

# 计算机操作系统

黄干平 陈洛资 等 编著

科学出版社

# 计算机操作系统

黄干平 陈洛资 等 编著

科学出版社

1989

## 内 容 简 介

本书是根据全国计算机教育与培训学会本科学组 1987 年推荐的“操作系统”教学大纲编写的。全书共十三章。第一章介绍了操作系统的概貌。第二章至第十二章分别讲述了并发进程、作业管理、存贮管理、虚拟存贮器、CPU 调度、设备管理、文件系统、死锁、并发程序设计、分布式系统和系统性能分析。第十三章是一个操作系统实例，详细介绍了较新版本的 UNIX 操作系统。此外，书末的附录还分别介绍了 VMS, MVS 和 DOS 三个操作系统。书中共附有 200 多道习题。

本书内容系统、全面，编排合理，论述清晰，集中讲述操作系统的基本概念、基本功能和实现方法，是计算机专业本科生的教材，也可供有关专业的教师和科技人员阅读、参考。

## 计算机操作系统

黄干平 陈洛资 等 编著

责任编辑 那莉莉

科学出版社出版

北京东黄城根北街 16 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

\*

1989 年 6 月第 一 版 开本：787×1092 1/16

1989 年 6 月第一次印刷 印张：22 3/4

印数：0001—4,270 字数：528,000

ISBN 7-03-001179-1/TP·73

定 价：12.00 元

# 前 言

操作系统是计算机系统基本的和重要的软件,是其它系统软件的支撑软件,因此“计算机操作系统”是各类大学计算机系或计算机专业的必修课程。为适应本科教学需要,我们编写了这本教材。

本书是根据中国计算机教育与培训学会本科学组1987年推荐的“操作系统”教学大纲编写的。全书共十三章。第一章至第十二章讲述操作系统原理,第十三章介绍一个操作系统实例——UniPlus<sup>TM</sup> UNIX 操作系统。原理部分集中讲述了各种操作系统的基本概念、基本功能、数据结构、基本算法和实现技巧。第一章是操作系统引论,讲述与操作系统全局有关的内容;第二章介绍操作系统中最重要的基本概念之一——进程;第三章至第八章,按照操作系统的资源管理观点,逐章讲述操作系统的各个组成部分:作业管理、存贮管理、虚拟存贮器、CPU调度、设备管理和文件系统;第九章讨论操作系统的特殊问题——死锁;第十章介绍并发程序设计语言和设计方法,这些语言和方法可用于操作系统的设计;第十一章介绍作为操作系统最新发展方向之一的分布式系统;第十二章介绍系统性能评估的理论方法和数学模型,主要是排队论和马尔科夫过程;第十三章介绍的 UNIX 操作系统是 UNIX 的较新版本,与 UNIX System 5 很接近。除第十三章外,每章后面都附有一定数量的习题。本书的附录介绍了目前几个较流行的操作系统。

本书的主要特点是做到了原理与实例的结合。因此,对不同类型的学校,在教学上可以作不同的处理:只讲原理,或只讲实例,或原理和实例都讲。

第一章至第十一章由黄干平执笔,第十二章由李琼章执笔,第十三章由陈洛资、陈济彪、李东生执笔。黄美绮、朱菊萍和周力等同志对本书的出版给予了大力支持和协助,北京计算机学院的王勇领同志对本书的编写提出了宝贵意见,在此谨向他们深表谢意。

由于作者水平有限,书中一定还会有错误和疏漏之处,欢迎读者批评指正。

编著者

1988年10月

# 目 录

第一章 绪论	1
1.1 什么是操作系统	1
1.2 操作系统的分类	3
1.3 操作系统的结构	4
1.4 操作系统的特征	10
1.5 中断在操作系统中的地位	11
1.6 网络环境下的操作系统	14
1.7 操作系统的启动和工作过程	14
习题	15
第二章 并发进程	16
2.1 进程的基本概念	16
2.2 进程的控制和管理	20
2.3 进程的互斥与同步	23
2.4 进程通信	36
习题	40
第三章 作业管理	43
3.1 作业的进入	44
3.2 作业的后备状态	44
3.3 作业调度	45
3.4 从作业到进程	50
3.5 作业的运行	51
3.6 作业的终止和完成	52
3.7 JSP 工作流程	52
习题	54
第四章 存贮管理	56
4.1 引言	56
4.2 分区管理	59
4.3 分页管理	67
4.4 分段管理	72
4.5 段页式管理	78
习题	80
第五章 虚拟存贮器	82
5.1 基本概念	82
5.2 请求分页的实现过程	83
5.3 页面置换算法	85
5.4 性能研究	88

