

国家精品课程配套教材

21世纪高等学校计算机规划教材

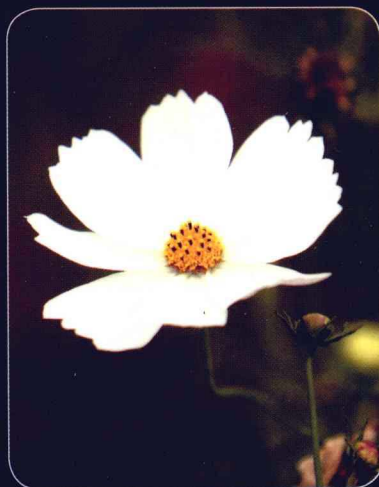
21st Century University Planned Textbooks of Computer Science

操作系统

Operating Systems

罗宇 文艳军 编著

- 从实用操作系统视角重组经典内容
- 从技术起源发展视角阐述技术内涵
- 从加强教学实践视角培养学生能力



精品系列



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

操作系统

Operating Systems

罗宇 文艳军 编著



精品系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

操作系统 / 罗宇, 文艳军编著. —北京: 人民邮电出版社, 2009. 5
21世纪高等学校计算机规划教材
ISBN 978-7-115-20569-8

I. 操… II. ①罗…②文… 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036860号

内 容 提 要

本书是国防科技大学国家精品课程配套建设教材。全书阐述了操作系统的基本概念、工作原理以及设计方法, 以多道程序技术为基础, 以实用操作系统设计思想为主线, 介绍操作系统涉及的关键内容, 并在最后一章中给出了具体的操作系统实例。本书依次介绍了操作系统的发展历史、操作系统运行机制、操作系统中的进程与线程管理、存储管理、文件管理和设备管理, 并对各种并发控制问题展开了讨论, 对前沿的分布式系统进行了介绍, 最后还详细介绍了 Windows NT 操作系统的结构和实现。

本书可作为高等院校计算机专业或计算机应用、通信与电子相关专业的教材和参考书, 也可供从事计算机设计、开发、维护和应用的专业人员阅读。

21 世纪高等学校计算机规划教材

操作系统

-
- ◆ 编 著 罗 宇 文艳军
责任编辑 滑 玉
执行编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
中国铁道出版社印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 14.75
字数: 383 千字 2009 年 5 月第 1 版
印数: 1—3 000 册 2009 年 5 月北京第 1 次印刷

ISBN 978-7-115-20569-8/TP

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

[illegible]

本书阐述了操作系统的基本概念、工作原理以及设计方法,以多道程序技术为基础,以实用操作系统设计思想为主线,介绍操作系统涉及的主要内容,并在最后一章中给出了当前流行的 Windows NT 操作系统设计实例。

本书可作为理工科院校计算机及相关专业的教材，也可作为计算机及应用专业自学考试教材，对于具有高级程序设计语言初步知识和对计算机有一定了解的专业人士，本书也不失为一本较全面的参考书。对于书中的不足之处，恳请专家及读者指正。

目 录

第 1 章 绪论.....1

- 1.1 什么是操作系统.....1
 - 1.1.1 操作系统的组成.....2
 - 1.1.2 操作系统的特征.....3
- 1.2 操作系统的发展历史.....5
 - 1.2.1 监督程序（单道批处理系统）.....6
 - 1.2.2 专用操作系统.....9
 - 1.2.3 多种方式的通用操作系统.....14
 - 1.2.4 并行与分布式操作系统及发展.....14
- 1.3 主要操作系统介绍.....15
 - 1.3.1 Windows 系列及 MS-DOS.....15
 - 1.3.2 UNIX 大家族（SVR4、BSD、Solaris、AIX、HP UX）.....18
 - 1.3.3 自由软件 Linux 和 FreeBSD 等.....22
- 习题.....26

第 2 章 操作系统运行机制.....27

- 2.1 中断和陷入.....27
 - 2.1.1 中断和陷入的区别.....28
 - 2.1.2 中断的分级与屏蔽.....28
- 2.2 中断/陷入响应和处理.....30
 - 2.2.1 中断/陷入处理基本概念.....30
 - 2.2.2 中断/陷入处理.....32
- 2.3 操作系统运行模型.....36
- 2.4 系统调用.....37
- 2.5 用户界面.....41
 - 2.5.1 命令语言.....41
 - 2.5.2 图形化的用户界面.....43
- 小结.....45
- 习题.....45

第 3 章 进程与处理机管理.....47

- 3.1 进程描述.....47
 - 3.1.1 进程的定义.....48
 - 3.1.2 进程控制块.....49

- 3.2 进程状态.....52
 - 3.2.1 进程的创建与结束.....52
 - 3.2.2 进程状态变化模型.....54
 - 3.2.3 进程的挂起.....56
- 3.3 进程控制与调度.....56
 - 3.3.1 进程执行.....57
 - 3.3.2 进程调度.....58
 - 3.3.3 调度算法.....62
- 3.4 作业与进程的关系.....66
- 3.5 线程的引入.....69
- 小结.....71
- 习题.....72

第 4 章 并发控制.....75

- 4.1 并发执行实现.....75
 - 4.1.1 并发编程方法.....76
 - 4.1.2 并发执行的实现.....76
- 4.2 同步与互斥.....78
 - 4.2.1 同步与临界段问题.....78
 - 4.2.2 实现临界段问题的硬件方法.....80
 - 4.2.3 信号量.....81
 - 4.2.4 同步与互斥举例.....84
- 4.3 消息传递原理.....89
 - 4.3.1 消息传递通信原理.....89
 - 4.3.2 消息传递通信示例.....90
- 4.4 死锁.....92
 - 4.4.1 死锁示例.....92
 - 4.4.2 死锁的定义.....95
 - 4.4.3 死锁的防止.....97
 - 4.4.4 死锁的避免.....99
 - 4.4.5 死锁的检测.....101
 - 4.4.6 死锁的恢复.....102
 - 4.4.7 死锁综合处理.....102
- 小结.....103
- 习题.....104

