

◆高等教育应用型本科人才培养系列教材

操作系统

CAOZUO XITONG

主 编 孙剑明

副主编 于舒春 刘 爽



哈尔滨工程大学出版社
Harbin Engineering University Press

本科人才培养系列教材

操作系统

主 编 孙剑明

副主编 于舒春 刘 爽

内 容 简 介

本书以 UNIX、Linux、Windows、Andriod 等操作系统为例,从操作系统的概念、类型、特征、功能、发展历史等方面谈起,主要介绍了计算机系统结构和操作系统接口、处理器管理、存储管理、文件管理、设备管理、进程同步与进程通信、死锁、计算机操作系统实验。

本书可以作为计算机专业的自学考试用书,也可供相关技术人员参考。

图书在版编目(CIP)数据

操作系统/孙剑明主编. —哈尔滨:哈尔滨工程大学出版社,2018. 7

ISBN 978 - 7 - 5661 - 2005 - 2

I. ①操… II. ①孙… 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(2018)第 150758 号

选题策划 夏飞洋

责任编辑 夏飞洋

封面设计 刘长友

出版发行 哈尔滨工程大学出版社
社 址 哈尔滨市南岗区南通大街 145 号
邮政编码 150001
发行电话 0451 - 82519328
传 真 0451 - 82519699
经 销 新华书店
印 刷 哈尔滨市石桥印务有限公司
开 本 787 mm × 1 092 mm 1/16
印 张 17.25
字 数 431 千字
版 次 2018 年 7 月第 1 版
印 次 2018 年 7 月第 1 次印刷
定 价 48.00 元

<http://www.hrbeupress.com>

E-mail: heupress@hrbeu.edu.cn

前 言

21 世纪对于职业技术应用型人才的需求在不断扩大,高职高专教育的发展是以应用型与专业理论型教育并存、共同发展为特征的教育模式。本科的教学通常偏重理论教育,学生实践能力普遍偏弱,与生产实践脱离较远,而专科又是本科的浓缩。因此,解决现阶段出现的教育现状与社会需求严重脱节问题的最好办法就是大力发展应用型本科、专科及职业教育。基于这种发展理念和顺应时代需求,我们编写了这本立足应用型人才培养的教材。

全书共分为 9 章,系统地反映了操作系统的发展现状和前沿动态。

第 1 章为绪论,介绍了操作系统的概念、历史、类型、特征和功能,进而分别对 UNIX 系统、Android 系统进行了阐述。第 2 章为计算机系统结构和操作系统接口,从计算机的系统结构和硬件组成开始,逐步阐述了计算机软硬件逻辑、操作系统的核心内容、操作系统的典型结构、操作系统接口,并以 UNIX 系统为例讲解了系统调用。第 3 章为处理器管理,阐述了进程的概念、进程的特征、进程的控制、UNIX 进程管理、处理器调度及相关算法。第 4 章为存储器管理,阐述了存储器的功能、连续分配方式、覆盖与对换管理、基本分页存储管理、段页式存储管理、虚拟存储器、请求分页存储管理、请求分段存储管理,进而以 UNIX 系统和 Linux 系统为例阐述了如何具体实施存储管理。第 5 章为文件管理,从文件和文件系统的概念谈起,对文件、目录、文件系统的实现、文件系统的管理和优化进行了详细论述,进而以 MS-DOS 系统和 UNIX 系统为例,讲解了文件系统的使用。第 6 章为设备管理,从设备管理的概念出发,对 Windows 的设备管理、DOS 的设备管理、磁盘的分区和格式化、硬盘容量限制及格式化参数设置进行了详细的论述。第 7 章为进程同步和进程通信,从进程同步的概念谈起,分析了经典的进程同步问题,进而类比地阐述了进程通信和线程通信的问题。第 8 章为死锁,从资源的分类和获取开始,阐述了死锁的条件、死锁检测与恢复、死锁避免及其他问题。第 9 章为实验,以 Linux 系统的使用、进程管理与通信、Shell 程序设计语言为内容,以实例的形式设置实验体会环境,更好地完成对操作系统相关知识的理解和运用。

在本书的编写过程中,得到了哈尔滨工程大学出版社编辑夏飞洋的大力支持与帮助,在此表示衷心的感谢。

我们对本书稿进行了反复的修改,就是希望能把它写得更好,但限于编者的水平,书中仍难免会有错误和不当之处,恳请读者批评指正。

编 者
2018 年 6 月

目 录

第 1 章 绪论	1
1.1 什么是操作系统	1
1.2 计算机和操作系统的协同发展	4
1.3 操作系统的基本类型	10
1.4 操作系统的特征	16
1.5 操作系统的功能	20
1.6 Linux 操作系统	23
1.7 Android 操作系统	24
第 2 章 计算机系统结构和操作系统接口	29
2.1 计算机系统结构	29
2.2 计算机硬件组成	32
2.3 计算机软硬件工作逻辑	41
2.4 操作系统的核心内容	52
2.5 操作系统的典型结构	58
2.6 操作系统接口	61
2.7 UNIX 系统接口	73
第 3 章 处理器管理	83
3.1 进程的概念	83
3.2 进程的描述	86
3.3 进程的控制	90
3.4 UNIX 进程管理	91
3.5 处理器调度	95
3.6 调度算法	97
3.7 UNIX 调度	104
第 4 章 存储器管理	106
4.1 存储器管理概述	106
4.2 连续分配方式	110
4.3 覆盖与对换管理	114
4.4 基本分页存储管理	115
4.5 分页式存储管理	118
4.6 分段式存储管理	119