习题 .....	91
<b>第六章 CPU 调度 .....</b>	<b>93</b>
6.1 概述 .....	93
6.2 CPU 调度算法 .....	96
6.3 调度的实现 .....	104
6.4 多处理机调度算法 .....	106
习题 .....	111
<b>第七章 设备管理 .....</b>	<b>113</b>
7.1 概述 .....	113
7.2 缓冲存储器及其管理 .....	115
7.3 设备分配 .....	119
7.4 I/O 的实现 .....	122
7.5 磁盘调度策略 .....	125
7.6 SPOOLing 技术 .....	129
习题 .....	132
<b>第八章 文件系统 .....</b>	<b>133</b>
8.1 概述 .....	133
8.2 目录结构 .....	137
8.3 文件的共享和保护保密 .....	142
8.4 文件的物理结构 .....	147
8.5 文件卷 .....	152
8.6 打开文件管理机构 .....	155
8.7 文件系统的执行过程 .....	159
习题 .....	160
<b>第九章 死锁 .....</b>	<b>162</b>
9.1 死锁基本概念 .....	162
9.2 死锁的预防 .....	165
9.3 死锁的避免 .....	167
9.4 死锁的检测和消除 .....	169
9.5 处理死锁的综合措施 .....	171
习题 .....	172
<b>第十章 并发程序设计 .....</b>	<b>174</b>
10.1 引言 .....	174
10.2 并发语言的结构及其功能 .....	174
10.3 一个并发程序的例子 .....	184
习题 .....	188
<b>第十一章 分布式系统 .....</b>	<b>190</b>
11.1 为什么要建立分布式系统 .....	190
11.2 拓扑结构 .....	191
11.3 通信 .....	194
11.4 系统类型 .....	197
11.5 文件系统 .....	198

11.6	计算模型 .....	200
11.7	事件定序 .....	201
11.8	同步 .....	202
11.9	死锁处理 .....	204
11.10	健壮性 (robustness) .....	205
11.11	小结 .....	208
	习题 .....	209
<b>第十二章</b>	<b>系统性能分析 .....</b>	<b>210</b>
12.1	性能评估方法 .....	210
12.2	排队论 .....	212
12.3	马尔可夫过程 (Markov process) .....	221
12.4	小结 .....	226
	习题 .....	227
<b>第十三章</b>	<b>一个操作系统实例——UniPlus™ UNIX 操作系统 V.7 .....</b>	<b>229</b>
13.1	UNIX 中的进程 .....	229
13.2	存贮器管理 .....	237
13.3	中断与陷入 .....	243
13.4	进程管理 .....	257
13.5	通信 .....	272
13.6	块设备管理 .....	286
13.7	字符设备管理 .....	303
13.8	文件系统 .....	313
<b>附录 A</b>	<b>VMS 操作系统 .....</b>	<b>334</b>
A.1	存贮管理 .....	334
A.2	进程调度 .....	335
A.3	输入/输出 .....	336
A.4	记录管理服务 .....	339
A.5	进程之间的通信和同步 .....	340
A.6	小结 .....	342
<b>附录 B</b>	<b>MVS 操作系统 .....</b>	<b>343</b>
B.1	作业管理 .....	343
B.2	存贮管理 .....	343
B.3	多处理机同步和管理 .....	345
B.4	数据管理 .....	347
B.5	系统资源管理 .....	348
B.6	小结 .....	348
<b>附录 C</b>	<b>DOS 操作系统 .....</b>	<b>350</b>
C.1	概述 .....	350
C.2	DOS 结构 .....	350
C.3	内存管理及其相应功能的应用 .....	351
C.4	文件系统 .....	352
	<b>参考资料 .....</b>	<b>356</b>

# 第一章 绪 论

本章介绍操作系统的基本概念,包括什么是操作系统,为什么要设置操作系统,操作系统的分类、结构、特征及其工作过程等等。

## 1.1 什么是操作系统

每一个使用过计算机的人都从用户的角度接触和使用过操作系统,得到过操作系统的帮助和服务,每一个计算机工作者都不同程度地学习、分析、解剖甚至设计过操作系统。那么,什么是操作系统呢?概括地说,操作系统(operating system)是一管理计算机系统的软件,它为用户使用计算机提供方便、有效和安全可靠的工作环境。这样一个解释,既说了操作系统是什么,又说了操作系统干什么,比较全面但不详尽,需要作进一步的说明。

