

● 高等学校教材

● 孙钟秀 谭耀铭 费翔林 谢立 衣文国

操作系统教程

(第二版)



高等教育出版社

TP316

S97-2

(2)

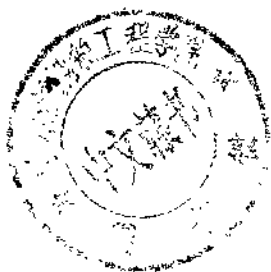
443498

高等学校教材

操作系统教程

(第二版)

孙钟秀 谭耀铭 费翔林 谢立 衣文国



00443498



高等教育出版社

(京)112号

JS147/03
内 容 提 要

本书重点介绍操作系统的基本概念、实现原理、设计方法和基本技巧。主要内容
包括:操作系统的形成与发展,处理器管理,存储管理,文件管理,设备管理,作业管理,
以及进程的互斥、同步、通信和死锁。另外,为使学生能对操作系统有一个整体概念,
本书还分别介绍了当前流行的 UNIX 操作系统和 MS-DOS 操作系统。

本书内容符合国家教育委员会 1991 年颁布的高等学校理工科计算机专业操作系
统课程的教学基本要求。可作为理工科计算机专业本科生教材,亦可供从事计算机工
作的有关人员阅读参考。

图书在版编目(CIP)数据

操作系统教程/孙钟秀等编著. — 2 版. — 北京:高等教
育出版社, 1998 重印
高等学校教材

ISBN 7-04-005381-0

I. 操… I. 孙… Ⅱ. 操作系统-高等学校-教材 N. T
P316

中国版本图书馆 CIP 数据核字(95)第 06679 号

*

高等教育出版社出版
新华书店总店北京发行所发行
高等教育出版社印刷厂印装

*

开本 850×1168 1/32 印张 10.75 字数 270 000
1989 年 5 月第 1 版
1995 年 12 月第 2 版 1998 年 12 月第 4 次印刷
印数 44 121-63 131
定价 10.50 元

前 言

操作系统是计算机系统软件中的一个不可缺少的重要组成部分,它出现于 50 年代末,至今已有三十余年。操作系统课程是有关计算机科学技术专业的一门专业基础课,因此,编写一本适用的操作系统教科书是十分需要的。70 年代以来,许多操作系统教科书相继问世,其中《操作系统教程》(作者:孙钟秀、谭耀铭、费翔林、谢立、衣文国)于 1989 年由高等教育出版社出版。该书出版后得到广大读者的欢迎和支持,许多学校选择该书作为教材或主要参考书。在 1992 年第二届全国高等学校优秀教材评选中《操作系统教程》被评为国家级优秀教材。

随着计算机科学技术的迅速发展,计算机应用的日益广泛,操作系统的新概念、新技术不断出现,为了适应这种发展的需要,必须对原教材进行更新。根据计算机科学技术的新发展和广大读者的反馈信息,我们对《操作系统教程》的内容作了适当修改、补充和调整,编写了这本《操作系统教程》(第二版)。

《操作系统教程》(第二版)保持了原教材的编写特点,力求做到:概念清晰,观点较高;深入浅出,便于自学;内容精选,取舍得当;难点分散,体系合理。全书着重讲解操作系统的基本原理和概念,以及设计方法和技巧,共分九章。第一章简述了操作系统的形成和发展历史;第二章至第六章叙述了操作系统的基本功能:处理器管理、存储管理、文件管理、设备管理和作业管理;第七章讨论了进程的互斥、同步、通信和死锁;第八、第九两章分别介绍了当前流行的 UNIX 操作系统和 MS-DOS 操作系统,希望能使读者有一个操作系统的整体印象。

