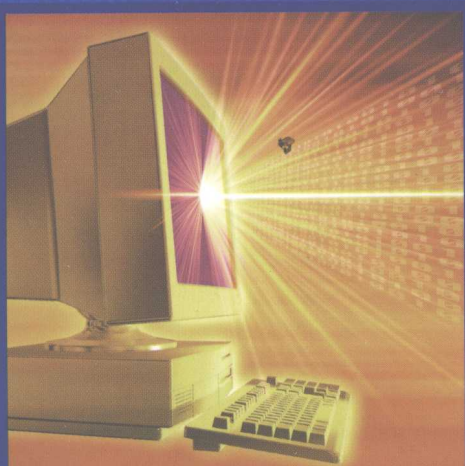




高职高专“十一五”规划示范教材



主 编 张 强

副主编 万 蓉 王 奇 柴 晟

程序设计基础



北京航空航天大学出版社



高职高专“十一五”规划示范教材

程序设计基础

主 编 张 强
副主编 万 蓉 王 奇 柴 晟

北京航空航天大学出版社

内容简介

本书以实际案例的分析和求解为主线,以 C 语言作为描述工具,串联起程序设计的各个方面。同时为了培养学生对实际问题的分析、理解和求解的能力,本书将阐释求解问题的技术作为核心内容,逐步介绍问题的起源与分类和求解问题的方法与技术,并始终强调流程图、伪代码和 C 语言 3 种算法表示方式间的对比和转化,使学生通过本课程的学习真正形成“以问题为中心”的思想。

全书分为计算机与程序设计基础知识、程序设计的工具、程序设计方法与扩展和学习进阶等四个部分,可作为程序设计入门类教材,同时也适用于计算机及相关专业的专科层次学生,也可作为培训班教材及自学手册。

图书在版编目(CIP)数据

程序设计基础/张强主编. —北京:北京航空航天大学出版社,2008.1

ISBN 978-7-81124-283-6

I. 程… II. 张… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 192435 号

程序设计基础

主 编 张 强

副主编 万 蓉 王 奇 柴 晟

责任编辑 董 瑞

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:13.75 字数:352 千字

2008 年 1 月第 1 版 2008 年 1 月第 1 次印刷 印数:4 000 册

ISBN 978-7-81124-283-6 定价:22.00 元

前 言

“程序设计基础”是一门十分重要的基础课程。长期以来,该课程的教学模式都过于注重语句、语法和一些细节,基本上是以某门高级语言自身的体系为脉络展开的,没有把逻辑思维与编程解题的思路放在主导地位上,对如何分析问题和解决问题讲得不够,对学生编程的能力、上机解题的能力训练不够。这样就给后续课程的学习带来了困难。很多学生在学习这门课时感到枯燥难学,学过之后,不能用来解决实际问题。

本书以实际案例的分析和求解为主线,以C语言作为描述工具,串联起程序设计的各个方面。为了培养学生对实际问题的分析、理解和求解的能力,本书将阐释求解问题的技术作为核心内容,逐步介绍问题的起源与分类、求解问题的方法与技术,并始终强调在流程图、伪代码和C语言3种算法表示方式间的对比和转化,使学生通过本课程的学习真正形成“以问题为中心”的思想。

全书分为四个部分:

第一部分介绍计算机与程序设计基础知识,仅包含第1章。该部分从计算机的基本结构和工作原理出发,介绍了计算机程序及程序设计的一些基本概念,并强调了编写计算机程序的目的是解决实际问题。

第二部分介绍进行程序设计的工具,包括第2章和第3章。该部分介绍了进行程序设计所必须的设计工具和程序设计语言,同时介绍了C语言的开发环境。

第三部分介绍程序设计方法,包括第4章到第9章。该部分详细介绍了如何利用C语言进行程序设计,并通过对一个完整应用案例的分析和设计过程的介绍,引导读者对实际问题进行分析和求解,将各章节的知识和技巧应用于实际。

