

UNIX

BASIC

# BASIC

## 语言

(三次修订本)

谭浩强 田淑清 编著

```
80 PRINT "COCK="; X; "RABBIT="; Y
```

科学普及出版社

《电脑报》计算机普及教程  
之 四

# C 语言速成

何宗琦 编著

成都科技大学出版社

(川)新登字号 015 号

## 内 容 简 介

本书以国家教委考试中心颁发的《全国计算机等级考试大纲》所规定为范围的依据,介绍了C语言的基本内容及C语言的基本内容及C语言程序设计的基本方法;还介绍了Turbo C2.0集成开发环境的使用。作者以C语言的特点出发把C语言的基本概念、语法规则、程序设计技巧和上机操作的方法有机地结合在一起。本书概念清晰,例题详尽,图文并茂,上机操作方法明确,各章都给出了小结和习题。初学者可在短期内掌握C语言的基本概念,编程方法,并掌握C的开发环境及上机操作的具体方法和步骤。以速成的方式达到全国计算机等级考试大纲提出的笔试和机试的要求。

本书特别适合于自学,可作为参加全国计算机等级考试的自学教材,也可作为高、中等学校和各类培训班C语言课程的教材使用。

### C 语言速成

何宗琦 编著

责任编辑: 哈 森 黄益民

技术编辑: 周 勃

封面设计: 易高原

\* \* \* \* \*

成都科技大学出版社出版

全国新华书店经销

《电脑报》社照排部 排版

达川新华印刷厂 印刷

开本: 787×1092 毫米 1/16 印张: 16.75 字数: 437 千字

1994 年 12 月第 1 版 1994 年 12 月第 1 次印刷

ISBN7-5616-2980-X/TP·114

---

定 价: 12.00 元

# 目 录

|                            |    |
|----------------------------|----|
| 第一章 C语言概论 .....            | 1  |
| § 1.1 C语言的发展过程 .....       | 1  |
| § 1.2 C源程序的结构特点 .....      | 1  |
| § 1.3 C语言的字符集和词汇 .....     | 5  |
| 1.3.1 C语言字符集 .....         | 5  |
| 1.3.2 C语言词汇 .....          | 6  |
| § 1.4 C源程序的编辑、编译和运行 .....  | 7  |
| 1.4.1 C语言程序的开发过程 .....     | 7  |
| 1.4.2 Turbo C的启动和退出 .....  | 8  |
| 1.4.3 Turbo C主菜单屏幕 .....   | 9  |
| 1.4.4 源程序的编辑方法 .....       | 11 |
| 1.4.5 源程序的编译和运行方法 .....    | 14 |
| 本章小结 .....                 | 17 |
| 思考题 .....                  | 19 |
| 上机题 .....                  | 19 |
| 第二章 数据类型、运算符与表达式 .....     | 20 |
| § 2.1 基本数据类型 .....         | 20 |
| 2.1.1 C语言的数据类型 .....       | 20 |
| 2.1.2 整型量 .....            | 21 |
| 2.1.3 实型量 .....            | 24 |
| 2.1.4 字符型量 .....           | 25 |
| 2.1.5 符号常量 .....           | 28 |
| 2.1.6 变量的初值和类型转换 .....     | 29 |
| § 2.2 基本运算符和表达式 .....      | 31 |
| 2.2.1 运算符的种类、优先级和结合性 ..... | 31 |
| 2.2.2 算术运算符和算术表达式 .....    | 32 |
| 2.2.3 赋值运算符和赋值表达式 .....    | 34 |
| 2.2.4 逗号运算符和逗号表达式 .....    | 36 |
| 本章小结 .....                 | 36 |
| 思考题 .....                  | 37 |
| 上机题 .....                  | 38 |
| 第三章 C程序设计初步 .....          | 39 |
| § 3.1 简单的C程序 .....         | 39 |
| 3.1.1 C程序的语句 .....         | 39 |
| 3.1.2 赋值语句 .....           | 40 |