第5章 存储管理	108	7.1.4 文件控制块.....	175
5.1 连续存储分配.....	108	7.2 文件目录结构.....	175
5.1.1 单道连续分配.....	108	7.2.1 一级目录结构.....	176
5.1.2 多道固定分区法.....	111	7.2.2 二级目录结构.....	176
5.1.3 多道连续可变分区法.....	113	7.2.3 树形目录结构.....	177
5.2 不连续存储分配.....	115	7.2.4 无环图目录结构.....	178
5.2.1 分页管理.....	115	7.3 文件存储器空间布局与管理.....	180
5.2.2 分段管理.....	120	7.4 文件访问系统调用.....	182
5.2.3 段页式管理.....	122	7.4.1 传统文件系统调用实现.....	182
5.3 虚存管理.....	124	7.4.2 Memory-Mapped 文件访问.....	184
5.3.1 请求分页虚存的基本思想.....	124	7.5 文件保护.....	185
5.3.2 请求分页虚存管理的实现.....	125	7.5.1 文件访问保护.....	185
5.3.3 页面置换策略.....	128	7.5.2 文件备份.....	186
小结.....	134	7.6 文件系统的基本模型.....	187
习题.....	135	小结.....	190
第6章 设备管理	139	习题.....	191
6.1 I/O 硬件概念.....	139	第8章 分布式系统	193
6.1.1 常见外部设备的分类.....	139	8.1 分布式系统的特点.....	193
6.1.2 设备控制器(I/O 部件).....	140	8.1.1 分布式系统的定义.....	194
6.1.3 I/O 控制方式.....	142	8.1.2 分布式系统的优势.....	194
6.1.4 I/O 控制方式的发展过程.....	145	8.1.3 分布式系统的特性.....	194
6.2 设备 I/O 子系统.....	145	8.1.4 分布式系统设计难点.....	196
6.2.1 设备的使用方法.....	146	8.2 几种分布式应用模型.....	196
6.2.2 I/O 层次结构.....	148	8.2.1 客户机/服务器模型.....	197
6.2.3 设备驱动程序.....	151	8.2.2 处理机池模型.....	198
6.2.4 缓冲技术.....	153	8.2.3 对等模型.....	199
6.3 存储设备.....	156	8.2.4 集群模型.....	199
6.3.1 常见存储外部设备.....	156	8.3 分布式系统实现模型.....	200
6.3.2 磁盘调度.....	159	8.4 分布式操作系统主要研究内容.....	202
6.3.3 磁盘阵列.....	163	8.5 分布式系统基础——通信协议概念 简介.....	203
小结.....	167	8.5.1 TCP/IP 简介.....	204
习题.....	168	8.5.2 远程过程调用.....	205
第7章 文件系统	170	小结.....	207
7.1 文件结构.....	170	习题.....	207
7.1.1 文件的概念.....	170	第9章 Windows NT 操作系统	208
7.1.2 文件的逻辑结构.....	171	9.1 历史.....	208
7.1.3 文件的物理存储.....	172	9.2 设计目标.....	209

9.3 系统结构.....	210	9.6 文件系统.....	223
9.4 系统组件.....	212	9.6.1 内部格式.....	223
9.4.1 硬件抽象层.....	212	9.6.2 恢复.....	225
9.4.2 内核.....	213	9.6.3 安全.....	225
9.4.3 执行体.....	215	9.6.4 压缩.....	225
9.5 环境子系统.....	222	小结.....	226
9.5.1 Windows 环境.....	222	习题.....	226
9.5.2 MS-DOS 环境.....	223	参考文献.....	227
9.5.3 登录和安全子系统.....	223		

第 1 章

绪论

计算机系统在经济建设和人们生活中起着越来越重要的作用。操作系统是计算机系统中不可或缺的系统软件，是计算机系统的控制中心。一方面，操作系统将裸机改造成为功能强大、系统各部件高性能运行、使用方便灵活、安全可靠的虚拟机，为用户提供了计算机系统的良好使用环境；另一方面，操作系统采用合理有效的方法组织多个用户任务共享计算机的各种资源，最大限度地提高资源的利用率。

自从世界上第一台计算机 ENIAC 于 1946 年问世以来，计算机在运算速度、存储容量、元器件工艺及系统结构等方面都有了惊人的发展。以前，人们按照计算机元件工艺的演变过程将计算机的发展划分为 4 个时代：电子管时代、晶体管时代、集成电路时代和大规模集成电路时代。与硬件发展相类似，人们也将操作系统的演变和发展过程划分为 4 个时代：单道批处理时代，多道批处理、分时、实时系统时代，同时具有多方面功能的多方式系统时代，并行与分布式系统时代。

近年来，设备仪器的智能化使得嵌入到各种设备仪器当中承担控制及数据处理的计算机系统发展很快，这些计算机最初只是担负简单的控制及处理任务，直接运行应用程序。但是随着设备仪器功能的扩展和效率的提高，支持多任务运行的操作系统也成了嵌入式系统必然的选择。

本章将介绍什么是操作系统及操作系统在计算机系统中的地位和作用，并通过阐述操作系统历史的演变过程，使读者对操作系统中基本概念、技术的产生和发展之类的问题有一个直观、形象的了解，从而使读者对不同类型操作系统的基本特征、今后的发展动向以及对现在流行的操作系统有更深刻的认识。