第四部分介绍一些扩展性知识和须进一步学习的内容,以使读者可以在本书所讲授内容的基础上,开阔眼界,进行深入学习。该部分包含第10章到第12章。

附录包括:ASCII码表、TC 2.0 关键字、TC 2.0 运算符、TC 2.0 库函数及TC 2.0 常见错误信息。

本书结构安排合理,课程内容精心策划,每一课的内容都依照难易程度做了合理分配。为了便于教学,在每一课的正文前面都配有学习目标导读,课后都配有思考与练习题。作为一种重要的程序设计语言和求解问题的工具,本书同时还介绍了C语言的一些语言特点和应用领域,以便学生进行深入学习。建议总课时数为64学时,其中实验部分24学时。

本书可作为程序设计入门类教材,适用于大中专院校、职业院校及培训学校的计算机与信息相关专业的教学。

本书由张强主编,其中张强编写第1章、第10章、第11章,银河编写第2章和第3章,陈蕾编写第4章至第6章,王奇编写第7章,万蓉编写第8章,柴晟编写第9章,罗传军编写第12章,其他参与本书编著、资料整理的人员有李明欣,在此对大家的辛勤工作表示衷心的感谢!虽然我们在编写本书的过程中倾注了大量心血,但恐百密之中仍有疏漏,恳请读者不吝指教,并欢迎广大专家和读者将好的思路和建议反馈给我们,以便修订时完善。

编 者

2007年8月

目 录

第1章 计算机与程序

1.1 计算机的基本结构与工作原理	1
1.1.1 计算机硬件的基本结构	1
1.1.2 计算机的工作原理	3
1.2 计算机程序	5
1.2.1 计算机程序的定义	5
1.2.2 计算机程序的执行过程	6
1.2.3 程序与程序设计语言	7
1.3 程序设计及其目标	8
1.3.1 为什么使用计算机	8
1.3.2 计算机可以做什么	9
1.3.3 问题求解过程	11
思考与练习	13

第2章 算法及其描述

2.1 算 法	14
2.1.1 什么是算法	14
2.1.2 算法的描述	15
2.1.3 算法设计举例	16
2.2 流程图	17
2.2.1 流程图及其分类	17
2.2.2 用流程图描述算法	18
2.2.3 流程图应用举例	20
2.3 伪代码	20
2.3.1 伪代码及其分类	21
2.3.2 用伪代码描述算法	23
2.3.3 伪代码应用举例	23
2.4 算法举例	24
2.4.1 问题分析	25
2.4.2 算法设计	26
2.4.3 用流程图和伪代码描述算法	26
2.5 实训一:用流程图和伪代码描述算法	27
思考与练习	27

第3章 程序设计语言基础

3.1 程序设计语言的用途	28
3.1.1 作为描述算法的工具	28

3.1.2	作为人一机交互的工具	29
3.1.3	作为人—人交流的工具	29
3.2	程序设计语言的演变	29
3.2.1	程序设计语言的历史	30
3.2.2	程序设计方法的演变	33
3.3	C语言基础	37
3.3.1	C语言的特点	37
3.3.2	C语言程序基本结构	38
3.3.3	C语言程序基本语法	39
3.4	C语言编程环境	39
3.4.1	程序设计过程与开发工具	39
3.4.2	TC 2.0 开发环境简介	40
3.4.3	在 TC 2.0 中编写 C 语言程序	43
3.5	实训二:TC 2.0 开发环境的使用	46
	思考与练习	50
第4章 数据的输入与输出		
4.1	数据输出方法	51
4.1.1	输出到屏幕	51
4.1.2	输出到文件	54
4.2	数据输入方法	54
4.2.1	从键盘输入	54
4.2.2	从文件输入	56
4.3	输入/输出设计	57
4.3.1	输入设计举例	57
4.3.2	输出设计举例	58
4.4	实训三:输入/输出设计	61
	思考与练习	63
第5章 数据的存储与处理		
5.1	数据和变量	64
5.1.1	数据的表示	64
5.1.2	数据类型	65
5.1.3	常量和变量	69
5.1.4	变量的声明和使用	71
5.1.5	数据在内存中的存储和使用	71
5.2	运算符与表达式	72
5.2.1	运算符概述	72
5.2.2	算术运算符与算术表达式	72
5.2.3	赋值运算符	74
5.2.4	关系运算符与逻辑运算符	74
5.2.5	表达式与语句	75