|                           |        |
|---------------------------|--------|
| 3.1.3 数据输出语句.....         | 41     |
| 3.1.4 数据输入语句.....         | 44     |
| 3.1.5 程序举例.....           | 48     |
| § 3.2 分支结构程序.....         | 49     |
| 3.2.1 关系运算符和表达式.....      | 49     |
| 3.2.2 逻辑运算符和表达式.....      | 50     |
| 3.2.3 IF 语句.....          | 51     |
| 3.2.4 SWITCH 语句.....      | 57     |
| 3.2.5 程序举例.....           | 59     |
| § 3.3 循环结构程序.....         | 60     |
| 3.3.1 WHILE 语句.....       | 60     |
| 3.3.2 DO--WHILE 语句.....   | 62     |
| 3.3.3 FOR 语句.....         | 63     |
| 3.3.4 转移语句.....           | 67     |
| 3.3.5 程序举例.....           | 69     |
| 本章小结.....                 | 70     |
| 思考题.....                  | 72     |
| 上机题.....                  | 73     |
| <br>第四章 数 组.....          | <br>74 |
| § 4.1 数组类型说明和数组元素的表示..... | 74     |
| 4.1.1 数组类型说明.....         | 74     |
| 4.1.2 数组元素的表示方法.....      | 75     |
| § 4.2 数组的赋值.....          | 75     |
| 4.2.1 数组初始化赋值.....        | 76     |
| 4.2.2 动态赋值.....           | 76     |
| § 4.3 二维数组.....           | 77     |
| 4.3.1 二维数组类型说明.....       | 78     |
| 4.3.2 二维数组元素的表示方法.....    | 79     |
| 4.3.3 二维数组的初始化.....       | 80     |
| § 4.4 字符数组与字符串.....       | 81     |
| 4.4.1 字符数组.....           | 81     |
| 4.4.2 字符串.....            | 82     |
| 4.4.3 字符串常用函数.....        | 84     |
| § 4.5 程序举例.....           | 87     |
| 本章小结.....                 | 90     |
| 思考题.....                  | 91     |
| 上机题.....                  | 92     |
| <br>第五章 函 数.....          | <br>93 |

|                         |     |
|-------------------------|-----|
| § 5.1 概述                | 93  |
| § 5.2 函数的定义和调用          | 94  |
| 5.2.1 函数定义的一般形式         | 95  |
| 5.2.2 函数调用的一般形式         | 96  |
| 5.2.3 函数的参数和函数的值        | 98  |
| 5.2.4 函数说明              | 99  |
| 5.2.5 数组作为函数参数          | 101 |
| § 5.3 函数的嵌套调用           | 106 |
| § 5.4 函数的递归调用           | 108 |
| § 5.5 变量的作用域和存储类型       | 112 |
| 5.5.1 变量的作用域            | 112 |
| 5.5.2 变量的存储类型           | 116 |
| § 5.6 内部函数和外部函数         | 121 |
| 本章小结                    | 122 |
| 思考题                     | 123 |
| 上机题                     | 125 |
| <b>第六章 指 针</b>          | 126 |
| § 6.1 指针和指针型变量          | 126 |
| 6.1.1 指针的基本概念           | 126 |
| 6.1.2 指针变量的类型说明         | 128 |
| 6.1.3 指针变量的赋值和运算        | 128 |
| § 6.2 数组指针变量            | 135 |
| 6.2.1 数组指针变量的说明和使用      | 135 |
| 6.2.2 数组名和数组指针变量作函数参数   | 138 |
| 6.2.3 指向多维数组的指针变量       | 139 |
| § 6.3 字符串指针变量           | 142 |
| 6.3.1 字符串指针变量的说明和使用     | 142 |
| 6.3.2 使用字符串指针变量与字符数组的区别 | 145 |
| § 6.4 函数指针变量            | 146 |
| § 6.5 指针型函数             | 147 |
| § 6.6 指针数组              | 149 |
| 6.6.1 指针数组的说明与使用        | 149 |
| 6.6.2 MAIN 函数的参数        | 154 |
| § 6.7 指向指针的指针变量         | 155 |
| 本章小结                    | 157 |
| 思考题                     | 159 |
| 上机题                     | 160 |
| <b>第七章 结构与联合</b>        | 161 |

