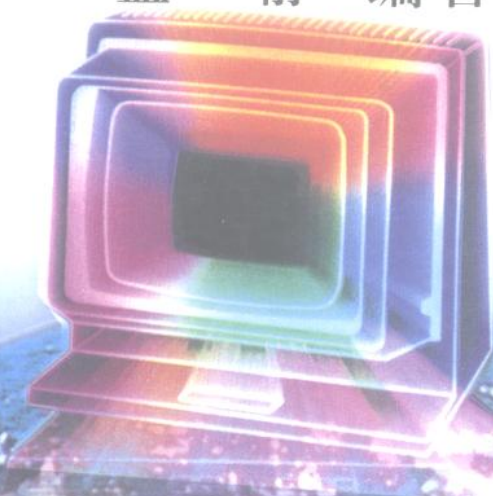


计算机操作系统教程

孟静 编著

计算机 操作系统 教程

孟 静 编著



中国人民大学出版社

316
59

出版社

410268

计算机操作系统教程

孟 静 编著

中国人民大学出版社

图书在版编目 (CIP) 数据

410208

计算机操作系统教程/孟静编著.

北京: 中国人民大学出版社, 1997/TP·1

ISBN 7-300-02337-1/TP·1

I. 计…

I. 孟…

II. 操作系统 (软件)-教材

IV. TP316

中国版本图书馆 CIP 数据核字 (97) 第 01360 号

计算机操作系统教程

孟 静 编著

出版 发行: 中国人民大学出版社

(北京海淀路 175 号 邮编 100872)

经 销: 新华书店

印 刷: 北京市丰台区印刷厂

开本: 787×1092 毫米 1/16 印张: 17

1997 年 11 月第 1 版 1997 年 11 月第 1 次印刷

字数: 397 000

定价: 22.00 元

(图书出现印装问题, 本社负责调换)



内 容 提 要

JS212

121

操作系统是计算机中最重要的系统软件。本书介绍其工作原理，包括其功能、外部使用接口、内部工作过程与结构、有关概念、技术、理论与实际系统。全书共八章，依次为总论、处理机管理和进程管理、内存管理、外存管理与文件系统、I/O设备管理、进程通信、实际系统介绍、分布式操作系统。本书是作者在大学计算机系多年教学基础上写成的，可作为计算机及其应用专业教材，还可作为计算机工作者、计算机应用人员、计算机爱好者、及各种计算机考试应试者的自学教材和参考用书。

前 言

操作系统是计算机中最重要的一种系统软件。本书介绍它的工作原理、评价技术和实际系统情况,旨在使读者了解它的外部接口功能和内部过程结构,以及相关的概念、技术、理论,从而获得对整个计算机系统原理和相关课程的理解基础,获得对各种实际操作系统的使用、维护的快速、高效掌握能力,获得对操作系统的评价、分析、选择能力和设计、实现、移植、改造的初步能力。

本书共八章:第一章介绍操作系统是什么,其作用和重要性,如何工作,其概念和技术的发展简史,操作系统课程内容,操作系统接口及其设置原则;第二章到第六章详细研究操作系统的内部工作原理,分别为CPU和进程管理、内存管理、文件系统、设备管理、进程通信;第七章介绍各种实际操作系统;第八章简单介绍分布式操作系统原理。

本书可作为计算机及其应用专业本科二三年级课程教材,要求先修汇编、硬件、高级语言等课程。还可作为计算机工作者、计算机爱好者、各种计算机考试的应试者的自学教材和参考用书。

本书针对系统化、教材化、标准化、全面新颖等方面做了大量努力,力求使本书在组织思路体现出操作系统各部分的内在统一本质的规律主线,在内容的选取上注意全面、新颖、基础、典型、有针对性,在讲述上力求由浅入深、深入浅出,并通过大量数字、图表、实例的使用,来增强形象和翔实程度,力求注意术语的标准性和定义的严格性。

