

高等学校教材

现代操作系统

教程

滕至阳 编著



高等教育出版社

高等学校教材

现代操作系统教程

滕至阳 编著

JS85/22

高等教育出版社

内容提要

本书把操作系统学科研究领域近年来的最新成果、主流操作系统的精华部分与传统教学内容有机融合,注重介绍该学科基本研究方法和基本发展规律。各章既有一般原理又有具体实例,既有传统内容又有新概念、新技术。与本书配套的多媒体教学课件生动有趣,便于自学。

本书可作为高等学校计算机及相关专业操作系统课程的教材,也可供计算机等级考试、成人自学考试的考生和计算机应用开发人员阅读参考。

图书在版编目(CIP)数据

现代操作系统教程/滕至阳编著. -北京:高等教育出版社,2000.2

高等学校教材

ISBN 7-04-007929-1

I. 现… II. 滕… III. 操作系统(软件)-高等学校-教材
IV. TP316

中国版本图书馆 CIP 数据核字(1999)第 64486 号

现代操作系统教程

滕至阳 编著

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

经 销 新华书店北京发行所

印 刷 国防工业出版社印刷厂

开 本 787×1092 1/16

版 次 2000 年 2 月第 1 版

印 张 18.75

印 次 2000 年 2 月第 1 次印刷

字 数 450 000

定 价 28.00 元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

序

在竞争已成为时代重要特征的今天，作为大学教材，我认为除了传授知识以外，更主要的作用是有利于学生能力的培养。

首先，教材要有利于培养学生获取知识的能力。因为我们面对科技迅猛发展、知识急剧增长的世界，从大学教材中学到的知识在十几年甚至数年后就必须更新，所以教材的重要任务是通过使用教材教会学生如何去获取知识。这就要求教材编著者对本门学科的认识论和方法论以及本门学科究竟培养学生一种什么样的学科素养有明确认识及透彻分析，并以此来贯穿教材编写的始末。这样，教材才能帮助学生构建一种正确的认识方法、思考方法、研究方法和学习方法，才能培养学生举一反三、自我拓展知识领域的能力。

其次，教材要有利于培养学生运用知识的能力。加强教材本身的应用性和实践性对学生运用知识能力的培养是相当重要的，这一特点被认为是“21世纪教材特点”。《现代操作系统教程》一书在重视学科理论知识的同时突出了理论的实用价值，同时注重由提出实际问题开始，随着问题的展开与解决，引入概念，介绍理论。书中许多例子和习题选自实际问题或有工程背景，要求学生以 Paper 或 Project 的形式自己去求解，达到培养动手能力的目的。

再次，教材要有利于培养学生的科技创新能力。当前高校学生接受或认识新事物的能力都已达到一定的高度，他们中的一些人已经不满足于呈现所有已知内容的教材形式。因此，教材内容不要总是面面俱到，有些内容只能点到为止，有些内容要联系学科发展的前沿。也就是说，教材编写要“留有余地”，为学生留下主动思考的空间。不经独立思考，是不可能创新思维的。另一方面，教材要反映学科间交叉以及学科的综合化趋势这一科学技术发展的主流。很难想象，一个没有横向思维、视野只局限于本专业的人在将来能有所创新。教材应在保有本学科一定的逻辑和体系的前提下，尽量地联系相邻、相关以及与之交叉的学科，明确本门课程知识在学科综合中的作用和地位以及向各方向上的渗透情况，也更应该论及其他学科的先进研究方法在本学科发展中的促进作用。扩大学生的视野并为学生接受新的、交叉领域的知识设有“接口”，有利于他们根据兴趣爱好选择研究方向。

最后，教材要主动适应现代教育模式，因为计算机辅助教学的兴起对教材形式提出了更高的要求。保证教材的时代感也是培养学生能力的一个重要方面。

通读《现代操作系统教程》，使人感到该书在有利于培养学生能力方面做了很多有益的尝试。在21世纪来临之际，我很高兴为此书作序。

顾冠群

1999. 9

于东南大学

前 言

随着计算机科学技术的迅速发展和计算机应用的不断深入，操作系统的新概念、新技术不断出现，迫切需要新的操作系统教材。另一方面，以计算机技术为核心的现代教育技术正催生出新的教学模式，也要求教材形式作相应变化。

《现代操作系统教程》一书是在“关于新操作系统教材内容的研究”（全国高等教育教材建设专业委员会第二批立项课题）的基础上写成的。与本书配套的多媒体教学课件是“九五”国家重点科技攻关 96-750 项目的研究成果。