|                             |     |
|-----------------------------|-----|
| § 7.1 结构类型定义和结构变量说明 .....   | 161 |
| § 7.2 结构变量的使用和初始化 .....     | 163 |
| 7.2.1 结构变量成员的表示方法 .....     | 163 |
| 7.2.2 结构变量的赋值 .....         | 164 |
| 7.2.3 结构变量的初始化 .....        | 164 |
| § 7.3 结构数组 .....            | 165 |
| § 7.4 结构指针变量 .....          | 168 |
| 7.4.1 结构指针变量的说明和使用 .....    | 168 |
| 7.4.2 结构数组指针变量 .....        | 170 |
| 7.4.3 结构指针变量作函数参数 .....     | 171 |
| § 7.5 动态存储分配与链表 .....       | 172 |
| 7.5.1 动态存储分配 .....          | 172 |
| 7.5.2 链表的概念 .....           | 174 |
| 7.5.3 链表的操作 .....           | 175 |
| § 7.6 联合 .....              | 183 |
| 7.6.1 联合类型的定义和联合变量的说明 ..... | 183 |
| 7.6.2 联合变量的赋值和使用 .....      | 184 |
| 本章小结.....                   | 186 |
| 思考题.....                    | 186 |
| 上机题.....                    | 188 |

|                             |     |
|-----------------------------|-----|
| <b>第八章 其它数据类型</b> .....     | 189 |
| § 8.1 枚举 .....              | 189 |
| 8.1.1 枚举类型的定义和枚举变量的说明 ..... | 189 |
| 8.1.2 枚举类型变量的赋值和使用 .....    | 190 |
| § 8.2 位运算与位域 .....          | 191 |
| 8.2.1 位运算 .....             | 191 |
| 8.2.2 位域 .....              | 194 |
| § 8.3 类型定义符TYPEDEF .....    | 197 |
| 本章小结.....                   | 198 |
| 思考题.....                    | 198 |
| 上机题.....                    | 198 |

|                      |     |
|----------------------|-----|
| <b>第九章 预处理</b> ..... | 199 |
| § 9.1 宏定义 .....      | 199 |
| 9.1.1 无参宏定义 .....    | 199 |
| 9.1.2 带参宏定义 .....    | 202 |
| § 9.2 文件包含 .....     | 206 |
| § 9.3 条件编译 .....     | 208 |
| 本章小结.....            | 210 |

|                              |            |
|------------------------------|------------|
| 思考题                          | 210        |
| 上机题                          | 211        |
| <b>第十章 文 件</b>               | <b>212</b> |
| § 10.1 文件与文件指针               | 212        |
| 10.1.1 文件的基本概念               | 212        |
| 10.1.2 文件指针                  | 212        |
| § 10.2 文件的打开与关闭              | 213        |
| 10.2.1 文件打开函数FOPEN           | 213        |
| 10.2.2 文件关闭函数FCLOSE          | 214        |
| § 10.3 文件的读写                 | 214        |
| 10.3.1 字符读写函数FGETC和FPUTC     | 215        |
| 10.3.2 字符串读写函数FGETS和FPUTS    | 218        |
| 10.3.3 数据块读写函数FREAD和FWRITE   | 220        |
| 10.3.4 格式化读写函数FSCANF和FPRINTF | 222        |
| § 10.4 文件的随机读写               | 223        |
| 10.4.1 文件定位                  | 223        |
| 10.4.2 文件的随机读写               | 224        |
| § 10.5 文件检测函数                | 225        |
| § 10.6 C库文件                  | 225        |
| 本章小结                         | 227        |
| 思考题                          | 227        |
| 上机题                          | 227        |
| <b>附录A Turbo C 热键</b>        | <b>228</b> |
| <b>附录B Turbo C 编辑命令</b>      | <b>229</b> |
| <b>附录C 运算符的优先级和结合性</b>       | <b>231</b> |
| <b>附录D C 语言语法汇总</b>          | <b>232</b> |
| <b>附录E Turbo C 库函数</b>       | <b>237</b> |
| <b>附录F 编译错误信息</b>            | <b>248</b> |

# 第一章 C 语言概论

## § 1.1 C 语言的发展过程

C 语言是在 70 年代初问世的。一九七八年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的“THE C PROGRAMMING LANGUAGE”一书。通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准 C 语言,后来由美国国家标准学会在此基础上制定了一个 C 语言标准,于一九八三年发表。通常称之为 ANSI C。