1.1 什么是操作系统

众所周知，处理机、主存、磁盘、终端、网卡等硬件资源通过主板连接构成了看得见摸得着的计算机硬件系统。为了能使这些硬件资源高效地、尽可能并行地被用户程序使用，也为了给用户程序提供易用的访问这些硬件的方法，我们必须为计算机配备操作系统软件。

操作系统是一种系统软件，它由许多程序模块组成。操作系统是计算机系统软硬资源的控制中心，它组织单个或多个用户以多任务并发或并行运行的方式，尽量合理、有效地共享使用计算机的各种资源。从某种意义上说，操作系统可以与政府类比，政府要管理社会资源，组织经济体及其他实体运行，提供对社会资源的使用服务等。

操作系统的功能就是管理计算机的处理机、主存、外设等硬件资源，提供并管理文件等逻辑资源，组织用户任务（如以进程的形式）高效使用这些资源。

1.1.1 操作系统的组成

操作系统在计算机系统中的地位与政府在国家中的地位一样非常重要。我们先来看一看操作系统在计算机系统的位置。图 1-1 描述了计算机系统的软件层次及构成。大体上，计算机系统可以粗分为三部分：最底层是硬件，包含处理机、主存、外设等资源；最上面是用户层软件，不论是系统配备的开发用软件（如程序编译器），还是服务器类软件（如数据库管理器），或者是用户自开发的应用程序，都看作是用户层软件；中间是我们将要学习的操作系统内核软件。操作系统内核软件实现了操作系统的主要功能。

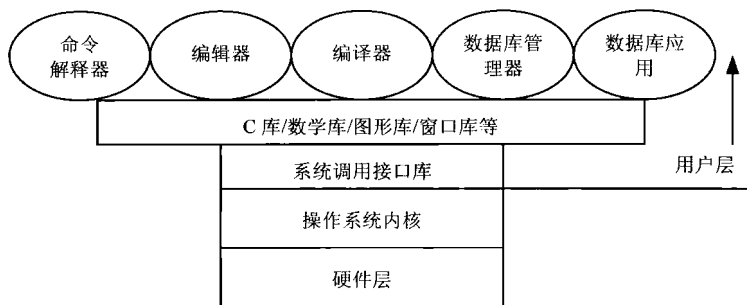


图 1-1 计算机系统中的软件构成

当用户在计算机中安装操作系统时，如图 1-1 中的操作系统内核、命令解释器、编辑器、Web 浏览器、编译器、各种库程序甚至数据库管理器、Web 服务器程序等都从安装介质拷贝到了计算机系统的磁盘上。大家往往以为它们都是操作系统的构成成分，但从操作系统历史及技术层面来说，操作系统主要由操作系统内核、命令解释器（或程序管理器）构成。

安装操作系统时，命令解释器（图形用户界面中对应的程序管理器，如 Windows 的 explorer.exe）是必不可少的一个程序，因为没有它，用户就无法操纵计算机，无法输入命令让计算机去执行。在现代操作系统实现中，命令解释器程序没有作为操作系统内核的组成部分，但它在使用计算机过程中是不可缺少的，用户在终端上输入命令就是由命令解释器程序接收并解释执行的。

其他的操作系统内核层之上的程序则是根据计算机的定位（定位为服务器或工作站）而选择安装的。如果将计算机定位成程序开发用的工作站，那么必须安装编辑器进行程序的编辑，安装编译器进行程序的编译。如果把计算机作为一个网络上的 Web 服务器，那么必须安装 Web 服务器程序。无论是用户开发的普通 C 语言程序还是数据库应用程序，都要在操作系统安装并运行后才能使用。这些操作系统内核层之上的程序，不管是命令解释器还是 Web 服务器或用户自编的程序，都是通过操作系统内核提供的进程机制来运行的。

图 1-1 中的各种库程序实际上就是一些可以重用、共用的用户子程序。它们提供了形形色色的功能。系统提供这些库程序是为了方便用户编程，用户不必为了实现一个较通用的功能再重写上述库程序的代码，而只要引用库程序中的函数即可。库程序可以看作一些通用的、公共的程序集合，利用内核提供的简单的资源管理功能实现复杂的复合功能。它位于用户层程序的底层，是在用户态下执行的基础的公共程序。之所以不将这些基础的公共程序放到操作系统内核中实现是因为它们不涉及系统公共资源的管理，或是为了减少内核的大小。

从现今的技术层面看，操作系统则只包含图 1-1 中的操作系统内核，它是一个非常重要的系统程序，管理着系统中所有公共的资源，并提供了实现程序运行的进程机制。由于操作系统内核

工作的重要性、特殊性,使得它必须在一种特殊的保护状态下运行,以免受到其他用户程序的干扰和破坏,它提供了一组称为系统调用的接口供上层程序调用,从而保证操作系统内核在特殊的保护状态下运行的需求,并且满足上层程序对系统资源的申请、使用、释放以及进程的创建、结束等诸多功能的需求。本书以后所要介绍的操作系统主要内容就是这里所说的操作系统内核。

操作系统内核位于硬件之上,是其他软件运行的关键支撑软件。构成操作系统内核的主要功能模块有:进程管理与调度、存储管理、输入/输出(I/O)与中断处理、文件系统等。这些将在以后的章节中介绍。

1.1.2 操作系统的特征

操作系统内核是一个在特殊保护状态下运行的系统软件,它为用户层程序提供尽可能多的系统服务(如文件访问、数据I/O等),它的主要目的是让用户层程序高效、安全地共享系统资源,同时内核必须提供多道程序并发运行的机制。

操作系统的主要特征如下:

- **并发(concurrent)** 多道程序可以轮流在单处理机上运行。
- **共享(sharing)** 各种并发程序正确共享使用系统软、硬件资源。

下面我们从不同的角度来理解操作系统内核特征。

1. 操作系统作为特殊子程序

从图 1-1 所示的计算机软件层次图中可以看出,操作系统内核位于计算机硬件之上。操作系统内核为用户层的程序提供了系统调用功能。系统调用可以看成是特殊的公共子程序,因为这些程序提供了一些可以被任意用户层程序调用的公共的功能,所以用户不需要再编写实现这些功能的程序,只要调用操作系统内核提供的相应“系统调用”即可。但是,要注意到系统调用的特殊性,即系统调用处理程序运行在一种特殊的保护状态下。在这种状态下程序可以执行一些特权指令,可以访问到用户层程序所访问不到的存储空间。系统调用之所以具有这样的特殊性是因为系统调用处理程序涉及系统共享资源的操作。

举例来说,因为 $\sqrt{x}$ 的求值是许多用户程序都要做的工作,就可以把它作为一个公共子程序实现。那么它需要作为系统调用在操作系统内核实现吗?回答是否定的。虽然 $\sqrt{x}$ 需要许多条机器指令来实现。但因为它不涉及系统的共享资源,而只需要对输入变量 x 进行操作,因此可以把它作为数学子程序库中的子程序来实现。

以计算机使用软盘为例,用户如果需要与软盘交换数据,可以通过对在软驱控制部件中的寄存器设置不同的值来实现设备初始化、移动磁头、读/写数据等命令功能。其中,最基本的命令是读/写命令,它们需要许多参数,如磁盘块地址、每个磁盘的扇区数、物理介质中所用的记录模式、扇区间距等。当操作完成时,软驱控制部件中的状态寄存器中有一些状态位,必须由程序判定是正常完成还是异常结束。在启动读/写命令前还需要判定软驱电动机是否已启动,若未启动,还需要先启动电动机。如果将这些操作都交给用户来编程处理,不仅复杂,而且每个用户都重复编程,多个用户使用还会引起混乱。因此操作系统提供给用户一个简单的文件使用界面,即软盘上包含多个文件,每个文件可以按照读/写方式打开,然后进行读/写,最后关闭文件。用户无需知道电动机如何启动、如何读写数据,也不需要知道要读写的数据存放在软盘上的哪一个扇区,只需要知道读/写哪个文件的哪一段数据,利用这个简单的文件使用界面就可以与软盘进行数据交换。这个文件使用界面由操作系统内核中的系统调用实现,因为软驱不是某个用户的私有资源,软盘上的文件可以供多个用户所访问,涉及软驱和文件的管理数据都应该受到保护,所以文件使用以

操作系统内核系统调用形式实现。

2. 操作系统作为资源管理者

计算机由处理机、主存、辅存、终端设备、网络设备等硬件资源所组成。处理机提供了程序执行能力；主存、辅存提供了程序和数据的存储能力；终端设备提供了人机交互能力；网络设备提供了机器间通信的能力。这些硬件资源需要高效地被计算机用户使用，因此必须有适合每种硬件资源特点的资源分配和使用机制。

为了使硬件资源充分发挥它们的作用，必须允许多用户或者单用户以多任务方式同时使用计算机，以便让不同的资源由不同的任务同时使用，减少资源的闲置时间。例如，当一个用户任务将文件内容从磁盘往主存的缓冲区读出时，另一个用户任务可以让自己的程序在处理机上运行。这样，处理机、主存、磁盘同时工作，也就提高了资源利用率。

要让每种资源被多用户任务充分地利用，就需要研究每种资源的特点。对于单处理机来说，它只能执行一个指令流。如果多个用户任务都要使用它，那只有让多个用户任务的程序分时地在处理机上运行，也就是说，处理机交替地运行多个用户任务中的程序。这意味着操作系统要合理调度多用户任务使用处理机。对于存储设备，它是为程序和数据提供存放空间的，只要多个用户的程序和数据按照规定的位置存放，互不交叉占用，它们是可以共存的，操作系统要做的工作就是管理存储空间，把适用的空间分配给用户的程序和数据使用，以便当用户任务访问这些程序和数据时能够找到它们。

针对不同资源的特点，资源管理包含两种资源共享使用的方法，即“时分”和“空分”。时分就是由多个用户进程分时地使用该资源，除了上述的处理机外，还有很多其他资源也必须分时地使用，如外设控制器、网卡等，这些控制部件包含了控制输入/输出的逻辑，必须分时地使用。空分是针对存储资源而言的，存储资源的空间可以被多个用户进程共同以分割的方式占用。

在时分共享使用的资源中，有两种不同的使用方法。

(1) 独占式使用

独占表示某用户任务占用该资源后，执行了对资源的多个操作，使用了一个完整的周期。例如，如果多用户任务使用打印机，那么对打印机的独占式使用是指多用户任务一定是分时地使用该打印机的，每个用户任务使用打印机时，执行了多条打印指令，打印了一个完整的对象（如完整的文件）。这里每个用户任务要执行多条打印指令，为了不让多条打印指令执行过程中断，用户任务需要在执行打印指令前申请独占该打印机资源，执行完所有打印指令后释放。

(2) 分时式共享使用

独占式使用不能发挥资源的利用率，为了提高资源的使用效率，希望对资源的使用尽可能“并发”共享。这种共享使用是指用户任务占用该资源无需使用一个逻辑上的完整周期。譬如说对处理机的使用，用户任务随时都可以被剥夺 CPU，只要运行现场保存好了，下次该用户任务再次占用 CPU 时就可以继续运行。再如对磁盘的输入/输出，当一个用户任务让磁盘执行了一条输入/输出请求后，其他用户任务又可向磁盘发出输入/输出请求，系统并不要求某个用户任务的几个输入/输出请求之间不能插入其他用户任务的输入/输出请求。

从上述使用方法可以看出，“独占式使用”资源利用率不如“分时共享使用”方式高，因此操作系统设计时应分析各种资源的特点，尽可能在保证资源使用正确的前提下改进对资源的使用方式，提高资源利用率。