首先,从概念上说,操作系统是软件而不是硬件。但操作系统这一软件的重要任务之一是管理计算机本身的机器硬件。因此,在操作系统运行和实现其功能的过程中,需要硬件的强有力的支持,而且操作系统的一部分功能就是由硬件直接完成的,这一点在随后各章可清楚地看出。从这个意义上说,操作系统又不完全是软件,它是一个软、硬结合的有机体——在软、硬件的相互配合下,共同完成操作系统所应完成的任务。

其次,操作系统是系统软件而不是应用软件,是由系统程序员研制和编写的,并把它“灌制”在外存(主要是硬磁盘或软磁盘)上,随计算机硬件而运行。对于任何一个现代计算机系统来说,可以没有其它软件,但决不可以没有操作系统。

操作系统不仅与应用程序不同,也与其它系统程序不同。我们知道,一个完善的机器系统通常都配有众多的系统程序,如编辑程序,编译、汇编程序,子程序库,各种命令的解释、执行程序,数据库管理系统,等等。所有这些程序,虽然与操作系统一样都属于系统软件,但它们都受操作系统的管理和控制,并得到操作系统的支持和服务,它们之间的关系大体如图 1.1 所示。

上述对操作系统的解释,说明了如何看待操作系统的两种观点和为什么要设置操作系统的两个目的。操作系统的两个观点,即资源管理观点和服务用户观点;设置操作系统的两个主要目的是,高效地利用计算机系统资源和为用户使用计算机创造良好的工作环境。

操作系统的资源管理观点是说操作系统是管理计算机系统资源的程序。如果把一个计算机系统比作一个“国家”,则操作系统可以说是这个国家的“政府机构”(government)。计算机有哪些系统资源?它有硬件资源和软件资源。硬件资源包括 CPU、内存贮器、外存贮器和辅助存贮器、各种外部设备,软件资源主要指的是文件。所有系统程序以及用户要求长期保存的程序和数据都是以文件的形式存放在系统内的。

现代计算机系统通常都是多道程序系统,即同时有多个程序在主机内运行的系统。因



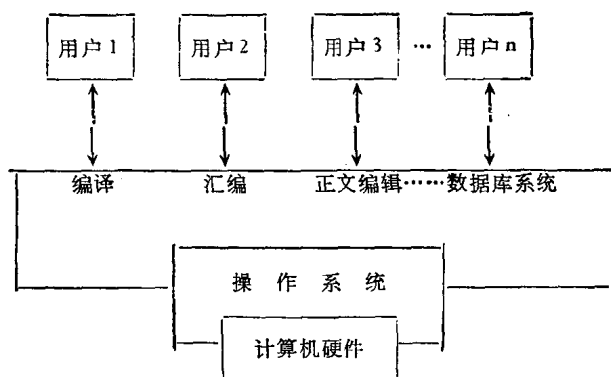


图 1.1 操作系统与其它软件的关系

此,所谓资源管理,主要指的是在多道程序之间合理地分配和回收各种资源,使资源得到合理有效的使用,使程序得以有条不紊地运行。

从资源管理观点出发,可以把整个操作系统分成以下几部分:存贮管理、CPU 调度(即处理机管理)、设备管理和文件系统。存贮管理主要是指在多道程序之间分配和回收内存空间,其中包括作为内存的扩展和延伸的辅存空间;CPU 调度负责在多道程序之间分配 CPU,使每道程序都能得以运行;设备管理统一管理各种已登入系统的外部设备并负责外设和主机之间的信息交换(即 I/O);文件系统统一管理各种以文件形式保存在外存上的信息,负责文件的建立、读、写和删除等等。

操作系统的管理职能也包括对作业和进程的管理,即对程序运行的管理。第二章将介绍进程的基本概念并讨论进程管理的细节。进程是计算机系统生动活泼的个体,是计算机王国的公民。第三章讲述作业管理。本书就将根据上述操作系统的管理职能及其划分原则讲述操作系统原理。

操作系统的服务用户观点告诉我们,操作系统作为软件,只不过是一个为用户服务的程序,当然是大型和复杂的程序。那么,一个用户在使用计算机的过程中需要哪些公共的服务呢?主要有以下一些:

1. 程序执行 任何一个用户程序都要在 CPU 上运行,而在此之前必须被首先装入内存,这是最起码的。
2. 输入/输出操作 任何一个在计算机上运行的程序都必须首先通过外部设备输入主机,程序在运行前和运行中都有可能输入其所加工的数据,任何程序都有其运行结果,这些结果无疑都要通过外部设备输出。
3. 信息保存 用户如果认为他的程序或数据是重要的,需要长期(或较长期)保存以便将来再次运行或进一步完善的,则要以文件的形式将程序存贮在外存上。系统要为用户的这种要求提供支持和服,既要使用户能方便存取和使用自己的文件,又要防止文件的泄密、被盗用或丢失。
4. 错误检测和处理 用户程序在运行过程中有可能出现各种错误,如算术运算溢出、程序中出现非法指令、访存地址越界等,操作系统应有能力对这些错误作出处理。此外,机器本身也有可能出现各种软、硬件故障,操作系统应及时地检测并修复它们,以保证用户程序的顺利运行。

上述服务功能是任何一个用户都需要的,是公共的和基本的。如果实现这些功能的程序都由每个用户去编写,这不仅增加了用户的工作量,而且这种重复性劳动往往容易产生各种各样的错误,使计算机系统的可靠性和有效性大大下降,甚至整个机器无法使用。因此,将这些功能集中起来,由系统程序员编写实现它们的程序是明智的、有效的、可靠的和易于实现的。实现这些功能的程序就是操作系统。

当然,操作系统不仅为用户程序的运行提供基本的服务,也为其它系统程序(编辑、编译、数据库管理等)的运行提供同样的服务,如果这些程序的运行需要这些服务的话。

操作系统的服务观点,不仅使我们从一个侧面对操作系统的功能有了进一步的了解,而且也给出了其工作流程的一个大致的轮廓,因为操作系统的工作流程也就是沿着用户程序的执行提供各种服务的过程。此流程可粗略表示如下: 程序和数据输入→作业收容→作业调度→作业运行→结果输出,如图 1.2 所示。

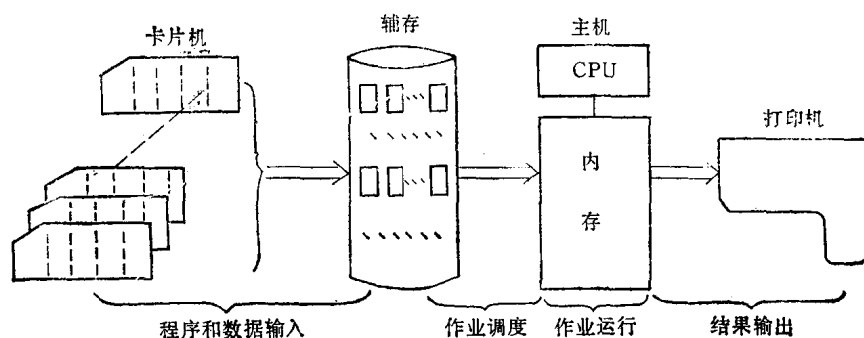


图 1.2 操作系统的服务流程

在程序和数据的输入阶段,操作系统提供输入服务,把它们从外设(例如卡片机)上输入到辅存,这时程序变成了计算机收容的作业(jobs)。然后,操作系统按一定的调度算法,把这些作业调度到主机(CPU + 内存)运行,这里涉及到为作业分配外设和内存等资源。作业调入主机后,在多道环境下运行,于是操作系统要在它们之间合理地分配 CPU,使它们都能得以运行。当一作业运行终止之后,总有一些结果需要输出,这时操作系统提供输出服务,把用户所需要的结果在不同的设备(如显示终端、打印机、绘图仪等)上输出。操作系统就是按照这样的流程周而复始地为用户服务的。

操作系统的管理观点和服务观点是一致的:通过管理而达到服务的目的,通过管理为用户提供良好的工作环境。良好的标准有两条:方便和安全。一个系统是否方便好用,这可以由用户的直觉来决定。至于“安全可靠”,主要指的是多用户、多作业共享同一计算机时互相不发生干扰以及尽量减小意外事件对用户程序和系统本身的破坏。

## 1.2 操作系统的分类

操作系统发展到今天,已经取得了辉煌的成果,各种功能完善,使用方便的系统正在各类大、中、小、微型计算机上运行。例如,由美国麻省理工学院等单位研制的 Multics 系统,由美国 Bell 实验室等单位研制并广泛使用于 PDP-11 小型机和 16 位微机的 UNIX

系统，IBM 公司为其系列机研制的 MVS 系统，运行在不同型号的 VAX-11 机上的 VMS，曾在 8 位微机上广泛盛行的 CP/M 以及在 IBM-PC 机上使用的 DOS 等，都是为人们所熟知的著名的操作系统。事实上，国内外比较著名的系统甚至多达几十种。

