

C语言程序设计及应用

邢馥生 刘志远 姜德森 编著



高等教育出版社

C语言程序设计及应用



C 语言程序设计及应用

邢馥生 刘志远 姜德森 编著

高等教育出版社

(京)112号

内 容 提 要

本书是教育部高等工程专科学校计算机基础课程“九五”规划教材。内容包括C语言的语法规则,各种数据类型、语句,程序的函数结构等。书中由始至终贯彻模块化程序设计方法,从开始就介绍了如何使用Turbo C集成环境编辑、调试、运行程序和如何编制文档。本书专门设置了应用性的章节,如图形、动画、菜单技术、汉字处理等,还简要介绍了面向对象程序设计技术的基本概念。每章都配有应用性的例题、习题和上机题。全书系统性强、循序渐进、重点突出,采用边学边练、边学边用方式,便于初学者学习,也便于教师操作。

本书可作为高等学校计算机专业程序设计的入门教材,也可作为非计算机专业及有关培训班的基礎课教材,还可以提供给高中以上文化程度的C语言学习者作为自学教材。

图书在版编目(CIP)数据

C语言程序设计及应用/邢馥生等编著. —北京:高等教育出版社,1998.6

ISBN 7-04-006406-5

I. C… II. 邢… III. C语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(98)第12432号

*

高等教育出版社出版

北京沙滩后街55号

邮政编码:100009 传真:64014048 电话:64054588

新华书店总店北京发行所发行

北京印刷二厂印刷

*

开本 787×1092 1/16 印张 24 590 000

1988年6月第1版 1998年6月第1次印刷

印数 0 001—5 102

定价 20.30元

凡购买高等教育出版社的图书,如有缺页、倒页、脱页等

质量问题者,请与当地图书销售部门联系调换

版权所有,不得翻印

前言

C语言自1972年诞生以来,随着它自身的不断发展,逐渐地被人们认识并广泛应用于编制各种系统软件和应用软件,显示出了它具有的强大生命力。C语言的生命力来源于它本身的特点:兼有高级语言和低级语言二者的特点;灵活、丰富的数据类型,表达力强;既可用于编写应用程序,又可用于编写系统程序;易于作为结构程序设计的工具。所以,C语言是一种较好的面向过程的程序设计语言。近年来,在C语言的基础上已设计出面向对象的程序设计语言 Turbo C++ 和 Borland C++,C语言又成为了实现面向对象程序设计技术的主要工具。据此,教育部高等工程专科学校计算机课程指导委员会将C语言定为各专业的必修课,同时制定了该课程的教学要求,并组织编写了本教材。

本书共分四篇。入门篇中第一章从简单的程序范例讲述C程序的概况:基本输入、输出函数的调用和如何在微机上调试程序;第二章、第三章讲述基本数据类型和基本控制语句。基础篇中第四章介绍了C语言的构造类型之一——数组;第五章讲述C程序的函数结构,并介绍 Turbo C 库函数。提高篇中第六章讲述了较复杂的数据类型:指针、结构体、联合体、文件,同时介绍了枚举类型;第七章讲述C程序设置在编译前的预处理功能;第八章介绍位操作。应用篇中第九章介绍图形函数及应用,还介绍了动画设计;第十章讲述指针类型的重要应用,即队列、栈、链表的实现及应用程序设计;第十一章介绍了面向对象程序设计方法的基本思想、基本概念,并举出实例予以说明;第十二章是汉字处理;第十三章是常用的菜单技术。

编者总结了C语言教学中碰到的问题:C语言语法规则灵活、难学、难记;学了很多算法、很多规则用不上;在少学时情况下很难学完。故本书抓住C语言的核心部分、难点部分,突出核心部分,尽量分散难点,同时边学边用。从一开始就强调上机调试程序,设置了应用型章节,提供上机题,并强调写好程序文档。考虑到教学需要,编者注意深入浅出地介绍新技术和注意全书的可操作性。有些章节前加*,可选讲。

本书由邢毓生主编,朱乃立主审。其中前言,第一、二、三章由邢毓生编写;第四、五、六、七、十二、十三章由刘志远编写;第八、九、十、十一章由姜德森编写。全书由邢毓生负责编纂、修改并定稿。