操作系统应针对不同的资源类型，实现不同的资源分配和使用策略，并为资源分配、释放、使用提供相应的系统调用接口。

3. 操作系统提供程序并发运行机制

用户可使用计算机进行科学计算、数据管理、通信、控制等工作。要实现这些任务，必须要由相应的程序实现。用户使用计算机的处理机来执行程序，用程序驱动外部设备来进行数据交换与控制，驱动网络设备来进行通信。用户的意图必须由程序及程序的输入参数表示出来，为了实现用户的意图，必须让实现相应功能的程序执行；为了能让程序执行，需要由操作系统为程序及程序数据安排存储空间；为了能提高资源利用率，增加并发度，还必须让多个用户程序能分时占用处理机；让一个程序还没运行完就让另一个程序占用 CPU 运行，就必须保存上一个程序的运行现场；因此必须要对实现各种用户意图的各个程序执行进程描述和控制。

要说明程序执行的状态、现场、标识等各种信息，有选择地调度某个程序占用 CPU 运行，必须由操作系统完成，这也是为了实现程序对 CPU 的分时使用。

操作系统一般使用进程机制来实现程序的执行。进程是指进行当中的程序，也就是指程序针对某一数据集合的执行过程。操作系统的进程调度程序决定 CPU 在各执行程序间的切换。操作系统为用户提供进程创建和结束等系统调用功能，使用户能够创建新进程、运行新的程序。操作系统在系统初始化后，会为每个可能的系统用户创建第一个用户进程，用户的其他进程则可以由先前生成的用户进程通过“创建进程”系统调用陆续创建，以完成用户的各种任务。

在支持交互使用计算机的系统中，用户的第一个进程往往运行命令解释器程序（或者图形用户界面中的程序管理器，如 Windows 操作系统的 explorer.exe），这个程序会从终端上获得用户输入的命令（或者由程序管理器得到用户单击图符的信息）再进行相应的处理，可能会调用操作系统的创建进程系统调用，创建新进程去运行实现命令功能的程序。例如，在 Linux 操作系统控制的终端上输入以下命令：

```
$ cp /home/ly/test.c /home/wq/hello.c
```

那么这一行字符串会由命令解释器程序获得，它会创建一个子进程，由子进程运行 cp 实用程序，由 cp 实用程序建立一个新文件/home/wq/hello.c，并把/home/ly/test.c 文件的内容读出来，写入 hello.c 中。

1.2 操作系统的发展历史

在计算机刚刚诞生的 20 世纪 40 年代，计算机系统仅由硬件和应用软件组成。在这一时期，整个计算机系统是由用户直接控制使用的，所以又称为手工操作阶段。当时的计算机不仅速度慢，存储容量小，而且外部设备简单，辅存主要借助磁带实现，如图 1-2 所示，整个计算机系统由单个用户独占使用。用户使用计算机的大致方法是：将程序和数据以穿孔方式记录在卡片或纸带上，把卡片或纸带装在输入设备上；然后在控制台上形成输入命令并启动设备，将卡片、纸带信息或磁带上的信息输入到指定的主存单元；接着在控制台上设置主存启动地址，并启动程序运行；

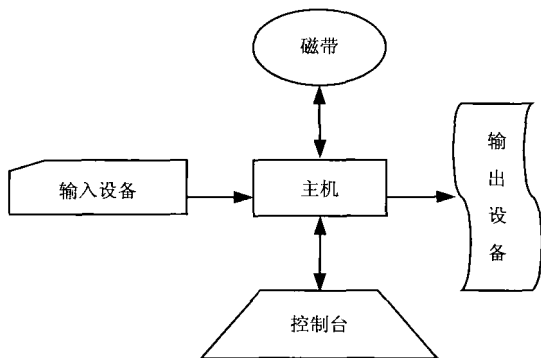


图 1-2 早期的计算机系统

最后在打印机等输出设备上取得程序运行的结果。

显然,在这种使用计算机的方式下,用户独占整机资源,使用机器语言编写程序,并且对计算机各部分的工作直接实施人工干预,或者通过用户自己编写的程序控制。在硬件各部分速度较低并且程序量较小的情况下,这种方式还能被人们所接受。但是,随着计算机速度的提高和 Fortran、COBOL 这类高级程序设计语言的问世,这种方式势必使人无法忍受。例如,用户如果想运行一段用 Fortran 语言编写的程序,必须首先把存有 Fortran 编译程序的磁带安装在磁带机上,将 Fortran 编译程序和用户编写的 Fortran 源程序调入主存对 Fortran 源程序进行编译;然后再安装链接程序所在的磁带,对编译好的程序进行链接形成目标程序;最后启动目标程序运行。由此可见,由于一批包括语言编译器在内的系统软件的问世,使用户的上机过程变得更繁杂,并增加了程序运行前的准备时间。由于计算机速度的提高,上述人工干预势必造成更大的资源浪费。为了缩短运行前的准备时间,提高计算机资源的利用率,人们提出了简单的改进措施,引入了“系统操作员”的概念。各用户将自己的程序以及程序的运行步骤(控制意图)交给系统操作员,系统操作员将这种形式的一批用户作业按类进行划分,每次处理一类作业。例如,将需要进行 Fortran 编译程序的作业组织成一类一起编译,并由系统操作员控制计算机运行用户程序。当然,这种使用计算机的方法仍旧停留在手工操作阶段,人的操作速度与机器运行速度相比仍存在着极不匹配的矛盾。由于人的操作缓慢,使得计算机资源大部分时间闲置,因此急需程序来代替手工操作。

1.2.1 监督程序(单道批处理系统)

