

实用Turbo C 程序设计

吴 陈 主编

东南大学出版社

实用 Turbo C 程序设计

吴 陈 主编

东南大学出版社

(苏)新登字第 012 号

内 容 提 要

本书分 17 章详细叙述 C 语言(Turbo C)的基本概念、语法结构、文件操作、串处理、文本窗口、图形功能、系统访问、库函数、与其他语言的接口及程序设计技术,内容系统、全面、实用、新颖。书中还配有大量的例题和练习题。

本书是一本实用 Turbo C 程序设计语言的读物,可作为计算机应用、自动控制、机械工程、建筑设计、管理工程、计算力学等专业大学生、研究生的教材,也可供工程、科技人员参考和广大计算机爱好者自学使用。

责任编辑 张 克

用 Turbo C 程序设计

吴 陈 主编

*

东南大学出版社出版发行

校 2 号 邮编 210018)

江 苏 省 新 华 书 店 经 销 南京京新印刷厂印刷

*

开本 787×1092 毫米 1/16 印张 23 字数 559 千

1995 年 3 月第 1 版 1995 年 3 月第 1 次印刷

印数:1—5000 册

ISBN 7-81023-201-0/TP·64

定价:19.50 元

(凡因印装质量问题,可直接向承印厂调换)

前 言

C语言是一种高级程序设计语言,亦可以实现用汇编语言所进行的几乎一切与计算机硬件有关的系统程序设计。由于它具有丰富的数据类型、简洁的表达式、极强的输入和输出(I/O)功能、灵活的数据构造能力、强有力的控制流程以及数值计算和符号处理能力、运行效率高、可移植性强等许多优点,目前愈来愈受到人们广泛的重视,受到世界上软件专家们的高度赞赏。C语言不仅具有较强的文本操作、串处理、窗口和图形功能,而且(特别是Turbo C)操作方便,可以交互工作,为用户提供在线帮助,程序调试极为简单,目前我国各理工院校几乎所有专业的教学以及工程科技人员的编程中得到广泛应用,并被专家们认定为是软件三种基本技能之一(一是C语言,二是数据库程序设计语言,三是图形程序设计语言)。实践证明,C语言具有强大的生命力和广阔的应用前景。

鉴于此,我们在原来多次使用的讲稿基础上,根据国家教委的本科教学大纲及国家程序员水平考试大纲和计算机应用能力等级考试大纲的要求编写了本书。

全书共分为17章。

第1章介绍C语言的脚本结构及C程序的编辑、编译和运行的过程和方法,读者通过本章的学习,可以很快地了解怎样使用Turbo C系统和书写简单的C语言程序,并可尽早上机练习实习。

第2章介绍Turbo C的各类基本数据类型和表达式以及基本的输入/输出函数,涉及整型、字符型、浮点型和双精度型变量数据类型;算术逻辑运算表达式、条件表达式;各种数据类型变量的输入/输出格式。

第3章介绍变量的存储类型,分析变量的存储性质,包括自动变量、全局变量、寄存器变量、静态存储变量及动态存储变量。

第4章介绍语句和流程控制,包括判定结构和循环语句。

第5章介绍Turbo C程序设计语言的预处理功能。这一功能可以方便用户编制复杂的大型C语言程序。

第6章介绍C语言的函数,包括库函数、用户自定义的函数、递归函数。

第7章介绍数组,包括字符数组、多维数组及数组的初始化。

第8章介绍结构、位域和联合,用结构来描述数据对象是C语言的一种极好的方法。

第9章介绍指针,它是一个相对来说较为复杂的概念,这里对它的各种情况进行了深入浅出的描述。

第10章介绍链表,它是指针和结构的一种应用,便于节省存储空间。

第11章介绍Turbo C的文件操作,对大量数据的长期保存具有特殊的作用。给出了一些实用的程序。

第12章介绍Turbo C的文本及窗口处理。通过几个例子,演示了其用法。

第13章介绍Turbo C的绘图,这是目前计算机在友好用户界面和直观图形显示设计中比较实用的内容。通过例程,对绘图及与鼠标的联结、图形菜单的设计作了较为细致的介绍。

