

21世纪网络平台大学计算机系列教材



C

程序设计实用教程

刘鸿翔 杨 莉 主编



科学出版社

www.sciencep.com

·21 世纪网络平台大学计算机系列教材·

C 程序设计实用教程

刘鸿翔 杨 莉 主编

科学出版社

北 京

内 容 简 介

本书按照《全国计算机等级考试二级考试大纲》中对 C 语言程序设计考试的要求编写,内容精练,概念清楚,结构合理,对读者可能遇到的疑难问题和易混淆概念作了详细的阐述。主要内容包括 C 语言程序设计基础知识、C 语言中涉及的各种数据类型和运算符、各种表达式、程序结构、数组、函数、指针、编译预处理和动态存储分配、结构体和共用体、文件等,有大量实例分析,以利于提高读者的编程技巧。每章之后都附有一定数量的练习题,供读者练习和自我检测。

本书可作为本科学生和大专、高职学生学习 C 语言程序设计的教材,也可以作为参加二级 C 语言程序设计考试者的自学和辅导教材。

图书在版编目(CIP)数据

C 程序设计实用教程/刘鸿翔,杨莉主编. - 北京:科学出版社,2004.8

(21 世纪网络平台大学计算机系列教材)

ISBN 7-03-014177-6

I .C… II .①刘…②杨… 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(2004)第 083301 号

责任编辑:冯贵层

责任印制:高 嵘/封面设计:深白广告

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉大学出版社印刷总厂印刷

科学出版社发行 各地新华书店经销

*

2004 年 8 月第 一 版 开本:787×1092 1/16

2004 年 8 月第一次印刷 印张:15 1/2

印数:1-6 000 字数:376 000

定价:22.50 元

(如有印装质量问题,我社负责调换)

前 言

C 语言是一门优秀的结构化程序设计语言,其结构严谨,数据类型完善,语句简练灵活,运算符丰富,一直受到计算机业内人士的好评。即使在面向对象程序设计语言广泛使用的今天,各类理工类专业,尤其是计算机专业,程序设计语言入门课程都首选 C 语言。

本书共分十三章,内容包括:C 语言概述,各种数据类型、运算符与表达式,简单的 C 程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,字符串,C 语言预处理,结构体、共用体和用户定义类型,位运算,文件等。

C 语言牵涉的概念比较复杂,规则繁多,使用灵活,容易出错。本书根据初学者的特点和认识规律,精选内容,通过通俗易懂的叙述阐明了复杂的概念,书中概念清晰,例题丰富,实用性强。在编写过程中严格按照《全国计算机等级考试二级考试大纲》中对 C 语言程序设计考试的要求,紧扣考试大纲,内容精练,结构合理,便于自学,对读者可能遇到的难题做了十分清楚和详细的阐述,也是一本系统的应试教材。

程序设计是一门实践性很强的课程,读者在学习过程中一定要重视实践环节。在掌握概念的基础上,要求能够动手编程,并要求能上机调试运行。与本书配套的《C 程序设计实训教程》主要用于上机实训和等级实训。这两本教材相辅相成,如果能够很好地掌握和消化这两本教材中的内容,完成习题,读者也就掌握了 C 语言的基础知识,完全可以达到全国计算机等级考试—C 语言所要求的水平。

参加本书编写工作的有刘鸿翔、杨莉、郑友莲、唐宏亮、姜娜,刘鸿翔、杨莉任主编,郑友莲、唐宏亮任副主编。书中所给出的实例程序全部在 Turbo C 2.0 环境下调试通过。