虽然操作系统的种类如此之多，但从功能特征上大致可分为三大类，即批处理操作系统，简称批处理系统 (batch system)；分时操作系统，简称分时系统 (time-sharing system) 和实时操作系统，简称实时系统 (real-time system)。

批处理系统，也叫作业流处理系统，主要装配在用于科学计算的大型机上。它的特点是采用 Spooling 技术将作业一批一批地输入主机进行处理。在作业处理的过程中，不需要用户的控制和干预，它所追求的目标是作业周转时间短、系统吞吐量大、资源的使用效率高等。

分时系统是多用户共享系统，它一般连接多个显示终端，每个用户都通过相应的终端键盘使用计算机。分时系统的主要特点是交互性，即用户使用键盘命令，以人-机会话的方式与系统打交道。操作系统按分时原则为每个用户服务，以合理的响应时间满足用户的要求，使每个用户感到机器好象是在为他一个人服务。

分时系统的工作方式如图 1.3 所示。

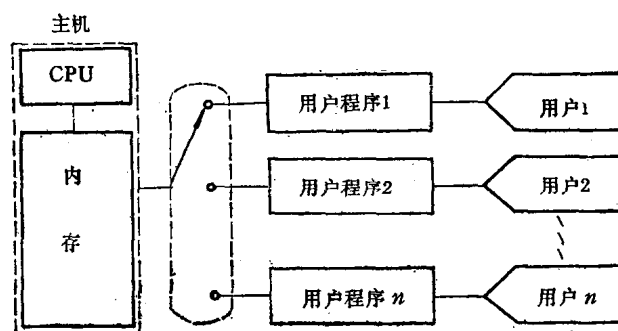


图 1.3 分时操作系统的工作方式

实时系统一般是专用控制系统。实时是“立即”、“马上”的意思。实时系统的特点是，它对外部输入信号作出反应的速度足以控制发出信号的对象。情报检索、飞机订票、银行业务、电报电话交换等都是实时系统的例子。

上述三类不同的操作系统，由于它们的侧重点不同，所使用的一些具体策略和算法也不相同，但它们的基本原理还是相同的。由于批处理系统规模较大，功能较强，且包含了操作系统的各种基本概念，故本书将较多地以批处理系统为例阐述操作系统原理。

### 1.3 操作系统的结构

操作系统的结构包括它的内结构和外结构。内结构指的是操作系统内部程序模块的层次结构，外结构指的是操作系统与其它系统软件的关系。

先说内结构。作为系统软件的操作系统的集合。这些模块各有自己的功能，它们共同完成操作系统的总任务。这些程序模块按照它们的功能及其之间的调用关系可以分成若干个层次，如图 1.4 所示。

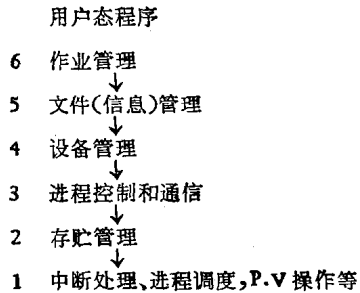


图 1.4 操作系统的内结构

上述每一层都由若干数量不等的程序模块组成。例如，以 UNIX 系统为例，内存管理只有 10 个左右的程序模块，而文件系统有多达 70~80 的模块。

上述层次构成一个半序结构。就是说，高层模块可以调用它的所有低层模块，同层次的模块可以互相调用，但低层的不能调用高层的。了解这样的层次结构是很重要的，它为我们学习、分析和设计操作系统提供了一种结构化方法。从设计上说，通常有三种顺序，即从上到下，从下到上以及从中间向上、下两个方向发展。