5.3 实训四:基本运算与表达式	77
思考与练习	77
第6章 流程和逻辑控制	
6.1 基本程序结构	80
6.1.1 顺序结构	81
6.1.2 分支结构	82
6.1.3 循环结构	82
6.2 分支结构	83
6.2.1 分支结构的逻辑	84
6.2.2 分支结构的实现	85
6.2.3 分支结构举例	86
6.3 循环结构	87
6.3.1 循环结构的逻辑	87
6.3.2 循环结构的实现	89
6.3.3 循环结构举例	90
6.4 算法的流程设计	91
6.4.1 问题的提出	91
6.4.2 问题分析与算法设计	91
6.4.3 程序流程分析与设计	91
6.5 实训五:算法的流程设计	94
思考与练习	95
第7章 数据组织形式	
7.1 构造数据类型	97
7.1.1 数组的声明与引用	97
7.1.2 指针的声明与引用	99
7.1.3 结构的声明与引用	101
7.1.4 枚举的声明与引用	103
7.2 常见数据存储结构	104
7.2.1 数组及其应用	104
7.2.2 指针及其应用	108
7.2.3 链表及其应用	109
7.2.4 堆栈及其应用	113
7.3 数据存储与处理的设计	115
7.3.1 问题的提出	115
7.3.2 问题分析与存储结构设计	115
7.3.3 关于效率的进一步讨论	116
7.4 实训六:数据组织形式的设计	118
思考与练习	118
第8章 编写高效的程序	
8.1 如何评价程序运行的效率	120

8.1.1 影响程序运行效率的因素	120
8.1.2 算法的时间消耗	121
8.1.3 算法的空间消耗	122
8.2 流程如何减少时间和空间的消耗	122
8.2.1 减少时间消耗的方法	122
8.2.2 减少空间消耗的方法	124
8.3 通过改进算法提高效率	125
8.3.1 对问题的再分析	125
8.3.2 算法的改进	125
8.3.3 进一步改进的思路	126
8.4 综合案例的设计	127
8.4.1 案例的背景说明	127
8.4.2 系统设计	128
8.5 实训七:改进算法提高效率	129
思考与练习	129
第9章 问题求解方法	
9.1 问题求解的一般步骤	130
9.1.1 问题的分类	130
9.1.2 问题求解的步骤	131
9.2 通用算法	131
9.2.1 穷举法与归纳法	131
9.2.2 递推法与递归法	134
9.2.3 其他方法	135
9.3 案例举例与分析	140
9.3.1 常见数值类问题的求解	140
9.3.2 常见非数值类问题的求解	140
9.3.3 综合案例的设计	141
9.4 实训八:设计算法求解数值类问题	145
9.5 实训九:设计算法求解非数值问题	146
思考与练习	146
第10章 C语言进阶	
10.1 指 针	147
10.1.1 间接运算符	147
10.1.2 指针的声明和使用	148
10.2 字符串	149
10.2.1 字符数组	149
10.2.2 指针与字符数组	151
10.3 函数进阶	152
10.3.1 编写函数	152
10.3.2 编写和使用一个简单的函数	153

10.3.3 函数的指针·····	155
10.4 实训十:字符串操作·····	161
思考与练习·····	161
第 11 章 大型程序开发	
11.1 程序的风格·····	162
11.1.1 逻辑风格·····	162
11.1.2 正 文·····	164
11.1.3 输入/输出·····	167
11.2 大型程序开发方法·····	167
11.2.1 标识符的连接属性·····	167
11.2.2 头文件的定义与使用·····	171
11.2.3 条件编译·····	172
11.3 程序项目管理器·····	173
11.3.1 一个程序项目的开发步骤·····	173
11.3.2 程序项目管理器的使用技巧·····	174
11.4 使用集成动态调试工具·····	174
11.4.1 概 述·····	175
11.4.2 步进执行·····	176
11.4.3 设置断点·····	177
11.5 实训十一:构造大型程序·····	178
思考与练习·····	178
第 12 章 常见应用领域	
12.1 用户界面与图形程序设计·····	179
12.1.1 文本模式下的用户界面设计·····	179
12.1.2 图形用户界面设计·····	182
12.1.3 用户界面设计总结·····	187
12.2 磁盘与文件管理·····	187
12.2.1 文件的概念·····	187
12.2.2 缓冲文件系统·····	188
12.2.3 文件的打开与关闭·····	188
12.2.4 文件的读写·····	190
12.2.5 磁盘与文件总结·····	191
12.3 实训十二:磁盘与文件操作·····	191
思考与练习·····	191
附录一:ASCII 字元表·····	192
附录二:TC 2.0 关键字·····	193
附录三:TC 2.0 运算符·····	195
附录四:TC 2.0 库函数·····	196
附录五:TC 2.0 常见错误信息·····	205

第 1 章 计算机与程序

欢迎进入精彩与神奇的计算机世界!