本书在编写过程中得到康乃真教授和教育部高等工程专科学校计算机课程指导委员会老师们的认真指导,高等教育出版社的大力支持,翁咏梅同志热情的具体帮助,在此表示诚恳的谢意。

限于编者水平,书中难免有不恰当的提法和不足之处,敬请读者批评指正。

编 者

1997年12月

目 录

入门篇

第一章 C 语言简介 2

1.1 C 语言的发展史和特点 2

1.1.1 C 语言的发展史 2

1.1.2 C 语言的特点 2

1.2 C 语言程序结构 ——几个 C 语言程序范例 3

1.3 C 语言程序的设计 4

1.4 C 语言程序的编程环境 6

1.5 写好程序文档 11

习题一 11

上机题一 12

第二章 数据类型(一) 13

2.1 标识符 13

2.2 常量和变量 14

2.2.1 常量 14

2.2.2 变量 17

2.3 用 typedef 定义类型 18

2.4 赋值语句 19

2.5 运算符和表达式 20

2.5.1 运算符 20

2.5.2 表达式 23

2.6 格式输出函数和格式输入函数 25

2.6.1 数据的格式输出函数 printf 25

2.6.2 数据的格式输入函数 scanf 29

2.7 标准函数和自定义函数 31

2.7.1 标准函数 31

2.7.2 自定义函数 35

2.7.3 函数调用 35

习题二 35

上机题二 36

第三章 控制语句 38

3.1 表达式语句等 39

3.1.1 表达式语句范例 39

3.1.2 空语句 40

3.1.3 复合语句(语句块) 40

3.2 if 语句 41

3.2.1 if 语句的一般格式 41

3.2.2 嵌套的 if 语句 42

3.2.3 用 if 语句实现的多分支选择 42

3.2.4 操作符?:的使用及与 if 语句 的比较 45

3.3 switch 开关语句 47

3.4 while 循环语句 49

3.5 do-while 循环语句 51

3.6 for 循环语句 55

3.7 break 语句和 continue 语句 59

3.8 goto 语句 60

习题三 61

上机题三 62

基础篇

第四章 数据类型(二) 64

4.1 一维数组 64

4.1.1 一维数组变量说明 64

4.1.2 一维数组变量的存储 64

4.1.3 一维数组变量的使用 65

4.2 字符数组和字符串 68

4.2.1 字符数组和字符串 68

4.2.2 字符串的使用 68

4.2.3 字符串函数(字符数组函数) 71

4.3 二维数组和多维数组 73

4.3.1 二维数组变量的说明 73

4.3.2 二维数组变量的存储 74

4.3.3 二维数组变量的使用 76

4.4 字符串数组 78

4.5 数组初始化 80

习题四 82

上机题四	83	上机题六(一)	145
第五章 程序结构	85	6.2 结构体(struct)	146
5.1 函数	85	6.2.1 结构体类型与结构体类型变量	146
5.1.1 函数的定义形式	85	6.2.2 结构体变量的引用与初始化	149
5.1.2 函数调用及函数的返回值	86	6.2.3 结构体与数组	151
5.1.3 形式参数的类型	88	6.2.4 结构体与指针	154
5.1.4 实在参数和形式参数之间的 数据传送	88	6.2.5 结构体与函数	157
5.1.5 函数原型	90	6.3 联合体(union)	159
5.2 数组作为函数参数	91	* 6.4 枚举	162
5.3 局部变量与全局变量	95	习题六(二)	165
5.4 动态存储变量与静态存储变量	97	上机题六(二)	166
5.4.1 动态变量(auto 型)	97	6.5 使用 typedef 定义新类型名 的讨论	168
5.4.2 外部变量(extern 型)	97	6.5.1 定义“替代”类型名	168
5.4.3 静态变量(static 型)	98	6.5.2 定义“构造”类型名	169
5.4.4 寄存器变量(register 型)	100	6.5.3 定义新类型名的一般步骤	169
5.5 内部函数与外部函数	101	6.6 文件(FILE)	171
5.5.1 内部函数(static)	101	6.6.1 文件、流的概念及文件流 的组织形式	171
5.5.2 外部函数(extern)	101	6.6.2 文件类型(FILE)与 标准流式文件	173
5.6 Turbo C 库函数简介	102	6.6.3 文件的打开/关闭	175
5.6.1 库函数与函数库	102	6.6.4 文件 I/O 操作	177
5.6.2 头部文件(*.H)	103	6.6.5 文件的重定位、测试与随机读写 (rewind(), feof(), ftell(), fseek())	183
5.6.3 程序连接	103	6.6.6 检错与处理 (ferror(), clearerr(), exit())	185
5.6.4 Turbo C 库管理程序——TLIB	104	6.6.7 DOS 文件调用与删除 (system(), remove())	187
习题五	106	习题题六(三)	188
上机题五	107	上机题六(三)	189
提高篇		第七章 编译预处理	192
第六章 数据类型(三)	110	7.1 包含指令#include	192
6.1 指针	110	7.2 宏定义指令#define 和 #undef	194
6.1.1 指针与指针变量	110	7.2.1 不带参数的宏定义指令	194
6.1.2 指针运算	112	7.2.2 带参数的宏定义指令	196
6.1.3 指针与数组	116	7.2.3 宏标识符	198
6.1.4 指针与字符串	122	7.2.4 宏终结指令#undef	198
6.1.5 指针与函数	124	7.3 条件编译指令#if、#else、#elif、	
6.1.6 Turbo C 动态存储分配	132		
6.1.7 指向指针的指针	135		
6.1.8 指针初始化	141		
习题六(一)	144		