4.7 虚拟存储器	120
4.8 请求分页存储管理	121
4.9 UNIX 存储管理	126
第5章 文件管理	130
5.1 文件系统概述	130
5.2 文件	131
5.3 目录	136
5.4 文件系统的实现	139
5.5 文件系统的管理和优化	149
5.6 文件系统实例	161
第6章 设备管理	165
6.1 设备管理的基本概念	165
6.2 Windows 的设备管理	172
6.3 DOS 的设备管理	174
第7章 进程同步与进程通信	177
7.1 进程同步	177
7.2 经典进程的同步问题	186
7.3 进程通信	192
7.4 线程	198
第8章 死锁	208
8.1 资源	208
8.2 死锁简介	211
8.3 死锁检测与恢复	214
8.4 死锁避免	219
8.5 死锁预防	223
8.6 其他问题	225
第9章 计算机操作系统实验	229
9.1 Linux 操作系统基本命令	229
9.2 用户界面与 Shell 命令	232
9.3 进程管理及进程通信	238
9.4 Shell 程序设计语言	246
9.5 Linux 编程系统调用	250
参考文献	257
操作系统自学考试大纲	258
Ⅰ 课程性质与设置目的	258
Ⅱ 课程内容与考核目标	259
Ⅲ 有关说明与实施要求	266
Ⅳ 题型举例	267

第 1 章 绪 论

1.1 什么是操作系统

一个完整的计算机系统是由硬件系统和软件系统两大部分组成的。操作系统 (Operating System, OS) 是所有软件中最基础、最核心的部分, 它为用户执行程序提供更加方便和有效的环境。从资源管理的角度来看, 操作系统对整个计算机系统内的所有资源进行管理和调度, 优化资源的利用, 协调系统内的各种活动, 处理可能出现的各种问题。

1.1.1 计算机系统组成

计算机系统包括硬件系统和软件系统。硬件系统是组成计算机系统的各种物理设备的总称, 是看得见、摸得着的实体部分。软件系统是为了运行、管理和维护计算机而编制的各种程序、数据和文档的总称。它们的区别犹如把一个人分成躯体和思想一样, 躯体是硬件, 思想则是软件。硬件系统和软件系统密不可分, 它们的有机结合才构成一个完整的计算机系统。

1. 计算机硬件系统

虽然现在的计算机系统从性能指标、运算速度、工作方式、应用领域和价格等方面与当时相比发生了很大变化, 但是基本上仍沿用冯·诺依曼体系结构, 采用存储程序和顺序执行的工作原理。冯·诺依曼体系结构的计算机主要由运算器、控制器、存储器、输入设备和输出设备组成, 其中运算器和控制器集成在一片大规模或超大规模集成电路上, 称为中央处理器 (Central Processing Unit, CPU)。这五大功能部件相互配合, 协同工作。

(1) 处理器。

CPU 是一台计算机的运算核心和控制核心, 它从内存中提取指令并执行它们。每种型号 CPU 的指令集都是专用的。CPU 内部都包含若干寄存器, 其中一类是通用寄存器, 用来存放关键变量和中间结果; 另一类是专用寄存器, 用来存放一些特殊的数据, 比如程序计数器 (Program Counter, PC) 和程序状态字 (Program Status Word, PSW)。程序计数器中保存下面要提取指令的内存地址。程序状态字用来存放两类信息: 一类是体现当前指令执行结果的各种状态信息, 称为状态标志, 如有无进位 (CF 位)、有无溢出 (OF 位)、结果正负 (SF 位)、结果是否为零 (ZF 位)、奇偶标志位 (PF 位) 等; 另一类是存放控制信息, 称为控制状态, 如允许中断 (IF 位)、跟踪标志 (TF 位)、方向标志 (DF 位) 等。

(2) 存储器。

存储器是计算机系统的主要部件之一。按照存取速度、存储容量和成本将存储器划分为一个典型的层次结构, 如图 1-1 所示。

第一层次	CPU内部寄存器
第二层次	高速缓存
第三层次	内存
第四层次	外存

图 1-1 计算机系统中存储器的层次结构

第一层次是 CPU 内部寄存器,其速度与 CPU 一样快,所以存取它们没有延迟,但是它们的成本高、容量小,通常都小于 1 KB。

第二层次是高速缓存(cache),它们大多由硬件控制。高速缓存的速度很快,它们放在 CPU 内部或者非常靠近 CPU 的地方。当程序需要读取具体信息时,高速缓存硬件首先查看信息是否在 cache 中,如果在其中,就直接使用;如果不在,就从内存中查找并获取该信息,并把信息放入 cache 中,以备以后再次使用。但 cache 成本很高,容量较小,一般小于 2 MB。

第三层次是内存,它是存储系统的主要部件,也称为随机存取存储器。CPU 可以直接存取内存及寄存器和 cache 中的信息,但是不能直接存取磁盘等外存上的数据。因此,机器执行的指令及所用的数据必须预先存放在内存及 cache 和寄存器中。但是,内存中存放的信息是易变的,当机器电源被关闭后,内存中的信息就全部丢失了。

第四层次是外存,包括磁盘、光盘和移动存储设备。外存记录的数据可以持久保存,而且根据需要可以随时更换。外存容量很大,现在常用的磁盘容量为 300 GB ~ 2 TB,价格低廉。外存中数据的存取速度低于内存。

(3) I/O 设备。

输入/输出设备也称为 I/O 设备,它是人机交互的工具,通常由控制器和设备本身两部分组成。

设备控制器是 I/O 设备的电子部分,它协调和控制一台或多台 I/O 设备的操作,实现设备操作与整个系统操作的同步。在小型机和微型机上,往往以印刷电路卡的形式插入计算机中。很多控制器可以管理 2 台、4 台甚至 8 台同样的设备。设备控制器本身有一些缓冲区和一组专用的寄存器,负责在外部设备和本地缓冲区之间移动数据。