参加本书修订工作的有：孙钟秀、谭耀铭、费翔林、谢立。

限于编著者的水平，错误与不妥之处定然难免，恳请读者批评指正。

编著者

1994年12月于南京

目 录

第一章 引言	1
1.1 操作系统概观	1
1.1.1 什么是操作系统	1
1.1.2 资源	1
1.1.3 一个系统的视图	1
1.2 操作系统的形成和发展	3
1.2.1 操作系统的形成	3
1.2.2 操作系统的发展	8
1.2.3 操作系统的产业	11
1.3 操作系统的功能	13
1.4 操作系统的分类	14
1.4.1 单用户操作系统	15
1.4.2 批处理操作系统	15
1.4.3 实时操作系统	17
1.4.4 分时操作系统	18
1.4.5 网络操作系统	18
1.4.6 分布式操作系统	19
习题	19
第二章 处理器管理	20
2.1 中断	20
2.1.1 中断装置	20
2.1.2 中断事件的处理	26
2.2 多道程序设计	33
2.2.1 多道程序设计的概念	33
2.2.2 多道程序设计的实现	37
2.3 进程	38
2.3.1 进程的概念	38
2.3.2 进程的建立	40
2.3.3 进程的创建和撤销	41

2.3.4 进程的状态及其转换	42
2.4 处理器调度	43
2.4.1 处理器调度的功能	43
2.4.2 队列机制	44
2.4.3 处理器调度策略	46
习题	48
第三章 存储管理	50
3.1 存储管理的功能	50
3.1.1 主存储器空间的分配和去配	51
3.1.2 主存储器空间的共享	51
3.1.3 存储保护	52
3.1.4 主存储器空间的扩充	52
3.2 连续存储空间管理	53
3.2.1 重定位	53
3.2.2 单连续存储管理	54
3.2.3 固定分区存储管理	56
3.2.4 可变分区存储管理	58
3.3 虚拟存储管理	65
3.3.1 概述	65
3.3.2 页式虚拟存储系统	66
3.3.3 段式和段页式虚拟存储系统	82
习题	93
第四章 文件管理	95
4.1 文件系统概述	95
4.1.1 文件的概念	95
4.1.2 文件系统及其功能	96
4.2 文件组织	98
4.2.1 文件的存储	98
4.2.2 文件的逻辑结构	100
4.2.3 文件的物理结构	110
4.3 文件目录	121
4.3.1 文件目录结构	121

4.3.2 文件目录的管理	124
4.4 文件的共享、保护和保密	131
4.4.1 文件共享	131
4.4.2 文件的保护和保密	132
4.5 文件的使用	135
4.5.1 存取方法	135
4.5.2 文件的使用	137
4.6 文件系统的实现	140
4.6.1 辅存空间管理	140
4.6.2 文件操作的执行过程	144
习题	145
第五章 设备管理	147
5.1 概述	147
5.1.1 设备管理的基本功能	147
5.1.2 I/O 控制方式	148
5.2 具有通道的 I/O 系统管理	152
5.2.1 I/O 指令	152
5.2.2 通道命令和通道程序	153
5.2.3 通道启动和 I/O 操作过程	155
5.2.4 输入输出程序设计	157
5.2.5 输入输出中断及其处理	160
5.3 缓冲技术	163
5.4 驱动调度	165
5.5 设备分配	173
5.5.1 设备独立性	173
5.5.2 设备分配	174
5.6 虚拟设备	177
5.6.1 问题的提出	177
5.6.2 斯普林系统的设计和实现	179
习题	183
第六章 作业管理	185
6.1 概述	185

6.1.1	作业和作业步	185
6.1.2	用户与操作系统的接口	186
6.1.3	作业管理的功能	187
6.2	批处理作业	187
6.2.1	作业控制语言	188
6.2.2	作业的组织	192
6.2.3	作业的装入	194
6.2.4	批处理作业的调度	195
6.2.5	作业调度与处理器调度	199
6.3	交互式作业	200
6.3.1	分时系统的优点	200
6.3.2	命令语言	201
6.3.3	终端用户的注册与注销	202
6.3.4	菜单技术	204
6.3.5	窗口技术	205
	习题	207
第七章	并发进程	208
7.1	并发性	208
7.1.1	顺序性和并发性	208
7.1.2	与时间有关的错误	209
7.2	进程的互斥	212
7.2.1	互斥和临界区	212
7.2.2	临界区的管理	214
7.2.3	用PV操作实现互斥	220
7.3	进程的同步	224
7.3.1	同步和同步机制	224
7.3.2	用PV操作实现同步	225
7.3.3	管程	228
7.3.4	管程的实现——汉森方法	232
7.3.5	管程的实现——霍尔方法	243
7.4	进程通信	247
7.4.1	进程通信的概念	247

