

职业中学计算机专业系列教材



C 语言 程序设计基础教程

职业中学计算机专业系列教材编委会

重庆大学出版社

内 容 简 介

全书共10章。详细介绍了C语言的基础知识,包括程序和程序设计语言的有关概念、C源程序结构、Turbo C集成环境的基本使用方法、基本数据类型、三种基本结构的程序设计、数组、函数以及文件,并介绍C语言对文本屏幕的一些特殊操作以及利用C语言进行屏幕作图的基本方法。对指针和结构体的有关概念作了概括而简洁的介绍。

书中共有100多道例题,注重实用性与趣味性相结合。所有例题程序全部采用中文注释。

各章的末尾都设有题型多样的习题,第2~8章还给出了目的明确、内容详尽的上机实习指导,极大地方便了教学和自学。

本书既适于作职业中学计算机专业的教材,又可作为初学者的自学参考书。

C语言程序设计基础教程

职业中学计算机专业系列教材编委会

责任编辑 王 勇

重庆大学出版社出版发行

新华书店经销

重庆通信学院印刷厂印刷

*

开本:787×1092 1/16 印张:16.5 字数:411千

1997年2月第1版 1997年2月第1次印刷

印数:1~3000

ISBN 7-5624-1428-9/TP·138 定价:17.00元

(川)新登字 020 号

职业中学计算机专业 系列教材编委会

主 任 牟维坤
副主任 包锦安 朱庆生
委 员 (以姓氏笔划为序)
文水华 王达恩 向才毅
李宝珠 李重辉 卓建业
赵元政 魏嗣富 张小毅
谭元颖

前 言

随着科学技术与现代社会的发展,电子计算机技术已成为当代新技术革命的前锋,广泛应用于国民经济各个领域。计算机文化也逐步成为现代文化的组成部分。因此,在中学教育中,实施计算机教育是一项面向现代化、面向世界、面向未来的重要举措,也是教育与现代科学技术接轨的重要途径,更是培育适应社会主义市场经济建设和具有国际参与和竞争能力人才的手段。

《中国教育改革和发展纲要》推动着计算机职业教育蓬勃发展。近年来,各地职业中学为国家培养和输送了大批计算机应用专业人才,受到社会赞誉。为了适应计算机应用、发展和普及的需要,适应计算机职业教育专业化、正规化的要求,重庆市教委职业教育处、重庆市教育研究所、重庆市劳动局培训处、重庆大学计算机系及重庆市计算机职业高级中学组织多年从教并具有丰富教学经验的特级教师、高级教师和计算机专家,编写了这套计算机职业中学系列教材。

本套教材是根据重庆市教育委员会重教职[1995]45号文颁发的职业高中计算机专业教学计划、教学大纲编写的。全套教材共11种:

《计算机导论》

《五笔字型汉字录入技术教程》

《PASCAL 语言程序设计教程》

《FOXBASE+教程》

《电子排版》

《常用软件及其应用》

《NOVELL 网络操作系统教程》

《操作系统》

《计算机故障判断与维修》

《计算机英语教程》

《C 语言程序设计基础教程》

本套教材是重庆市教委推荐的职业高中教材。

本套教材编写的原则是:保证基础,突出应用,既照顾当前教学的实际,又考虑未来发展的需要。编写中力求做到“精、用、新”、“浅、简、广”,既适于职业高中、技工学校使用,也可供大中专、程序设计人员和各类计算机培训班选用。

本套教材各课程课时分配如下表,仅供参考。

为适应计算机教学需要,我们还将推出这套系列教材的配套上机实习手册,练习册及教学光盘,以满足教学需要,欢迎广大读者提出宝贵建议。

愿本套教材的推出,为职业教育计算机专业的发展作出贡献。

教学课时分配表

序 号	学 年		一学年		二学年	
	学 时 课程名称	学 期	1	2	3	4
1	计算机导论		36			
2	PASCAL 语言程序设计教程		108	108		
3	五笔字型汉字录入技术教程		72			36
4	基本工具软件及其应用				72	
5	FOXBASE+ 教程				108	108
6	电子排版				72	
7	网络操作系统教程					72
8	操作系统			72		
9	计算机故障判断与维修				90	90
10	计算机英语教程					72
11	C 语言程序设计基础教程		108	108		