为满足 21 世纪培养创新人才的需要，本书注重计算机操作系统学科的研究方法和发展规律的阐述。作者有如下考虑：第一，用通俗的文字向读者介绍操作系统学科中的重要概念和结论，并指出这些结论对于了解身边操作系统产品的性能和在计算机系统中发生的种种现象上的意义。同结论一道，说明得出结论的关键环节和关键方法，使读者获得学科知识的同时也能了解和掌握这门学科的基本研究方法。第二，努力吸纳该学科研究领域的最新成果和主流操作系统中的精华部分，并把它们与传统教学内容有机融合，强调操作系统学科知识体系的主根、主干及其基本发展规律，引导读者踏上寻找和判断新的知识生长点之路。第三，配置一定数量的思考题和课程设计大型作业，训练读者从模仿（根据例题和实例）走向创造的能力。第四，编制并提供多媒体教学课件，使教、学双方跟上并适应信息时代教学模式的发展与变化。

全书共分 13 章。各章既有一般原理又有具体实例，既有传统内容又有新概念、新技术，力求做到：内容精选，取舍得当；难点分散，体系合理。

限于作者水平，错误与不妥之处，恳请读者批评指正。

作者衷心感谢计算机科学家、中国工程院院士顾冠群教授为本书作序。

作 者

1999. 6

于南京玄武湖畔

责任编辑 鲍 涌
封面设计 王凌波
责任印制 杨 明

11

第1章 引言

1.1 什么是操作系统

操作系统尚未有一个被普遍接受的定义。但通常认为：操作系统既是计算机系统资源的管理员，又是计算机系统用户的服务员。资源管理以提高资源利用率为目标，以给用户服务提供尽可能多的服务项目和最大的方便为宗旨。管理与服务的功能用一组程序来描述，这组程序通过事件驱动以并发执行方式发挥作用。人们把这组程序称为操作系统，它是计算机系统中极为重要的系统软件。为了全面理解操作系统的概念，下面对它的基本功能、基本特征及它在系统软件中的地位作一介绍。

操作系统的基本功能是管理计算机系统的硬件、软件资源。计算机硬件资源有处理机(CPU)、主存、辅存、输入输出设备、通信设备；软件资源指除操作系统之外且存在于计算机系统内的程序、数据、信号量等信息。

处理机管理指操作系统根据一定的调度算法分配处理机。在单处理机系统中，进程管理也属于这一范围。在采用多道程序设计技术的操作系统中，处理机在操作系统的控制下交替地被多道程序所占用。在多处理机系统中，处理机的管理要复杂一些。一方面必须了解每台处理机的功能，以便把适合于该处理机的任务分配给它；另一方面，也应确切地知道一个作业中诸任务之间的串、并行关系。处理机集概念较好地解决了多处理机调度问题。

存储管理指分配、收回与保护存储单元。一道程序在启动执行前必须装入主存，操作系统应当根据程序的大小和当前主存空间的实际情况，为每一道程序分配使之能运行的必要的存储空间。当程序执行完后，操作系统把该程序所占用的全部存储空间收回，以作别用。显然，一道程序未必要全部装入主存才能运行。如果主存空间宽余，可以把程序全部装入。如果主存空间紧缺，则把程序的一部分装入，在运行过程中需要访问未装入的程序片段时，再把这些程序片段装入主存。由此产生了虚拟存储管理算法。一般认为，主存管理算法可推广到辅存(磁盘、磁带、光盘)空间的管理上。计算机系统中，辅存属于外围设备。

设备管理主要是对设备进行分配、回收与控制。通信设备在工作过程中的控制尤为重要。用户程序使用外围设备的步骤是先申请，再使用，最后释放。与此相应，操作系统必须把有限的设备进行优化调度，满足多个用户程序的要求，对分配出去的设备进行有效控制，发挥设备的最大能力，最后还要对用完的设备及时收回。为了提高设备效率，设备管理中引入了虚拟设备的概念和 SPOOLing 技术。

文件管理涉及文件存储空间的分配与回收、文件目录管理、文件读写与保护等。由于现代计算机系统内的信息资源常以文件的形式出现，因此管理好文件就能使软件资源发挥更好的效益。目前，一个全方位的以文档为中心的资源管理方法受到用户的欢迎。用户希望只需要对文档进行操作，而不必关心被操作的文档究竟是文件、程序、目录、声像数据、磁盘卷

标还是操作系统中的其他一些对象。

作业管理、通信事务管理是操作系统提供给用户的最直接的服务。一个作业从进入系统到运行结束，一般要经过收容、运行、完成三个阶段。相应地，操作系统提供作业注册、作业调度、作业终止等服务。电子邮件服务、文件传送与远地存取服务、作业的传送与操作服务是最常见的通信服务。

