业？这是一个处理机调度问题，需要实现有效的处理机调度机制和调度算法。随着硬件技术的发展，计算机越来越复杂，可以包含多个处理机，而处理机又可以包含多个 CPU 核。因此，处理机调度一直是操作系统领域中的热门研究课题。

（3）I/O 设备竞争

多个作业同时并发执行，必然会带来对系统资源的竞争。在早期，这种竞争主要表现在对 I/O 设备的使用上。当时的 I/O 设备主要是读卡机和打印机，它们的速度相对于 CPU 而言非常慢，而且一台计算机能够挂接的 I/O 设备有限，作业运行中的大部分时间是在等待数据的输入/输出，或者等待使用 I/O 设备。输入/输出成为当时计算机系统的主要性能瓶颈。利用磁盘设备和 I/O 通道实现的 Spooling（假脱机）系统（见图 1-6），能够很好地解决这个问题。

Spooling 系统的基本思想是利用相对快速的磁盘作为输入/输出数据的缓存。具体地说，在磁盘上划出两块空间，分别用作“输入井”和“输出井”。由 I/O 通道控制读卡机，将作业的程序、数据和作业控制说明书提前读入并存放在“输入井”中。操作系统从“输入井”中读入作业说明书和程序，启动作业运行。作业在运行过程中，也从“输入井”中输入数据，进行处理。同时，作业的输出数据都写入到“输出井”中。当作业运行结束后，由 I/O