#endif 和 #ifdef、#ifndef ...	199	10.1.4 静态数据结构和动态数据结构 ...	246
7.3.1 #if、#else、#elif 和 #endif 指令...	199	10.1.5 C 语言的动态存储分配	247
7.3.2 #ifdef、#ifndef 指令	201	10.2 栈	248
* 7.4 错误信息指令 #error	202	10.2.1 栈的概念、结构及其操作	248
* 7.5 行控制指令 #line 及预定义宏		10.2.2 栈的程序实现	249
__LINE__、__FILE__ ...	203	10.2.3 应用举例	251
习题七	205	10.3 队列	254
上机题七	205	10.3.1 队列的概念、结构及操作	254
第八章 位操作	207	10.3.2 队列的程序实现	254
8.1 位逻辑运算	207	10.3.3 应用举例	256
8.1.1 按位求反运算	207	10.4 链表	260
8.1.2 按位与运算	208	10.4.1 引言	260
8.1.3 按位或运算	209	10.4.2 单向链表	261
8.1.4 按位异或运算	210	10.4.3 双向链表	276
8.2 移位操作	211	10.5 二叉树及应用	285
8.2.1 左移位操作	212	10.5.1 引言	285
8.2.2 右移位操作	212	10.5.2 遍历二叉树	286
8.3 位操作赋值运算	214	10.5.3 应用举例	288
习题八	215	习题十	299
上机题八	216	上机题十	299
第九章 图形函数的应用	217	第十一章 面向对象程序设计概念简介 ...	300
9.1 微机图形系统基础	217	11.1 引言	300
9.1.1 图形与显示器基础	217	11.2 面向对象的基本概念与特征 ...	300
9.1.2 图形显示模式	218	11.3 面向对象的程序举例	303
9.1.3 调色板和颜色	219	11.3.1 C++ 的类定义	303
9.2 图形函数及其用法	220	11.3.2 C++ 的对象说明	305
9.2.1 图形系统控制和屏幕操作函数 ...	221	11.3.3 程序举例	305
9.2.2 绘图和填充函数	224	习题十一	312
9.2.3 图形存取函数	234	上机题十一	312
9.2.4 图形方式下的文本输出函数	235		
9.3 动画程序举例	237	应用篇	
习题九	243	第十二章 汉字处理	314
上机题九	244	12.1 汉字编码与点阵汉字字模	314
第十章 常用数据结构的实现方法及		12.1.1 汉字编码规则	314
 应用程序设计	245	12.1.2 点阵汉字字模及其	
10.1 引言	245	存储格式、汉字库	314
10.1.1 数据和数据结构	245	12.2 汉字的显示与放大	315
10.1.2 逻辑结构和存储结构	245	12.2.1 汉字操作系统下汉字的显示 ...	315
10.1.3 线性表及其存储运算	246	12.2.2 西文操作系统下汉字的	