设备本身的对外接口较简单,实际上它们隐藏在控制器的后面。因而,操作系统常常是和设备控制器打交道,而不是与设备直接作用。设备的种类很多,因此设备控制器的类别也很多,需要不同的软件来控制它们。这些向设备控制器发送命令并接收其回答信息的软件称为设备驱动程序。不同操作系统上的不同控制器对应不同的设备驱动程序。计算机系统常常把设备驱动程序以核心态的方式来运行。

(4) 总线。

在计算机系统中,为了简化硬件电路设计、简化系统结构,常用一组线路,配置以适当的接口电路与各部件和外围设备连接,这组共用的连接电路称为总线。按照总线上传输信息的种类,可将总线划分为数据总线、地址总线和控制总线三部分。

数据总线用于 CPU 与内存或 I/O 设备之间的数据传递,它的宽度取决于 CPU 的字长。数据总线是双向总线,两个方向都能传送数据。

地址总线用于传送存储单元或 I/O 接口的地址信息,信息传送是单向的。它的位数决

定了计算机内存空间的范围大小,即 CPU 能够管辖的内存数量控制总线用于传送控制器的各种控制信息,它的位数由 CPU 的字长决定。

在计算机系统中有多个设备要向总线发信号时,在传送数据之前,先要监听总线是否有空闲,空闲时才能占用总线,使用之后要释放总线。

2. 计算机软件系统

在计算机软件系统中,软件通常分为系统软件和应用软件两大类,但是这两类软件的界限并不十分明显。系统软件是指控制计算机的运行、管理计算机的各种资源并为应用软件提供支持和服务的一类软件。系统软件通常包括操作系统、语言处理程序和各种实用程序。应用软件是指利用计算机的软、硬件资源为某一专门的应用目的而开发的软件,常见的应用软件有办公软件、图形图像处理软件等。

1.1.2 操作系统的地位

1. 操作系统的地位

操作系统在计算机系统的位置如图 1-2 所示。

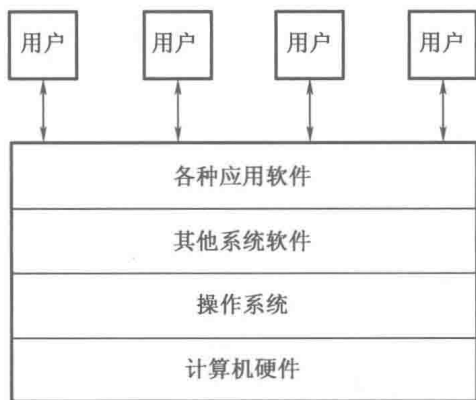


图 1-2 综合软、硬件的计算机系统层次结构

操作系统是一种系统软件,它是配置在计算机硬件之上的第一层软件。它在计算机系统中占据特别重要的地位,是整个计算机系统的控制管理中心。其他系统软件如汇编程序、编译程序等,以及各种应用软件都将依赖于操作系统的支持,取得它的服务。操作系统对它们既具有支配权力,又为其运行构造必备的环境。

没有任何软件支持的计算机称为裸机。操作系统是裸机之上的第一层软件,它只在核心态模式下运行,受硬件保护,与硬件关系尤为密切。它不仅对硬件资源直接实施控制、管理,而且其他很多功能的完成也是与硬件动作配合起来实现的。

通过图 1-2 可以看出,在裸机上每加一层软件后,计算机系统的功能变得更加强大,用户使用起来更加方便。通常把经过软件扩充后的计算机称为虚拟机。

2. 操作系统的作用

操作系统在计算机系统中起三个方面的作用。

(1) 操作系统作为用户接口和公共服务程序。

操作系统作为用户接口是指 OS 处于用户与计算机硬件系统之间,用户通过 OS 来使用计算机系统。或者说,用户在 OS 帮助下,能够方便、快捷、安全、可靠地操纵计算机硬件并

运行自己的程序。从内部看,操作系统对计算机硬件进行了改造和扩充,为应用程序提供强有力的支持;从外部看,操作系统提供友好的人机接口,使得用户能够方便、可靠、安全和高效地使用硬件和运行应用程序。同时,用户可以通过“系统调用”使用操作系统提供的各种公共服务,无须了解软硬件本身的细节。所以操作系统可以看作是友善的用户接口和各种公共服务的提供者。

(2)操作系统作为资源的管理者和控制者。

在计算机系统中,能够分配给用户使用的各种软硬件设施总称为资源。资源包括两大类,即硬件资源和软件资源。其中硬件资源有处理器、存储器、外部设备等;软件资源则包括程序和数据等。为了使应用程序正常运转,操作系统必须为其分配足够的资源;为了使系统效率提高,操作系统必须支持多道程序设计,合理调度和分配各种资源,充分发挥并行部件的性能,使它最大限度地重叠操作和保持忙碌。

作为资源的管理者,操作系统要对资源进行研究,找出各种资源的共性和个性,有序地管理计算机中的软硬件资源,记录资源的使用情况,确定资源分配策略,实施资源的分配和回收,满足用户对资源的需求,提供机制来协调应用程序对资源的使用冲突,研究资源利用的统一方法,为用户提供简单、有效的资源使用手段,在满足应用程序需求的前提下,最大限度地实现各种资源的共享,提高资源利用率,从而提高计算机系统的效率。