考察操作系统的基本特征，能帮助人们从更深的层次上认识操作系统。并发、共享、虚拟、不确定性和可重构性是现代操作系统的基本特征。

并发性(concurrency)是指两个或两个以上的进程的执行在时间上有重叠，即一个进程的第一个操作是在另一个进程的最后一个操作完成之前开始。一个程序只有在操作系统为其建立进程之后，它的执行才具有并发性^①。有关进程的概念将有详细的说明。现代操作系统是并发活动的管理机构，其本身也是并发执行的。并发执行有效地改善了系统资源的利用效率，提高了系统的吞吐量，但却使系统复杂化。操作系统必须具有控制和管理各种并发活动的的能力，无论这些活动是用户的还是操作系统本身的。

共享(sharing)是指系统中的硬件和软件资源能为多个用户共同使用。根据资源属性，通常有互斥共享和同时共享两种方式。一种资源(如打印机或内部链表)只有被使用者释放之后才能被另一使用者使用，这样的资源称为临界资源，它们要求互斥共享。用重入码编制的程序及公告栏信息等资源可被多个用户同时共享。由于资源数量有限，共享必然引发竞争。

虚拟(virtual)的本质含义是把物理上的一个变成逻辑上的多个。例如，多道程序设计技术能把一台物理 CPU 虚拟为多台逻辑上的 CPU，SPOOLing 技术能把一台物理 I/O 设备虚拟为多台逻辑上的 I/O 设备(虚拟设备)，动态地址变换机构能把一个统一编址的物理主存变成多个逻辑上独立编址的存储器(虚存)。此外，还有虚拟信道，虚拟计算机系统(虚拟机)甚至虚拟现实(拟实)等。

不确定性(nondeterministic)不是说操作系统本身的功能不确定，也不是说在操作系统控制下运行的用户程序的结果不确定(即同一程序对相同的输入数据在两次或两次以上运行中有不同的结果)，而是说在操作系统控制下的多个作业的运行顺序和每个作业的运行时间是不确定的。具体地说，同一批作业，例如 3 个作业 J1, J2, J3，两次或多次运行的顺序可能是不相同的，而且同一个作业(如 J1)这次运行需要 1 min，下次运行却需要 4 min 等。系统内部的各种活动是错综复杂的，与这些活动有关的事件，例如从外围设备来的中断、I/O 请求、程序运行时发生的故障等都是不可预测的。这是造成不确定性的基本原因。这种不确定性对系统是个潜在的危险，它与资源共享一起将可能导致与时间有关的错误。

可重构性(reconstruction)支持对计算机系统可靠性和配置动态化的考虑。在多处理机系统中应使操作系统具有这样的能力：当系统中某台处理机或存储模块等资源发生故障时，系统能自动切除故障资源，换上备份资源，对系统进行重构使之继续工作。现代操作系统允许在系统运行过程中动态地改变硬件的配置。当配置变更时，操作系统截获变更信息，然后自动重构系统配置信息。例如，Windows 95 的即插即用(plug and play)方案允许用户将其便携式系

^① 在并行处理、并行计算、并行程序设计等文献中，并行(parallel)有三种含义：同时性、并发性、流水线。但在操作系统领域，多数人使用“并发”一词，认为并行是并发的特例。

统从台式停靠站卸下并迅速移到另一位置。在此期间用户不必关闭计算机。当用户再度用机时，即插即用子系统已经重构好了 Windows 95 的新配置，用户马上就可以继续原来的工作。现在，用户对动态配置的要求越来越强烈。如用户有时用一个 PCMCIA 卡上的调制解调器作为 COM1 端口插入到便携机上，而有时又需要将便携机安装到停靠站上。又如，无线网络要支持可移动工作站，也必须能够进行动态配置。即插即用是一个受到关注的方案。

最后，再考察一下操作系统与计算机硬件、应用软件的关系。图 1-1 展示了这种关系。由图可以看出，应用软件(包括实用软件)以操作系统为支撑环境，换言之，操作系统为应用层提供服务。操作系统紧贴裸机，把裸机改造成功能更强大、使用更方便的虚拟机器。操作系统本身又分为内核和外核两部分。操作系统本身的活动在外核是并发的，而在内核通常是顺序执行的。例如，UNIX 的核心程序在执行期间，只要它自己不调用 sleep 原语就不会引起处理机的切换。这样做的好处是简化核心程序的编制。传统的内核包括中断处理、设备驱动、CPU 切换以及实施进程控制与通信等功能，这些功能由原语实现。内核微型化既能提高操作系统的并发程度又使操作系统能方便地被移植到不同芯片的计算机系统上，这就是所谓的微核(microkernel)技术。