20 世纪 50 年代,为了减少系统操作员工作所花的时间,提高资源利用率,人们开始利用计算机系统软件来代替系统操作员的部分工作,从而产生了最早的操作系统——早期批处理系统。

它的基本思想是:设计一个常驻主存的程序(监督程序 Monitor),操作员有选择地把若干用户作业合成一批,安装在输入设备上,并启动监督程序。然后由监督程序自动控制这批作业运行。监督程序首先把第一道作业调入主存,并启动该作业。一道作业运行结束后,再把下一道作业调入主存启动运行。待一批作业全部处理结束后,系统操作员则把作业运行的结果一起交给用户。按照这种方式处理作业,各作业间的转换以及各作业的运行完全由监督程序自动控制,从而减少了部分人工干预,有效地缩短了作业运行前的准备时间。

作业(Job)就是用户在一次上机活动中要求计算机系统所做的一系列工作的集合。从执行的角度看,作业由一组有序的作业步组成,如编译、运行分别称为不同的作业步。

监督程序取代系统操作员的部分工作后,用户也应以某种方式告知监督程序其作业的处理步骤。因此,在早期批处理系统中引出了“作业控制语言”和“作业控制说明书”的概念。作业控制说明书是利用作业控制语言编写的用以控制作业运行的一段描述程序。在组织一道作业时,通常将作业控制说明书放在被处理的作业前面(或插入适当位置),监督程序则通过解释执行作业控制说明书中的语句来控制作业运行。典型的卡片作业结构如图 1-3 所示。

卡片叠中某些卡片表示了作业控制语句,监督程序通过逐条解释执行该说明书中的作业控制语句自动控制作业运行。`$JOB` 语句说明了该作业的名字,预计最大执行时间等信息。解释 `$FORTRAN` 的结果是把 Fortran 编译程序调入主存,并启动编译程序编译后面的源程序。编译结束后,控制返回到监督程序。监督程序解释 `$LOAD` 的结果是通过链接程序把经过编译的程序链接起来,形成可执行程序。最后解释 `$RUN`,从而启动可执行程序执行。

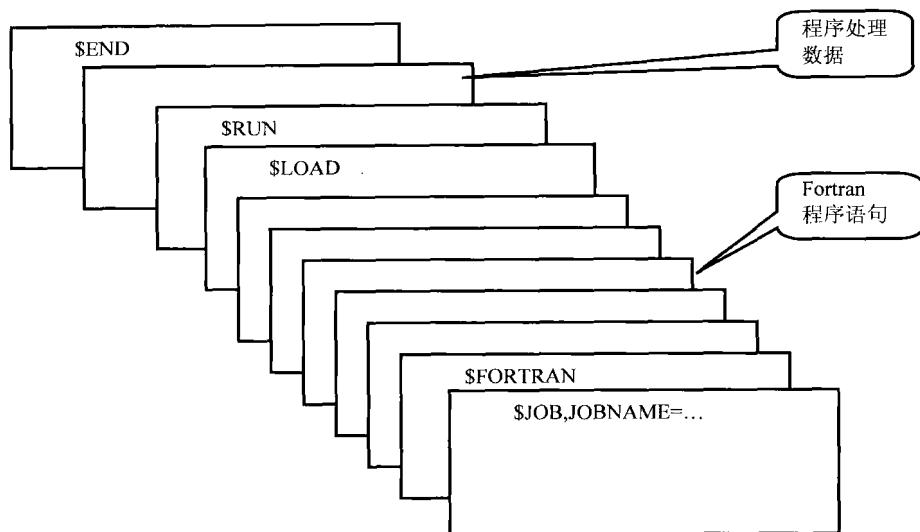


图 1-3 典型的卡片作业

监督程序内专设一个作业控制程序 (Job-Controller) 以控制作业的运行。批处理作业的控制意图被描述在作业说明书中。作业控制程序在控制某一道作业运行时, 其实质性工作是解释执行作业说明书中的语句, 实现对作业的控制。从逻辑上看, 一个作业由三部分组成: 源程序 (或程序)、数据以及“加工”步骤。监督程序一旦接收到一道作业后, 根据“加工”步骤所规定的动作逐步完成对作业的加工活动。

如果用户可以使用全部的机器指令, 可以直接控制和使用系统资源 (如主存、外部设备等), 用户编程中的错误往往可能导致各种预想不到的后果。为了避免这类错误发生, 人们将机器指令分为普通指令和特权指令, 并且引入了“模式/态 (Mode)”的概念。把有关输入/输出的指令、对特殊寄存器的访问等列为特权指令, 并且规定只有监督程序才有权执行特权指令, 用户程序则只能执行普通指令。将输入/输出指令列为特权指令后, 用户便不能直接控制设备进行传输。如果用户希望进行输入/输出, 则必须向监督程序提出请求; 监督程序通过调用系统内部的程序段来完成用户的输入/输出请求。由此又引出了系统调用 (System Call) 或称广义指令的概念。

监督程序为用户提供了系列分别完成各种不同功能的系统调用程序段。用户程序中可以用一条特殊的硬件转移指令请求一次特定的系统调用。当处理机执行到用户程序的系统调用指令时, 硬件通过产生“自陷”并借助转换机制将处理机执行模式从当前的用户模式转变为监督模式, 控制也随之转入监督程序。监督程序根据用户提供的调用参数进行相应的处理, 完成设备输入/输出等功能。处理结束后, 监督程序则根据自陷前所保存的现场将模式改变为用户模式, 退回用户程序继续执行。