(3)操作系统实现了计算机资源的抽象。

资源抽象是指通过创建软件来屏蔽硬件资源的物理特性和接口细节,简化对硬件资源的操作、控制和使用,即不考虑物理细节而对资源执行操作。资源抽象用于处理系统的复杂性,重点解决资源的易用性。资源抽象软件对内封装实现细节,对外提供应用接口,这意味着用户不必了解更多的硬件知识,只需通过软件接口即可使用和操作物理资源。例如,为了方便用户使用I/O设备,在裸机上覆盖一层I/O设备管理软件,由该软件实现对设备操作的细节,并向上提供一组操作命令。用户可以利用操作命令进行数据的输入/输出,而无须关心I/O设备是如何实现的。这里,I/O管理软件实现了对设备的抽象。

1.1.3 操作系统的概念

根据操作系统的地位和作用,可以从以下方面理解它的定义。

(1)操作系统是系统软件,由一整套程序组成。

(2)它的基本职能是控制和管理计算机系统内的各种资源,合理地组织工作流程。

(3)它提供众多服务,方便用户使用,扩充硬件功能。

通常可以这样定义操作系统:操作系统是控制和管理计算机系统内的各种硬件和软件资源,合理地组织计算机工作流程,并为用户提供使用计算机提供方便的一种系统软件。

1.2 计算机和操作系统的协同发展

操作系统已经存在许多年了。在本小节中,我们将简要分析一些操作系统历史上的重要之处。操作系统与其所运行的计算机体系结构的联系非常密切。我们将分析连续几代的计算机,看看它们的操作系统是什么样的。把操作系统的分代映射到计算机的分代上有些粗糙,但是这样做确实有某些作用,否则还没有其他好办法能够说清楚操作系统的历史。

下面给出的有关操作系统的发展主要是按照时间线索叙述的,且在时间上是有重叠的。每个发展并不是等到先前一种发展完成后才开始。存在着大量的重叠,也存在不少虚假的开始和终结时间。请读者把这里的文字叙述看成是一种指引,而不是盖棺定论。

第一台真正的数字计算机是英国数学家 Charles Babbage (1792—1871) 设计的。尽管 Babbage 花费了几乎一生的时间和财产,试图建造他的“分析机”,但是始终未能让机器正常运转,因为它是台纯机械的数字计算机,他所在时代的技术不能生产出他所需要的高精度轮子、齿轮和轮牙。毫无疑问,这台“分析机”没有操作系统。

有一段有趣的历史花絮,Babbage 认识到他的“分析机”需要软件,所以他雇用了个名为 Ada Lovelace 的年轻妇女,作为世界上第一个程序员,而她是著名的英国诗人 Lord Byron 的女儿。程序设计语言 Ada 是以她的名字命名的。

1.2.1 第一代计算机和操作系统

Babbage 失败之后一直到第一次世界大战,数字计算机的建造几乎没有什么进展,第二次世界大战刺激了有关计算机研究的爆炸性开展。IOWA 州立大学的 John Atanasoff 教授和他的学生 Clifford Berry 建造了据认为是第一台可工作的数字计算机。该机器使用了 300 个真空管。大约在同一时期,Konrad Zuse 在柏林用继电器构建了 Z3 计算机,英格兰布莱切利园的一个小组在 1944 年构建了 Colossus, Howard Aiken 在哈佛大学建造了 Mark I,宾夕法尼亚大学的 William Mauchley 和他的学生 J. Presper Eckert 建造了 ENIAC。这些机器有的是二进制的,有的使用真空管,有的是可编程的,但是都非常原始的,甚至需要花费数秒时间才能完成最简单的运算。

在那个年代,同一个小组的人(通常是工程师们)设计、建造、编程、操作并维护一台机器。所有的程序设计是用纯粹的机器语言编写的,甚至更糟糕,需要通过将上千根电缆接到插件板上连接成电路,以控制机器的基本功能。没有程序设计语言(甚至汇编语言也没有),操作系统则从来没有听说过。使用机器的一般方式是,程序员在墙上的机时表上预约一段时间,然后到机房中将他的插件板接到计算机里,在接下来的几小时里,期盼正在运行中的两万多个真空管不会烧坏。那时,所有的计算问题实际都只是简单的数字运算,如制作正弦、余弦以及对数表等。

到了 20 世纪 50 年代早期,这种情况有了改进,出现了穿孔卡片,这时就可以将程序写在卡片上,然后读入计算机而不用插件板,但其他过程则依然如旧。

1.2.2 第二代计算机和操作系统

20 世纪 50 年代,晶体管的发明极大地改变了整个状况。计算机已经很可靠,厂商可以成批地生产并销售计算机给用户,用户可以通过计算机长时间运行,完成一些有用的工作。此时,设计人员、生产人员、操作人员、程序人员和维护人员之间第一次有了明确的分工。

这些机器,现在被称作大型机(Mainframe),锁在有专用空调的房间中,由专业操作人员运行。只有少数大公司、重要的政府部门或大学才能承受数百万美元的标价。要运行一个作业(JOB,即一个或一组程序),程序员首先将程序写在纸上(用 FORTRAN 语言或汇编语言),然后穿孔成卡片,再将卡片盒带到输入室,交给操作员,接着就喝咖啡等待输出完成。

计算机运行完当前的任务后,其计算结果从打印机上输出,操作员到打印机上撕下运算结果并送到输出室,程序员稍后就可取到结果。然后,操作员从已送到输入室的卡片盒

中读入另一个任务。如果需要 FORTRAN 编译器,操作员还要从文件柜把它取来读入计算机。当操作员在机房里走来走去时,许多机时被浪费掉了。

由于当时的计算机非常昂贵,人们很自然地要想办法减少机时的浪费。通常采用的解决方法就是批处理系统(Batch Processing System)。其思想是:在输入室收集全部的作业,然后用一台相对便宜的计算机,如 IBM 1401 计算机,将它们读到磁带上。IBM 1401 计算机适用于读卡片、复制磁带和输出打印,但不适用于数值运算。另外用较昂贵的计算机,如 IBM 7094 来完成真正的计算。