第14章介绍 Turbo C 与其它高级语言(Fortran, dBASE III, Turbo Prolog)之间的接口。对 Turbo C 如何与其它程序混合编程及其应用作了较系统的描述。

第15章介绍 Turbo C 的端口访问、中断处理及磁盘管理函数、Turbo C 与 Transputer 网络开发环境之间的通讯。使读者对使用 Turbo C 进行系统程序设计有十分重要的帮助。

第16章复习 C 的语法结构,概括了 Turbo C 的全貌。

第17章给出了部分参考练习程序。供读者模仿编程使用。

附录 I 给出了有关 Turbo C 应用问答和错误信息。

附录 II、III 分别给出了常用的 Turbo C 系统库函数及 ASCII 码字符对照表。

另外,全书还配有大量的例程,可帮助读者理解和学习。这些例程全部在 Turbo C 下运行通过,并储存在一张软盘中,读者可直接与编者联系。

本书所包含的内容极其丰富,在内容的组织上立意求新,并遵循一般读者的常端思维编排章序,简明而又精细。本书自成系统,读者学习时几乎可以不需要参考其它的资料,即可阅读使用。在教学中,第3章可放在稍后讲,以便学生能尽早上机编程。一些重要的概念,如指针,希望在教学中自始至终反复讲解。如果时间不充足的话,后面的第13、14、15章可作为初学者进一步参考自学的内容。基本部分的教学时数可在70~100学时之间,上机基本时数不应少于24学时。在条件许可的情况下,应尽可能多上机练习,这样才能充分领会和把握 Turbo C 的内容,达到运用自如的目的。在各自的教学中,读者也可根据具体情况选用。

此外,为方便读者阅读清楚起见,在本书中所出现和使用的符合如+、-、*、<、=、/、...等,大部分都采用中文下的符合形式编排,但读者在编程使用时,均应直接使用西文下的符合形式(除在注释中的以外)。编者在此特殊说明。

本书由吴陈主编,夏祖勋、刘同明主审。参加本书部分章节或内容编写的还有:顾建业(提纲目录、部分原稿审阅及第7章)、王顺林(第16章、附录 I 和 II)、生佳根(第4章)、归宝琪(第8章)、翁妙凤(第13章)、陈保香(第15章)、余松庆(附录 I)、吴世义和黄嘉颖(第17章)、邓靖和夏琼华(附录 II)等。

本书在编写过程中,得到我院、系领导的大力支持和鼓励,也得到我院、系教师的大力帮助,特将主审人对本书提出了许多宝贵的建设性意见,使本书增色不少,在此致以衷心的感谢。

编者

1994年12月

目 录

1 C语言概述

1.1 C语言的发展历史简介	(1)
1.2 C语言的特点	(1)
1.3 C程序入门与一般形式	(2)
1.4 C程序的上机编辑、编译、连接和执行过程	(7)
习 题	(9)

2 数据类型、运算符与表达式

2.1 数据类型	(10)
2.2 常量	(11)
2.3 变量	(14)
2.4 标识符	(18)
2.5 运算符与表达式	(19)
2.6 混合运算及其类型转换	(24)
2.7 枚举类型及其变量说明	(25)
2.8 输入/输出方式	(27)
2.9 编程风格	(34)
习 题	(35)

3 变量的存储类型

3.1 变量的存储类型概述	(37)
3.2 局部变量	(38)
3.3 全局变量	(39)
3.4 寄存器变量	(44)
3.5 静态存储变量	(45)
3.6 动态存储变量	(49)
3.7 变量存储类别小结	(48)
习 题	(51)

4 语句和流程控制

4.1 最简单的语句	(52)
4.2 复合语句与变量的作用域	(52)
4.3 if 语句	(54)
4.4 循环语句	(56)
4.5 break 语句	(59)
4.6 switch 语句	(60)
4.7 continue 语句	(62)
4.8 return 语句	(62)
4.9 循环的退出及死循环的避免	(63)
4.10 程序举例	(64)