7.4.2	直接通信	248
7.4.3	间接通信	249
7.4.4	有关问题的讨论	252
7.5	死锁	254
7.5.1	死锁的产生	254
7.5.2	死锁的防止	257
7.5.3	死锁的避免	259
7.5.4	死锁的检测	261
7.5.5	混合策略	263
7.6	并发程序设计	266
7.6.1	并发程序设计语言	266
7.6.2	并发程序设计技术	270
	习题	279
第八章	UNIX 操作系统简介	283
8.1	概述	283
8.2	UNIX 系统结构	284
8.2.1	Shell	284
8.2.2	系统调用	286
8.2.3	进程及进程调度	287
8.2.4	存储管理	289
8.2.5	文件管理	290
8.2.6	设备管理	292
8.3	UNIX 操作系统的使用	293
8.3.1	注册与注锁	293
8.3.2	Shell 命令的使用	294
第九章	MS-DOS 操作系统简介	302
9.1	概述	302
9.2	MS-DOS 的结构	303
9.2.1	MS-DOS 的层次结构	303
9.2.2	文件系统	306
9.2.3	设备驱动	311
9.2.4	命令处理	312

9.3 MS-DOS 的使用	315
9.3.1 MS-DOS 系统启动	315
9.3.2 键盘命令	316
9.3.3 批处理	321
9.3.4 软中断和系统功能调用	324
9.4 CCDOS 简介	329
9.4.1 CCDOS 的结构	329
9.4.2 汉字的编码	329
9.4.3 CCDOS 的编码转换	332
9.4.4 CCDOS 的操作模式	333

第一章 引 言

1.1 操作系统概观

1.1.1 什么是操作系统

操作系统是计算机系统的一种系统软件,它用于管理计算机的资源和控制程序的执行。一个程序只有在通过操作系统获得必需的资源后才能执行。例如,程序在执行前必须获得主存储器资源才能装入,它的执行要依靠处理器,它在执行中还可能要用外围设备输入或输出数据,或者使用计算机系统中的文件以及调用子程序等等。计算机配置了操作系统后可以提高效率,便于使用。现在,操作系统已成为计算机系统中不可缺少的一种软件。

1.1.2 资源

计算机系统的资源包括两大类:硬件资源和信息资源。

硬件资源包括中央处理器(也称处理器),存储器(例如主存储器、磁盘、磁带等)和输入输出设备(例如键盘输入设备、显示设备和打印机等)。信息资源包括程序和数据、程序库和共享文件等都是信息资源的例子。信息资源存放在存储器中供用户存取使用。

1.1.3 一个系统的视图

一个计算机系统可以被认为是由硬件和软件按层次方式构成。图 1-1 表示了一个四层结构,每层表示一组功能和一个接

口。接口是用于在该层内实现功能的一组可见的约定。我们把接口的这些特性称为系统的一个视图。

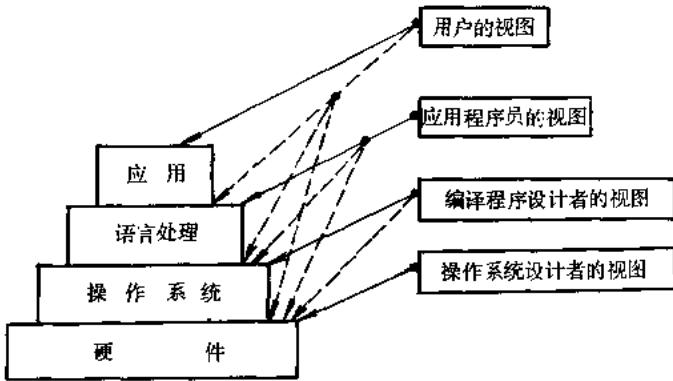


图 1-1 一个系统的分层和视图

硬件层表示机器的可见结构,它包括:可执行一组指令的处理器,若干个供程序使用的寄存器和用于访问存储器的寻址模式。此外,还包含诸如通道、控制器、处理器和存储器之间的关系。它是操作系统工作的基础。因而,对操作系统的设计者来说,他所看见的一个系统视图就是硬件层。