在收集了大约一小时的批量作业之后,这些卡片被读进磁带,然后磁带被送到机房里并装到磁带机上。随后,操作员装入一个特殊的程序(现代操作系统的前身),它从磁带上读入第一个作业并运行,将输出写到第二盘磁带上,而不打印。每个作业结束后,操作系统自动地从磁带上读入下一个作业并运行。当一批作业完全结束后,操作员取下输入和输出磁带,将输入磁带换成小一批作业,并把输出磁带拿到一台 1401 机器上进行脱机(不与主计算机联机)打印。

典型的输入作业结构如图 1-3 所示。一开始是一张 MYMJOB 卡片,它标识出所需的最大运行时间(以分钟为单位)、计费号以及程序员的名字。接着是 MYMFORTRAN 卡片,通知操作系统从系统磁带上装入 FORTRAN 语言编译器。之后就是待编译的源程序,然后是 LOAD 卡片,通知操作系统装入编译好的目标程序。接着是 MYMRUN 卡片,告诉操作系统运行该程序并使用随后的数据。最后,MYMEND 卡片标识作业结束。这些基本的控制卡片是现代 Shell 和命令解释器的先驱。

第二代大型计算机主要用于科学与工程计算,例如,解偏微分方程。这些题目大多用 FORTRAN 语言和汇编语言编写。典型的操作系统是 FMS (Fortran Monitor System, FORTRAN 监控系统)和 IBSYS(IBM 为 7094 机配备的操作系统)。

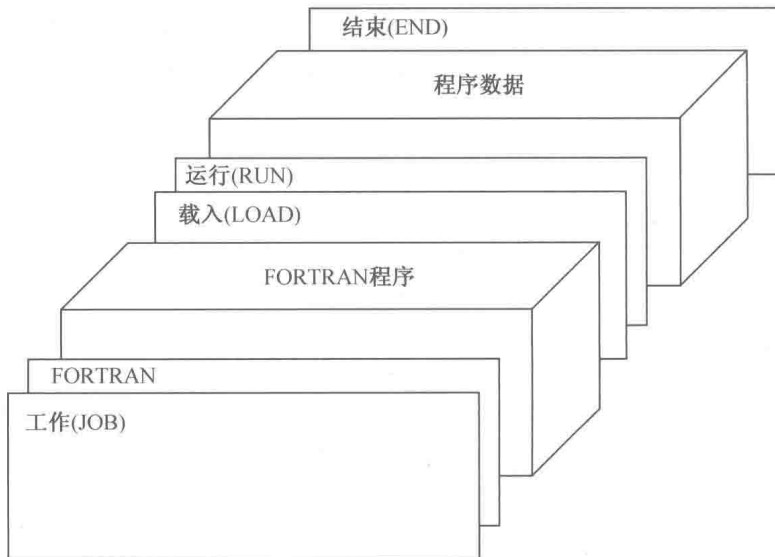


图 1-3 典型的输入作业结构

1.2.3 第三代计算机和操作系统

20 世纪 60 年代初期,大多数计算机厂商都有两条不同并且完全不兼容的生产线。一

条是面向字的、大型的科学用计算机,诸如 IBM7094,主要用于科学和工程计算。另一条是面向字符的、商用计算机,诸如 IBM1401,银行和保险公司主要用它进行磁带归档和打印服务。

开发和维护两种完全不同的产品,对厂商来说是昂贵的。另外,许多新的计算机用户开始时只需要一台小计算机,后来可能又需要一台较大的计算机,而且希望能够更快地执行原有的程序。

IBM 公司试图通过引入 System360 来一次性地解决这两个问题。360 是一个软件兼容的计算机系列,其低档机与 1401 相当,高档机则比 7094 功能强很多。这些计算机只在价格和性能(最大存储器容量、处理器速度、允许的 I/O 设备数量等)上有差异。由于所有的计算机都有相同的体系结构和指令集,因此,在理论上,为一种型号机器编写的程序可以在其他所有型号的机器上运行。而且 360 被设计成既可用于科学计算,又可用于商业计算。这样,一个系列的计算机便可以满足所有用户的要求。在随后的几年里,IBM 使用更现代的技术陆续推出了 360 的后续机型,如著名的 370,4300,3080 和 3090 系列。zSeries 是这个系列的最新机型,不过它与早期的机型相比变化非常之大。

360 是第一个采用(小规模)芯片(集成电路)的主流机型,与采用分立晶体管制造的第二代计算机相比,其性能/价格比有很大提高。360 很快就获得了成功,其他主要厂商也很快采纳了系列兼容机的思想。这些计算机的后代仍在大型的计算中心里使用。现在,这些计算机的后代经常用来管理大型数据库(如航班订票系统)或作为 web 站点的服务器,这些服务器每秒必须处理数千次的请求。

“单一家族”思想的最大优点同时也是其最大的缺点。原因在于所有的软件,包括操作系统 OS/360,要能够在所有机器上运行。从小的代替 1401 把卡片复制到磁带上的机器,到用于代替 7094 进行气象预报及其他繁重计算的大型机;从只能带很少外部设备的机器到有很多外设的机器;从商业领域到科学计算领域等。总之,它要有效地适用于这些不同的用途。