显示与放大	317	13.2 中文窗口式菜单	328
12.2.3 汉字放大	322	13.3 中文下拉式菜单	332
12.3 立体汉字和空心汉字	325	附录一 ASCII 字符编码一览表	339
12.3.1 立体汉字	325	附录二 Turbo C 2.0 库函数	340
12.2.2 空心汉字	325	附录三 Turbo C 2.0 编译出错信息 ...	364
第十三章 菜单技术	327		
13.1 窗口与菜单	327		

入 门 篇

第一章 C 语言简介

1.1 C 语言的发展史和特点

1.1.1 C 语言的发展史

C 语言是与 UNIX 操作系统互相依存的,二者相互促进得以发展。UNIX 操作系统是美国贝尔实验室的 K. Thompson 和 D. M. Ritchie 于 1971 年设计成功的,第 1 版是在 GE635 机器上产生并通过纸带把可执行代码传送到 PDP-7 上的,后又以汇编语言编写,由于汇编语言不可移植,描述问题效率不高,且可读性差,故 K. Thompson 决定开发一种高级语言来描述 UNIX 操作系统。这就是 C 语言产生的历史背景。

1972 年 C 语言研制成功并投入使用,1973 年 K. Thompson 和 D. M. Ritchie 把 UNIX 操作系统用 C 语言改写了一遍。UNIX 操作系统由于使用了 C 语言而取得成功,几乎成为 16 位微机的标准操作系统;C 语言一诞生就由于编制 UNIX 操作系统的成功而引起人们的关注。

1973 年之后,C 语言的发展也是很迅速的:1975 年 UNIX 第 6 版公布,1977 年又研制成功不依赖具体机器的 C 语言编译文本——“可移植 C 语言编译程序”,推动了 C 语言在各种机型上的广泛应用。UNIX 第 7 版在 1978 年研制成功,以其中的 C 编译系统为基础,美国国家标准协会于 1983 年制定了一个 C 语言标准草案,即 83 ANSI C,1987 年又公布了 87 ANSI C。

目前 C 语言编译系统有多种版本,在微机上常用的有 Microsoft C, Turbo C, Quick C 等。

美国 Borland 公司在 Turbo C 的基础上在 IBM PC 上实现了面向对象的程序设计语言 Turbo C++。Turbo C++ 与 Turbo C 高度兼容,在功能方面前者有更多的扩充。

1.1.2 C 语言的特点

C 语言是一种高级语言,同时它又提供了类似于汇编语言的低级语言的功能,如它可以直接访问物理地址并能进行位操作,为编写系统程序提供了手段。

C 语言程序生成的目标代码质量高,效率只略低于汇编语言,所以执行效率高。

C 语言程序可移植性好,即容易从一种类型计算机系统移植到另一种类型的计算机系统。

C 语言设有预处理功能,它设置了预处理命令,如 `#define`, `#include` 等,在编译源程序时先处理这类命令,以提高程序的可读性和可移植性,并且为调试程序提供了方便。

C 语言源程序是由一个或多个函数组成的,函数段之间是相对独立的。C 语言还提供了丰富的库函数,包括图形函数等,可供用户调用。用户自编函数可使用系统提供的九种控制语句,程序结构清晰、易编、易读。

C 语言提供的数据类型丰富,可以实现现代化语言的各种数据结构,如指针类型使用十分灵活,用它可以构成链表、树、栈等。

C 语言提供了四十种运算符,包括位运算符,可以完成其它高级语言难以完成的运算。

综上所述,C 语言是一种功能很强的语言。但是,它也有一些不足之处,C 语言语法限制不严谨,对于熟练程序员编程灵活,但安全性低;运算符丰富,完成功能强,但难记、难掌握。因此,

学习、使用 C 语言不妨先学基本部分,先用起来,用熟练后再学不规范的语法现象,进而全面掌握 C 语言。

1.2 C 语言程序结构——几个 C 语言程序范例

本节中将举出三个用 C 语言写的程序,请配合实例读懂以下几点说明:

(1) 程序的总体结构。一个 C 语言源程序是由一个主函数或者再加上若干个函数段组成的,程序的开始部分可以根据程序的需要写出以“#”为首字符的编译预处理行和全程序使用的变量说明,其后是若干函数段。每个段的最前面是函数名,主函数名是 main,其它函数段由程序员命名。函数名后是由圆括号‘()’括起的形式参数表,圆括号的后面是每一个形式参数的类型说明。再后就是由花括号‘{ }’括起的函数的主体了。