从 1946 年第一台电子计算机诞生至今,计算机的发展已经六十多年了。在这六十多年的时间里,计算机领域已经发生了翻天覆地的变化,促进了科学技术的飞速发展,其影响深入到了人类生活的方方面面。程序设计语言作为赋予计算机灵魂的基本工具,也在与这个时代共同进步着。本教程将一步一步地向读者展示如何利用计算机求解问题,以及所涉及的程序设计理论、工具和方法。同时,全书以一个用 C 语言描述的,连续、完整的实例,即课程评测系统贯穿始终,各章的内容与这个实例紧密关联,逐步递进,并在各章的实验中为实例的各种功能陆续编写出成功的代码,最终将各种功能组合为所要求的应用程序。仔细研究本书所提供的思路和方法,将有助于读者在学习过程中早日领悟程序设计的真谛。而要进行程序设计,首先需要理解计算机的基本结构和工作原理、计算机与程序的关系以及程序设计的目标和基本方法。

通过本章的学习,要求读者达到的目标是:

- 了解计算机的基本结构和工作原理,理解计算机与程序、程序与程序设计语言的关系;
- 理解编写计算机程序的目的,了解利用计算机求解问题的全过程。

1.1 计算机的基本结构与工作原理

计算机是一个能自动进行信息处理的系统,即它接收数字化的输入信息,根据存储在计算机内的程序对输入信息自动进行处理,并将处理结果输出。因此,计算机又是一个信息处理机。它由硬件和软件两大部分组成。本节介绍通用计算机硬件的基本结构和工作原理。

1.1.1 计算机硬件的基本结构

计算机硬件是由电子的、磁性的和机械的器件组成的装置,是计算机的物理基础。计算机硬件虽然有不同的构成形式,但都有其相同的特点。1944 年,美国数学家冯·诺伊曼提出了计算机应具有 5 个基本组成部分:运算器、控制器、存储器、输入设备和输出设备,描述了这五大部分的功能和相互关系,并提出了“采用二进制”和“存储程序”这两个重要的基本思想。采用二进制即计算机中的数据和指令均以二进制的形式存储和处理;存储程序即将程序预先存入存储器中,使计算机在工作时能够自动地从存储器中读取指令并执行。采取以上典型结构的计算机称为冯·诺伊曼机,目前绝大部分计算机仍然采取这样的体系结构。图 1-1 所示是冯·诺伊曼提出的计算机硬件组成的示意图。

1. 运算器

运算器是对二进制数进行运算的部件。它在控制器的控制下执行程序中的指令,完成各种算术运算、逻辑运算、比较运算、移位运算以及字符运算等。

运算器由算术逻辑部件(ALU)与寄存器组成。算术逻辑部件完成加、减、乘、除等四则运算以及与、或、非、移位等逻辑运算。寄存器用来暂存参加运算的操作数或中间结果。常用的寄存器有累加寄存器、暂存寄存器、标志寄存器和通用寄存器等。

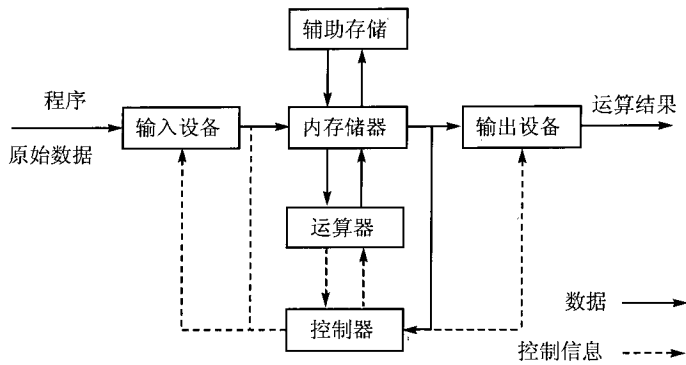


图 1-1 计算机的硬件组成示意图

运算器的主要技术指标是运算速度,其单位是 MIPS(百万指令/秒)。由于执行不同的指令所花费的时间不同,因此某一台计算机的运算速度通常是指按照一定的频率执行各种指令的统计值。