第三代计算机的另一个特性是,卡片被拿到机房后能够很快地将作业从卡片读入磁盘。于是,任何时刻当一个作业运行结束时,操作系统就能将一个新作业从磁盘读出,装进空出来的内存区域运行。这种技术称为同时的外部设备联机操作(Simultaneous Peripheral Operation on Line, Spooling),该技术同时也用于输出。当采用了 Spooling 技术后,就不再需要 IBM1401 机,也不必再将磁带搬来搬去了。

第三代操作系统很适于大型科学计算和繁忙的商务数据处理,但其实质上仍旧是批处理系统。许多程序员很怀念第一代计算机的使用方式。那时,他们可以几小时独占一台机器,可以即时地调试他们的程序。而对第一代计算机而言,从一个作业提交到运算结果取回往往长达数小时,更有甚者,一个逗号的误用就会导致编译失败,而可能浪费了程序员半天时间。

程序员们的希望很快得到了响应,这种需求导致了分时系统(timesharing)的出现。它实际上是多道程序的一个变体,每个用户都有一个联机终端。在分时系统中,假设有 20 个用户登录,其中 17 个在思考问题、与人聊天或喝咖啡,则 CPU 可分配给其他三个需要的作业轮流执行。由于调试程序的用户常常只发出简短的命令(如编译一个 5 页的源文件),而很少有长的费时命令(如上百万条记录的文件排序),所以计算机能够为许多用户提供快速的交互式服务,同时在 CPU 空闲时还可能在后台运行一个大作业。第一个通用的分时系

统——兼容分时系统(Compatible Time Sharing System, CTSS)是 MIT(麻省理工学院)在一台改装过的 7094 机上开发成功的(Corbas 等人, 1962 年)。但直到第三代计算机广泛采用了必需的保护硬件之后, 分时系统才逐渐流行。

另一个第三代计算机的主要进展是小型机的崛起, 以 1961 年 DEC 的 PDP-1 作为起点。PDP-1 计算机只有 4 000 个 18 位的内存, 每台售价 120 000 美元(不到 IBM 7094 的 5%), 该机型非常热销。对于某些非数值的计算, 它和 7094 几乎一样快。PDP-1 开辟了一个全新的产业。很快有了一系列 PDP 机型(与 IBM 系列机不同, 它们互不兼容), 其顶峰机型为 PDP-11。

一位曾参加过 MULTICS 研制的贝尔实验室计算机科学家 Ken Thompson, 后来找到一台无人使用的 PDP7 机器, 并开始开发一个简化的、单用户版 MULTICS。他的工作后来决定了 UNIX 操作系统的诞生。

对 UNIX 版本免费产品(不同于教育目的)的愿望, 促使芬兰学生 Linus Torvalds 编写了 Linux。这个系统直接受到在 MINIX 开发的启示, 而且原本支持各种 MINIX 的功能(例如 MINIX 文件系统)。尽管它已经通过多种方式扩展, 但是该系统仍然保留了某些与 MINIX 和 UNIX 共同的低层结构。对 Linux 和开放源码运动具体历史感兴趣的读者可以阅读 Glyn Moody 的书籍(2001)。

1.2.4 第四代计算机和操作系统

随着 LSI(大规模集成电路)的发展, 在每平方厘米的硅片芯片上可以集成数千个晶体管, 个人计算机时代到来了。从体系结构上看, 个人计算机(最早称为微型计算机)与 PDP-11 并无二致, 但就价格而言却相去甚远。以往, 公司的一个部门或大学里的一个院系才配备一台小型机, 而微处理器却使每个人都能拥有自己的计算机。

1974 年, 当 Intel 8080——第一代通用 8 位 CPU 出现时, Intel 希望有一个用于 8080 的操作系统, 部分是为了测试。Intel 请求其顾问 Gary Kildall 编写。Kildall 和一位朋友首先为新推出的 Shugart Associates 8 英寸软盘构造了一个控制器, 并把这个软磁盘同 8080 相连, 从而制造了第一个配有磁盘的微型计算机。然后 Kildall 为它写了一个基于磁盘的操作系统, 称为 CP/M(Control Program for Microcomputer)。由于 Intel 不认为基于磁盘的微型计算机有什么前景, 所以当 Kildall 要求 CP/M 的版权时, Intel 同意了他的要求。于是 Kildall 组建了一家公司(Digital Research), 进一步开发和销售 CP/M。

1977 年, Digital research 重写了 CP/M, 使其可以在使用 8080、Zilog Z80 以及其他 CPU 芯片的多种微型计算机上运行。

在 20 世纪 80 年代的早期, IBM 设计了 IBM PC 并寻找可在上面运行的软件。

另一个微软操作系统是 Windows NT(NT 表示新技术), 它在一定的范围内同 Windows 95 兼容, 但是内部是完全新编写的。它是一个 32 位系统。Windows NT 的首席设计师是 David Cutler, 他也是 VAX VMS 操作系统的设计师之一, 所以有些 VMS 的概念用在了 NT 上。事实上, NT 中有太多的来自 VMS 的思想, 所以 VMS 的所有者 DEC 公司控告了微软公司。法院对该案件判决的结果引出了一大笔需要用多位数字表达的金钱。微软公司期待 NT 的第一个版本可以消灭 MS-DOS 和其他的 Windows 版本, 因为 NT 是个巨大的超级系统, 但是这个想法失败了。只有 Windows NT 4.0 踏上了成功之路, 特别在企业网络方面取得了成功。1999 年初, Windows NT 5.0 改名为 Windows 2000。微软期望它成为 Windows 98 和