其次是外结构，即操作系统与其它系统软件的关系。此关系大体上可以用图 1.5 表示。

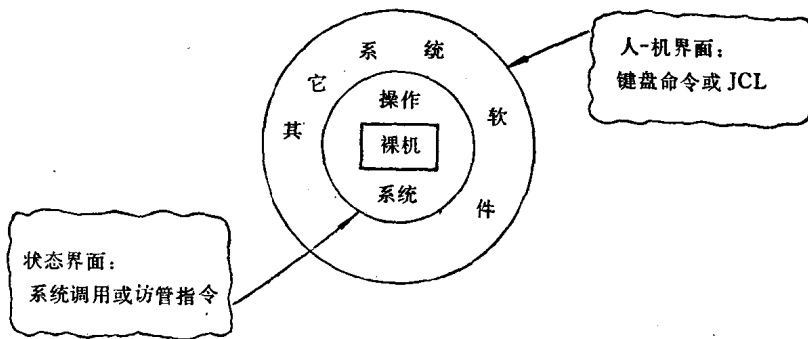


图 1.5 操作系统的外结构

任何计算机系统的基础是由硬件构成的物理计算机。装入了操作系统之后，就好象给计算机穿上了一件“衣服”，使计算机的功能更完善，使用更方便，工作更有效。

如上节所说，操作系统以外的系统软件还很多，它们与操作系统之间的关系已如图 1.1 所示。关于它们之间的关系，这里须进一步说明几个问题。

### 1.3.1 两种机器状态

任何一个计算机系统都有两种运行状态，即核心态(也叫系统态、管态)和用户态(也叫目态)，二者必居其一。当操作系统内的程序模块运行时，机器处于核心态，其它程序运行时机器处于用户态。值得注意的是，操作系统以外的其它系统程序与用户程序一样都是运行在用户态下而不是运行在核心态下。为此，我们把它们统统叫做用户态程序。请注意，用户程序只是用户态程序的一部分。

在实际系统中，之所以要区分机器的两种运行状态，目的是要给操作系统以某些特权。例如，改变状态寄存器和地址映射寄存器的内容，存取外部设备局部寄存器的内

容,等等。这些特权是通过执行特权指令实现的。仅当在核心态下才能执行特权指令,若在用户态下执行特权指令则为非法。

### 1.3.2 操作系统的用户态界面

操作系统是计算机系统的核心软件,它支持和管理用户态程序的运行并为它们提供基本的服务。所谓操作系统的用户态界面指的是操作系统与其它用户态程序之间的分界面。分界面以内的程序(即操作系统)运行在核心态下,分界面以外的程序运行在用户态下,因此,有时也叫状态界面。这个界面由系统调用指令 SCI (System Call Instructions) 组成,有的系统叫访管指令 SVC。例如,UNIX 系统提供了 50 多条系统调用指令。任何一个用户态程序,要想执行操作系统内的程序模块,以获得操作系统提供的服务,唯一的办法就是执行相应的系统调用指令——系统调用指令是操作系统与用户态程序之间的唯一接口。系统调用指令通常是面向汇编语言的,有的也可面向象 C 这样的高级语言。

每一条系统调用指令与操作系统中一个可直接由用户态程序调用的程序模块相对应。因此,作为上述界面的系统调用指令集合应包含哪些指令,是由操作系统中的哪些功能、哪些程序模块可由用户态程序直接调用决定的。一般来说,操作系统中的以下一些功能(因而实现这些功能的程序模块)是可以由用户态程序直接调用的:

第一,进程控制。

- END, Abort
- Load, Execute
- Create Process, Terminate Process
- Get Process Attributes, Set Process Attributes
- Wait for Time
- Wait Event, Signal Event

第二,文件操作。

- Create File, Delete File
- Open, Close
- Read, Write, Reposition
- Get File Attributes, Set File Attributes

第三,设备管理。

- Request Device, Release Device
- Read, Write, Reposition
- Get Device Attributes, Set Device Attributes

第四,信息保存。

- Get Time or Date, Set Time or Date
- Get System Date, Set System Date
- Get Process, File, or Device Attributes
- Set Process, File, or Device Attributes

与操作系统中实现上述功能的程序相对应的指令都应该是系统调用指令,都应是界面的一分子。用户态程序若要请求操作系统的服务,唯一的办法就是执行相应的系统调

用指令,以此来调用并执行操作系统中的相应程序模块。

值得注意的是,用户态程序经系统调用指令调用操作系统内部的程序模块与一般的子程序调用是有原则区别的。这是因为,一般的子程序调用并不涉及到机器运行状态的改变,而用户态程序调用操作系统内的程序模块则涉及到机器运行状态的改变:用户态程序是在用户态下运行,操作系统是在核心态下运行。因此,在实际系统中,系统调用指令的执行是经中断机构“陷入”(trap)操作系统的,它导致从用户态转入核心态,保存调用程序的现场,并调用相应的模块。当被调模块执行完毕后,经中断机构返回,机器由核心态变回到用户态。如图 1.6 所示。