本书可作为大学本科、高职、高专 C 语言程序设计课程的教材,也可作为参加全国计算机二级等级考试——C 程序设计的教材用书。

由于编者水平有限,书中难免会有欠妥之处,恳请广大读者提出宝贵意见。

编者

2004 年 5 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言简介	1
1.1.1 计算机语言	1
1.1.2 C 语言的发展过程和特点	3
1.1.3 C 程序的结构和书写特点	4
1.2 Turbo C 2.0 的基本操作	6
1.2.1 Turbo C 2.0 的集成开发环境	6
1.2.2 用 Turbo C 运行 C 程序的步骤	7
1.3 习题	8
第 2 章 数据类型、运算符与表达式	10
2.1 C 的数据类型	10
2.2 常量与变量	11
2.2.1 标识符	11
2.2.2 关键字	11
2.2.3 常量与符号常量	12
2.2.4 变量	13
2.3 整型数据	13
2.3.1 整型常量	13
2.3.2 整型变量	14
2.4 实型数据	15
2.4.1 实型常量	15
2.4.2 实型变量	16
2.5 字符型数据	17
2.5.1 字符常量	17
2.5.2 转义字符	17
2.5.3 字符串常量	19
2.5.4 字符变量	19
2.6 变量赋初值	20
2.7 算术运算符和算术表达式	21
2.7.1 C 运算符简介	21
2.7.2 算术运算符和算术表达式	22
2.8 赋值运算符和赋值表达式	25
2.8.1 赋值运算符	25
2.8.2 赋值表达式	27
2.9 逗号运算符和逗号表达式	27

2.10 习题	28
第 3 章 简单的 C 程序设计	29
3.1 流程结构和语句	29
3.1.1 三种基本结构	29
3.1.2 C 语言中的语句	29
3.2 格式输入输出函数	30
3.2.1 标准输出函数 printf	30
3.2.2 标准输入函数 scanf 函数	35
3.3 字符输入输出函数	37
3.3.1 字符输出函数 putchar	37
3.3.2 字符输入函数 getchar	38
3.4 顺序结构程序设计举例	38
3.5 习题	39
第 4 章 选择结构程序设计	43
4.1 关系运算符和关系表达式	43
4.1.1 关系运算符及其优先次序	43
4.1.2 关系表达式	43
4.2 逻辑运算符和逻辑表达式	44
4.2.1 逻辑运算符及其优先次序	44
4.2.2 逻辑表达式	45
4.3 条件语句和条件运算符	45
4.3.1 if 语句	45
4.3.2 条件运算符	49
4.4 switch 语句	50
4.5 程序举例	52
4.6 习题	56
第 5 章 循环结构程序设计	59
5.1 goto 语句	59
5.2 while 语句构成的循环结构	60
5.3 do-while 语句构成的循环结构	62
5.4 for 语句	63
5.5 循环的嵌套	65
5.6 break 语句和 continue 语句	66
5.6.1 break 语句	66
5.6.2 continue 语句	67
5.7 应用综合举例	67
5.8 习题	71
第 6 章 数组	74
6.1 一维数组	74
6.1.1 一维数组的定义	74

6.1.2 一维数组的初始化	75
6.1.3 一维数组元素的引用	76
6.1.4 一维数组程序举例	76
6.2 二维数组	78
6.2.1 二维数组的定义	78
6.2.2 二维数组的初始化	80
6.2.3 二维数组元素的引用	81
6.2.4 二维数组程序举例	81
6.3 习题	84
第 7 章 函数	87
7.1 概述	87
7.2 函数的定义	88
7.2.1 无参函数的定义	88
7.2.2 有参函数的定义	88
7.3 函数的参数和函数的返回值	89
7.3.1 形参和实参	89
7.3.2 数组作为函数参数	91
7.3.3 函数的返回值	92
7.4 函数的一般调用	93
7.4.1 函数调用的一般形式	93
7.4.2 函数调用的几种常用方法	94
7.4.3 函数的声明	94
7.5 函数的嵌套调用	96
7.6 函数的递归调用	99
7.7 局部变量和全局变量	101
7.7.1 局部变量	101
7.7.2 全局变量	103
7.8 变量的存储类别	104
7.8.1 动态存储方式与静态存储方式	104
7.8.2 auto 变量	105
7.8.3 用 static 声明的变量	105
7.8.4 register 变量	107
7.8.5 用 extern 声明外部变量	108
7.9 内部函数与外部函数	110
7.9.1 内部函数	110
7.9.2 外部函数	110
7.9.3 多个文件的运行方法	111
7.10 程序举例	111
7.11 习题	114
第 8 章 指针	117