2. 存储器

存储器是用来存储数据和程序的部件。由于计算机的信息都是以二进制形式表示的,所以必须使用具有两种稳定状态的物理器件来存储信息。这些物理器件主要有磁芯、半导体器件、磁表面器件和光盘等。

位(bit)是存储器的最小存储单位,8 位为 1 个字节(byte,B)。若干位组成一个存储单元,其中可以存放一个二进制的数或一条指令。一个存储单元中存入的信息称为一个字(word),一个字所包含的二进制数的位数称为字长。小型机或微型机的字长一般为 16 位或 32 位,表示一个存储单元中的信息由 16 位或 32 位的二进制代码组成。计算机的字长越大,其精确度越高。存储器所包含的存储单元的总数称为存储容量,其单位为 KB(1KB=1 024B)。

根据功能的不同,存储器一般可分为内存储器和外存储器两种类型。

(1) 内存储器

内存储器又称主存储器,简称内存或主存,用来存放现程序的指令和数据,具有存取速度快、可直接与运算器及控制器交换信息等特点,但容量一般不大。按照存取方式,内存储器又可分为随机存储器(Random Access Memory, RAM)和只读存储器(Read Only Memory, ROM)两种。随机存储器用来存放正在执行的程序及所需要的数据,具有存取速度快、集成度高、电路简单等优点;但断电后信息不能保存。只读存储器用来存放监控程序、操作系统等专用程序。只读存储器按照功能和特点又可以分为掩膜 ROM、可编程 PROM 和可改写 EPROM 等。

(2) 外存储器

外存储器又称辅助存储器,简称外存或辅存,用来存放需要长期保存的信息,其特点是存储容量大、成本低。但它不能直接和运算器、控制器交换信息,需要时可成批地与内存储器交换信息。目前广泛使用的外存储器主要有软磁盘、硬磁盘及光盘等。

3. 控制器

控制器是指指挥计算机的各个部件按照指令的功能要求协调工作的部件,是计算机的“神经

中枢”。控制器的主要特点是采用内存程序控制方式,即在使用计算机前必须预先编写(或由编译程序自动生成)由计算机指令组成的程序,并将其存入内存存储器,由控制器依次读取并执行。

控制器由程序计数器(PC)、指令寄存器(IR)、指令译码器(ID)、时序控制电路以及微操作控制电路等组成。其中,程序控制器用来对程序中的指令进行计数,使得控制器能够依次读取指令;指令寄存器用来在指令执行期间暂时保存正在执行的指令;指令译码器用来识别指令,并分析指令的操作要求;时序控制电路用来生成时序信号,以协调在指令执行周期内的各部件的工作;微操作控制电路用来产生各种控制操作命令。

4. 输入/输出设备

输入/输出设备简称 I/O 设备,又被称为外部设备,它是外部与计算机交换信息的渠道。

(1) 输入设备

用于输入程序、数据、操作命令、图形、图像以及声音等信息。常用的输入设备有键盘、鼠标、扫描仪、光笔、数字化仪以及语言输入装置等。

(2) 输出设备

用于显示或打印程序、运算结果、文字、图形及图像等,也可以播放声音。常用的输出设备有显示器、打印机、绘图仪以及声音播放装置等。

1.1.2 计算机的工作原理

1. 计算机的指令系统

(1) 指令及其格式

指令是能被计算机识别并执行的二进制代码,它规定了计算机能完成的某一种操作。例如,加、减、乘、除、存数、取数等都是一个基本操作,可以分别用一条指令来实现。一台计算机所能执行的所有指令的集合称为该台计算机的指令系统。注意:指令系统是依赖于计算机的,即不同类型的计算机的指令系统不同;另外,计算机硬件只能够识别并执行机器指令,用高级语言编写的源程序必须由程序语言翻译系统把它们翻译为机器指令后,计算机才能执行。

某一种类型的计算机指令系统中的指令都具有规定的编码格式。一般的,一条指令可分为操作码和地址码两部分。其中,操作码规定了该指令进行的操作种类,如加、减、存数、取数等;地址码给出了操作数、结果以及下一条指令的地址。