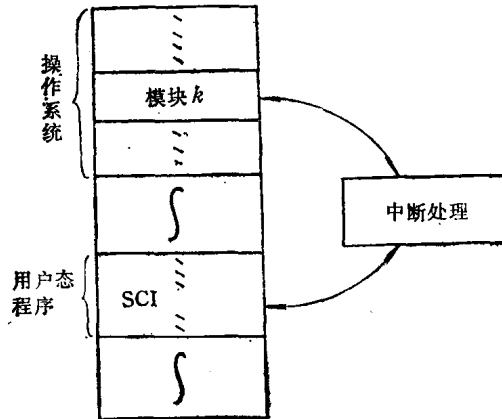


图 1.6 系统调用指令执行过程

### 1.3.3 人-机界面

人-机界面是人与计算机之间的接口,也是人与操作系统之间的接口。任何一个用户只有通过这个界面才能与计算机系统打交道,获得机器的服务。对于批处理系统,这个界面就是作业控制语言 JCL (Job Control Languages),也就是说,用户使用这个语言(或类似语言)编制作业控制程序,以脱机方式控制作业的运行;对于多用户分时系统,这个界面由键盘命令组成。终端用户使用这些命令以交互会话方式与计算机打交道并运行各自的程序。不同的系统,键盘命令的多少和功能的强弱是各不相同的。下面是在某 UNIX 分时系统上使用键盘命令与机器会话、上机的一个简单例子。

```

DASTRA SYSTEMS 68000 MULTIBOX WITH UNISOFT UNIX
:LOGIN: ROOT                                (“挂号”,进入系统)
PASSWORD:                                    (输入“口令”)
# DATE                                        (询问上次上机时间)
SAT NOV 28 15:04:14 CST 1987
# DATE 8801010830.25                          (送入新时间)
FRI JAN 1 08:30:25 CST 1988
# WHO                                         (询问哪些用户在用机)
ROOT CONSOLE JAN 6 05:43
NAME4 TTY?? NOV 28 15:00
NAME4 TTY2 JAN 6 06:26
NAME4 TTY3 NOV 28 15:01
    
```

NAME4 TTY6 JAN 6 06:11

# PS-AXL

(查询系统内的进程及有关信息)

```

F S UID PID PPID CPU PRI NICE ADDR SZ WCHAN TTY TIME CMD
3 S 0 0 0 0 0 20 34 1 104B8 ? 1:04 SWAPPER
0 S 0 1 0 0 30 20 14B 8 11F10 ? 0:01 /ETC/INIT
0 S 0 31 1 0 28 20 11B 6 19868 CO 0:07 -SH
0 S 40 222 1 0 30 20 9F 6 11F6C 1 0:01 -SH
1 S 0 24 1 0 40 20 5D 6 1FF000 ? 0:03 /ETC/UPDATE
1 R 0 26 1 104 56 20 67 8 ? 4:07 /ETC/CRON
0 S 40 93 1 0 30 20 20F 6 11FF6 2 0:02 -SH
0 S 40 215 1 0 30 20 65 6 12024 3 0:04 -SH
1 S 0 240 1 0 30 20 35 6 12052 4 0:06 -SH
0 S 0 36 1 0 28 20 E3 7 193C4 5 0:01 -9

```

# PWD

(查询工作目录)

/

# LS

(列表根目录下的文件名)

```

.CSHRC BIN F LIB TMP USR
.EXRC DEV FDLOAD LOST + FOUND UNIX
.LOGIN ETC HUO T UNIX1.8.2:KO

```

# CD/USR/NAME1

(改变工作目录)

# LS

(列表/USR/NAME1 下面的文件名)

```

?l CC GAON KK LWM.PAS MWL.C WWW.C
?L CCC GDT.C L0.PAS LX.C MWL.PAS XO.PAS
1 CY.C GGG.C LHB.PAS LX.PAS MYD22.PAS XJ.PAS
A.C D.C H LL LY01.PAS NN XX.C
A.PAS DD.PAS H.PAS LLL.C LY011.PAS PLM.PAS XXX
AAA DDD HH.PAS LLL.PAS LY02.PAS PP.PAS XXX.C
BBB G.C HUANG.PAS LWD.PAS LY022.PAS TTY4 YXD.C
BHL.PAS G1.C HUANG1.PAS LWD46.PAS LY023.PAS WML.PAS ZWG.C
BLK.C GAO.C HUANG2 LWM.C LY03.PAS WW ZWG.PAS
BLK.PAS GAO1.C JHB.PAS LWM.LST LY04.PAS WWW

```

# EX HUANG.PAS

(编辑名为 HUANG.PAS 的文件)

"HUANG.PAS" 8 LINES, 110 CHARACTERS

```

:1, $
PROGRAM ABC(INPUT, OUTPUT);
  VAR A, B, C :INTEGER;
  BEGIN
    READLN(A);
    READLN(B);
    C:=A+B;
    PRINT C
  END.
:7
PRINT C