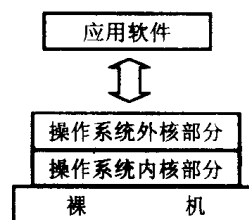


图 1-1 操作系统与硬件软件的关系

1.2 操作系统类型

1.2.1 单用户操作系统

个人计算机(PC)上配置的操作系统大多属于这种类型，它提供联机交互功能，用户界面特别友好。目前单用户操作系统采用了很多传统操作系统技术，例如虚存、并发、多任务等。联网、移动通信功能已成为它的一个重要组成部分。常见的单用户操作系统有 Windows 95，OS/2 等。

1.2.2 多道批处理系统

在这种操作系统控制下，用户作业逐批地进入、处理、离开系统。作业与作业之间的过渡不需要用户的干预。多道即在主存内同时有几道相互独立的作业在单 CPU 情况下交替地运行或在多 CPU 情况下并行运行。批处理系统的优点是提高了系统资源的利用率和作业的吞吐量，缺点是无交互性。它主要装配在用于科学计算的大型计算机上。

1.2.3 分时系统

它一般连接多个终端，用户通过相应的终端使用计算机。它为每个用户提供适当大小的时间片，采用轮转的方法为用户服务。若能保证足够快的响应时间并提供交互会话功能，则每个用户都感觉到好像独占一台计算机一样。在分时系统中必须注意响应时间的改善。减少

用户数目和减少时间片的大小无疑会改善响应时间，但这会损害系统的性能。采用重入码和虚存技术都能减少内外存之间的对换信息量，从而使 CPU 有更多的时间去处理终端用户作业，响应时间能显著缩短。

UNIX 系统是当今著名的分时操作系统。

1.2.4 实时系统

在这种操作系统的控制下，计算机系统能及时地响应外部事件的请求，在规定的时间内完成对该事件的处理并有效地控制所有实时设备和实时任务协调地运行。实时系统往往是专用的，在设计时首先要保证它的实时性和可靠性，其次才是系统效率。它常有两种类型即实时控制和实时信息处理。前者如生产过程、武器系统的实时控制，后者如银行业务、航空订票等实时事务管理。

但实际的系统往往同时具有批处理、分时、实时三种或两种功能。在此情况下，批处理作业往往作为后台任务。

1.2.5 网络操作系统

计算机网络中的各台计算机配置各自的操作系统，而网络操作系统把它们有机地联系起来，用统一的方法管理整个网络中的共享资源。因此，网络操作系统除了具备存储管理、处理机管理、设备管理、信息管理和作业管理外，还应具有高效可靠的网络通信能力和多种网络服务能力。网络用户只有通过网络操作系统才能享受网络所提供的各种服务。

1.2.6 分布式操作系统

分布式计算机系统分为两类：一类是建立在多处理机上的紧密耦合分布式系统；另一类是建立在计算机网络基础之上，称为松散耦合分布式系统。分布式操作系统是为分布式计算机系统配置的操作系统。它与网络操作系统相比更着重于任务的分布性，即把一个大任务分为若干个子任务，分派到不同的处理站点上去执行。它有强健的分布式算法和动态平衡各站点负载的能力。它是网络操作系统的更高级形式，具有强大的生命力。最近网络计算机与高速网络的发展，商品化的分布式操作系统必然出现。

1.3 操作系统运行环境

计算机硬件所提供的支持构成现代操作系统的硬件环境，其中最为重要的是中断机构。事件引发中断，中断必须加以处理，操作系统由此被驱动。操作系统是一个众多程序模块的集合。根据运行环境，这些模块大致分为三类：第 1 类是在系统初启时便与用户程序一起主动参与并发运行的，如作业管理程序、输入输出程序等。它们由时钟中断、外设中断所驱动。第 2 类是直接面对用户态(亦称常态或目态)程序的，这是一些“被动”地为用户服务的程序。这类程序的每一个模块都有一条系统调用指令与其对应，仅当用户执行系统调用指令时，对应的程序模块才被调用、被执行。系统调用指令的执行是经过陷入中断机构处理的。因此从

这个意义上说,第2类程序也是由中断驱动的。第3类是那些既不主动运行也不直接面对用户程序的、隐藏在操作系统内部的、由前两类程序调用的模块。既然前两类程序是由中断驱动的,那么第3类程序也是由中断驱动的。应当注意,操作系统本身的代码运行在核心态(亦称管态、特态)。从用户态进入核心态的唯一途径是中断。UNIX中的trap指令是专供用户程序进入UNIX核心而设置的。