在一条指令中,操作码是必须有的。地址码可以有多种形式,如四地址、三地址和二地址等。四地址指令的地址部分包括第一操作数地址、第二操作数地址、存放结果的地址和下一条指令地址,如图 1-2 所示。三地址指令的地址部分只包括第一操作数地址、第二操作数地址和存放结果的地址,下一条指令的地址则从程序计数器中获得。计算机每执行完一条指令后,PC 将自动加 1,从而可形成下一条指令的地址。二地址指令的地址部分是使存放操作结果的地址与某一个操作数地址相同,即在执行操作之前该地址存放操作数,操作结束后该地址存放操作结果,这样可以去掉结果的地址部分。

操作码	第一操作数地址	第二操作数地址	结果的地址	下一条指令的地址
-----	---------	---------	-------	----------

图 1-2 四地址指令的一般格式

(2) 指令的分类与功能

指令系统中的指令条数因计算机的不同类型而异,少则几十条,多则数百条。无论哪一种类型的计算机一般都具有以下功能的指令:

- 数据传送型指令 其功能是将数据在存储器之间、寄存器之间以及存储器与寄存器之间进行传送。例如,取数指令将存储器某一存储单元中的数据取入寄存器,存数指令将寄存器中的数据存入某一存储单元。
- 数据处理型指令 其功能是对数据进行运算和变换。如加、减、乘、除等算术运算指令,与、或、非等逻辑运算指令,大于、等于、小于等比较运算指令等。
- 程序控制型指令 其功能是控制程序中指令的执行顺序。如无条件转移指令、条件转移指令、子程序调用指令和停机指令等。
- 输入/输出型指令 其功能是实现输入/输出设备与主机之间的数据传输。如读指令、写指令等。
- 硬件控制指令 其功能是对计算机的硬件进行控制和管理。

2. 计算机的工作原理

计算机工作时,有两种信息在流动,即数据信息和指令控制信息。数据信息是指原始数据、中间结果、结果数据、源程序等,这些信息从存储器读入运算器进行运算,计算结果再存入存储器或传送到输出设备。指令控制信息是由控制器对指令进行分析、解释后向各部件发出的控制命令,指挥各部件协调地工作。

下面以指令的执行过程认识计算机的基本工作原理。在图 1-3 中给出了指令的执行过程。其中,左半部是控制器,包括指令寄存器、指令计数器、译码器和逻辑线路等;右上部是运算器,包括累加器、算术与逻辑运算部件等;右下部是内存储器,其中存放程序和数据(为简单起见,数据用十进制表示,指令操作码和地址码用八进制表示);带有圆圈的数字表示指令的执行步骤。

指令的执行过程可分为以下 4 个步骤:

(1) 取指令

即按照指令计数器中的地址从内存储器中取出指令,并送往指令寄存器中。图 1-3 中,指令计数器中的地址为 0102,所指向的指令为 070250。

(2) 分析指令

即对指令寄存器中存放的指令进行分析。图 1-3 中的指令为 070250。由操作码(07)确定执行什么操作,由地址码(0250)确定操作数的地址。

(3) 执行指令

即根据分析的结果,由控制器发出完成该操作所需要的一系列控制信息,去完成该指令所要求的操作。

(4) 准备执行下一条指令

上述步骤完成后,指令计数器加 1,为执行下一条指令做好准备。如果遇到转移指令,则须将转移地址送入指令计数器。

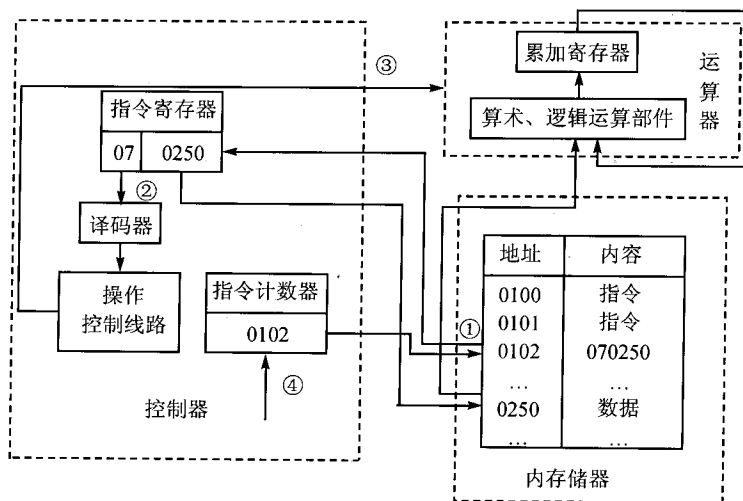


图 1-3 指令的执行过程

1.2 计算机程序

计算机程序也称为软件。要理解软件开发,就需要知道什么是计算机程序以及程序是如何工作的。计算机具有快速而精确地完成数学计算的能力,要执行这些计算,计算机需要运行专门为某个任务编写的计算机程序。计算机程序定义了一组指令集,计算机根据指令集中的指令要求对输入数据进行处理,处理的结果将作为输出数据。计算机程序员就是为各种任务编写指令集的专家。本节将学习计算机程序的定义和计算机程序可以实现的功能。

1.2.1 计算机程序的定义

计算机程序是一组精确地告诉计算机执行什么操作和什么时候执行操作的连续指令集。由于各种任务的复杂程度和时间长度存在差异,因此计算机程序的大小也各不相同。

程序可使计算机具备特定的功能。任务可以由单个程序或协同工作的多个程序完成。当几个程序一起工作可以提供更为强大的功能时,常常把这些程序组合成一个软件包。例如,常见的 Word 和 Excel 就组合在一起成为 Office 软件包的一部分一同出售。

计算机程序可以分为 3 种类型:

- **应用程序** 应用程序是最常见的计算机程序。例如,商务问题的解决方案通常由一个或多个小任务组成,要解决一个商务问题,需要多个程序一起构成一个应用程序,如财务应用程序通常包含员工工资处理程序、员工工资单生成程序和报表打印程序。
- **操作系统** 任何应用程序都需要一个可以支持其运行的操作系统。操作系统负责与硬件通信以支持各种程序命令的执行。操作系统向应用程序提供其请求的资源,比如内存、磁盘空间,并协助应用程序访问诸如显示器和打印机等输出设备。
- **设备驱动程序** 操作系统需要通过设备驱动程序与硬件设备通信。设备驱动程序是帮助操作系统与计算机中的硬件进行通信的应用程序。设备驱动程序管理硬件设备,

比如打印机、显示器、调制解调器和磁盘驱动器等。设备驱动程序与硬件和操作系统直接交互,因此编写设备驱动程序时需要特别小心,设备驱动程序中的缺陷很容易导致计算机无法响应。

1.2.2 计算机程序的执行过程

可以把计算机程序的执行过程分成3个不同的阶段,即输入、处理和输出阶段。无论如何使用计算机,所有计算机都按照相似的过程工作:把数据输入计算机,计算机处理该数据,然后计算机产生输出,该输出在显示器上显示或在纸上打印。这样的过程称为输入—处理—输出循环或I—P—O循环。

本节将简单介绍输入、处理和输出这3个阶段。

1. 输入阶段如何工作

输入阶段是计算机程序执行的第一个阶段。在输入阶段中,需要向计算机提供数据。事实上,不同的工作任务和目标需要不同的数据,而数据也具有很多种不同的存在形式和表现方式,因此有很多不同的数据输入方式。例如,要在影像租赁店中租借影碟,顾客必须提供关于影片要求的信息,比如影片类型、导演或者主演者等条件。确定要租借某张影碟后,工作人员负责将该影碟的唯一标号输入计算机,并记录租借时间。上述操作就是影像制品租赁程序的输入阶段。

2. 处理阶段如何工作

计算机程序执行的第二个阶段是处理阶段。在处理阶段中,计算机在输入阶段提供的数据上执行操作。显然,不同的任务和目的,对于输入数据的处理方法是不同的。例如,在影像制品租赁程序的输入阶段完成后,计算机用输入的数据来判断在当前的碟片库中是否还有该影片的拷贝,如果已经没有拷贝了,则租借过程无法进行。