(2) 函数段内的内容。最简单的函数段是空函数段,执行它什么也不做。一般来说,根据程序功能的需要,函数段内可以有两部分内容:说明部分和可执行部分(语句部分)。说明部分中可以包括多个说明项,用以说明函数中使用的变量和需调用的其它函数等。语句部分可以包括多个可执行语句。

(3) 各语句之间用分号‘;’分隔开;并列的标识符或项之间用逗号‘,’分隔;两个关键词相邻时,中间用空格(至少一个)相间。

(4) C 语言在书写时是比较自由的,几个说明项或几个语句可以写在一行,一个语句可以分为几行写,但是一个词或一个数不能分两行写。写什么格式主要考虑程序易读和便于程序的维护。

(5) 为了使程序易读,在程序中设置注释部分,注释的内容写在“/*”和“*/”之间,程序运行时系统忽略其间的内容。

例 1.1 一个没有功能的程序。

程序如下:

```
main()
{
}
```

运行这个程序,什么也不做。

例 1.2 编写 C 语言程序,打印一行字符。

程序如下:

```
main()
{
    printf("I WISH YOU SUSSESS!")
}
```

运行这个程序将输出两个双引号中间的一行字符。printf 是函数库中的一个函数,此处是调用库函数完成输出功能。

例 1.3 输入 a、b、c 三个整型量,输出其中最大者。编 C 语言程序实现之。

```
#include "stdio. h"          /* 编译预处理:标准输入输出头文件 */
main()
```

```
{
    int smax();
    int a,b,c,max;
    scanf("%d,%d,%d",&a,&b,&c); /* 标准输入函数调用 */
    max=smax(a,b,c);
    printf("a,b,c=%d,%d,%d; max=%d",a,b,c,max); /* 标准输出函数调用 */
}

int smax(x,y,z) /* 用户自定义函数 */
int x,y,z; /* 形式参数说明 */
{
    int k;
    if(x>y) k=x;
    else k=y;
    if(z>k) k=z;
    return(k);
}
```

运行结果:

输入:8,4,9 ↵ (Enter 键)

输出:a,b,c=8,4,9; max=9

本程序中包括两个函数:main 和 smax。在函数 main 中,前两行是说明部分,后面的多行是可执行的语句部分。在函数 smax 中,在花括号前是形式参数 x、y 和 z 的类型说明,花括号中第一行是变量的类型说明,后面四行是可执行的语句部分。

从以上三个程序可以看出,要学习使用 C 语言编程序,首先要学习程序的结构:如何将函数组成一个完整程序,程序中使用的参数(形式参数和实在参数)、变量如何说明,还要学习使用各种可执行的语句,了解它们如何控制计算机完成运算。

1.3 C 语言程序的设计

上节举出的三个用 C 语言编制的程序虽然繁简不同,但都称 C 语言源程序、C 语言程序或 C 语言的用户程序。下面将介绍程序员编制 C 语言的用户程序的过程,随着读者对语法规则的学习,将逐步加深对此步骤的理解。

需要用 C 语言编制程序来处理某一问题时,无论问题的繁简,都应经过以下几个步骤:

1. 问题定义:首先要回答的关键问题是“要解决的问题是什么”、“要处理的对象(数据)是什么”,即使是初学者的练习题,也要把问题搞清楚。如果是用户提出的问题,程序员还需要和用户协调,甚至多次的协调以达到共识。要将问题写成清楚的文字材料,这是程序文档的第一部分。

2. 可行性研究:对前面所确定的问题“是否有行得通的解决办法”。这个阶段只需在高层次、较抽象级上研究,研究此问题是否可能解决和是否值得解决,并估计研制程序或软件系统的成本效益。

3. 需求分析:在上述 1、2 步骤的基础上,程序员与用户反复研讨程序必须具备哪些功能,从而得出问题的逻辑模型,通常用功能图表和文字说明形成“文档”。所谓“文档”就是程序的有关文字资料和程序。