C 语言是一种结构化、模块化的通用程序设计语言，它兼有高级语言和低级汇编语言的优点，适宜于编制各种系统软件和应用软件。所以，现在学习 C 语言和使用 C 语言编程的人越来越多。

开设 C 语言程序设计课程的目的在于让学生以 C 语言为有力的编程工具，培养初步的程序设计方法和良好的程序设计风格，为下一步深入的程序设计课程和相关理论课程打下基础。近年来，国内出版了很多介绍 C 语言的书籍，但其中面向职业中学计算机专业这一层次的书籍却是凤毛麟角。为此，编者从职业中学计算机专业的 C 语言教学要求出发，根据自己多年的教学经验编写了这本书。

本书具有以下几个特点：

①本书的讲述是以具有高效集成开发环境的 Turbo C 为代表，书中的例题程序全部在 Turbo C 的集成开发环境中调试通过。

②全书共 10 章。其中，第 1~8 章详细介绍 C 语言的基础知识，包括程序和程序设计语言的有关概念、C 源程序结构、Turbo C 集成环境的基本使用方法、基本数据类型、三种基本结构的程序设计、数组、函数以及文件，这一部分是 C 语言课程的必学内容；第 9 章介绍 C 语言对文本屏幕的一些特殊操作，以及利用 C 语言进行屏幕作图的基本方法，该章有助于学生在程序的界面设计环节中开阔思路；第 10 章介绍了指针和结构体的概念，考虑到本层次的教学要求，该章的讲述概括而简洁，旨在让有余力的学生有更上一层楼的机会。第 9 章和第 10 章都是

选学内容。

③在有关程序设计方法的讲述中,渗入了结构化程序设计的思想。

④书中所有例题,都是编者针对职业中学学生的知识面精心设计和安排的。所有的例题都不涉及对复杂数学问题的处理,而是把重点放在实用性与趣味性相结合上,力求简明易懂,既有利于启发思维,又能提高学生的学习兴趣。所有例题程序都采用中文注释,而且程序中用了大量的汉字字符串作为输入输出的提示信息。所有的程序均可在 UC DOS 汉字系统下运行。

⑤各章的末尾都设有题型多样的习题,第 2~8 章还给出了目的明确、内容详尽的上机实习指导,极大地方便了教与学。

⑥本书既适于作职业中学计算机专业的教材,又可作为初学者的自学参考书。

本书的第 1、2、7、8、9 章以及附录由向华编写,第 3、4、5、6、10 章由吴新编写。

重庆大学的王铮副教授对书稿提出了不少的宝贵意见,重庆市龙门浩职业高级中学的李伟老师编排了第 2 章的所有图示,在此表示诚挚的感谢。本书在编写的过程中,得到了重庆市教科所、重庆市龙门浩职业高级中学电脑部、重庆市小龙坎职业中学计算机教研室的领导和同事们的支持、鼓励和帮助,在此一并致以衷心的感谢!