8.1 指针变量的定义与应用	117
8.1.1 变量的地址和指针的概念	117
8.1.2 指针变量的概念	117
8.1.3 指针变量的定义	118
8.1.4 指针变量的赋值	119
8.1.5 对指针变量的操作	120
8.1.6 指针变量的运算	123
8.1.7 指针变量作为函数参数	124
8.2 一维数组的指针和指向一维数组的指针变量	127
8.2.1 数组元素的指针	127
8.2.2 通过指针引用数组元素	128
8.2.3 数组名作函数的参数	132
8.3 二维数组的指针和指向数组的指针变量	139
8.3.1 二维数组的指针	139
8.3.2 指向由多个元素组成的一维数组的指针变量	142
8.3.3 指针数组	143
8.3.4 二维数组名和指针数组作为函数的参数	144
8.3.5 指向指针的指针变量	145
8.3.6 main 函数的参数	147
8.4 函数指针和指向函数的指针变量	148
8.5 返回指针值的函数	149
8.6 有关指针的数据类型的小结和 void 指针类型	151
8.6.1 指针的数据类型小结	151
8.6.2 void 指针类型	152
8.7 习题	152
第9章 字符串	155
9.1 用一维数组存放字符串	155
9.1.1 C 语言对字符串的约定	155
9.1.2 通过赋初值的方式给一维数组赋字符串	156
9.1.3 程序执行时给一维数组赋字符串	156
9.2 指向字符串的指针	157
9.2.1 使指针指向一个字符串	157
9.2.2 使用字符串指针变量与字符数组的区别	157
9.3 字符数组的输入和输出	159
9.3.1 用格式说明符%s 进行整串输入和输出	159
9.3.2 调用 gets、puts 函数,从终端进行字符串的输入和输出	160
9.3.3 其他字符串处理函数	162
9.4 字符串数组	164
9.5 程序举例	165
9.6 习题	166

第 10 章 C 语言预处理	169
10.1 宏定义	169
10.1.1 不带参数的宏定义	169
10.1.2 带参数的宏定义	172
10.2 文件包含	175
10.3 条件编译	177
10.4 习题	179
第 11 章 结构体、共用体和用户定义类型	182
11.1 结构体类型	182
11.1.1 结构体的基本概念与基本操作	182
11.1.2 结构数组	187
11.1.3 结构与指针	190
11.1.4 结构与函数	192
11.2 链表基础	194
11.2.1 链表概述	194
11.2.2 动态链表	196
11.3 共用体	202
11.4 枚举类型	204
11.5 用 typedef 定义数据类型	205
11.6 习题	206
第 12 章 位运算	209
12.1 位运算符和位运算	209
12.1.1 按位与——&	209
12.1.2 按位或——	210
12.1.3 异或运算符——^	210
12.1.4 取反运算符——~	211
12.1.5 左移运算符——<<	211
12.1.6 右移运算符——>>	211
12.1.7 不同长度的数据进行位运算时的规则	211
12.2 位运算举例	212
12.3 习题	213
第 13 章 文件	215
13.1 C 文件概述	215
13.2 文件类型指针	216
13.3 文件的打开与关闭	216
13.3.1 文件的打开 (fopen 函数)	216
13.3.2 文件的关闭 (fclose 函数)	217
13.4 文件的读写	218
13.4.1 fputc 函数和 fgetc 函数 (putc 函数和 getc 函数)	218
13.4.2 fgets 函数和 fputs 函数	221

13.4.3 fread 函数和 fwrite 函数	222
13.4.4 fprintf 函数和 fscanf 函数	225
13.5 文件的定位	226
13.5.1 rewind 函数	226
13.5.2 fseek 函数	226
13.5.3 ftell 函数	227
13.6 文件检测函数	228
13.7 习题	228
附录	231
附录 I C 语言关键字	231
附录 II C 语言运算符的优先级与结合性	231
附录 III ASCII 字符编码表	232
附录 IV C 常用库函数浏览	232