4. 一般设计:这个阶段由程序员考虑应该“如何解决这个问题”。有的人说:“计算机是万能的,什么问题都可以解决,怎么还要我来想解决问题的办法呢?”这种说法是不准确的,因为即使计算机处理数据速度快,存储量大——记忆好,可以处理各种数据,解决很多问题,但这一切必须在程序的控制下工作,程序是由程序员编制的。

程序员考虑设计程序时必须具备如下条件:熟悉 C 语言语法,熟悉可能利用的库函数,熟悉待解决的问题。

下面讲述考虑具体解决方案时的指导思想:

(1) 如果要解决的问题比较简单,则用一个模块(一个函数)即可以解决。如果要解决的问题比较复杂,则考虑分为多个模块(例如在例 1.3 中分为 main 和 smax 两个函数),根据题目的复杂程度,还可以继续分下去,这就是目前推荐的逐步求精的程序设计方法。这样设计的程序符合结构化程序设计的思想,得到的程序结构性好。

(2) 现代程序要求要有友好的人机界面,要充分利用库函数中的输出函数和图形函数等,使程序在运行过程中能给用户醒目的提示和必要的干预,并且输出中间结果和最终结果。

(3) 在设计一个系统时,设计方案不止一种,功能要求越完美,程序将越复杂,软件成本越高。在设计实际程序系统时必须估计每种方案的成本和效益,充分考虑到成本/效益比,权衡利弊,选取一种最佳方案。

5. 详细设计和编码,根据第 4 步骤中确定的方案,进一步解决“具体怎么实现”。这一步包括设计出程序的详细规格说明和编写程序,并对编写的每个模块进行测试。

6. 综合测试:这一阶段主要是通过总体测试以解决该程序是否能完成预期的功能。这一阶段是把已通过了单元测试的模块按既定的策略装配起来,在装配过程中对程序进行必要的测试,即集成测试。最后,按系统功能说明书由用户(或在用户参与下)进行验收测试。在测试中要注意选取那些覆盖面大的数据组。测试资料——数据和结果都要做为文档保存。测试之后,软件可以交付使用。

7. 软件维护:程序的编制难于尽善尽美,测试的数据也难于覆盖全部可能情况,而且客观情况也在不断发生变化,因此软件的维护是十分必要的。在软件的维护阶段主要进行两种维护工作:第一种是改正性维护,就是改正在使用过程中发现的错误。第二种是当软件应用的环境发生了变化或用户提出改进功能的新要求时,需要完成的适应性维护。此时需要先分析维护要求,提出维护方案,确定维护计划,然后进行修改程序、测试程序、复查、验收等一系列工作。在此项工作中也应书写文档。

虽然,对于初学者来说,完成一个小题目似乎用不着这么繁琐,但是为了更快地编制实用程序,这种工作方式的建立却是必需的。以上程序(一般指大型程序系统)生成的过程称软件的“生命周期”,这种工作方式称“软件工程”,即软件生成过程工程化。

1.4 C 语言程序的编程环境

从编写完成程序到在计算机上运行、得到结果需要经历几个阶段。首先,将源程序输入机器,形成源程序文件;然后,由 C 编译系统对源程序进行编译,将源程序翻译成机器代码语言的目标程序模块;再后,连接系统将目标程序模块(如果使了用库函数等)连接成为可执行程序文件,运行可执行程序则可输出结果。总结起来,可以说是编辑、编译、连接、运行四步。

以上只是大致地画出了程序的调试过程,在程序调试的过程中还可能出现意想不到的错误,需要程序员排除。从图 1.1 可以看到:在调试程序时可能中断,其原因是程序中有语法错误;连接时也可能中断,这常常是由于存储器容量不够等原因引起的;运行时输入数据错或程序中有逻辑错(程序算法上的错)、甚至由于程序员书写程序时由于笔误引起的错误等都将导致输出结果不正确。

图 1.1 是按调试程序过程的顺序画出的,这种图称“流程图”。在设计程序时常常先根据思路画出流程图再编程序。流程图使用的符号含义如图 1.2 所示。

下面将介绍在本书使用的 Turbo C 集成环境下运行 C 语言程序的操作步骤:

1. 建立 Turbo C 编程环境(本步骤可由机房工作人员完成)