系统调用概念的引入大大提高了监督程序在整个系统中的地位, 丰富了监督程序的功能。监督程序不仅要负责对作业的处理流程进行自动控制, 而且还要负责为用户程序的运行提供各种功能的服务。系统调用的引入也为用户提供了使用计算机系统的新界面, 可以使用户从直接使用物理处理机的繁杂束缚中解脱出来, 呈现在用户面前的是一台功能强、使用方便的虚拟处理机。引入系统调用后, 用户对系统内部各种资源的使用均由监督程序代为完成, 因而也使得系统更加安全, 可以避免用户在使用资源时可能出现的某些错误, 也有利于提高资源利用率。

在手工操作阶段,存储器全部由用户支配使用。引入监督程序后,存储器不再由用户独占,常驻主存的监督程序必须占据部分主存。通常,监督程序占用主存的 $0 \sim k$ 单元, $k+1 \sim n$ 单元供用户程序占用。监督程序所在的存储空间称为系统空间,用户程序所在的存储空间称为用户空间。为了避免用户程序执行时有意或无意地对系统空间进行存取访问,硬件提供一个界地址寄存器,用以存放系统空间与用户空间的分界地址。当系统处于用户模式时,每访问一次主存,硬件自动进行地址越界检查,从而保证了监督程序不被破坏。这种保护称为存储保护。

在早期批处理系统中,系统动态运行时,一段时期处于监督模式,一段时期又处于用户模式。从用户模式进入监督模式主要是由于用户程序中的系统调用而引起的。比如,用户请求输入/输出或者请求结束运行。但是,若用户程序执行过程中永不出现系统调用,或者永不出现请求结束运行的系统调用(例如用户程序进入了“死循环”),系统监督程序便失去了作用。为了防止这种情况发生,人们设置了“定时器中断”。定时器(Timer)是一个硬件计数器,计时长度可以根据需要而调整。计数器根据硬件的计时周期自动计时。计数器满后便发生定时器中断。用户程序执行时若碰到定时器中断,则无条件进入监督模式。监督程序根据当前程序说明(或规定)的最大运行时间值来判断该程序是否进入了死循环,从而有效地防止了某个用户程序长期“垄断”系统处理机的现象。

引入上述概念后,早期批处理系统中的监督程序工作流程如下所述:

- (1) 判断输入设备上是否有待输入的作业,如果没有,则等待作业输入。
- (2) 从设备上输入一道作业。
- (3) 控制作业运行。

① 取出作业说明书中的一条语句,解释执行。如果是一条“作业终止”语句,则删除该作业,转第1步。

② 如果当前是一条“执行性语句”(如请求编译、请求运行用户程序等),则在主存中建立相应程序的运行环境,并分配 CPU 开始在用户态执行该程序。

③ 在用户态的程序执行过程中,如果发生中断事件(如 I/O 中断、系统调用、程序执行错误等),硬件将控制转入监督程序。中断事件处理结束后,返回用户态,用户程序继续执行。

④ 用户程序执行结束后,进入监督程序,控制转第1步,取出下一条作业说明书语句执行。

监督程序如同一个系统操作员,它负责批作业的输入/输出,并自动根据作业控制说明书以单道串行的方式控制作业运行,同时在程序运行过程中通过提供各种系统调用,控制使用计算机资源。虽然监督程序并不能被称为操作系统(它与操作系统的本质差别在于监督程序不具有并发控制机制),但它与操作系统有许多相似的特征。监督程序在系统中的地位和作用、追求实现的基本目标以及管理资源的基本方法与操作系统是类似的。真正的操作系统就是在此基础上进一步发展和完善的。

与手工操作阶段相比,监督程序的引入有效地减少了人工干预时间,减少了作业运行前的准备时间,相对提高了 CPU 的利用率。但是在计算机速度大幅度提高的形势下,用这种方法管理计算机远不能适应需要。首先,在一个 CPU 上运行的程序启动输入/输出操作时 CPU 被迫处于空闲状态或忙等待(busy_wait)状态,也就是说, CPU 启动输入/输出操作后再循环判断输入/输出是否完成,而没有做实质性工作,这将导致高速的 CPU 受到慢速设备的牵制,从而使 CPU 无法被充分利用。

利用脱机输入/输出改善系统性能

由于作业的输入/输出与作业的运行是串行的,所以受卡片机、光电机、打印机这类慢速输入

/输出设备的影响, CPU 的利用率难以提高。为了进一步提高系统的工作效率, 必须解决低速输入、输出的问题。磁带机的传输速度比卡片机、光电机和打印机的速度快, 若用磁带机来代替这类低速设备便可进一步缩小 CPU 与外设间速度上的差异。历史上人们曾采用脱机输入/输出技术实现作业输入/输出, 如图 1-4 所示。

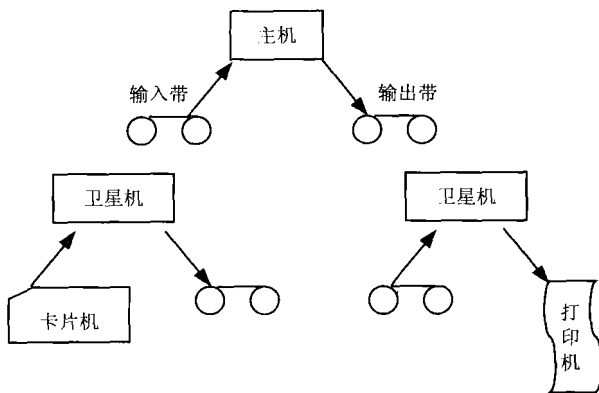


图 1-4 脱机输入/输出系统模型

在采用脱机输入/输出技术的系统中, 主机的所有输入/输出操作都是通过磁带机进行的。用户的作业由另外的一台能力较弱、价格较低的卫星机负责从卡片机传输到磁带上(称为输入带), 然后操作员将输入带安装到与主机相连的磁带机上。主机在处理输入带上的作业时, 将产生的输出结果直接送到输出带上。操作员再将输出带安装到卫星机上, 由卫星机负责将输出带上的信息从打印机上输出。由于磁带机比慢速输入/输出设备(如卡片机、打印机)的速度快, 因而按照这种脱机方式控制作业的输入/输出, 可以减少作业输入/输出所花费的时间, 有效地提高 CPU 的利用率。如果将一台主机与多台卫星机有机地组合, 使速度得到最好的匹配, 则可以大幅度地提高系统的处理能力。在 20 世纪 50 年代末到 60 年代初, 这种脱机处理方式被广泛地应用于批处理系统中。