3. 输出阶段如何工作

计算机程序执行的第三个阶段是输出阶段。在输出阶段中,计算机将输入后经过处理的信息以某种方式提供出来。一般的,程序的输出内容都在显示器上显示出来,但也有一些程序在执行后其结果不会直接输出到显示器,而是通过打印机或者计算机的其他输出端口输出。例如,在影像制品租赁程序的处理阶段处理了输入阶段指定的数据后,计算机将在显示器上或在打印的文档中显示租赁的结果。

4. I—P—O 循环

编写计算机程序是为完成某个特定的任务或解决某个特定的问题。一般来说,随着计算机硬件性能的飞速提高,要求计算机解决的问题的规模逐步扩大,复杂性也迅速提高。因此,为控制问题的规模和复杂性,保证软件的质量,一般将程序划分为多个逻辑块或功能块,其各自执行全部任务中相对独立的一部分。同时,多个逻辑块或功能块之间按照任务要求的逻辑关系,一起连接协同执行。一个块的输出刚好作为下一个块的输入,最终完成所有操作。因此,一个完整的计算机程序中,一般包含着多个协同工作的子程序,其各自的I—P—O阶段相互衔接形成了I—P—O循环,以完成整个应用程序的功能要求,如图1-4所示。

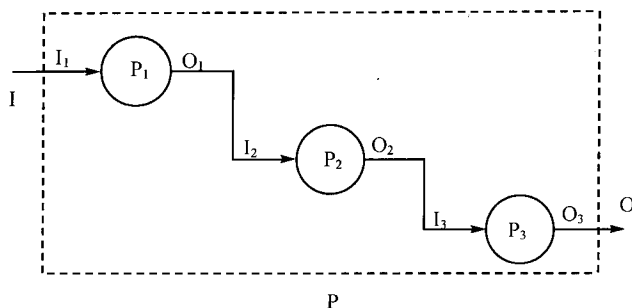


图 1-4 多个程序的 I—P—O 衔接

1.2.3 程序与程序设计语言

1. 计算机与程序

虽然计算机被称为“电脑”，但事实上，它只是一台快速执行指令的机器。这些指令非常简单（简单的四则运算、逻辑运算、数据传送和跳转指令），但它们的组合却能完成非常复杂的任务。要让计算机做出符合人们意图的事，只须把这些简单的指令排成一个顺序执行的序列，例如针织毛衣时，两根毛衣针虽然只有上针、下针、收针和放针四个操作，但根据不同的口诀却可以织出各色花样的毛衣，这个口诀就是针织花样的程序。世界上第一位程序员 Ada 就是这样以穿孔卡片为 Babbage 的差分引擎计算机编制织花程序的。所以，程序是计算机指令的序列，编制程序就是为计算机安排指令的序列。把程序装入计算机内存，按顺序逐条执行，全部执行完就达到所需的要求。

计算机程序设计的核心问题是设计解决问题的方法和步骤，即确定算法。一旦算法确定下来了，就可以按照程序设计语言的语法规则将其表述为程序代码的形式，这就是所谓的程序源代码或源程序。源程序提交给计算机后，经过语言编译—解释程序的处理，形成可以直接操作的计算机二进制指令，既可立即执行，又可以将二进制指令保存为可执行文件，在需要时执行。

2. 程序与程序设计语言

使用计算机时，要让计算机能按人的规定完成一系列的工作，就要求计算机具备理解并执行人们给出的各种指令的能力。因此，程序设计语言就是在人和计算机之间进行信息交流的工具，是人和计算机沟通的桥梁。程序设计，既是使用程序设计语言编写程序的过程，又是人们把对某种事务的处理方法用程序设计语言的形式重新表达的过程，相当于文章的创作以及翻译的过程。

随着计算机技术的发展，程序设计语言经历了由低级向高级发展的多个阶段，程序设计方法也随之得到了不断发展和提高。计算机语言按其发展程度可以分为机器语言、汇编语言和高级语言。表 1-1 中列出了常见的程序设计语言的有关信息。关于程序设计语言的历史及其发展演变的趋势，将在第 3 章中进行详细的叙述。

本书将向读者介绍程序设计的基本思想及方法，并详细介绍 C 语言的语法规则，以及如何利用 C 语言进行程序设计的详细步骤。在此之前，先来看看程序设计的目标及其基本思想和方法。通过这些介绍可以理解人们为什么要使用计算机，进行程序设计的目的是什么，为什