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识,到了八十年代, C 开始进入其它操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用。成为当代最优秀的程序设计语言之一。

C 语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构。它还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于 C 语言实现了对硬件的编程操作,因此 C 语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发,也适合于应用软件的开发。

此外, C 语言还具有效率高,可移植性强等特点。因此广泛地移植到了各类各型计算机上,从而形成了多种版本的 C 语言。

目前最流行的 C 有以下几种:

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- AT&T C

这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便,完美。

本书以微机上最为流行的 Turbo C 2.0 为开发环境,介绍 C 语言及其程序设计。

在 C 的基础上,一九八三年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。C++ 目前流行的最新版本是 Borland C++ 4.0, Symantec C++ 6.1, 和 Microsoft Visual C++ 1.5。C++ 提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性,掌握起来有一定难度。但是, C 是 C++ 的基础, C++ 语言和 C 语言在很多方面是兼容的。因此,掌握了 C 语言,再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。

## § 1.2 C 源程序的结构特点

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,表现了 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到组成一个 C 源程序的基本部分和书写格式。

[例 1.1] 输出字符串 HELLO, WORLD

```
main()
{
    printf("HELLO WORLD\n");
}
```

运行结果:

HELLO WORLD

本程序的功能是在屏幕上输出字符串“HELLO.WORLD”。程序共有四行。第一行main 是主函数的函数名,表示这是一个主函数。第二行和第四行为{和},这是一对括号,表示在括号中的内容是主函数体。在定义一个函数时,其函数体必须用{}括起来。本程序的主函数体中只有一行,这一行的功能就是输出字符串。它是一个函数调用语句,printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数,可在程序中直接调用。例1.1 是一个最简单的C 语言源程序。每一个C 源程序都必须有,且只能有一个主函数(main 函数)。

[例1.2]求一个数的正弦值。

```
#include<math.h>                                /* 预处理命令 */
#include<stdio.h>
main()
{
    double x,s;                                    /* 变量说明 */
    printf("input number:\n");                    /* 以下为执行部分 */
    scanf("%lf",&x);
    s=sin(x);
    printf("sine of %lf is %lf\n",x,s);
}
```

运行结果:

```
input number:
5
sine of 5.000000 is -0.958924
```

例1.2 程序的功能是从键盘输入一个数x,求x 的正弦值,然后输出结果。和例1.1 不同的是在main()之前增加了两行。这两行称为预处理命令(详见第九章)。预处理命令还有其它几种,这里的include 称为文件包含命令,其意义是把尖括号<>或引号“”内指定的文件包含到本程序来,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h。因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此,凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了三个库函数:输入函数scanf,正弦函数sin,输出函数printf。sin 函数是数学函数,其头文件为math.h 文件,因此在程序的主函数前用include 命令包含了math.h。scanf 和printf 是标准输入输出函数,其头文件为stdio.h,在主函数前也用include 命令包含了stdio.h 文件。需要说明的是,C 语言规定对scanf 和printf 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第二行的包含命令#include<stdio.h>。同样,在例1.1 中使用了printf 函数,也省略了包含命令。

在例1.2 的主函数体中又分为两部分,一部分为说明部分,另一部分为执行部分。说明是指变量的类型说明。例1.1 中未使用任何变量,因此无说明部分。C 语言规定,源程序中所有用到的变量

都必须先说明,后使用,否则将会出错。这一点是编译型高级程序设计语言的一个特点,与解释型的BASIC语言是不同的。说明部分是C源程序结构中很重要的组成部分。本例中使用了两个变量x,s,用来表示输入的自变量和sin函数值。由于sin函数要求这两个量必须是双精度浮点型,故用类型说明符double来说明这两个变量。

说明部分后的四行为执行部分或称为执行语句部分,用以完成程序的功能。执行部分的第一行是输出语句,调用printf函数在显示器上输出提示字符串,请操作人员输入自变量x的值。第二行为输入语句,调用scanf函数,接受键盘上输入的数并存入变量x中。第三行是调用sin函数并把函数值送到变量s中。第四行是用printf函数输出变量s的值,即x的正弦值。程序结束。

运行本程序时,首先在显示器屏幕上给出提示串input number,这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数,如5,按下回车键,接着在屏幕上给出计算结果。