无论如何, 由于 CPU 与 I/O 设备是以串行方式工作的, 也就是说, CPU 工作时 I/O 设备空闲或者 I/O 设备工作时 CPU 在忙等待, 这就限制了设备的利用率; 其次, 从方便用户的角度, 采用这种批量处理的控制方法后, 用户不能以交互的方式使用计算机, 从而限制了对计算机的灵活使用。随着对这些问题不断深入的研究和解决, 逐步形成了第二代操作系统。

1.2.2 专用操作系统

20 世纪 60 年代初计算机硬件有了很大的发展。例如, 主要元件由电子管变成了晶体管, 出现了磁盘、通道、终端等部件。而这些硬件的发展为监督程序提出了新的研究课题, 也为操作系统的形成提供了重要的物质基础。这个时期是操作系统形成的重要时期。随着计算机应用发展的巨大牵引, 不仅批处理系统得到充分的发展, 而且还出现了实时(real time)、分时(sharing time)等不同类型的系统。

1. 多道批处理系统

在早期批处理系统(也称单道批处理系统)中, CPU 与 I/O 设备以串行方式工作, 故两者的利用率较低。为了提高资源利用率, 人们开始使用输入/输出缓冲、SPOOLing 等技术, 尤其是引入了“多道程序设计”(Multiprogramming)的思想, 使单道批处理系统发展为多道批处理系统。

(1) 利用输入/输出缓冲异步编程

在单道批处理系统中作业的处理过程是单道串行的，在监督程序的控制下，CPU 与外设按串行方式工作。为了改变这种串行工作方式，人们首先采用了缓冲（buffering）技术使两者在一定程度上并行操作。例如，在主存中建立两个长度相同的缓冲区：B0、B1。对于一批待输入的信息，首先将其中的一个记录从设备上读入 B0，读入完后接着将下一个记录从设备上读入 B1，与此同时，CPU 开始处理 B0 中的记录。待 CPU 处理工作与输入工作均结束后，则又将下一个记录读入 B0，CPU 同时处理 B1 中的记录。如此重复，直至将信息全部输入。这种利用双缓冲区实现的 I/O 操作在一定程度上实现了 CPU 与外设并行工作。这类并行的实现要求 I/O 设备有较强的功能，能不依赖于 CPU 实现外设与主存独立交换数据。

(2) SPOOLing 技术

通道技术的引入，使得 CPU 与外设并行操作成为可能。通道是指专门用来控制输入/输出的硬件装置，它可以实现外设与主存直接交换数据，在相当长的时间里不用“打扰”CPU，因此这时 CPU 可以去干别的事情。为了能够消除脱机输入/输出带来的人工干预的麻烦，又要保持脱机输入/输出系统中作业高速出入主存的特点，人们借助通道和磁盘成功地实现了著名的 SPOOLing 系统。通道也可看成是专门的 I/O 处理机。磁盘则是一种比磁带更快且能够随机存取的外部存储设备。

所谓 SPOOLing（Simultaneous Peripheral Operation On Line）的含义是：并发的外部设备联机操作。利用 SPOOLing 技术控制批处理系统中作业输入/输出（见图 1-5）的基本思想是：利用磁盘（或一组磁盘）设备作为主机的直接输入/输出设备，即系统直接从磁盘上选取作业运行，作业的运行结果也直接存入磁盘；相应的通道（在设备驱动程序控制下）则负责并行地将卡片机上的用户作业输入到磁盘或者将磁盘上作业的运行结果从打印机上输出。

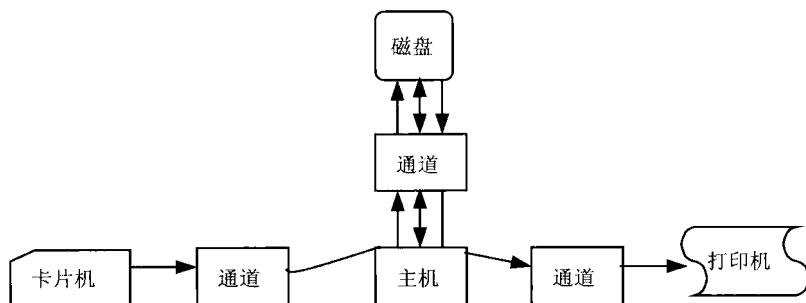


图 1-5 SPOOLing 系统

通道直接受主机控制，主机与通道之间借助中断机制相互通信。例如，只要卡片机上有用户作业，操作系统驱动程序便启动设备通道，通道被启动后便将作业信息输入到主机主存缓冲区，然后操作系统启动磁盘通道将缓冲区作业信息放入磁盘，在作业输入期间主机可以并行地从事其他工作；类似地，只要磁盘中存在等待输出的信息且打印机空闲，则操作系统通过启动通道将信息从打印机上输出。所以，SPOOLing 技术又被称为“伪脱机输入/输出”技术，被广泛地用于后来的批处理系统中。采用 SPOOLing 技术实现输入/输出的系统通常又简称为 SPOOLing 系统。SPOOLing 技术主要是通过加快作业的输入/输出提高系统性能。

(3) 多道程序设计技术

如前所示，采用 SPOOLing 技术，利用主机和通道间的并行性，可以使作业的输入/输出与主