第1章 C语言概述

1.1 C语言简介

1.1.1 计算机语言

计算机是按照人的要求进行计算的。我们必须以计算机所能接受的语言——计算机语言与其交流,告诉它要进行什么运算,对哪些数据运算,以及运算的先后次序。

计算机语言的发展经历了以下几个阶段。

1. 第一代语言

第一代语言是机器语言,它由计算机的指令系统组成。由于计算机只能识别和存储二进制的数据和指令,机器语言中的每一条语句(机器指令)实际就是由0和1组成的二进制代码,它由操作码的二进制编码和操作数的二进制编码组成。机器指令通常随计算机类型不同而不同,编写效率较低。例如计算:

$$a = \frac{2.58 + 5.36 \times 1.64}{1.27}$$

某类型计算机的机器语言如下:

数 据		程 序		
地 址	数	指 令		注 释
		操作码	地址码	
0500	2.58	0600	0501	取 5.36
0501	5.36	0601	0502	乘 1.64
0502	1.64	0602	0500	加 2.58
0503	1.27	0603	0503	除以 1.27
0504	a	0604	0504	存储 a
		0605	0504	打印 a
		0606	0000	停止

计算机可以直接识别和执行机器语言程序,执行效率高,但是人工编写机器语言程序却较繁琐,且易出错。

2. 第二代语言

第二代语言是汇编语言,也称为符号语言,它是用符号代替机器语言中的二进制编码,这样看起来较直观,不易出错。例如上例的汇编语言程序如下:

```
K: LDA    5.36
   MUL    1.64
   ADD    2.58
   DIV    1.27
   STA    a
```

```
PRINT a
```

```
STOP
```

与机器语言一样,不同类型的计算机具有不同的汇编语言。计算机不能直接识别和执行汇编语言程序,必须经过一个汇编程序(系统软件)转换成机器语言(目标程序)后才能执行,如图 1.1 所示。

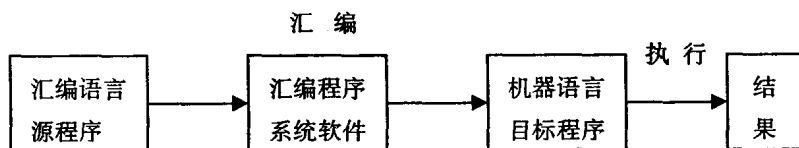


图 1.1 汇编语言的汇编与执行

汇编语言与机器语言是一一对应的,一个复杂的程序将包含许多汇编语言指令,写起来仍较繁琐。

机器语言和汇编语言都是面向机器的语言,因此人们称之为“低级语言”。

3. 第三代语言

第三代语言也称为算法语言。算法语言中的语句较为接近自然语言的英文字句,数据用十进制来表示,从而更容易为人们掌握和理解,并且它独立于计算机系统。用算法语言编写的程序可以在任何其他类型的计算机上执行,因此算法语言也称为“高级语言”。

常用的高级语言有 BASIC, FORTRAN, ALGOL, COBOL, PASCAL, C 语言等。

如上例,用 BASIC 语言可以写成如下程序。

```
10 a=(2.58+5.36*1.64)/1.27
```

```
20 PRINT a
```

```
30 END
```

高级语言书写的程序称为“源程序”,计算机也不能直接识别和执行,需要经过编译程序(系统软件)编译成目标程序(*.obj),再经过连接程序(系统软件)连接成可执行文件(*.exe)后,才能执行,如图 1.2 所示。