时间仓促,水平有限,竭诚欢迎读者提供宝贵意见。作者通讯地址:100872,北京,中国人民大学信息学院孟静。

目 录

第一章	总论	1
	1.1 操作系统简介	1
	1.2 操作系统的第一个工作——启动和结束程序	7
	1.3 操作系统的第二个工作——用户程序对操作系统的调用.....	16
	1.4 操作系统的第三个工作——为常用基本操作提供现成程序.....	23
	1.5 操作系统的第四个工作——解决效率和安全问题.....	23
	1.6 操作系统的各个方面总结.....	29
	1.7 操作系统发展简史.....	33
	1.8 小结.....	52
第二章	CPU 管理和进程管理	53
	2.1 CPU 管理概述	53
	2.2 进程模型.....	54
	2.3 小结.....	60
第三章	内存管理	62
	3.1 内存管理概述.....	62
	3.2 无管理模式和单一分区模式.....	75
	3.3 固定分区模式.....	79
	3.4 可变分区模式.....	82
	3.5 页模式.....	89
	3.6 段模式	100
	3.7 段页模式	109
	3.8 小结	111
第四章	外存管理与文件系统	114
	4.1 文件系统概述	115
	4.2 文件系统的基本工作过程与数据流程	123
	4.3 文件系统的基本结构与算法	125
	4.4 性能改善与功能增强	136
	4.5 小结	146
第五章	I/O 设备管理	147
	5.1 I/O 设备硬件资源及其原始使用方式的特点	149
	5.2 I/O 设备管理功能的任务、结构、过程	168
	5.3 小结	178

第六章	进程通信	181
	6.1 背景	181
	6.2 临界区问题	183
	6.3 同步硬件	189
	6.4 信号量	190
	6.5 经典的进程协同问题	193
	6.6 进程间通信	196
	6.7 小结	201
第七章	实际系统介绍	203
	7.1 UNIX 概述	203
	7.2 MS-DOS 概述	209
	7.3 其他传统操作系统	214
	7.4 小结	218
第八章	分布式操作系统	219
	8.1 目标	219
	8.2 硬件概念	221
	8.3 软件概念	227
	8.4 实现策略	236
	8.5 线程	242
	8.6 分布式文件系统	254
	8.7 Mach 概述	260
	8.8 小结	263

第一章 总 论

读者肯定是带着许多问题开始读这本书的，这些问题可以概括为下面六个：

1. What—操作系统是什么？做什么？
2. Why—为什么需要操作系统（其作用和重要性）？
3. How—操作系统如何工作？如何使用操作系统？
4. What—本书学习什么内容？
5. Why—学了有何用处？
6. How—如何学？

本书第一章就来解答这六个问题。当然，其中第三个问题只是大致的解答，其系统深入的答案构成全书主要内容和目标。其余五个问题的答案也需要在全书学习过程中逐步加深了解和理解。实际上，在学习任何一门知识前，我们都有类似的这六个问题（简称6W或3W），对任何一门知识的学习也就是对这六个问题的解答，当然主要内容也是第三个问题。

本章共分八节。1.1节讲述操作系统是什么，做什么，为什么需要操作系统（即其作用和重要性），操作系统是一个多大规模的软件；1.2节到1.5节介绍操作系统大致是如何工作的；1.6节从前4节中总结出操作系统的接口、过程、结构、特点、类型，并介绍操作系统的各种别名、比方、观点；1.7节简介操作系统发展历史；1.8节为本章小结。本章重点是1.1节到1.5节。

1.1 操作系统简介

操作系统是什么？做什么？

也许你说不出来操作系统是什么，但你肯定用过操作系统，至少一个——如果你用过计算机的话。那么我们先吧“操作系统是什么”这个问题放一放，先来看一看你用过的那（几）个操作系统。

1.1.1 你所用过的操作系统

我们已经学过一些计算机课程，用过或听说过一些程序或软件。下面这些软件中哪些是操作系统？

TETRIS（搭积木） DOS WPS Windows Turbo-C WORD FOXPRO

UNIX 你自己编的一个C语言源程序 Turbo-Assembler vi

虽然你不能概括地说出操作系统的定义，但你知道上述软件中只有DOS、Windows和UNIX是操作系统，其余的软件都是用户程序和其他系统软件（表1.1）。实际存在的操作系统有许多种（如果包括已不再使用的，大约有几百种），其中目前最常用的就是这三种。

其他常用的还有 Netware (网络操作系统)、OS/2 (IBM 为其高档个人计算机设计的操作系统)、OS/400 (IBM 小型机上的操作系统)、Macintosh (苹果计算机上的操作系统) 等等。

表 1.1 下述软件中哪些是操作系统

软件名	说明
TETRIS (搭积木)	游戏软件。属于应用软件。
DOS	操作系统。是 IBM 个人计算机系列上最常用的操作系统。
WPS	编辑软件。是目前国内在 DOS 下最常用的编辑软件。
Windows	操作系统。是 IBM 个人计算机系列上最常用的窗口多任务操作系统。
Turbo-C	编译程序。属于系统软件。
WORD	编辑软件。是 Windows 下的编辑软件。
FOXPRO	数据库管理软件。
UNIX	操作系统。是多用户计算机上最常用的操作系统。
你自己编的一个 C 语言源程序	严格来说, 这不算程序, 只能算数据。这个源程序经编译连接后产生的才是程序——可执行目标程序。
Turbo-Assembler	汇编程序。属于系统软件。
vi	编辑软件。是 UNIX 下的编辑软件。

根据你的使用经验 (虽然可能是有限的), 你能说一说你所用过的 DOS、Windows、UNIX 这些操作系统做些什么事吗?