职业中学计算机专业系列

教材编委会

1996 年 12 月

目 录

1 程序和程序设计语言	1
1.1 计算机程序	1
1.2 计算机中数的表示方法	1
1.2.1 二进制数的特点	2
1.2.2 二进制数与十进制数之间的转换	2
1.2.3 十六进制数	3
1.2.4 八进制数	4
1.3 程序设计语言	5
1.3.1 程序设计语言的发展	5
1.3.2 C语言概述	7
小结1	8
习题1	8
2 C语言基础	10
2.1 C语言的基本符号和源程序结构	10
2.1.1 C语言的基本符号和标识符	10
2.1.2 C语言的源程序结构	11
2.2 Turbo C集成环境简介	13
2.2.1 启动 Turbo C	13
2.2.2 建立 C 源程序	14
2.2.3 运行 C 程序	16
2.2.4 源程序的修改	17
2.2.5 保存 C 程序	18
2.2.6 装入一个已存在的 C 程序	19
2.2.7 结束当前程序,开始编辑另一个 C 程序	19
2.2.8 退出 Turbo C 系统	19
2.3 基本数据类型	19
2.3.1 数据类型	19
2.3.2 常量与变量	20
2.3.3 整型数据	22
2.3.4 实型数据	23
2.3.5 字符型数据	23
2.4 运算符与表达式	26
2.4.1 算术运算符和算术表达式	26
2.4.2 赋值运算符和赋值表达式	28
2.4.3 逗号运算符和逗号表达式	29

小结2.....	30
习题2.....	31
上机实验2.....	32
3 顺序结构程序设计.....	37
3.1 算法的概念和表示方法.....	37
3.1.1 算法的概念	37
3.1.2 流程图	37
3.2 结构化程序设计方法.....	39
3.3 C 语言的表达式和语句	40
3.3.1 表达式	40
3.3.2 语句.....	42
3.4 数据输出.....	43
3.4.1 输入输出的概念	43
3.4.2 printf 函数	44
3.4.3 字符输出函数 putchar	49
3.5 数据输入.....	49
3.5.1 scanf 函数.....	50
3.5.2 字符输入函数 getchar()	52
3.6 综合程序举例.....	53
小结3.....	55
习题3.....	55
上机实验3.....	56
4 选择结构程序设计.....	59
4.1 if 语句	59
4.1.1 最基本的 if 语句.....	59
4.1.2 if else 语句	60
4.1.3 多重选择:if else if 语句	63
4.1.4 else 与 if 的匹配问题	65
4.1.5 if 语句的嵌套	67
4.2 关系运算符和关系表达式.....	68
4.2.1 关系运算符和表达式	68
4.2.2 “真”与“假”	70
4.3 逻辑运算符和逻辑表达式.....	71
4.4 switch 语句	74
4.4.1 switch 语句	74
4.4.2 多重标号的使用	77
小结4	78
习题4	79

上机实验4	83
5 循环结构程序设计	85
5.1 while 语句	85
5.1.1 while 语句	85
5.1.2 循环结构的术语	87
5.2 do while 语句	89
5.3 for 语句	91
5.3.1 for 语句	91
5.3.2 for 的灵活性	93
5.4 循环嵌套	95
5.4.1 二重循环	95
5.4.2 循环嵌套的几种形式	97
5.5 循环的辅助语句:break 语句和 continue 语句	98
5.5.1 break 语句	98
5.5.2 continue 语句	99
5.6 综合示例	102
小结5	105
习题5	105
上机实验5	109
6 数组	111
6.1 一维数组	111
6.1.1 一维数组的定义和引用	111
6.1.2 一维数组的初始化	113
6.1.3 数组元素的赋值	114
6.2 一维数组常见处理算法	115
6.3 二维数组	122
6.3.1 二维数组的定义和引用	122
6.3.2 二维数组的初始化	124
6.4 字符数组和字符串	125
6.4.1 字符数组和字符串	125
6.4.2 字符串的输入输出	129
6.4.3 字符串处理函数	131
6.4.4 综合举例	137
小结6	139
习题6	139
上机实验6	142
7 函数	145