1.4 操作系统回顾与展望

1946年第1台计算机问世时并没有操作系统,甚至没有任何软件。人们用手工操作的方法使用计算机。20世纪50年代出现了监督程序,它使作业与作业之间的过渡摆脱了人的干预,提高了计算机操作的自动化程度。监督程序成为现代操作系统的雏形。20世纪50年代末和60年代初,单道批处理取得成功。20世纪60年代中期出现的多道程序设计的操作系统和分时系统是操作系统发展的第2阶段。操作系统的许多基本特征在这一阶段已充分显示出来。这一阶段设计出来的操作系统功能较强、规模较大。在该阶段,对操作系统理论和结构进行了研究,取得了丰硕的成果。1969年,著名的UNIX系统问世。此时,已开始使用高级程序设计语言编写操作系统。20世纪70年代中期,操作系统进入第3发展阶段。1975年,UNIX系统成为真正的多用户分时系统。与此同时研制了网络操作系统和分布式操作系统。此外,有关操作系统理论的研究进一步深入。20世纪80年代以来,网络操作系统和分布式操作系统是发展的主导方向。由此带来的计算机系统的安全问题引起操作系统研制者的普遍关注。Carnegie Mellon大学1984年研制的Mach操作系统引入了线程(thread)概念,在多机操作系统中最为引人注目。20世纪80年代后期,随着计算机尤其是个人计算机的普及,操作系统的界面几乎全部采用了窗口技术。X Window是配置在UNIX系统上的图形用户界面,它独立于硬件厂家,既可运行在IBM PC机、大型机以及巨型机上,又可运行在X终端上。进入20世纪90年代,Windows 95, Windows NT, OS/2的窗口界面已为大家所熟悉,特别是IBM的OS/2,几乎全部采用了面向对象的设计方法。用户通过窗口操作可以相当方便地把PC机接入因特网(Internet)或接入移动通信网。窗口界面系统采用事件驱动方式。用户对键盘或鼠标进行的操作就是一个事件(实际上是向操作系统发出一个消息),操作系统内部有一个事件驱动控制进程,它负责接受输入事件并驱动相应的事件处理程序,最后给用户反馈信息。

一般认为,新一代操作系统应是智能型的。所谓智能,一方面是指在操作系统的设计过程中以知识为基础,要有知识获取和学习功能并能建立相应的知识库和推理机制;另一方面是建立更加友好、更加方便的智能化人机接口。采用倍受欢迎的多媒体技术,使计算机系统具有直接看(读)、听、说、写、图案识别、颜色识别、气味区分等能力。新的人机交互模式甚至能指导或提示用户作出正确或明智的决策。

新一代操作系统面对的将是一个超大规模的、多处理机并行的、开放的、分布的硬件体系结构。因此,面向对象与智能技术必然被广泛地用于21世纪的操作系统。

回顾操作系统的发展历史,清楚地看到,计算机硬件体系结构的革新和使用方便性的承诺,是操作系统发展的动力,抓住这一点,就能把握现代操作系统发展的方向。

1.5 操作系统启动过程简介

操作系统的初启是比较复杂的，随着版本或计算机类型而不同。大体上说，有以下过程：首先执行 ROM 中的自举程序，然后操作系统本身的引导程序取得控制权。系统引导程序把操作系统的其余全部代码装入主存，并把控制权交给操作系统的初启程序(如 UNIX 中的 /etc/init)。接着，初启程序对系统作必要的检查(如 UNIX 中执行 fsck 以保证根盘是好的)，置系统参数的初值，创建作业流管理进程随时准备为批处理作业提供服务，创建联机终端进程(在 UNIX 中由 /etc/rc 创建/etc/getty 终端进程)随时准备与终端用户交互，创建操作系统内部其他系统进程。最后初启程序把控制权转给 CPU 调度程序，此时 CPU 运行一个“闲逛”进程。至此整个系统便在操作系统的管理和控制下有条不紊地运转起来了。如果恰好从终端键入“终端接收”进程将与用户建立会话；如果恰好输入机上有作业等待操作系统服务，那么“作业管理”开始工作。下一章即作业管理。本教材将以作业管理作为讲解操作系统原理的切入点。读者从使用者的角度来理解操作系统，容易入门。