- 我们用过 dir、copy、cd、format 等等, 它们有一个共同的名称: DOS 命令。这些命令都属于 DOS 这个操作系统的功能。同样, 类似的 UNIX 命令也属于 UNIX 操作系统的功能。在 Windows 中, 类似的功能是通过鼠标来体现的。
- 你要用计算机做任何事, 都需先运行某个程序或某条命令, 运行程序的最典型方法是在 DOS 命令提示符后打入程序名。而开机后如果不出现命令提示符, 你就什么命令也打不了, 什么程序也运行不了, 也就什么事都做不了。这个命令提示符, 就是 DOS 显示的。显示命令提示符、接受命令行、启动执行相应的程序, 也属于 DOS 的功能。UNIX 中也是这样的。在 Windows 中, 类似的功能是通过鼠标和菜单命令体现的。
- 我们在用汇编语言编程序时, 都用过 INT 语句, 通常使用这些语句的原因都是要做一些 I/O 工作, 比如文件读写或打印等。INT 语句实际上是一种特殊的调用语句, 它所调用的都是 DOS 的内部功能。INT 语句是 DOS 向所有程序提供的调用接口或调用方式。在 Windows 和 UNIX 中也有类似的内部功能和相应的对外调用接口。
- 在 Windows 中, 我们可以同时运行多个程序 (即多任务), 比如我们在等着一个文件

打印完成的同时可以编辑另一个文件,这就提高了用户效率和机器利用率。多任务方式是 Windows 提供的。DOS 中就没有类似的功能,因此我们在用 DOS 打印一个文件时就只能等待。而 UNIX 可以让多个用户同时使用一个计算机,这就更进一步提高了机器利用率。

.....

至此,你已有了一个印象:操作系统的功能很多、很杂、很散,很难概括。好像什么事都做。但操作系统又不可能什么事都做,否则还要别的那么多软件干什么。操作系统不做什么事?你能列举出来吗?

- 操作系统不做天气预报,这是由专门的天气预报软件来做的。
- 操作系统不做房屋设计,这是由专门的建筑 CAD 软件来做的。
- 操作系统不是编译程序,你用什么语言编的源程序就用相应语言的编译程序。

.....

总之,操作系统不直接解决最终具体应用问题,也不负责编译源程序,等等。

那么操作系统到底做什么,不做什么?它的准确的功能范围?它到底是一个怎样的软件?有没有内在的统一的本质?

1.1.2 操作系统是什么?做什么

操作系统为你完成所有硬件相关、应用无关的工作,以给你方便、效率、安全。

操作系统 (operating system, 简称 OS) 是计算机中最重要的系统软件,是这样的一组系统程序的集成:这些系统程序在用户对计算机的使用中,即在用户程序中和用户操作中,负责完成所有与硬件因素相关的(硬件相关)、和任何用户共需的(应用无关)基本使用工作,并解决这些基本使用工作中的效率和安全问题,为使用户(操作和上层程序)能方便、高效、安全地使用计算机硬件和其中存储的信息,而从最底层统一提供所有通用的帮助和管理(图 1.1)。

那么具体来说,在用户操作和用户程序中,哪些工作内容是硬件相关和应用无关的?主要是以下四方面工作。也就是说,操作系统是完成以下工作的诸系统程序的集成(表 1.2):

(1) 负责启动执行每个程序,并负责完成每个程序的结束处理工作。使每个程序可以很方便灵活地开始执行和中止。

(2) 在任何用户程序中,负责完成程序在使用硬件及硬件中存放的信息时的所有硬件相关和应用无关的基本使用工作,每当用户程序中涉及这些工作时,就通过调用操作系统来完成(这种调用称为系统调用),从而为程序中对计算机各种硬件和硬件中存放的信息的方便使用提供统一帮助和管理。

(3) 为用户对硬件及所存放信息的基本操作提供现成的实用程序和相应的管理,以使这些操作系统能很方便和有效地进行和完成。这里所说的基本操作是指任何应用或开发背

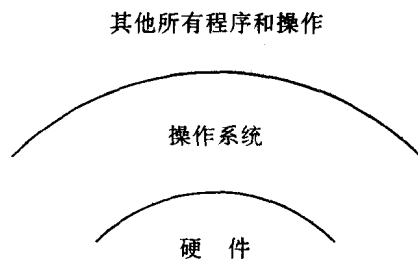


图1.1 操作系统在计算机系统中的地位

景下都通用的和普遍需要、经常发生的操作（如拷贝文件，删除文件，显示磁盘目录内容，显示文件内容，格式化磁盘等）。

(4) 改善上述三方面基本使用的效率和安全问题，使计算机系统的各部分和整个计算机系统得到高效安全地利用。

以上四个工作似乎互无关联，但它们其实都具有这样的共性：硬件相关和应用无关（见表 1.2，详见 1.2 节—1.5 节）。反过来说，计算机使用中的硬件相关和应用无关工作就是这四个方面。