```

```

:D
  END.
:
  END.
:I
  WRITELN C
.
:1, $
PROGRAM ABC (INPUT, OUTPUT);
  VAR A, B, C :INTEGER;
  BEGIN
    READLN(A);
    READLN(B);
    C:=A+B;
    WRITELN C
  END.
:W
"HUANG.PAS" 8 LINES, 112 CHARACTERS
:Q
# PC HUANG.PAS                                (编译该文件)
MC68000 PASCAL COMPILER V2.0    22-AUG-83
(C) COPYRIGHT 1983 SILICON VALLEY SOFTWARE, INC.
  7. WRITELN C<<<<<
***** ERROR NUMBER 20 IN LINE 7 OF FILE HUANG.PAS*****
1 ERRORS. 8 LINES. FILE HUANG.PAS
# EX HUANG.PAS                                (再次编辑该文件)
"HUANG.PAS" 8 LINES, 112 CHARACTERS
:7
  WRITELNC
:D
  END.
:I
  WRITELN(C)
.
:W
"HUANG.PAS" 8 LINES, 113 CHARACTERS
:Q
# PC HUANG.PAS                                (再次编译该文件)
MC68000 PASCAL COMPILER V2.0    22-AUG-83
(C) COPYRIGHT 1983 SILICON VALLEY SOFTWARE, INC.
0 ERRORS. 8 LINES. FILE HUANG.PAS
MC68000 CODE GENERATOR V2.0    22-AUG-83
(C) COPYRIGHT 1983 SILICON VALLEY SOFTWARE, INC.
MC68000 UNIX OBJECT CODE FORMATTER V2.0    22-AUG-83

```

} 编辑过程

} 该文件有一个错误,在第七行

} 编辑过程



```

(C) COPYRIGHT 1983 SILICON VALLEY SOFTWARE, INC.
LD: FILE /LIB/CRT0.O NOT FOUND
# A.OUT                (运行该文件)
12345
45678
# MAIL ROOT           (向 ROOT 用户通讯)
THANK YOU FOR YOUR HELP.
# LOGOUT              (退出系统)
DASTRA SYSTEMS 68000 MULTIBOX WITH UNISOFT UNIX

```

值得注意的是，人-机界面的实现，即 JCL 程序或键盘命令(和键盘命令程序)的解释和执行，从严格意义上说，并不是操作系统的功能，而是操作系统的外壳(在 UNIX 系统中叫 Shell)的功能。这个外壳也是一个众多程序模块的集合。JCL 中的语句或键盘命令所指定的具体功能都是由外壳中的相应程序模块实现的。这些程序模块是运行在用户态下的，它们是系统软件的一部分。当然，这些程序模块在运行过程中，也可以通过系统调用指令调用操作系统中的程序模块以获得操作系统的支持和服务。

那么，通常所说的操作系统包不包括其外壳？这就涉及到对操作系统的两种理解：狭义的和广义的。从广义上说，操作系统包括状态界面以内的程序以及外壳的全部程序，从狭义上说，操作系统仅指状态界面以内的程序，即运行在核心态下的程序。本书取后一种理解，即狭义理解。请读者注意这一点。在随后各章里，我们只讲述状态界面以内的操作系统的基本概念，基本功能和实现细节，基本上不涉及其外壳的内容，除非是在非常必要时。

## 1.4 操作系统的特征

现代操作系统有以下四个主要特征。

第一，并发 (concurrency)。

从最一般的意义上讲，所谓并发就是多个事件的同时发生。就计算机而言，并发是指在一段时间内，有多道程序在宏观上同时运行。这样的系统叫并发系统 (concurrent system)。显然，并发和多道是同一事情的两个方面，正是由于多道程序设计才导致多个程序的并发执行。“宏观上”同时运行是什么意思？这是因为多数计算机系统一般只有一台处理机，于是，在任一指定时刻，只能有一道程序在真正运行，其它参与并发执行的程序就只能说是在宏观上处于运行状态了，即它们都是处在已经开始运行和尚未结束运行的过程中。

现代操作系统是并发系统的管理机构，其本身也是并发执行的，是与用户程序以及其它用户态程序一起并发执行的。程序的并发执行带来许多程序串行执行时所没有的新问题，这便导致了操作系统的复杂化。并发是操作系统的第一个重要特征。

第二，虚拟 (virtual)。

虚拟的本质含义，是把物理上的一个变成逻辑上的多个。例如，把一台物理 CPU 变成多台逻辑上独立的 CPU，把一个统一编址的物理存贮器变成多个逻辑上独立编址的存贮