思考与习题

1. 什么是操作系统？它有哪些基本功能与基本特征？
2. 怎样理解“操作系统本身是一个并发系统”？
3. 什么是批处理、实时、分时系统？它们有什么特征？各适用哪些场合？
4. 中断有哪些类型？它在操作系统中起的作用如何？
5. 网络操作系统最基本的功能是什么？它最使你感兴趣的是什么？
6. 分布式操作系统与网络操作系统有什么不同之处？
7. 你使用过 UNIX 或 Windows 95 或 Windows NT 或 Netware 或其他操作系统吗？它或它们给你提供了哪些服务？
8. 操作系统发展的动力是什么？你对 21 世纪的操作系统有什么见解？
9. 进程“并发”与计算机系统中 CPU 的个数有无关系？为什么？
10. 阅读 UNIX 系统中 etc/rc 源代码，画出流程图。

第2章 作业管理

2.1 什么叫作业

操作系统是为用户服务的。用户交给计算机做的工作称为作业。例如，解一个方程组、发送一个电子邮件等均可称为作业。有些工作只需数条操作系统命令就能完成，也称为作业。不过管理这样的作业相对容易些。有些作业由程序、数据、作业说明书三部分组成。程序是问题求解的算法描述，数据是程序加工的对象，但有些程序未必使用数据，作业说明书是告诉操作系统本作业的程序和数据按什么样的控制要求使之执行。作业说明书主要包括三方面内容：其一，作业基本情况描述，如作业名、用户名、所使用的编程语言名等。其二，作业的控制描述，例如，对 C 语言程序先装入哪些模块，后再装入另外的模块等各作业步的操作顺序以及某步不能正常执行时的出错处理方法等。其三，资源要求描述，包括估计的主存需要量、计算时间、外设类型及数量、作业优先级等。

按作业提交的方式可把作业分为两类：批处理作业和交互式作业。批处理系统要求用户预先把作业的三个部分准备好后交给系统。这样的作业一般是运行时间比较长的计算任务，称为批处理作业。脱机是批处理作业的主要特征，因为用户将作业交给系统后便可离开机房，数天后来取结果。与批处理作业相反，交互式作业以联机为主要特征。用户独占终端实施交互式控制，特别适合于对程序的动态调试和修改。容易理解，这类作业放到批处理系统中处理是不适宜的，因为从作业提交到作业完成为止，批处理需要较长的周期，尤其是当作业程序本身有错时，一个作业需要多次修改和多次提交。

毫无问题，为制作一个作业，批处理作业的用户不但会用某种编程语言编写程序和准备数据，而且还要会用某种控制语言书写作业说明书；对交互式作业的用户，要会使用联机终端命令与系统交互。作为操作系统，应为作业的制作提供支持，这些支持组成了操作系统的用户界面。

2.2 操作系统提供的用户界面

界面有时又称为接口，意指用户是通过“界面”看待操作系统的，通过“接口”来使用计算机的。操作系统提供程序界面、联机用户界面、脱机用户界面。

2.2.1 程序界面

它由一组系统调用命令(简称系统调用或广义指令或访管指令)组成。每条系统调用命令都对应一个由操作系统设计者事先编制好的、能完成某些特定功能的例行程序。系统调用命

令有汇编语言和高级编程语言两种形式。在编制程序时，用户像使用其他过程调用语句一样使用系统调用命令，以便得到操作系统的服务。系统调用在程序一级上为用户提供支持，所以称为程序界面。

系统调用命令通常分为以下几类：

- (1) 与文件相关的命令。如创建文件、打开文件、读写文件、关闭文件等；
- (2) 与进程相关的命令。如进程创建、撤消、唤醒以及进程间通信等；
- (3) 与系统状态有关的命令。如取日历时间、取或设置终端信息等；
- (4) 与资源相关的命令(在 UNIX 系统中设备归入文件类)。如请求或释放指定大小的主存空间等。

显然，系统提供的系统调用命令越多，系统的功能越强，用户使用起来也就越方便。

2.2.2 UNIX 系统调用执行过程

通过系统调用命令调用操作系统内部的例行程序与一般的子程序调用虽然在形式上相同，但在实现机制上有原则区别。一般子程序调用并不涉及处理机运行状态的改变，而执行系统调用例行程序模块时涉及到处理机运行状态的改变：从用户态转到核心态，执行完例行程序后再返回用户态。UNIX 的系统调用命令是通过中断机构“陷入”(trap)操作系统核心的。无论是汇编形式的还是高级语言形式的系统调用命令，经过汇编或编译后都被转换成带有参数的 trap 机器指令。trap 指令是用户态程序进入核心态取得操作系统服务的唯一途径。trap 指令的八进制表示为 1044XX~1047XX(PDP-11 系统)，最末两个 XX 占 6 比特位，最多可表示 64 种不同的系统调用，它的值作为系统调用的编号。