表 1.2 操作系统完成的四方面工作和它们的共性

操作系统负责的所有工作有以下四方面	这些工作的具体内容和完成方式举例。	这些工作是硬件相关的			这些工作是应用无关的			如果没有操作系统来完成这些工作，用户操作和用户程序会怎么样？	由操作系统完成这些工作后，用户操作和用户程序会怎么样？
		实现该工作的过程代码和硬件因素密切相关。（即需要设置与测试、使用物理地址、设备接口寄存器等。）	硬件相关必然复杂繁琐代码量大。	硬件相关的工作，其实现代码不通用，当硬件变化时，需要重新编制或编译相应代码。	任何应用（使用）都需要该工作。	在不同应用中，该工作的过程都是相同的。	与具体应用无直接关系。（即与用户所关心的应用目标无直接关系。）		
负责启动每个程序执行，并负责每个程序的结束处理工作。	例如：通常启动一个程序只需在命令提示符后打入程序名即可。此外，操作系统还提供了批exec等多种启动方式（详见1.2节）	启动和结束过程涉及到对IO设备、外存、内存的使用与处理，这些使用与处理涉及物理地址与物理地址接口寄存器。	一个程序的启动工作和结束处理工作所需的代码要与外存接口寄存器、内外存物理地址打交道，其代码量可能比程序主体代码量还大。	例如，如果程序从软盘移到磁带上，其启动代码就需要重新编制。	用计算机做任何事都需运行相应程序，而每个程序都需要启动过程和结束处理工作。	所有程序的启动过程都是雷同的（就同一种启动方式而言，详见1.2节）。所有程序的结束处理工作都是雷同的（就同一种结束原因而言，详见1.2节）。	例如：一个财务软件，其启动过程和结束处理工作同财务问题没有任何关系。	所有程序都必须自启动程序，即自包含引导装入代码（详见1.2节），所有程序的启动操作都相当于一关开机操作——麻烦、费时、不灵活。	用户程序中不必包含启动自己执行的装入引导代码，不必包含与本程序无关的那些异常结束处理代码，所有结束出口处只需一条“返回系统”指令。从而用户不必考虑程序启动与结束问题，了解相关硬件细节，频繁重复编制相应代码。

续前表

2	<p>每当用户程序中使用 IO 设备, 所存储的信息、内存等, 其中的硬件相关和应用工作都通过操作系统来完成。</p>	<p>例如: 在个人计算机上用户程序中需打印字符时, 只需一条 int 指令调用操作系统即可。</p>	<p>例如: 打印一个字符的过程中需设置和查看打印机接口的每一位, 需要了解和使用打印机的接口 IO 物理地址。</p>	<p>用户程序对 IO 设备的使用所涉及的代码经常占程序主体代码的大部分。这些代码中多处涉及接口寄存器、物理地址等复杂繁琐的硬件细节。</p>	<p>例如: 计算机系统换了一台型号的打印机, 则程序中涉及打印输出的那些代码都要重新编制。</p>	<p>任何程序的执行过程中都需要使用内存、IO 设备、外存信息。只要使用 IO 设备, 就要与这些设备的接口寄存器打交道; 只要使用内存, 就要与物理地址打交道。</p>	<p>例如: 打印字符 A 与打印字符 E 的过程是一样的, 只不过处理的数据 (A 与 E) 不同。</p>	<p>财务软件中, 打印一个数据的过程与财务问题上无任何联系。</p>	<p>所有程序中都包含硬件相关代码。程序员必须了解相应硬件细节与编制大量复杂、繁琐代码。当硬件变化时, 必须重新编制程序。</p>	<p>用户程序中不包含硬件相关代码, 即不包含与接口寄存器、物理地址打交道的那些代码。用户不必了解相关硬件细节知识和编制相应的复杂繁琐代码。</p>
3	<p>为用户的常用基本操作提供现成程序。这里的常用基本操作是指任何应用都普遍需要和通用的操作。</p>	<p>例如: DOS 中为文件拷贝提供的现成程序有 copy、xcopy 等。用户需要拷贝时只需运行这些现成程序, 不用自行编程。</p>	<p>这些操作大量涉及对外存、IO 设备的使用。</p>	<p>同上</p>	<p>同上</p>	<p>不管是财务应用还是人口统计应用, 任何应用都需要进行这些工作。</p>	<p>拷贝一个财务总帐文件和拷贝一个人口清单文件, 其拷贝过程都是相同的, 只不过数据不同。</p>	<p>例如: 拷贝的过程与拷贝的数据无关。</p>	<p>用户要为这些常用操作自行编制程序。</p>	<p>用户不必为这些常用操作亲自编程。</p>
4	<p>解决以上工作中的安全性和效率问题。</p>	<p>例如: 操作系统提供多任务方式可以提高效率。</p>	<p>例如: 多任务方式的实现过程中需管理内存和 IO 设备</p>	<p>例如: 提供多用户方式的操作系统比不提供多用户方式的操作系统要大得多, 复杂的多。</p>	<p>同上</p>	<p>每个用户都希望提高效率, 每个机器主人都希望提高机器利用率。</p>				