操作系统对硬件层作第一次功能扩充,以便为编译程序的设计者和应用程序员有效的服务。它提供接口以便容易地开发系统程序。操作系统提供的接口并不能完全隐藏硬件特性,因此,一个编译程序的设计者可能需要某些机器特性的知识。

编译工作的基础是被操作系统扩充了功能的机器,它由软件定义的操作系统接口和硬件指令集合的某些部分组成。类似地,一个使用汇编语言的程序员将利用操作系统和硬件提供的复合功能。因而,向编译程序的设计者所展示的一个系统视图除了操作系统外,还应加上操作系统未能隐藏的硬件特性,它们在图 1-1 中分别用实线箭头与虚线箭头表示。

同样,应用程序员的视图除了语言处理层外,还有未被隐藏的

部分操作系统和硬件的特性。而用户的视图除了应用层外,还有未被隐藏的部分语言处理层、操作系统层及硬件层的特性。它们在图 1-1 中也分别用实线箭头和虚线箭头表示。

1.2 操作系统的形成和发展

1.2.1 操作系统的形成

第一代计算机速度较低,外围设备较少,因而,编制和运行一个程序也比较简单。那时候,程序员往往直接使用机器语言来编制一个程序,这种“目标程序”被人为地穿在卡片(或纸带)上,并用一个引导程序装入主存储器。程序员通过控制台开关来调试和操作运行程序。在这期间,整个计算机都是被一个程序员所占有。因而,不需要专门的操作员,程序员身兼两职——既是操作员,也是程序员。

随着计算机的发展,协助用户使用计算机软件——原始汇编系统产生了。在这样的系统中,数字操作码被记忆码所代替,程序按一个固定格式的汇编语言书写。程序员(或系统程序员)预先编制一个汇编解释程序,它把汇编语言书写的“源程序”解释成计算机能直接执行的用机器语言书写的“目标程序”。因而,在这样的计算机系统中,首先需要把这个汇编解释程序和源程序都穿在卡片或纸带上,然后再装入和执行。如图 1-2 所示,整个计算分两个阶段,六个计算步,每步的功能如下:

- (1) 通过引导程序把汇编解释程序装入到计算机;
- (2) 通过汇编解释程序读入源程序,并执行汇编过程;
- (3) 产生一个目标程序,并输出到卡片或纸带上;
- (4) 通过引导程序把目标程序装入计算机;
- (5) 目标程序读入卡片数据或纸带数据;

(6) 产生计算结果,并输出到卡片或打印纸上。其中 1—3 的三个计算步是汇编阶段,4—6 的三个计算步是执行阶段。

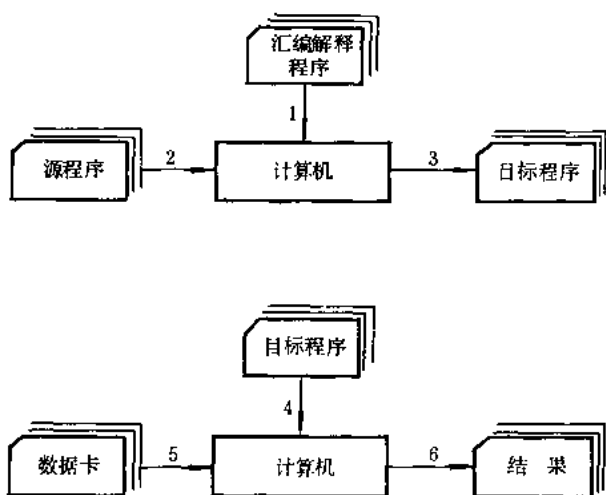


图 1-2 原始汇编和执行过程

这样的工作方式尽管比第一种工作方式前进了一步,但本质上是一样的,都是手工操作方式。手工操作方式的主要特点是:用户独占计算机;用户上机算题是一个个“串行地”进行的;调试和控制程序的运行采用“亮灯显示,开关控制”。为了让程序员能通过开关控制程序装入和运行,设计了一个解释开关命令的程序。此外,为了方便用户,还配置了一些控制外围设备的程序,供用户调用。这些控制程序可以看成是原始的操作系统。