系统调用通过系统内部的寄存器 R0, R1 来传递参数，或通过 trap 指令自带参数。R0, R1 既可保存输入参数又可存放返回值。自带参数只能用于输入，此时又可分为两种情况：①直接自带参数——所带的参数 1 和参数 2 紧跟在 trap 指令的后面，如图 2-1(a)所示；②间接自带参数——当 trap 指令的低 6 位为 00 时，表示其下一条是指针，它指向真正要执行的 trap 指令，在该指令下面的为间接自带参数，如图 2-1(b)所示。UNIX 规定自带参数最多 5 个。为使系统调用能方便地转至相应的例行程序，在系统中设置了一张系统调用入口表。该表有 64 个表目如图 2-2，每个表目有两项：一项是该系统调用自带参数个数，另一项是系统调用例行程序入口地址。当系统执行 trap 指令时，硬件中断机构产生的自陷中断(从用户态进入核心态)被操作系统俘获后转入系统调用中断处理：取 trap 指令低 6 位得到系统调用编号，根据编号

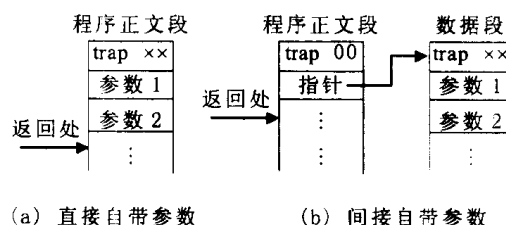


图 2-1 自带参数的两种形式

编号	自带参数个数	例行程序入口地址
0	0	&nullsys
1	0	&renit
2	0	&fork
3	2	&read
4	2	&write
	⋮	⋮
63	0	&nosys

图 2-2 系统调用入口表

直接查找系统调用入口表便可转至相应的例行程序。例如当系统调用的编号为 3 时，相当于 trap 指令的低 6 位为八进制 03，查得该系统调用有两个自带参数且例行程序入口地址为 &read。执行完例行程序后，退出中断，返回到用户程序的断点处(图 2-1 的返回处)继续执行。此时处理机又回到了用户态。

2.2.3 联机用户界面

联机用户界面又称为交互式界面，它由键盘命令和屏幕命令组成。键盘命令是由联机用户在交互式终端上通过键盘键入的命令，其特点是键入一条便即刻执行并把执行结果反馈到屏幕上，充分体现人机之间的交互性。键盘命令的格式随不同的操作系统而有差异，但功能基本上是相同的。UNIX 系统的键盘命令包括以下几类：

- (1) 系统管理类，如 login(注册)、logout(注销)等；
- (2) 编辑和文件管理类，如 vi(全屏编辑)、cp(拷贝文件)、ls(列目录)等；
- (3) 编译、链接类，如 cc(C 语言的编译)、link(目标模块链接)等；
- (4) 系统维护类，如 chown(改变文件主)、fsck(根盘检查)等；
- (5) 调试类，如 dbx；
- (6) 记账、日期等；
- (7) 作业控制方式转换，如 &(在命令结尾使用，该命令转后台执行)；
- (8) 其他，如管道命令、重定向命令等。

屏幕命令作为一种交互式手段在近年广为流行。它的出现使用户界面更加友好和开放。因为它独立于硬件厂商，所以既可运行在各种大型机上，又可运行在 PC 机上。屏幕命令包括多窗口命令、按钮命令、菜单命令、图标(icon)命令、滚动条命令等。屏幕命令的输入主要是靠鼠标器的点击、拖曳和移动。若点击选中特定菜单项，则系统即刻执行该菜单项所对应的实用程序而完成相应的功能。这种用法十分方便，因为它不要求用户记住任何命令。Windows 95, OS/2 等 PC 机上的操作系统已在图形用户界面上做得相当成功。X Windows 是 UNIX 系统上为用户提供的多窗口图形界面，已被移用于 UNIX 系统之外的 MS DOS, VMS 等多个不同的操作系统。

2.2.4 键盘命令执行过程

操作系统初启时，它为每个在线终端创建了联机终端处理进程——指执行终端处理程序的活动。首先，当无键盘输入时，该活动处于等待状态，一旦有键盘输入就执行终端处理程序中的键盘监督程序，使之读入一个键符、回送屏幕同时存入命令串缓冲区，直到遇到回车键，表示一个完整的命令已键毕。接着，“启动”终端处理程序中的命令解释程序。这里“启动”有两种含义：①对较为简单的命令，如列目录、拷贝文件等，命令解释程序本身便能完成，则由命令解释程序直接加以处理；②对于比较复杂的命令，如编译一个 C 语言程序，命令解释程序本身不能处理，则终端处理进程创建一个子进程，该子进程作为编译 C 语言程序的一个活动完成编译工作。最后，终端处理进程转回键盘监督程序，等待后续命令的键入，如此循环直到终端被注销。