何谓硬件相关和应用无关？

在这里，我们说一个工作是硬件相关的，是指这个工作的实现代码中包含（内外存及设备的）物理地址，包含对设备接口寄存器和设备接口缓冲区的读写，等等。硬件相关的代码必然随硬件的变化而变化。这样的硬件变化包括内外存物理存储空间大小的变化，程序和数据在内外存物理存储空间中的存放位置的变化，设备数量和类型的变化，等等（但不包括 CPU 指令集的变化，否则所有可执行目标程序都是硬件相关的了）。我们说操作系统承担硬件相关工作，使其上程序都是硬件无关的，是指当程序所使用的硬件发生变化（除 CPU 指令集以外）时，程序不必改变，人的操作更不必改变。

一个工作是应用无关的，是指不管用计算机来做什么，不管在计算机上运行什么程序，只要使用相应硬件或相应信息就要涉及到的工作，是用户共需的，且工作过程都是相同的，是有共性可循的，却又与应用本身的问题没有直接关系的。

操作系统为用户操作和用户程序完成所有硬件相关和应用无关的工作，目的和益处何在？硬件相关，必然意味着复杂繁琐、大代码量（大到经常占代码比例分量的大部分）、代码不通用和变化大，需要用户投入大量的精力来设计实现和维护修改，以及了解相应的大量硬件细节知识，有必要统一管理，使用户摆脱负担。应用无关，就意味着更有必要统一管理（因为普遍和频繁涉及，以及与具体应用无直接关系）和能够统一管理（因为工作过程都是相同的）。越是对计算机硬件和信息的使用中的底层的、基本的使用工作，越具有硬件相关和应用无关的特点，对用户和系统的方便、效率、安全影响越大，越需要、值得、可能由操作系统来完成这些工作并解决其中的效率和安全问题，从而使用户程序成为硬件无关的（即独立于硬件，不包括硬件相关的代码，当硬件改变时程序不必改变），使用户不必考虑这些底层基本使用过程和了解相关的硬件细节知识，避免频繁重复编制（或编译）与应用问题本身关系不大的大量复杂繁琐代码，专心于应用本身，更好地达到最终的具体应用要求和目标。总之，操作系统为保证用户操作和用户程序最终使用的方便、效率、安全，而承担了所有硬件相关和应用无关的工作，从最底层提供统一帮助和管理。

1.1.3 操作系统有多大？有多少个？有多重要

从操作系统的上述四个工作可以看出，操作系统必然是个庞然大物。实际正是如此。

想想你用过的那（几）个操作系统。在硬盘上的 DOS 目录下，通常有上百或几百个文件。DOS 6.2 需要四张高密软盘；UNIX 需要十几张、几十张高密软盘；Windows 3.1 中文版需要 8 张高密软盘；Windows 95 需要几十或上百张高密软盘，以至通常用光盘安装。

在 IBM 个人计算机上，经常运行的操作系统有 DOS、Windows、UNIX 等等。UNIX 操作系统，可以运行在 SUN 工作站上（SUN OS、Solaris 等等），也可以运行在 IBM 个人计算机上（XENIX、SCO UNIX 等等），可以运行在众多不同机器上。

从 1955 年开始出现操作系统到现在的 40 多年中，在各种机器上实际运行的操作系统有过几百个，它们中最小的有上千万行代码，最大的达几十万行（30 万行 PL/1）甚至几百万行（汇编，几千名程序员编制）代码，甚至更多。它们在实际运行时占用了大量的机器时空资源：它们占用了 20%—45% 的 CPU 时间，本身占内存空间最大达 4 兆—6 兆，占用了大量的外存空间，等等。这些数字都说明了操作系统的规模巨大。另外，不同机器上的操