手工操作方式对第一代计算机还是适应的,但是,随着第二代计算机的出现,这种方式就很不适合了。首先,手工操作只能作一些简单的控制,但是,对于规模比第一代计算机大得多的第二代计算机,只作一些简单的控制已不能满足需要。其次,手工操作要求用户熟悉机器的很多细节,既带来了麻烦,又增加了出错机会。第

三,手工操作速度慢,对于速度较快的第二代计算机来说,这样做会降低计算机的效率。例如,在一台第一代计算机上花了一个小时计算的一个问题,在计算过程中,人工操作可能花了三分钟,仅占全部时间的5%。但是,当计算机的速度提高了十倍以后,这个问题可能只要计算六分钟,而人工操作如果仍需三分钟,就占了全部时间的50%。最后,对于规模较大,设备配置较多的第二代计算机,许多小型的计算问题无需使用系统的全部功能,“独占”全机的做法不能充分利用系统资源。所以,随着第二代计算机的诞生,不久就出现了管理程序——初级的操作系统。

50年代末60年代初,计算机发展进入了第二代,不仅计算机的速度有了很大提高,而且存储容量也有了很大的增长,这就给软件的发展奠定了物质基础。在这期间,先后出现了FORTRAN和ALGOL等程序设计语言与相应的编译程序以及程序库的使用等。同时,出现了对计算机硬件和软件进行管理与调度的软件——管理程序。例如,美国IBM360系列计算机系统的初级控制程序和英国1900系列计算机系统的执行程序等,就是这种管理程序。

管理程序的主要功能是:向用户提供多个共享资源来运行他们的程序;帮助操作员控制用户程序的执行和管理计算机的部分资源。管理程序为用户提供一套控制命令,并以一定格式穿在卡片上,称为控制卡。图1-3列出了某些简单的控制卡,用户用控

控制卡	功能
* Job	启动一个程序
* Fortran	调用 Fortran 编译程序
* Assemble	调用汇编解释程序
* Load	调用装入程序
* Data	表明随后是数据卡
* End	发出数据结束信号

图 1-3 一个典型管理程序的某些控制卡

制卡来表达对程序进行控制的要求。管理程序通过读入和执行这些控制卡而成批地处理用户的程序,构成了程序自动定序,即在一个程序处理完毕后,管理程序再自动启动下一个程序而无需操作员进行干预。

有了管理程序以后,用户不必亲自上机操作,而是由专业化的操作员代劳。操作员只要通过控制台打字机打入控制命令就可以操纵计算机,操作员打入的命令由管理程序来识别和执行。这样,不仅操作速度快,而且操作员可以方便地进行一些较为复杂的控制。当计算机运行中发生错误或意外时,管理程序通过计算机从控制台打字机上输出信息向操作员报告。这种输出信息不仅比“亮灯显示”所表达的更为丰富,而且操作员也易于理解。总之,用这种半自动方式来控制计算机不仅提高了效率,而且方便了使用。

在管理程序控制下的一个典型模式如图 1-4 所示。

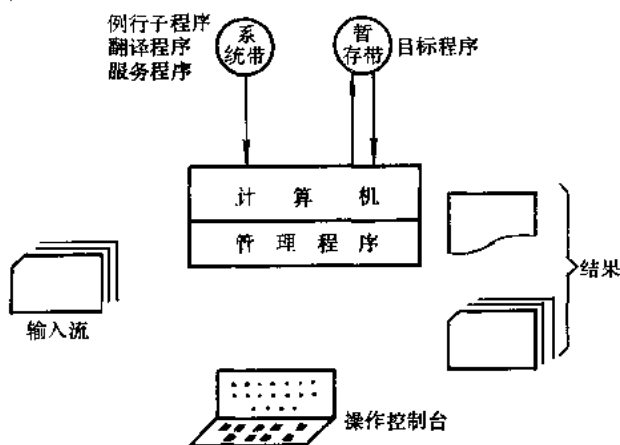


图 1-4 具有管理程序的计算机系统

一般地说,管理程序的主要部分需要预先装入计算机,它的某些例行程序和语言编译程序、装入程序可以一起存放在系统带上。假设,用户使用图 1-3 中的控制卡来表达对某个运行程序的要