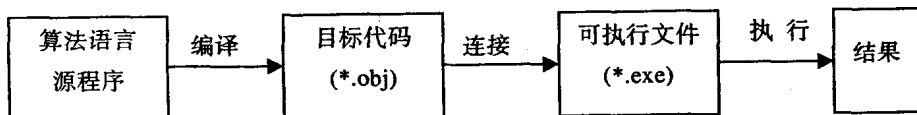


图 1.2 高级语言的编译与执行

4. 第四代语言

第三代语言是过程化语言,它必须描述问题是如何求解的。第四代语言是非过程化语言,它只需要描述需求解的问题是什么。

例如,需要将某班学生的成绩按高低的次序输出,用第四代语言只需写出这个要求即可,而不必写出排序的过程。数据库查询语言就可以做到这一点,所以我们可以把数据库查询语言看成是最简单的第四代语言。

5. 第五代语言

第五代语言主要是为人工智能领域设计的,称为智能化语言。例如,知识库系统、专家系统、推理工程、自然语言处理等等。在这些领域内将复杂的知识进行编码,使得计算机能从中

得出推论,从而显出较高的智力。

第五代语言还处于萌芽状态,PROLOG语言可以看做是它的一个例子。

1.1.2 C语言的发展过程和特点

C语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。在需要对操作系统和对硬件进行操作的场合,C语言明显优于其他高级语言,许多大型应用软件都是用C语言编写的。C语言的发展过程如下:

(1) 1970~1973年,C语言的诞生。C语言是1972年由美国的Dennis M. Ritchie设计发明的,并首次在使用UNIX操作系统的DEC PDP-11计算机上使用。它由早期的编程语言BCPL(basic combinational programming language)发展演变而来,既保持了BCPL和B语言简练的优点,又克服了它们的缺点(过于简单,数据无类型等)。

(2) 1973~1988年,C语言的发展。为了使UNIX操作系统推广,1977年Dennis M. Ritchie发表了不依赖于具体机器系统的C语言编译文本《可移植的C语言编译程序》。1978年,Brian W. Kernighan和Dennis M. Ritchie出版了名著*The C Programming Language*,从而使C语言成为目前世界上流行使用最广泛的高级程序设计语言。

(3) 1988年,C语言的成熟。随着微型计算机的日益普及,出现了许多C语言版本。由于没有统一的标准,使得这些C语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为C语言制定了一套ANSI标准,成为现行的C语言标准。

C语言发展迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件,如DBASE III PLUS、DBASE IV都是由C语言编写的。用C语言加上一些汇编语言子程序,就更能显示C语言的优势了,像PC-DOS、WORDSTAR等就是用这种方法编写的。归纳起来,C语言具有下列特点:

(1) 简洁、紧凑、方便、灵活。C语言一共只有32个关键字,9种控制语句,程序书写自由,主要用小写字母表示,并压缩了一切不必要的成分。

(2) 运算符丰富。C语言的运算符包含的范围很广泛,共有34个运算符。C语言把括号、赋值、强制类型转换等都作为运算符处理,从而使C的运算类型极其丰富,表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(3) 数据结构丰富。C语言的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂的数据类型的运算,并引入了指针概念,使程序效率更高。另外,C语言具有强大的图形功能,支持多种显示器和驱动器,且计算与逻辑判断能力强大。

(4) 是结构式语言。结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C语言是以函数形式提供给用户的,这些函数可方便调用,并具有多种循环、条件语句可控制程序流向,从而使程序完全结构化。

(5) 语法限制不太严格,程序设计自由度大。一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误,而C语言允许程序编写者有较大的自由度。

(6) 允许直接访问物理地址,可以直接对硬件进行操作。C语言把高级语言的基本结构和语句与低级语言的实用性结合起来,因此也有人称它是中级语言,因为它既具有高级语言的优越,又具有低级语言的许多功能。例如,C语言能够像汇编语言一样对位、字节和地址进

行操作,而这三者是计算机最基本的工作单元,C 语言可以用来写系统软件。这是一般高级语言不具备的。

(7) 程序生成代码质量高。程序执行效率高,一般只比汇编程序生成的目标代码效率低 10%~20%。

(8) 适用范围大,可移植性好。C 语言的一个突出优点就是适合于多种操作系统,如 DOS、UNIX,也适用于各种不同类型的计算机。

1.1.3 C 程序的结构和书写特点

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,体现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解组成一个 C 源程序的基本部分和书写格式。

例 1.1

```
main()  
{  
    printf("This is a c program.\n");  
}
```

程序的运行结果是在屏幕上输出一行字符:

This is a c program.