作系统一般是不同的，同一种机器上可以有不同的操作系统，同一种操作系统可以有在不同机器上的不同版本或变种，同一机器同一操作系统也有功能升级导致的不同版本（基本上每一到五年推出一次新版），这些因素又都导致了操作系统的数量众多。

这些众多庞然大物的产生，不可避免地由用户各自开发，发展到厂家提供，从免费配给或提供，发展到商品化和独立注册，它们的产生（实现）与维护支持使厂家耗费了巨大的财力：1975年估计每个厂家每个操作系统每年花费200万美元~1000万美元，而当时一个厂家同时支持13个操作系统（5个产品线）、每个操作系统要支持2—3版本的情况并不少见；1981年估计过去25年中所有厂家与用户的OS总费用几百亿美元。这样的庞然大物的使用和维护，也给用户（包括机器拥有者和使用者）带来了烦难的学习负担和管理负担，用户机构每年要付出大笔的培训费用和维护费用（尤其在系统升级时）。

是什么导致人们花费如此多的财力和时间在操作系统上？因为操作系统带给用户方便、效率和安全（从而带给厂家效益），虽然为它付出的代价是很大的，但与它带来的方便、效率、安全、效益比，这种代价是值得的。所以操作系统虽是令人头痛的庞然大物，但它却是每台计算机都离不了的、凡与计算机打交道的每个人（不管是厂家还是用户）都离不了的庞然大物。今天，从最小的个人计算机到最大的巨型机，操作系统可以说是无处不在，缺之不可。用户很少使用一台没有操作系统的机器，或说很少不在操作系统的帮助下使用计算机，以至到了对操作系统反而熟视无睹的地步。计算机产生初期没有操作系统时的对计算机使用的艰繁低效已经成了遥远陌生的事情。

在下面的1.2到1.5节中，我们来详细地分别看一下操作系统的上述四个工作，了解它们的具体内容，它们为什么是硬件相关和应用无关的，它们的内部实现的大致过程，用户如何使用它们，然后在1.6节中总结出操作系统的接口、过程、结构、特点、类型，并简单介绍操作系统的各种别名、比方、观点。

1.2 操作系统的第一个工作——启动和结束程序

我们首先来看程序如何开始执行。从硬件知识可知，任一（可执行目标）程序的开始执行有两个前提条件：（1）程序在内存（即程序已装入内存）；（2）程序计数器PC被置为程序在内存的入口地址。一旦CPU中的程序计数器PC被置为该程序在内存的首地址，该程序就开始运行了。那么一个程序是由谁来装入和启动它执行的呢？关于“启动一个程序执行”的申请又是怎样提出的呢？按照这两个问题的答案，形成了五种启动方式，1.2.1节到1.2.5节就分别来讨论这五种方式。

1.2.1 程序的第一种启动方式——命令方式

在IBM个人计算机上，如果你要启动执行一个程序，通常你怎么做？当然是在命令提示符后打程序名。比如要启动执行编辑软件WPS，就在命令提示符后打WPS并按回车。

以命令方式来启动程序，就是在命令提示符后打入程序名，就可以启动程序执行。命令提示符是操作系统在屏幕上向用户提供的一种提示标志，它的出现就表示“你可以在命令提示符后打入程序名来启动下一个程序的执行了”。DOS的命令提示符的最常见形式是

C), UNIX 的命令提示符的最常见形式是 \$ 或 % 等。通常把在命令提示符后打入的程序名和其参数 (到回车为止) 称为命令行或命令。任一命令行 (即任一条命令) 的本质都是申请某一程序执行。命令方式是启动程序运行的最主要方式, 在命令方式下的用户工作过程如图 1.2 (1) 所示。

通常把操作系统中实现命令方式 (及批方式, 见下文) 的那部分程序称为命令解释程序、命令解释器、作业管理程序或作业控制程序、SHELL。其内部工作过程如图 1.2 (2) 所示。作业与作业步的概念如图 1.3 所示。DOS 的命令解释器是 COMMAND.COM, 可以在启动盘根目录下看到这个文件。UNIX 的命令解释器惯称为 SHELL, 有 BSH、CSH、KSH 等多种。在 Windows 中, 命令方式是以窗口菜单显示和鼠标操作来体现的, 这种窗口界面形象生动, 操作简单而有规律, 用户不必记忆命令, 现在已普遍应用。