在前两个例子中用到了输入和输出函数scanf和printf,在第三章中我们要详细介绍。这里我们先简单介绍一下它们的格式,以便下面使用。

scanf和printf这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此,这两个函数在括号中的参数表都由以下两部分组成:

“格式控制串”,参数表

格式控制串是一个字符串,必须用双引号括起来,它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第三章。在printf函数中还可以在格式控制串内出现非格式控制字符,这时在显示屏幕上将原文照印。参数表中给出了输入或输出的量。当有多个量时,用逗号间隔。例如:

```
printf("sine of .%lf is %lf\n",x,s);
```

其中%.lf为格式字符,表示按双精度浮点数处理。它在格式串中两次出现,对应了x和s两个变量。其余字符为非格式字符则照原样输出在屏幕上。

[例1.3]输入两个整数,输出其中的大数。

```
main()                                /* 主函数 */
{
    int max(int a,int b);              /* 函数说明 */
    int x,y,z;                         /* 数量说明 */
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);               /* 输入x,y 值 */
    z=max(x,y);                       /* 调用max 函数 */
    printf("maximum=%d",z);            /* 输出 */
}

int max(int a,int b)                  /* 定义max 函数 */
{
    if(a>b)return a;
    else return b;                    /* 把结果返回主调函数 */
}
```

运行结果为:

```
input two numbers:
5 27
maximum=27
```

例1.3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。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它函数。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第五章介绍。在程序的每行后用/\* 和 \*/括起来的内容为注释部分,程序不执行注释部分。

例 1.3 程序的执行过程是,首先在屏幕上显示提示串,请用户输入两个数,回车后由 scanf 函数语句接收这两个数送入变量 x,y 中,然后调用 max 函数,并把 x,y 的值传送给 max 函数的参数 a,b。在 max 函数中比较 a,b 的大小,把大者返回给主函数的变量 z,最后在屏幕上输出 z 的值。

[例 1.4] 含有两个源文件的源程序

C 语言允许一个源程序由多个源文件组成。但只能有一个主函数 main。例 1.3 是由一个源文件组成的源程序,也可把它改为两个源文件。

源文件一,文件名为 max1.c。

```
main()
{
    extern int max(int a,int b);          /* max 是本文件之外的外部函数 */
    int x,y,z;
    printf("input two numbers:\n");
    scanf("%d,%d",&x,&y);
    z=max(x,y);
    printf("maximum=%d",z);
}
```

源文件二,文件名为 max2.c。

```
int max(int a,int b)
{
    if(a>b) return a;
    else return b;
}
```

这两个源文件可以分别编辑,编译后再链接在一起生成可执行程序。其结果与例 1.3 完全相同。但在源文件 max1.c 中的主函数内必须说明 max 函数是外部函数(加说明符 extern),即被调函数在本文件之外的其它文件中。

由于 C 语言允许由多个源文件组成一个源程序,因此十分便于组织模块化程序设计,并且便于调试和维护,从而提高了编程的效率。

从对上述例题的分析,可以归纳 C 源程序的结构特点如下:

1. 一个 C 语言源程序可以由一个或多个源文件组成;
2. 每个源文件可由一个或多个函数组成。
3. 一个源程序不论由多少个文件组成,都有一个且只能有一个 main 函数,即主函数。
4. 函数由函数头和函数体两部分组成。函数头由函数名和参数表两部分组成。例如,

max(int a,int b);

其中 max 为函数名,括号内为参数表,该参数表中给出两个整型变量 a,b。函数体内可以含有两个部分即说明部分和执行(语句)部分。只有经过说明的变量和函数(用户自定义函数)才能在语句中被使用。

函数分为两种,一种为 C 库函数,由 C 系统提供,使用前虽不必说明,但必须在文件前把该函数

所在的头文件用包含命令include 包含到文件中去。另一种是用户自定义函数,它们在被调用之前应先予以说明。各函数之间是并列的,没有从属关系,函数之间可以互相调用(但主函数不能被其它函数调用)。