1) Turbo C 2.0 在目前的微机硬件和 DOS 条件下运行书中图形处理内容时,要求显示器不低于 VGA 方式。

2) 在硬盘上安装 Turbo C 2.0 系统,源系统中包含安装文件 INSTALL,在 DOS 下运行 INSTALL 程序按屏幕提示信息操作即可完成安装。

在安装结束后,在 C 盘根目录下,建立了如下子目录:

```

C:\TC
├── BIN
├── INCLUDE—SYS
├── LIB
└── OUTPUT
  
```

其中 BIN 子目录中为 TC. EXE、TCC. EXE、TLIB. EXE 和其它的系统文件;

INCLUDE 子目录中的文件为头文件;

LIB 子目录中的文件为系统的函数库文件;

OUTPUT 子目录用于存放程序编译、调试、运行时产生的. OBJ、. MAP、. EXE 形式的文件。

在软盘上安装 Turbo C 2.0 系统:准备三张格

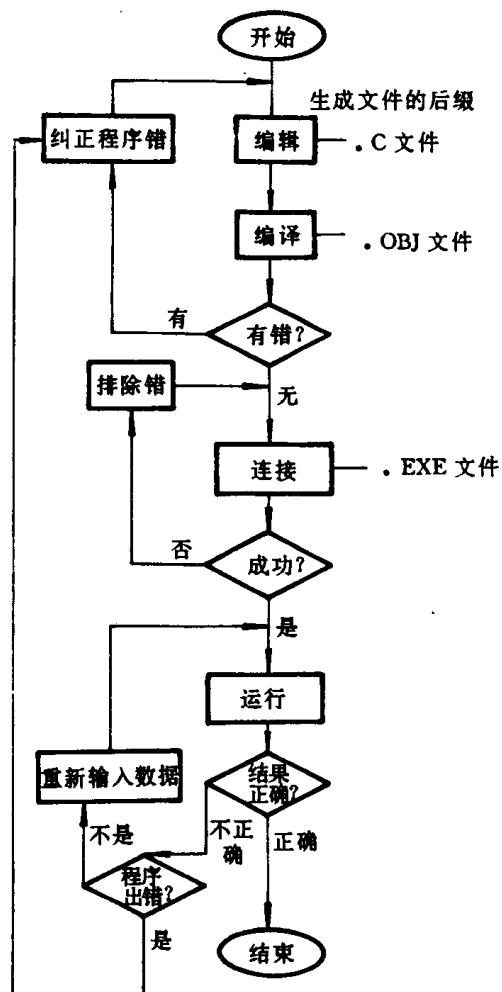


图 1.1 程序调试过程流程图

式化了的空盘,之后,进行如下操作:

(1) 将标签为 INSTALLATION
DISK 的源盘插入 A 驱动器。

(2) 键入 A:

(3) 键入 INSTALL

遵循屏幕提示进行操作直至安装完
毕。

2. 使用 Turbo C 编程环境(程序员
必须熟练下面程序调试过程)

首先进入 Turbo C 集成环境:

C:\>CD C:\TC\BIN ✓

(进入 bin 子目录)

C:\>TC\BIN\>TC ✓



开始、结束



执行框



判断框(根据条件选择可能情况之一执行下去)



沿箭头方向执行

图 1.2 流程图使用符号的含义

进入 Turbo C 集成环境后,屏幕显示如下:

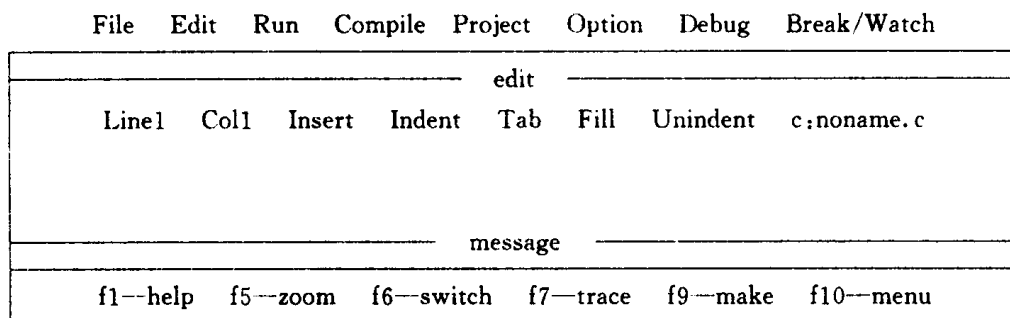


图 1.3 Turbo C 2.0 界面示意图

屏幕上面的内容可以分为四个区域:第一行是主菜单;下面以 edit 作为首行的方框窗口是编辑窗口;在编辑窗口内,message 开始的下半部分是信息窗口;编辑窗口后,屏幕的最后一行是使用功能键的提示。下面就四个窗口的功能和操作给以说明。