习 题	(65)
5 C语言预处理功能	
5.1 宏替换	(67)
5.2 文件包含	(69)
5.3 条件编译	(70)
5.4 行控制	(72)
习 题	(72)
6 函数	
6.1 函数的分类	(73)
6.2 函数的定义和调用	(74)
6.3 函数的类型	(75)
6.4 关于函数参数的讨论	(77)
6.5 递归	(78)
6.6 字符串的存取与字符串函数的使用	(80)
习 题	(82)
7 数组	
7.1 一维数组	(84)
7.2 字符型数组	(85)
7.3 二维数组及多维数组	(85)
7.4 数组的初始化	(87)
7.5 函数调用中的数组	(89)
习 题	(92)
8 结构、位域和联合	
8.1 结构的概念及定义	(93)
8.2 结构与函数	(98)
8.3 位域	(100)
8.4 联合	(102)
8.5 类型定义 typedef	(103)
习 题	(104)
9 指针	
9.1 指针的概念	(106)
9.2 指针变量	(106)
9.3 指针的运算	(106)
9.4 指针的函数参数	(109)
9.5 指针和数组	(110)
9.6 字符指针和字符数组之间的区别与联系	(111)
9.7 指针与多维数组	(113)
9.8 结构指针	(115)
9.9 返回指针的函数	(117)
9.10 带多个函数值的返回	(118)
9.11 命令行参数	(119)
9.12 指向函数的指针	(120)

习 题	(122)
10 链表	
10.1 链表的创建方法	(125)
10.2 链表的遍历	(125)
10.3 链队列	(127)
10.4 链表中元素的插入与删除	(128)
10.5 双向链表	(130)
习 题	(132)
11 Turbo C 的磁盘文件操作	
11.1 流和文件	(134)
11.2 缓冲 I/O 系统的函数概述	(136)
11.3 文件操作函数	(138)
习 题	(155)
12 文本及窗口处理	
12.1 字符显示模式及控制	(157)
12.2 字符显示属性及控制	(158)
12.3 开窗口及定位	(159)
12.4 与窗口字符输入和输出相关的一组函数	(160)
12.5 窗口函数应用举例	(160)
习 题	(161)
13 Turbo C 绘图	
13.1 图形显示模式	(162)
13.2 图形系统控制	(164)
13.3 色彩控制	(165)
13.4 绘图和着色	(167)
13.5 图形模式下的文本输出	(168)
13.6 图形模式下的定位	(169)
13.7 图形函数应用举例	(169)
13.8 用鼠标交互式绘图	(176)
13.9 绘图机绘图	(185)
习 题	(192)
14 Turbo C 与其它程序语言间的接口	
14.1 存储模式	(194)
14.2 Turbo C 的内部汇编	(205)
14.3 Turbo C 与汇编语言的接口	(208)
14.4 Turbo C 与 Turbo Prolog 语言间的接口	(218)
14.5 Turbo C 与 Fortran 语言的接口	(224)
14.6 Turbo C 与 dBASE Ⅲ 的接口	(229)
14.7 协处理器的使用	(232)
15 直接端口、中断处理及磁盘管理函数	
15.1 直接端口处理函数	(234)
15.2 中断处理函数	(236)

15.3 BIOS 服务函数	(240)
15.4 通讯程序设计	(240)
15.5 磁盘管理系统函数	(247)
15.6 进程管理函数	(257)
15.7 时间处理函数	(261)
15.8 Turbo C 与 Transputer 网络开发环境的通讯	(262)
16 C 语言语法结构	(266)
17 部分练习程序	
17.1 输入/输出语句	(285)
17.2 表达式语句	(287)
17.3 判定结构语句	(288)
17.4 循环结构	(292)
17.5 数组	(299)
17.6 字符数组	(305)
17.7 结构类型	(308)
17.8 指针类型	(310)
17.9 综合应用	(311)
附录 I Turbo C 应用问答与错误信息	(317)
附录 II 常用的 Turbo C 系统库函数	(322)
附录 III 常用字符与 ASCII 代码对照表	(355)
参考文献	(356)

1 C 语言概述

1.1 C 语言的发展历史简介

C 语言是目前国际上流行的一种结构化、模块化、可编译的计算机高级程序设计语言。它既适合于系统程序的开发,也适合于应用程序的开发。例如,著名的 Unix 操作系统,微机上流行的 C-DBASE III 等都是用 C 语言开发的。很多图形软件接口应用程序,以及目前被人们广泛研究的神经网络应用软件等大都采用 C 语言来编写。数据处理(如企业管理)以及数值计算等都可以很方便地使用 C 语言。