5. 源程序中可以有预处理命令(include 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

6. 每一个说明,每一个语句都必须以分号结尾。但预处理命令,函数头和花括号}之后不能加分号。

7. 标识符,关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

C 源程序的结构可如图 1. 1 所示

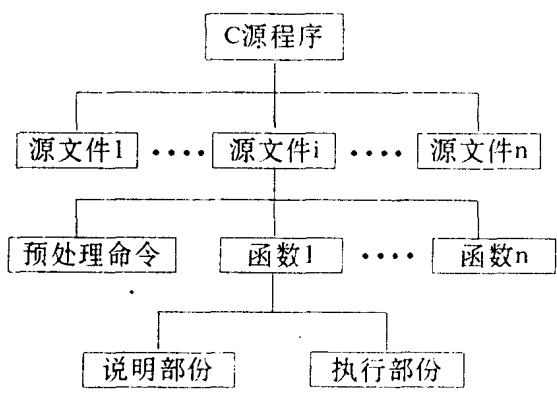


图1. 1 C源程序的结构

在前面几个例题中,都是每一个说明或语句占用一行,甚至花括号{和}也占用一行。实际上C语言允许在一行中连续书写多个说明和语句,也允许一个说明或语句占用多行。例如,例1. 1 也可改写为:

```
main(){printf("HELLO,WORLD\n");}
```

但从书写清晰,便于阅读,理解,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,并单独占一行。
- 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. 3 C 语言的字符集和词汇

#### 1. 3. 1 C 语言字符集

字符是组成语言的最基本的元素。C 语言字符集由字母,数字,空格,标点和特殊字符组成。在字符常量,字符串常量和注释中还可以使用汉字或其它可表示的图形符号。

##### 1. 字母

小写字母a~z 共26个

大写字母A~Z 共26个

## 2. 数字

0~9 共10个

## 3. 空白符

空格符,制表符,换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其它地方出现时,只起间隔作用,编译程序对它们忽略不计。因此在程序中使用空白符与否,对程序的编译不发生影响,但在程序中适当地使用空白符将增加程序的清晰性和可读性。

## 4. 标点和特殊字符

C语言允许使用多种标点符号和特殊字符,通常用来表示运算符或格式符,表1.1列出了这些符号。这些符号在不同的地方其含义不一定相同,在此后各章中将陆续介绍。

表1.1 C语言标点和特殊字符

| 字符 | 名称   | 字符 | 名称   | 字符 | 名称  | 字符 | 名称   | 字符 | 名称   |
|----|------|----|------|----|-----|----|------|----|------|
| ,  | 逗号   | .  | 点号   | ;  | 分号  | :  | 冒号   | '  | 单引号  |
| "  | 双引号  | (  | 左括号  | )  | 右括号 | [  | 左方括号 | ]  | 右方括号 |
| {  | 左花括号 | }  | 右花括号 | <  | 小于符 | >  | 大于符  | ?  | 问号   |
| !  | 惊叹号  |    | 竖线   | /  | 斜线  | \  | 反斜线  | ~  | 非运算符 |
| _  | 下划线  | #  | 数字符  | %  | 百分号 | &  | 和号   | ^  | 位异或符 |
| *  | 乘号   | +  | 加号   | -  | 减号  | =  | 等号   |    |      |

### 1.3.2 C语言词汇

在C语言中使用的词汇分为六类:标识符,关键字,运算符,分隔符,常量,注释符等。

#### 1. 标识符

在程序中使用的变量名,函数名,标号等统称为标识符。除库函数的函数名由系统定义外,其余都由用户自定义。C规定,标识符只能是字母(A~Z,a~z),数字(0~9),下划线(\_)组成的字符串,并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

a, x, \_3x, BOOK \_1, sum5

以下标识符是非法的:

3s                      以数字开头

s \* T                    出现非法字符 \*

-3x                      以减号开头

bowy - 1                出现非法字符-(减号)

在使用标识符时还必须注意以下几点:

(1)标准C不限制标识符的长度,但它受各种版本的C语言编译系统限制,同时也受到具体机器的限制。例如在某版本C中规定标识符前八位有效,当两个标识符前八位相同时,则被认为是同一个标识符。

(2)在标识符中,大小写是有区别的。例如BOOK和book是两个不同的标识符。

(3)标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名应尽量有

相应的意义,以便于阅读理解,作到“顾名思义”。

## 2. 关键字

关键字是由C语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C语言的关键字分为以下几类:

### (1)类型说明符

用于定义、说明变量、函数或其它数据结构的类型。如前面例题中用到的int,double等

### (2)语句定义符

用于表示一个语句的功能。如例1.3中用到的if else就是条件语句的语句定义符。

### (3)预处理命令字

用于表示一个预处理命令。如前面各例中用到的include。

## 3. 运算符

C语言中含有相当丰富的运算符。运算符与变量,函数一起组成表达式,表示各种运算功能。运算符由一个或多个字符组成。

## 4. 分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中,分隔各个变量。空格多用于语句各单词之间,作间隔符。在关键字,标识符之间必须要有一个以上的空格符作间隔,否则将会出现语法错误,例如把int a;写成inta;C编译器会把inta当成一个标识符处理,其结果必然出错。

## 5. 常量

C语言中使用的常量可分为数字常量,字符常量,字符串常量,符号常量,转义字符等多种。在第二章中将专门给予介绍。

## 6. 注释符

C语言的注释符是以“/\*”开头并以“\*/”结尾的串。在“/\*”和“\*/”之间的即为注释。程序编译时,不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来,使翻译跳过不作处理,待调试结束后再去掉注释符。

# § 1.4 C源程序的编辑,编译和运行

## 1.4.1 C语言程序的开发过程

C语言程序的开发过程指的是提出一个问题,用C语言编程,然后将编好的程序进行编译链接和运行,最后获得结果的全过程。这个开发过程包含了如下阶段:

1. 根据提出的问题,找出解决问题的具体算法。确定算法是解决问题的关键所在。算法是求解问题的具体计算过程,通常可用流程图表示。

2. 按照确定的算法,用C语言编写出源程序。并上机编辑成源程序文件。源程序文件是文本文件,其扩展名为.C。可用DOS命令type显示源程序文件。

3. 对编辑好的源程序进行编译,并反复修改调试,直到没有编译错误为止,生成正确的目标文件。目标文件是二进制代码文件,扩展名为.OBJ。

4. 对目标文件进行链接,把目标文件代码和库函数或其它目标文件链接起来。生成可执行文件。可执行文件是二进制代码文件,其后缀为.EXE。可在DOS系统下直接执行.EXE文件。

上述各步开发过程可表示如图1.2。设源程序文件名为test。

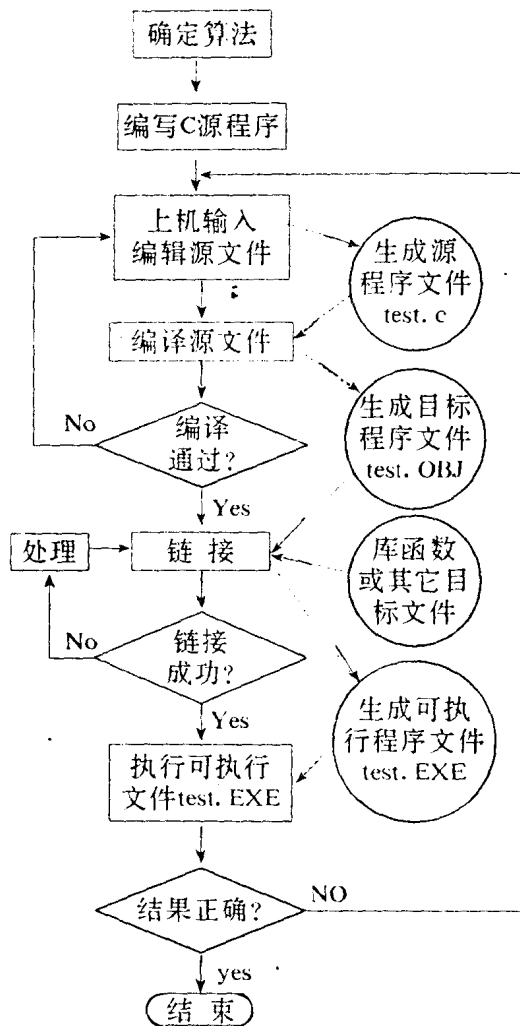


图1. 2 程序开发过程

为了使读者能掌握上机操作的具体方法,在以下几个小节中,介绍微机中最流行的Turbo C 集成开发环境下的编辑,编译和运行方法。

应该说明的是,Turbo C 集成开发环境的功能非常丰富,命令也很多。为了突出重点,本章只介绍必须使用的基本命令及其意义,以便上机操作。所介绍的命令需要反复上机实践摸索,才能真正理解、掌握。