(1) 主菜单窗口

当光标在主菜单上(当击 F10 键时可将光标移至主菜单),称主菜单处于激活状态。可用左、右箭头键使光标左右移动,选取主菜单项(用 Enter 键确认),也可以使用快捷键来激活主菜单,即按住 Alt 键,再按 F、E、R、C、P、O、D 或 B(各项首字母),就可选取主菜单中的八个项,每项都有一个下拉菜单(Edit 除外),提供子菜单功能。

(2) 编辑窗口

编辑窗口顶部的一行信息给出此时正在窗口内编辑的程序的状态。这些参数的含义为:

Line n 表示光标所在行号。

Col n 表示光标所在列号。

Insert 插入模式开关,用 Ctrl+V 或 Insert 键进行切换。

Indent 自动缩进开关,用 Ctrl+OI 切换。

Tab 制表开启开关,用 Ctrl+OT 切换。

- Fill** 当 Tab 为 on 时,编辑程序将制表及空格符优化每一行的开始,用 Ctrl+OF 切换。
- Unident** 当光标在一行中的第一个非空字符上或在空行上时,退格键退一格,用 Ctrl+OF 切换。

c:noname.c 表示正在编辑的程序未命名而使用系统的代用名。

注:文件修改后尚未存盘时文件名前出现 *。

(3) 信息窗口

在对文件进行编辑、连接、跟踪等调试时,若出现问题则在此窗口输出警告信息或出错信息,同时在编辑窗口内以闪动的光标标出出错位置。

(4) 功能提示窗口

这是一组主要热键,任何时候按下热键,计算机立即执行相应功能。见表 1.1。

表 1.1 Turbo C 热键

F1 激活求助窗口	Alt+F 选择激活 File 菜单
F2 保存当前正在编辑文件	Alt+E 选择激活 Edit 菜单
F3 加载文件(出现输入框)	Alt+R 选择激活 Run 菜单
F4 程序运行到光标所在行	Alt+C 选择激活 Compile 菜单
F5 放大/缩小活动窗口	Alt+P 选择激活 Project 菜单 0
F6 开关活动窗口	Alt+O 选择激活 Option 菜单
F7 跟踪程序运行,深入函数内部	Alt+D 选择激活 Debug 菜单
F8 跟踪程序运行,跳过函数调用	Alt+B 选择激活 Break\Watch 菜单
F9 执行 Make 命令	Ctrl+F1 对编辑器调用函数求助
F10 激活主菜单窗口	Ctrl+F2 重启运行程序
Alt+F1 显示上次访问的帮助	Ctrl+F3 显示调用栈
Alt+F3 选择文件加载	Ctrl+F4 计算表达式值
Alt+F5 用户屏幕与 TC 屏幕切换	Ctrl+F7 设置监视表达式
Alt+F6 转换活动窗口的内容	Ctrl+F8 断点设置开关
Alt+F7、F8 确定上下错误位置	Ctrl+F9 运行程序

下面将进一步介绍七个主菜单项目的下拉子菜单的功能(Edit 没有下拉菜单)。

(1) File 处理文件

包括文件操作(装入、存盘、选择、建立、换名、写盘),目录操作(列表、改变工作目录),退出程序及调用 DOS。下面逐条解释下拉菜单的功能。

Load(加载) 装入一个文件,可用类似 DOS 的匹配符(如 *.C)来进行列表选择,也可用装入某一个文件,只要给出文件名即可。

Pick(选择) 将最近装入编辑窗口的八个文件列成一个表,让用户选择其一,再次装入编辑器,光标置在上次修改过的地方。

New(新文件) 为用户提供一个空白编辑窗口,编辑新文件,其缺省文件名为 NON-