程序中的 main 被称作“主函数”,在任何一个 C 程序中都必须有一个且只有一个 main 函数,该函数中可以有很多语句,这些语句要用一对花括号{ }括起来。本例中,主函数只有一条输出语句,printf 是 C 语言中的输出函数,作用是向屏幕上输出信息,它将引号之间的内容按原样输出到屏幕。“\n”是换行符,作用是在输出“This is a c program”后回车,使光标换行。

例 1.2

```
main()                                /* 主函数 */  
{  
    int x,y,sum;                      /* 定义变量 */  
    x=1999; y=1;                     /* 给变量赋值 */  
    sum=x+y;                          /* 求和 */  
    printf("The sum is %d\n",sum);    /* 输出结果 */  
}
```

本程序的作用是求 x+y 的和,求和的结果赋值给变量 sum,并将结果输出到屏幕上。程序的运行结果:

The sum is 2000

在程序中,/*……*/ 为注释部分,不参与编译,也不影响程序的运行,这些注释只是用来帮助人们阅读程序的,注释的具体内容夹在“/*”和“*/”之间,注释部分可以加在程序的任何部分;第 3 行是声明部分,指定了 x, y 以及 sum 为整型(int)变量;第 4 行是赋值语句,使变量 x 和 y 的值分别为 1999 和 1;第 5 行将 x+y 的值送入变量 sum 中;第 6 行是利用 printf 函数将结果输出到屏幕,输入语句中的“%d”为输入输出的“格式字符串”,用来指明数据类型和格式,“%d”表示“十进制整数类型”,在执行输出时,“%d”的位置将用一个十进制的

整型数值代替, 本例中用 sum 的值 2000 来代替 “%d” 这个位置上的内容。

例 1.3

```
main( )                                /* 主函数 */
{
    int x,y,z;                          /* 变量说明 */
    int max(int a,int b);               /* 函数说明 */
    scanf("%d, %d",&x,&y);             /* 输入变量 x 和 y 值 */
    z=max(x,y);                         /* 调用 max 函数, 将得到的值赋给 z */
    printf("maxnum=%d",z);             /* 输出 z 的值 */
}
int max(int a,int b)                   /* 定义 max 函数, 函数值为整型, 形式参数 a, b 为整型 */
{
    int c;                             /* 在此定义 max 函数中要用到的变量 c */
    if(a>b) c=a;
    else c=b;
    return c;                          /* 把结果返回主调函数 */
}
```

上面例中程序的功能是由用户输入两个整数, 程序执行后输出其中较大的数。程序运行情况:

```
3, 5↵          (输入 3 和 5 给 x 和 y, ↵表示回车)
maxnum=5        (输出 z 的值)
```

本程序由两个函数组成, 主函数和 max 函数, 函数之间是并列关系。可从主函数中调用其他函数。max 函数的功能是比较两个数, 然后把较大的数返回给主函数。max 函数是一个用户自定义函数, 因此在主函数中要给出说明 (程序第三行)。可见, 在程序的说明部分中, 不仅可以有变量说明, 还可以有函数说明, 关于函数的详细内容将在以后的章节介绍。

上例中程序的执行过程是:

(1) 首先由用户输入两个数。

(2) 由 scanf 函数接收这两个数并送入变量 x, y 中, scanf 函数与 printf 函数一样, 也是 C 语言提供的标准输入输出函数。语句中 &x 和 &y 的 “&” 符号表示 “取地址”, 用户输入的 3 和 5, 将分别输入到 x 和 y 所标志的存储单元中。

(3) 调用 max 函数, 并把 x, y 的值传送给 max 函数的形式参数 a, b, 在 max 函数中比较 a, b 的大小, 把大者 (用变量 c 表示) 作为函数返回值。