## 1.4.2 Turbo C 的启动和退出

### 1. 启动

Turbo C 集成开发环境的主控文件为TC. EXE。在DOS 状态下,键入:

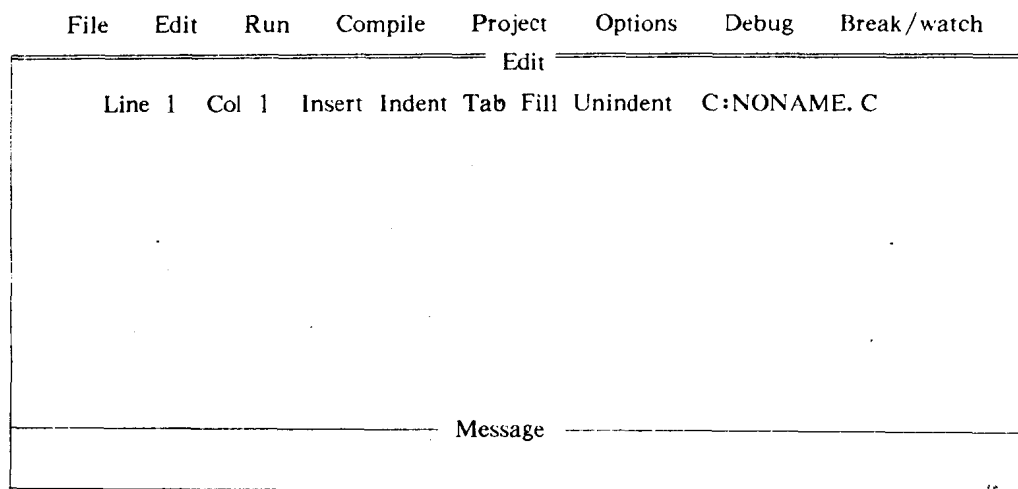
\路径\TC

即可进入Turbo C 集成开发环境。其中的“路径”为TC. EXE 所在的子目录。TC 启动后将在屏幕上显示图1.3 所示的主菜单屏幕。

### 2. 退出

退出Turbo C 有两种方法,一种是按热键ALT-X(同时按下ALT 和X 两键);另一种是选择主菜

单的File/Quit项(即选File进入子菜单后,再选Quit项)。



F1—Help F5—Zoom F6—Switch F7—Trace F8—Step F9—Make F10—Menu NUM

图1.3 主菜单屏幕

### 1.4.3 Turbo C 主菜单屏幕

Turbo C 主菜单屏幕分为四个区(见图1.3)。即主菜单行,编辑窗口,信息窗口和功能键行。下面分别予以介绍。

#### 1. 主菜单行

主菜单行位于屏幕顶部,共有八个选项。当主菜单处于活动状态时,有一光条落在某项上,用←或→键可移动光条到要选择的项上,按下回车键即选中该菜单项。在任何状态下均可按下热键F10来激活主菜单。此外也可以通过热键直接进入主菜单的某一项,即按下ALT键的同时按下某主菜单项的第一个字母。例如按下ALT+E就直接进入Edit状态,按下ALT+O就直接进入Option子菜单。

主菜单各项的基本意义如下:

**File** 文件操作。该子菜单中又包括了文件的装入,选择,建立,存盘,更名存盘以及列文件目录,改变当前目录,暂时退出TC,退出TC等多个子项。

**Edit** 建立和编辑源文件。

**Run** 编译,链接和运行程序。

**Compile** 编译,链接生成目标文件和可执行文件。

**Project** 允许在源程序中含有多个源文件。

**Options** 设置编译器的任选项及工作环境。

**Debug** 设置调试选择项。

**Break/Watch** 编辑监视表达式,设置或清除断点。用于调试程序。

这八个选项除Edit外,其它各项被选中时都进入相应的下拉子菜单,供操作员进一步选择使用。不论处于哪一级菜单均可用ESC键退回到上一级。

#### 2. 编辑窗口

主菜单行下面是编辑窗口。编辑窗口中的第一行是编辑状态行:例如