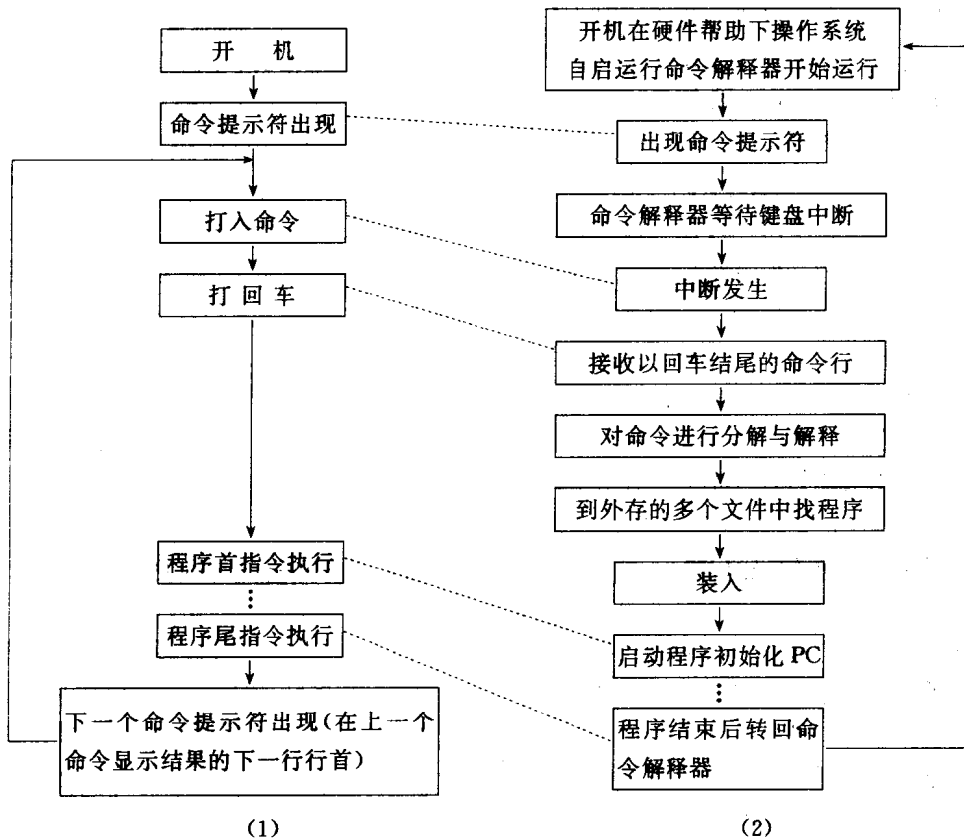


图 1.2 命令方式下的申请执行和装入启动过程

如果通过命令行所启动的程序是操作系统提供的现成程序 (即操作系统的第三个工作, 见 1.1.2 节), 则该程序、该命令 (行) 称为操作系统命令。

在 DOS 中, 当在命令提示符下打入一个命令而启动执行一个程序后, 怎么知道该程序运行完了没有? 是从下一个命令提示符的出现知道的。因此 DOS 命令提示符有两重含义: 一是标志着上一个程序的结束 (也可以说上一条命令的结束), 一是标志着可以启动运行下一个程序 (即可以打下一条命令)。而在 UNIX 中, 某些情况下也是这样的, 某些情况下则

只有第二重含义（例如后台作业）。

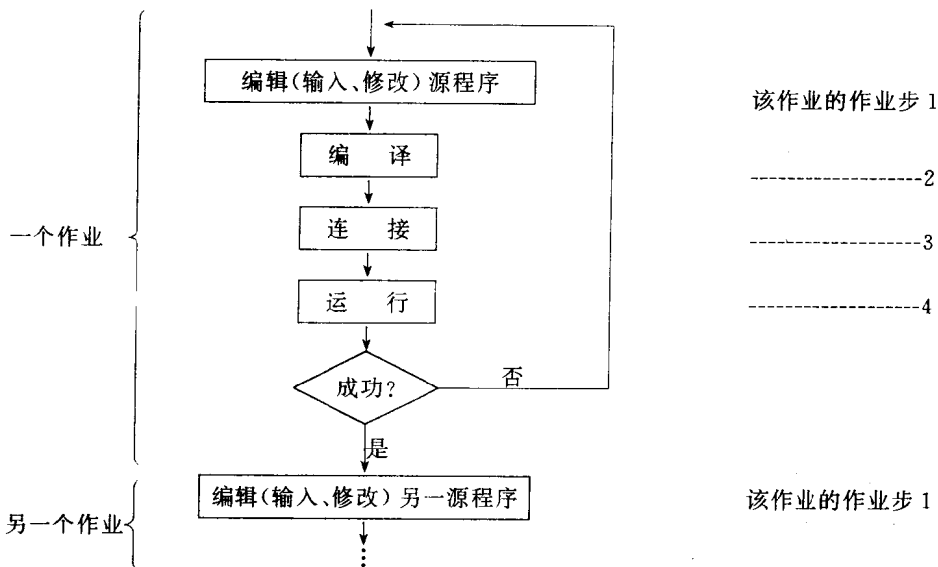


图 1.3 典型的作业与作业步

思考题：既然开机通电状态任一时刻都有程序在运行（指令在执行），那么在 PC 上当 C) 后为空（即光标紧挨在 C) 后闪烁）而操作者走开已久未归时，机器上是在运行程序吗？什么程序？