Windows NT 4.0 的接替者。

不过这两个方面都不太成功,于是微软公司发布了 Windows 98 的另一个版本,名为 Windows Me(千年版)。2001 年,发布了 Windows 2000 的一个稍加升级的版本,称为 Windows XP。这个版本的寿命比较长(6 年),基本上替代了 Windows 所有原先版本。在 2007 年 1 月,微软公司发布了 Windows XP 的后继版,名为 Vista。它有一个新的图形接口 Aero,以及许多其他新的成升级的用户程序。

在个人计算机世界中,另一个主要竞争者是 UNIX(和它的各种变体)。UNIX 在网络和企业服务器等领域强大,在台式计算机上,特别是在诸如印度和中国这些发展中国家里,UNIX 的使用也在增加。在基于 Pentium 的计算机上,Linux 成为学生和不断增加的企业用户们代替 Windows 的通行选择。顺便提及在本书中,我们使用 Pentium 这个名词代表 Pentium I, Pentium II, Pentium III 和 Pentium IV,以及它们的后继者,诸如 Core 2 Duo 等。术语 X86 有时仍旧用来表示 Intel 中的包括 8086 的 CPU,而 Pentium 则用于表示从 Pentium I 开始的所有 CPU。很显然,这个术语并不完美,但是没有更好的方案。人们很奇怪,是 Intel 公司的哪个天才把半个世界都知晓和尊重的品牌名(Pentium)扔掉,并替代以 Core 2 Duo 这样一个几乎没有人立即理解的术语,“2”是什么意思,而“Duo”又是什么意思?也许 Pentium 5(或者 Pentium 5 dual core)太难于记忆吧。至于 FreeBSD,一个源自 Berkeley 的 BSD 项目,也是一个流行的 UNIX 变体。所有现代 Macintosh 计算机都运行着 FreeBSD 的一个修改版。在使用高性能 RISC 芯片的工作站上,诸如 Hewlett - Packard 公司和 Sun Microsystems 公司销售的那些机器上,UNIX 系统也是一种标准配置。

尽管许多 UNIX 用户,特别是富有经验的程序员们更偏好基于命令的界面而不是 GUI,但是几乎所有的 UNIX 系统都支持由 MIT 开发的称为 X Windows 的视窗系统(如众所周知的 X11)。这个系统处理基本的视窗管理功能,允许用户通过鼠标创建、删除、移动和变比视窗。对于那些希望有图形系统的 UNIX 用户,通常在 X11 之上还提供一个完整的 GUI,诸如 Gnome 或 KDE,从而使得 UNIX 在外观和感觉上类似于 Macintosh 或 Microsoft Windows。

1.2.5 第五代计算机和操作系统

自从 20 世纪 40 年代连环漫画中的 Dick Tracy 警探对着他的“双向无线电通信腕表”说话开始,人们就在渴望一款无论去哪里都可以随身携带的交流设备。第一台真正的移动电话出现在 1946 年并且重达 40 kg,你可以带它去任何地方,前提就是你有一辆拉它的汽车。

第一台真正的手持电话出现在 20 世纪 70 年代,重约 1 kg,绝对属于轻量级。它被人们爱称为“砖头”。很快,每个人都想要一块这样的“砖头”。现在,移动电话已经渗入全球 90% 人口的生活中。我们不仅可以通过便携电话和腕表打电话,在不久的将来还可以通过眼镜和其他可穿戴设备打电话。而且,手机这种东西已不再那么引人注目。我们在车水马龙间从容地收发邮件、上网冲浪、给朋友发信息、玩游戏,一切都是那么习以为常。

虽然在电话设备上将通话和计算合二为一的想法在 20 世纪 70 年代就已经出现了,但第一台真正的智能手机直到 20 世纪 90 年代中期才出现。这部手机就是诺基亚发布的 N9000,它真正地做到了将通常处于独立工作状态的两种设备:(手机和个人数字助理)合二为一。1997 年,爱立信公司为它的 GS888“Penelope”手机创造出术语——智能手机。

随着智能手机变得十分普及,各种操作系统之间的竞争也变得更加激烈,并且形势比个人电脑领域更加模糊不清。谷歌公司的 Android 是最主流的操作系统,而苹果公司的 iOS

也牢牢占据次席,但这并不是常态,在接下来的几年时间里可能会发生很大变化。在智能手机领域,唯一可以确定的是,长期保持在巅峰并不容易。

毕竟,在智能手机出现后的第一个十年中,大多数手机自首款产品出厂以来都运行着 Symbian OS, Symbian 操作系统被许多主流品牌选中,包括三星、索尼爱立信和摩托罗拉,特别是诺基亚也选择了它。然而,其他操作系统已经开始侵吞 Symbian 市场份额,例如 RIM 公司的 Blackberry OS 和苹果公司的 iOS。很多公司都预期 RIM 能继续主导商业市场,而 iOS 会成为消费者设备中的王者。然而, Symbian 的市场份额骤跌。2011 年,诺基亚放弃 Symbian 并且宣布将 Windows Phone 作为自己的主流平台。在一段时间里,苹果公司和 RIM 公司是市场的宠儿,但谷歌公司 2008 年发布的基于 Linux 的操作系统 Android,并没有花费太长时间就追上了它的竞争对手。

对于手机厂商而言,Android 有着开源的优势,获得许可授权后便可使用。于是,厂商可以修改它并轻松地适配自己的硬件设备。并且,Android 拥有大量软件开发人员,他们大多数通晓 Java 编程语言。即使如此,最近几年也显示出 Android 的优势可能不会持久,并且其竞争对手及其渴望从它那里夺回一些市场份额。