应当指出，用键盘命令与计算机交互之初，用户打入的第 1 条命令必须是 login(注册)命

令，并给出用户名(注册名)和正确的口令(password)，只有注册成功后系统才接收其他命令，向系统管理员申请注册权。当结束终端工作时，用户必须键入 logout(注销)，退出系统后方能离开终端。如果忘了注销，系统不仅继续记账，而且存放在系统内的信息难保不被别人篡改。用户在系统内的活动特别是企图攻击系统的行为与时间都被记录在案以备审计。

2.2.5 屏幕命令执行过程

Apple 公司的 Macintosh 操作系统首推事件驱动控制方式来执行屏幕命令。一切外部触发被认为是一种事件，通过发送消息把事件告知系统，系统接收消息并根据消息类别作相应处理。每个消息都与一个特定的窗口联系在一起，并且每个窗口都有一个与其相关的“窗口过程”。图 2-3 是 Windows 95 的消息队列。Windows 95 把来自硬件设备的消息先放到原始消息队列。为了提供抢先多任务环境，系统把原始队列中的消息分派到 3 个相关队列。多队列技术保证单个出错的应用程序不会引起整个系统死锁。

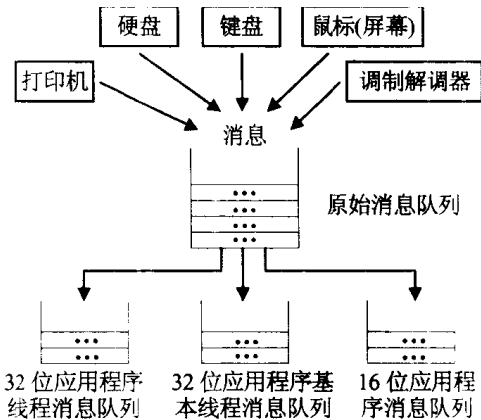


图 2-3 Windows 95 下多个消息队列的分派

当鼠标器光标在多窗口屏幕上点击、拖曳、移动时，相应的屏幕命令被选中，与屏幕事件相应的消息被分派到消息队列。“窗口过程”负责处理消息。

下面是系统内事件驱动控制程序的片段。它负责接收事件并在事件接收后从相应的队列中取出事件进行处理，最后从队列中清除事件。

```
// Start of fragment...
// Acquire and dispatch messages until a WM_QUIT message is received.
while (GetMessage(&msg,          // Message structure
             NULL,             // Handle of window receiving the message
             0,                // Lowest message to examine
             0)) {             // Highest message to examine
    if (!TranslateAccelerator (msg.hwnd, hAccelTable, &msg)) {
        TranslateMessage(&msg);    // Translates virtual key codes
        DispatchMessage(&msg);    // Dispatches message to window
    }
}
//...end of fragment

// Start of fragment...
```



```

switch (message){
    case WM_COMMAND: // Message:command from application menu
        #if defined (_WIN32)
            wParam = LOWORD(uParam);
            lParam = HIWORD(uParam);
        #else
            wParam = uParam;
            lParam = HIWORD(lParam);
        #endif
        switch (wParam) {
            case IDM_ABOUT:
                lpProcAbout=MakeProcInstance((FARPROC)About,hInst);
                DialogBox(hInst , // Current instance
                    "AboutBox", // Dlg resource to use
                    hWnd, // Parent handle
                    (DLGPROC)lpProcAbout): // About() instance address
                    FreeProcInstance(lpProcAbout);
                break;
            case IDM_EXIT:
                DestroyWindow (hWnd);
                break;
            default:
                return (DefWindowProc(hWnd, message, uParam, lParam));
        }
        break;
    case WM_DESTROY: // Message: window being destroyed
        PostQuitMessage(0);
        break;
    default: // Passes it on if unprocessed
        return(DefWindowProc(hWnd, message, uParam, lParam));
}
return (0);
//...end of fragment

```

IBM OS/2 采用面向对象的方法实现屏幕命令，它把屏幕上的界面元素看作对象(object)，屏幕命令就是用鼠标器操纵界面元素，相当于激活对象。这是一种比事件驱动更为先进的实现方法。新一代操作系统将全面采用面向对象的程序设计方法。