7.1 模块化程序设计方法	145
7.2 Turbo C 的库函数	146
7.2.1 Turbo C 库函数分类	146
7.2.2 库函数的调用方法	146
7.3 函数的定义和调用	147
7.3.1 函数的定义方法	147
7.3.2 函数的调用方法	153
7.3.3 函数的嵌套调用	155
7.3.4 函数的递归调用	156
7.4 函数的参数传递	158
7.4.1 基本数据类型作函数参数	158
7.4.2 数组作函数参数	159
7.5 局部变量和全局变量	164
7.5.1 局部变量	165
7.5.2 全局变量	167
7.6 一个有关函数应用的实用程序	171
7.6.1 问题分析	171
7.6.2 程序的模块结构	172
7.6.3 程序清单	173
7.6.4 一点启发	177
小结7	178
习题7	178
上机实验7	181
8 文件	183
8.1 文件概述	183
8.1.1 文件的概念	183
8.1.2 从程序设计的角度看使用数据文件的优点	183
8.1.3 文件的特点	184
8.2 文件的打开与关闭	184
8.2.1 文件的打开(fopen()函数)	184
8.2.2 文件的关闭(fclose()函数)	185
8.3 文件的读写	186
8.3.1 fputc()函数和 fgetc()函数	186
8.3.2 fprintf()函数和 fscanf()函数	190
8.3.3 fputs()函数和 fgets()函数	194
8.4 文件的定位	195
8.4.1 rewind()函数	195
8.4.2 fseek()函数和 ftell()函数	196
8.5 一个有关文件应用的实用程序	198

8.5.1 程序分析.....	198
8.5.2 程序的模块结构.....	199
8.5.3 程序清单.....	199
小结8	205
习题8	206
上机实验8	206
9 字符屏幕和图形屏幕	208
9.1 文本模式下的字符屏幕	208
9.1.1 文本窗口的定义.....	208
9.1.2 文本窗口颜色的设置.....	208
9.1.3 文本窗口的有关操作.....	209
9.1.4 利用文本窗口函数设计一个漂亮菜单的实例	210
9.2 图形屏幕	211
9.2.1 图形模式的设置与关闭.....	212
9.2.2 基本图形函数.....	214
9.2.3 图形的填充	219
小结9	221
10 C语言的高级特性.....	222
10.1 指针的概念.....	222
10.1.1 变量的地址和指针	222
10.1.2 指针变量的定义和引用	223
10.1.3 指针变量的初始化	225
10.1.4 指针作函数参数	225
10.2 指针与数组.....	228
10.2.1 指向数组元素的指针变量的定义与赋值	228
10.2.2 指向字符数组的指针	230
10.3 结构.....	232
10.3.1 结构的定义	232
10.3.2 结构与函数	234
10.3.3 结构与指针	236
小结10	239
附录.....	240
附录1 Turbo C 常见编译错误信息	240
附录2 Turbo C 常用库函数	244
附录3 C语言的关键字	250

1 程序和程序设计语言

本章重点

- 计算机与程序、程序与程序设计语言之间的联系
- 计算机中数据的存储形式
- 二进制、十进制和十六进制三种数制之间的转换
- 程序设计语言的发展
- C语言的特点

1.1 计算机程序

现在,“计算机”对几乎所有的人来说都不是一个陌生的名词,从高科技领域到日常生活到处都有它的影子。计算机太聪明了,不管多么复杂的工作,它都可以快速而精确地完成,以至于人们称它为“电脑”!电脑真的可以胜过人脑吗?当然不!因为再聪明的机器都是由人控制的。确切地说,计算机是受人所编写的程序的控制,不论一台计算机的速度有多么快,功能有多么强,它的一切行动都要听程序的指挥。任何计算机,都只能做程序安排它做的事情,对程序“亦步亦趋”。离开了程序,即使是最好的计算机也将会“寸步难行”!

对计算机来说,程序是如此的重要!那么,程序究竟是什么呢?