(4) 将函数返回值赋值给主函数 main 中的变量 z, 最后在屏幕上输出 z 的值。

通过观察上面的三个程序, 可归纳出 C 语言的基本特点:

(1) C 程序由函数构成, 函数相当于其他语言中的子程序, 是 C 语言调用的基本单位。一个 C 程序至少包含 main 函数, 也可包含一个 main 函数和其他若干个函数。被调用的函数可以是系统提供的库函数如 printf 和 scanf 函数, 也可以是用户自己编制设计的函数, 像上面的 max 函数。标准 C 中提供了 100 多个库函数, Turbo C 和 MS C 4.0 提供了 300 多个库函数。利用这些现成的库函数, 我们可以非常轻松编写功能强大的程序。C 程序的函数式结构使得 C 程序非常容易实现模块化, 便于阅读和维护。

(2) 函数由两大部分组成。第一部分是函数的首部,即函数的第一行,包括函数的名称、函数的返回值类型、函数的参数名和数据类型。第二部分是函数体,由大括号括起来的若干变量定义和语句构成。

(3) 一个 C 程序总是从 main 函数开始执行,不论 main 函数在整个程序中的位置如何。也就是说,可以将 main 函数放在程序的任何位置。

(4) C 程序书写格式自由,一行内可以包含若干条语句,也可以将一条语句写成多行。

(5) 每个语句和数据定义的最后必须加一个分号“;”。

(6) C 语言没有专门的输入输出语句,输入输出的操作是通过 scanf 和 printf 等库函数实现的。

(7) 可以用“/*……*/”对 C 程序中的任何部分作注解,好的程序都有必要的注释用来提高程序的可读性。

从书写清晰,便于阅读、理解和维护的角度出发,在书写程序时应遵循以下规则:

(1) 一个说明或一个语句占一行。

(2) 用{}括起来的部分,通常表示了程序的某一层结构。{}一般与该结构语句的第一个字母对齐,可单独占一行。

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便看起来更加清晰,增加程序的可读性。

在编程时应力求遵循这些规则,以养成良好的编程风格。

1.2 Turbo C 2.0 的基本操作

1.2.1 Turbo C 2.0 的集成开发环境

我们在上一节已经介绍,C 语言是一种高级语言,用户用 C 语言编写的程序称为源程序,C 语言所编写的源程序文件扩展名一般为“c”。计算机不能直接执行 C 的源程序,必须将源程序翻译成二进制目标程序。翻译工作是由一个系统程序完成的,这个程序称为编译程序。翻译的过程称为编译,编译的结果称为目标程序,目标程序文件名的后缀为“obj”或“o”。程序翻译成目标程序后,便可进行连接。连接的目的是使程序变成在计算机上可以执行的最终形式。在这一阶段,目标程序(.obj)将与系统的库函数和其他目标程序连接起来,连接的结果称为可执行目标程序,文件名一般以“.exe”结尾。

Turbo C 是微机上广泛采用的集成开发环境,它具有方便、直观的界面,将文本编辑、程序编译、连接以及程序运行一体化,大大方便了程序的开发。本节主要介绍运用 Turbo C 进行程序设计的基本步骤及如何编译、调试和运行源程序。

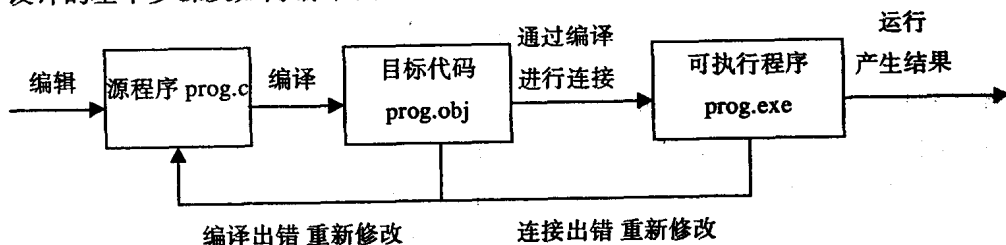


图 1.3 源程序编辑运行过程

例如,我们已经完成源程序的编辑工作,并命名为 prog.c,通过图 1.3 我们可以清楚地看到该程序在微机内的运行步骤。

1.2.2 用 Turbo C 运行 C 程序的步骤

为了能直接使用 Turbo C,必须先将 Turbo C 编译程序装入磁盘某一目录下。

(1) 调用 Turbo C。启动机器后,进入 DOS 操作界面,将当前目录设定为 Turbo C 的安装目录,假设 Turbo C 被安装在 C 盘的 TC 目录下,则应将当前目录设定为 C:\TC>,然后键入 tc 命令即可:

C:\TC>tc ↵

进入 Turbo C 2.0 集成开发环境中后,屏幕上显示:

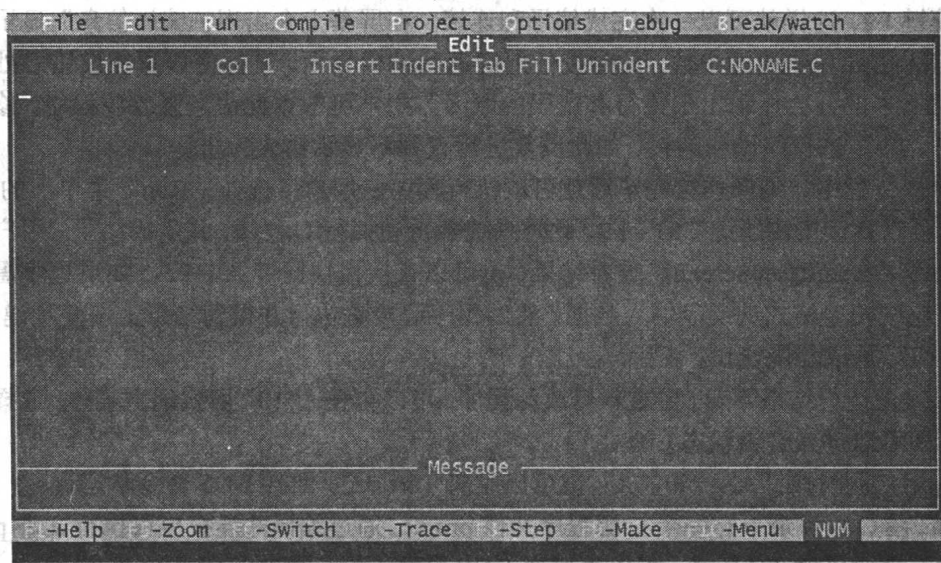


图 1.4 Turbo C 2.0 的操作界面

从图 1.4 可以看出,最上一行为 Turbo C 2.0 主菜单,中间窗口为编辑区,接下来是信息窗口,最底下一行为参考行。这 4 个窗口构成了 Turbo C 2.0 的主屏幕,以后的编程、编译、调试以及运行都将在这个主屏幕中进行。

主菜单在 Turbo C 2.0 主屏幕最上一行,显示下列内容:

File Edit Run Compile Project Options Debug Break/watch

分别代表文件操作、编辑、运行、编译、项目文件、选项、调试、中断/观察等功能菜单。除 Edit 外,其他各项均有子菜单,只要用 Alt 加上某项中第一个字母,就可进入该项的子菜单中。

用户可以通过选择“File”菜单下的“New”命令新建一个 C 源文件,缺省文件名为 NONAME.C,存盘时可改名。通过该菜单下的“Load”命令可以调用一个已存在的源文件,也可以用类似 DOS 的通配符(如*.C)来进行列表选择,也可以装入其他扩展名的文件,只要给出文件名(或只给路径)即可。该项的快捷键为 F3,即只要按 F3 即可进入该项,而不需要先进入 File 菜单再选此项。

源文件编辑结束后,用户可以通过“File”菜单下的“Save”命令将编辑区中的文件存盘,若文件名是 NONAME.C 时,将询问是否更改文件名,其快捷键为 F2。

(2) 编辑源文件。在编辑状态下(Edit)可以根据需要输入或修改源程序。