1.3 操作系统的基本类型

按照操作系统的功能,可将操作系统分成以下几类:多道批处理系统、分时系统、实时系统、单用户操作系统、网络操作系统、分布式操作系统和嵌入式系统等。下面分别对这些系统进行阐述。

1.3.1 多道批处理系统

早期的批处理系统中只有一道作业在主存,系统资源的利用率仍然不高。为了提高资源利用率和系统吞吐量,在 20 世纪 60 年代中期引入了多道程序设计技术,形成多道批处理系统。

1. 多道程序设计

早期的批处理系统因为每次只调用一个用户作业程序进入主存并运行,故称为单道批处理系统,其主要特征如下:

(1) 自动性。在顺利的情况下,磁带上的一批作业能自动地、逐个作业依次运行而无须人工干预。

(2) 顺序性。磁带上的各道作业是顺序地进入主存的,各道作业完成的顺序与它们进入主存的顺序之间在正常情况下应当完全相同,亦即先调入主存的作业先完成。

(3) 单道性。主存中仅有一道程序并使之运行,即监督程序每次从磁带上只调入一道程序进入主存运行,仅当该程序完成或发生异常情况时才调入其后继程序进入主存运行。

图 1-4 所示说明了单道程序运行时的情况。图中说明用户程序首先在 CPU 上进行计算,当它需要进行 I/O 传输时,向监督程序提出请求,由监督程序提供服务并帮助启动相应的外部设备进行传输工作,这时 CPU 空闲等待。当外部设备传输结束时发出中断信号,由监督程序中负责中断处理的程序做处理,然后把控制权交给用户程序继续计算。

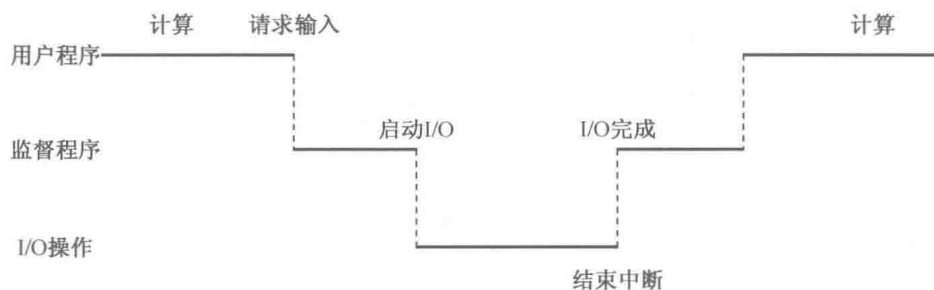


图 1-4 单道程序运行示例

从图 1-4 中可以看出,当外部设备进行传输工作时,CPU 处于空闲等待状态;反之当 CPU 工作时,设备又无事可做。这说明计算机系统各部件的效能没有得到充分的发挥,其原因在于主存中只有一道程序。在计算机价格十分昂贵的 20 世纪 60 年代,提高设备的利用率是首要目标。为此,人们设想能否在系统中同时存放几道程序,这就引入了多道程序设计概念。

多道程序设计是指允许各个作业(或程序)同时进入计算机系统的内存并启动交替计算的方法。也就是说,内存中多个相互独立的程序均处于开始和结束之间,从宏观上看是并行的,多道程序都处于运行过程中,但尚未结束;从微观上看是串行的,各道程序轮流占用 CPU,交替执行。引入多道程序设计技术,可以提高 CPU 的利用率,充分发挥计算机硬件部件的并行性。现代计算机系统都采用多道程序设计技术。

多道程序运行情况如图 1-5 所示。图示中用户程序 A 首先在处理器上运行,当它需要从输入设备输入新的数据而转入等待时,系统帮助它启动输入设备进行输入工作,并让用户程序 B 开始计算。程序 B 经过一段计算后,需要从输出设备输出一批数据,系统接受请求,并帮助启动输出设备工作。如果此时程序 A 的输入尚未结束,也无其他用户程序需要计算,处理器就处于空闲状态直到程序 A 在输入结束后重新运行。若程序 B 的输出工作结束时程序 A 仍在运行,则程序 B 继续等待直到程序 A 计算结束再次请求 I/O 操作,程序 B 才能占用处理器。

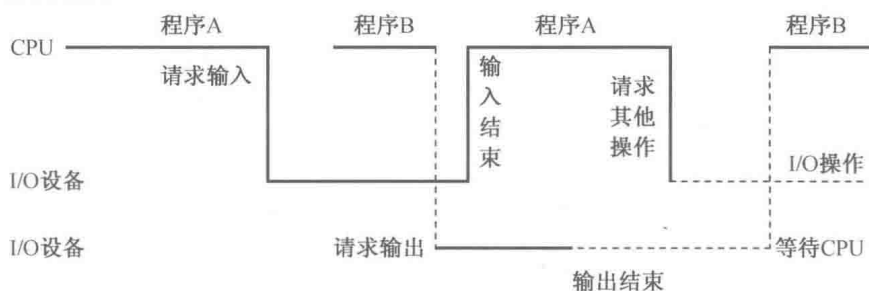


图 1-5 多道程序运行示例

操作系统中引入多道程序设计的优点:一是提高 CPU、主存和设备的利用率;二是提高系统的吞吐量,使单位时间内完成的作业数增加;三是充分发挥系统的并行性,设备与设备之间、设备与 CPU 之间均可并行工作。其主要缺点是延长了作业的周转时间。

2. 多道批处理系统

在批处理系统中,采用多道程序设计技术就形成了多道批处理系统。在多道批处理方式下,交到机房的许多作业由操作员负责将其从输入设备转存到辅存设备(比如磁盘)上,