从广义上说,程序就是完成某项工作所需要的一步一步的计划。日常生活中的许多工作都可以被看成是程序。比如,人们可以把煮面条的过程写成下面的程序:

- ①按各自的口味准备好调料放在一个碗里;
- ②把洗净的汤锅放到火上,加入清水烧开;
- ③把面条放入水中;
- ④面条煮熟后挑于碗中,即成。

由此可知,程序的概念很简单,它就是一系列的操作步骤。而计算机程序是指由人事先规定的计算机的操作步骤。每一步骤的具体内容由专门的指令或语句来描述,这些指令或语句,将告诉计算机“做什么”和“怎样做”。所以,当你需要计算机完成某项任务时,首先要有一个按照该任务的要求编写好的程序,把这个程序输入计算机之后,计算机就能遵照执行了。

1.2 计算机中数的表示方法

计算机程序可以用很多种不同的语言工具来描述,但不管你的程序和要处理的数据具有什么样的面孔,最后输入到计算机内部之后都将转化为二进制数,这是电子计算机的物理特性所决定的。在二进制数中只有两个符号,即0和1,而在电学中具有两种稳定状态以代表0和1的东西是很多的,如:电压的高和低,电灯的亮和灭,电容器的充电和放电,脉冲的有和无,等

等。所以,二进制数在电气元件中容易实现和运算,所有的程序和数据在计算机内部都采用二进制的形式表示。

1.2.1 二进制数的特点

①只具有两个基本的数字符号:0和1。

②运算时逢2进位。例如:

$$1+1=10$$

$$11+1=100$$

十进制数的0和1对应的二进制数也是0和1,十进制数的2对应的二进制数是10(注意不要读作“拾”,而应读作“壹零”),十进制数的3对应的二进制数是11($10+1=11$)。

二进制数中的每个0或1称为一个二进制位(Bit),“位”是计算机中信息存储的最小单位。每8个二进制位称为一个字节(Byte)。

1.2.2 二进制数与十进制数之间的转换

虽然计算机内部以二进制形式表示数据,且以二进制形式对数据进行算术运算和逻辑运算,但由于人们习惯使用十进制,在程序中也通常采用十进制表示数据,所以,有必要了解如何在二进制数与十进制数之间进行转换。

1)十进制数转换成二进制数

一个十进制整数要化为二进制整数只需将它反复除以2(直到商为0为止),每次除以2之后都得到一个余数,从最后一次的余数开始将每个余数连在一起,就得到了二进制表示的数。下面是把十进制数10转化成二进制数的过程:

$$\begin{array}{r} 2 \overline{) 10} \quad (0 \\ 2 \overline{) 5} \quad (1 \\ 2 \overline{) 2} \quad (0 \\ 2 \overline{) 1} \quad (1 \\ 0 \end{array}$$

从下往上取余数,最后得到: $(10)_{10}=(1010)_2$ 。这里,括弧外的10或2表示括弧内的数是十进制数或二进制数。

2)二进制数转换成十进制数

一个二进制整数转换成十进制数的方法是,将二进制数的最后一位乘以 2^0 ,最后第二位乘以 2^1 ,……依此类推,如果这个二进制数的位数为n,那么第一位就应该乘以 2^{n-1} 。最后将乘得的各项相加就得到用十进制表示的数。例如:

$$\begin{aligned} (1010)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 8 + 0 + 2 + 0 \\ &= (10)_{10} \end{aligned}$$

1.2.3 十六进制数

程序中往往用人们习惯的十进制来表示数,但某些情况下需要从二进制的角度来理解和处理问题,而二进制表示数又非常不方便,这时可以把二进制数转换成与之有着非常直接关系的十六进制数。

1)十六进制数的特点

①具有16个数字符号,它们分别是0~9和A~F。这16个数字符号与十进制数和二进制数之间的关系如表1.1所示。

表 1.1

十 进 制 数	二 进 制 数	十 六 进 制 数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

②逢16进位。

2)十六进制与二进制之间的转换

一位十六进制数可以用四位二进制数表示,它们之间存在着直接的对应关系。所以,二进制数与十六进制数之间的转换是十分简捷而方便的。

(1)二进制转换为十六进制

一个二进制整数转换为十六进制数时,将二进制数从低位向高位每四位组成一组,最后一组若不足四位则可以在最前面补0,如,1101110可分为0110、1110两组,然后把每四位二进制数用相应的十六进制数代替,即可转换为十六进制数。例如,二进制数1101110可用下面的方法转换为十六进制数:

0110, 1110

↓ ↓
6 E

所以, $(1101110)_2 = (6E)_{16}$ 。

(2) 十六进制转换为二进制

与二进制转换为十六进制相反,十六进制转换为二进制时,只需将十六进制数的每一位用四位相应的二进制数代替就行了。例如,十六进制数 A5 可用下面的方法转换为二进制数:

A	5
↓	↓
1010	0101

所以, $(A5)_{16} = (10100101)_2$ 。

3) 十六进制与十进制之间的转换

十六进制与十进制之间的转换方法类似于二进制与十进制之间的转换。

(1) 十进制转换为十六进制

将十进制数反复除以 16,从最后一次的余数开始把每次得到的余数排列在一起,即得到了相应的十六进制数。注意,对大于 9 的余数,要写成十六进制的字母符号形式。例如:

16	250 (10(A))
16	15 (15(E))
	0

所以, $(250)_{10} = (EA)_{16}$ 。

(2) 十六进制转换为十进制

与二进制整数转换为十进制数相似,一个位数为 n 的十六进制整数转换为十进制数时,从高位向低位将每一位分别乘以 16^{n-1} 、 16^{n-2} 、…… 16^0 ,然后将所乘的各项相加,即得到相应的十进制数。例如:

$$\begin{aligned}(2B)_{16} &= 2 \times 16^1 + 11 \times 16^0 \\ &= 32 + 11 \\ &= (43)_{10}\end{aligned}$$

1.2.4 八进制数

程序中也可以使用八进制数。八进制数的特点是具有 0~7 这八个数字符号,运算时逢八进位。

1) 八进制与二进制之间的转换

二进制数转换为八进制数的方法是,将二进制数由低向高每三位组成一组,再分别把每组数用一位八进制表示。例如,二进制数 1101110 可用下面的方法转换成八进制数:

1,	101,	110
↓	↓	↓
1	5	6

所以, $(1101110)_2 = (156)_8$ 。

相反, 八进制数要化为二进制数, 只需将每位八进制数用三位二进制数表示即可。例如, 八进制数 716 用二进制数表示:

$$\begin{array}{ccc} 7 & 1 & 6 \\ \downarrow & \downarrow & \downarrow \\ 111 & 001 & 110 \end{array}$$

所以, $(716)_8 = (111001110)_2$ 。

2) 八进制与十进制之间的转换

十进制数化为八进制数的方法是, 将十进制数反复除以 8, 从最后一次的余数开始把每次得到的余数排列在一起, 即为八进制表示的数。例如:

$$\begin{array}{r} 8 \overline{) 93} \quad (5 \\ 8 \overline{) 11} \quad (3 \\ 8 \overline{) 1} \quad (1 \\ 0 \end{array}$$

所以, $(93)_{10} = (135)_8$ 。

反之, 八进制整数化为十进制数时, 是将八进制数的最后一位乘以 8^0 , 最后第二位乘以 8^1 , ……依此类推, 如果这个八进制数的位数为 n , 那么第一位就应该乘以 8^{n-1} , 最后将各项相加即可。例如:

$$\begin{aligned} (13)_8 &= 1 \times 8^1 + 3 \times 8^0 \\ &= (11)_{10} \end{aligned}$$

1.3 程序设计语言

前面已经提到, 计算机是受程序控制的, 它只能做程序规定做的事, 一点也不多做和少做。而任何一个计算机能执行的程序都是由某种程序设计语言来描述的。

1.3.1 程序设计语言的发展

1) 机器语言

这是人们最早使用的程序设计语言。在机器语言中, 所有的指令都采用二进制代码, 例如, 可以用二进制数 10110110 代表加法运算, 用 10110101 代表减法运算等等。在上一节中已经知道, 计算机内部只能识别二进制代码。所以, 用机器语言编写的程序能够被计算机直接理解和执行, 这是机器语言的最大优点。