答案：从图 1.2 可知，此时是负责命令的接受解释启动执行的命令解释器在运行。既然命令解释器负责接受、解释、执行命令行，而用户何时打命令是不确定的，因此为了在用户一旦打命令后尽快地作出响应，命令解释器必须随时随刻准备接受命令，尤其在命令提示符出现后（因为命令提示符的出现含义就是通知用户可以打命令）。因此，当 PC 上 C) 后为空地，是命令解释器正在运行——正在等待用户打命令（是通过一个循环来等待的，这个循环不停地检测是否有键盘中断，即是否有用户开始打命令）。

思考题：当你为了以参数 P 执行程序 A，而在 C) 后打入一条命令“A P 回车”，从你开始打命令到打下了命令尾的回车，直到程序 A 内的第一条指令执行前，计算机在做些什么？

思考题：（假定在 PC 机上）从程序的最后一条指令执行完到下一 C) 出现之前，计算机在做些什么？

1.2.2 程序的第二种启动方式——批方式

例 1：在 IBM 个人计算机上，用户经常使用 .BAT 文件来启动一个中文环境。也就是说，通常你为启动一个中文环境所打的那条命令，所启动执行的实际上是一个 .BAT 文件。例如要启动 UC DOS 这个中文环境，通常在命令提示符后打 UC DOS 回车或 UP 回车，即可成功启动 UC DOS。这条命令所启动执行的实际上是 UC DOS.BAT 或 UP.BAT 这个文件。该文件的内容通常为：


```
c: \ucdos\rd16 %1
c: \ucdos\knl %2
c: \ucdos\py
c: \ucdos\wb
```

类似地，要启动 SPDOS 这个中文环境，通常在命令提示符后打 SPDOS 回车，即可成功启动 SPDOS。这条命令所启动执行的实际上是 SPDOS.BAT 这个文件。该文件的内容通常为：

```
c:
cd\spdos
splib/1
spdos
wbx
```

通常启动一个中文环境需要多条命令：首先要装入字库（如上两个文件中的 rd16 和 splib），然后启动主程序（如上两个文件中的 knl 和 spdos），最后要装入启动所需的汉字输入法软件（如上两个文件中的 py（拼音）、wb（五笔）、wbx（五笔）等）。每次启动中文环境都要打这么多条命令，很麻烦，而且一般用户不知道要打这些命令，也不知道怎么打，需要了解相应知识。把这些命令放在一个 .BAT 文件中，只需打该文件的名字就可启动中文环境，简单方便而且不必要求用户了解那么多知识。

例 2：对 IBM 个人计算机略为熟悉的人都知道 AUTOEXEC.BAT。该文件直接放在启动盘的根目录下，在开机时第一个命令提示符出现前自动执行该文件中的命令。该文件的典型内容为：

```
path c:\dos
prompt $p$g
```

该文件的内容可以由用户修改，用户可以把在每次开机时都需要执行的命令放在该文件中。

批方式是指将若干条命令（请注意命令即启动程序的请求）放在一个文件（批文件）中通知计算机，当在命令提示符后打入该文件的名字（在 DOS 中打不打扩展名均可）时，就由计算机自动连续执行该文件中的这组命令。

批方式的工作过程见图 1.4。

与图 1.3 中作业与作业步的概念相对应，一个批文件中的所有命令合称为一个作业，其中的每条命令称为一个作业步。

批方式的优点是：（1）程序可以连续执行，不须人工干涉；（2）命令可以设计成有控制结构的形式，称为作业控制语言，使程序可以组合执行。

批文件的名字和位置是用户自定的，但扩展名在有些操作系统中是固定的（例如在 DOS 下批文件的扩展名必须均为 .BAT）。要启动执行批文件，就在命令提示符后打入该批文件的名字（在 DOS 中打不打扩展名均可）。但有一种批文件的名字和位置是固定的，可以不经用户启动而自动执行。在 DOS 下名为 AUTOEXEC.BAT，必须直接放在启动盘根目录下，在 DOS 启动时自动执行；在 UNIX 下名为 .profile、.cshrc、.login 等，只能放在用户主目录下，在注册或 SHELL 启动时自动执行。