但也正是因为机器语言中只有 0 和 1 两种符号, 才使得用机器语言编写程序成了一件十分繁琐的工作。可想而知, 机器语言与人们习惯使用的自然语言相差太大, 用它编写的程序该会是多么的冗长而且难以理解! 不仅如此, 每种计算机都有自己的机器语言, 不同型号的计算机的机器语言是互不通用的(称为“面向机器”)。我们无法把为某种型号的计算机编写的机器

语言程序移到另一种型号的计算机上使用,显然这是不方便的。

2) 汇编语言

为了解决机器语言的上述缺陷,人们想出了用一些帮助记忆的符号来代替机器语言中二进制代码的方法,设计出了“汇编语言”。在汇编语言中,表示各种操作的代码全部用符号表示,如可以用 ADD 表示加法操作等等,这比机器语言中的一长串二进制数简洁直观且好记多了。

汇编语言其实是机器语言的一种助记表示,汇编语言中的每一条指令都有一条机器指令与之对应。所以,汇编语言与机器语言一样,也是面向机器的语言,它们称为是低级语言。

3) 高级语言

现在人们编写程序用得最多的是高级程序设计语言。高级语言面向用户,基本上与所用的计算机无关,即用高级语言写的程序可以适用于不同的计算机,这就给使用者带来了极大的方便。高级语言最大的优点是,形式上非常接近于人们习惯使用的算术语言和英语语言。例如,求 A、B 两数之和 C,在 C 语言中可以用一个简单的语句 $C=A+B$; 来表示。由此可见,高级语言不仅易学、易用,而且写出的程序更加简练。自 50 年代末高级语言问世以来,人们已经设计了几十乃至上百种高级语言,其中如 C, FORTRAN, ALGOL60, BASIC 与 PASCAL 等,都是它们中的著名代表。

既然计算机内部只能识别 0 和 1 两个符号,那为什么它又能读懂高级语言编写的程序中“ $A+B$ ”这样的式子呢?这里有一个翻译的过程。事实上,高级语言写的程序(称为“源程序”)并不能直接被计算机理解和执行,必须先翻译成机器指令表示的目的程序后,再让计算机执行机器指令。就像不懂英语的中国人与不懂汉语的美国人之间,需要翻译才能彼此交流一样。翻译的方式通常有两种,即编译方式和解释方式。

(1) 编译方式

在编译方式中需要有一个编译程序,先把编译程序装入计算机的内部存储器(简称内存)中,再把用高级语言写的源程序输入计算机,编译程序便把源程序整个地翻译成用机器语言表示的目的程序,然后再执行该目的程序,最后得到处理结果。图 1.1 以 C 语言为例,表示了编译的过程。

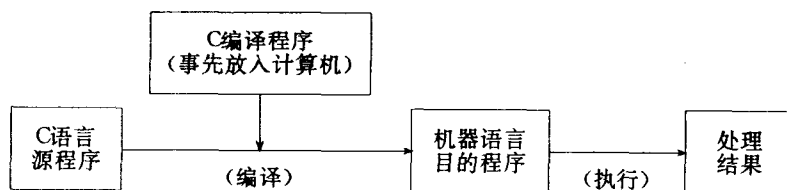


图 1.1 源程序的编译与执行过程

(2) 解释方式

有些高级语言如 BASIC 语言写的源程序,并不是像编译方式那样整个地被译成目的程序后再执行,而是由一个解释程序逐句地翻译,译出一句立即执行,即边解释边执行。与编译程序一样,解释程序也需要事先放入计算机的内存中。图 1.2 以 BASIC 语言为例,表示了解释的过程。