C 语言最初是在 1972 年由 Dennis Ritchie 为 DEC 公司 PDP-11 计算机配置 Unix 操作系统而设计的,这个操作系统以及所有的基本 Unix 应用程序都用 C 语言编写而成。虽然 C 语言是在 Unix 系统上发展起来的,但它并不局限于某一种操作系统或机器。用 C 语言编写的程序可以很方便地移植到任何支持 C 语言的机器上。

C 语言既具有高级语言的所有特点,也具有实现汇编语言的某些功能(如对内存地址的操作、位操作、端口访问等直接对硬件进行操作的功能),因而既克服了用汇编语言编写的程序可读性是和可移植性差的缺点,同时又增强了高级语言的处理能力。

随着微型计算机的流行和发展,C 语言也成为微机上颇有影响的一种程序设计语言。目前在微型机上流行使用的有 Microsoft C、Turbo C 和 Quick C 等。各种版本的 C 语言编译系统的基本部分是相同的,但也有一些细微的差异。因此,读者在选用了自己的 C 语言编译系统后,应了解自己所选用的计算机系统的特点和规定(可以参考有关的手册)。本书主要以 Turbo C 为蓝本来介绍 C 语言的使用。

1.2 C 语言的特点

C 语言能在短短的 20 年内得到广泛的发展和流行应用,显示出其强大的生命力,主要是由于它具有以下独到的特点。

(1) 以英文小写字母为基础。这符合人们日常阅读和书写的习惯。同时在不使用小写字母的微型机上允许使用大写字母,使用的范围不受影响;

(2) 程序语言表达简洁、紧凑,书写格式自由,使用方便灵活,因而程序短小精悍让人易写易懂。

(3) 程序由函数的集合组成,因此程序功能结构比较清楚。而且每个函数都是独立的,可以单独编译,对设计一个大的程序来说,有利于分工编译和调试;

(4) 运算符丰富,有利于编写程序。它把括号、赋值、强制类型转换等都作为运算符处理,并增加了如增 1(++)和减 1(--),以及条件运算等其它高级语言没有的运算符。因而

使运算符多达 34 种以上。同时也使 C 的运算类型极其丰富,表达式类型多样化。能够通过这些运算符的灵活使用,即可以较短的代码,完成其他高级语言难以实现的运算。

(5) 数据类型多。C 语言的数据类型比 Fortran, Algol 等多,与 Pascal 语言相当。具有整型、实型、字符型、数组类型、指针类型、结构类型和联合类型等现代化语言所应有的各种数据类型。能够实现各种复杂的数据结构(如链表、栈、队列和树等)的运算。

(6) 指针作为数据来处理。可以用指针指向各类数据结构。

(7) 具有结构化程序设计所应有的控制语句(如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句),是理想的结构化程序设计语言,符合现代编程的风格和要求。

(8) 语法的限制不死板,程序设计比较自由。例如,整型、字符型以及逻辑型数据可以通用,变量的类型限制不严格,数组的下标越界不作检查等。因而要求程序员自己仔细检查程序,保证其正确,不要过分依赖 C 编译程序。允许程序设计者充分使用 C 语言的自由度。

(9) 输入输出功能通过函数程序来实现。

(10) 允许直接访问物理地址,进行位(Bit)操作,可实现汇编语言的大部分功能,可以直接对硬件进行操作。C 语言体现了高级语言的很多优越性。

(11) 可移植性好(与汇编语言相比)。用 C 语言编写的程序基本上不作修改就能用于各种型号的计算机和各种操作系统中。

(12) 通过预处理可以进行宏调用。使大型程序的设计可以分工编写,因而是一种模块化的程序设计语言。

(13) 没有作为语言组成部分的文件类型,但能通过操作系统完成对文件的操作,等等。

C 语言的上述特点,读者目前可能还不能深刻理解,待学完了 C 语言的全部主要内容后,再回过头来重温一下,就会有比较深刻的体会。当然,这里只是介绍 C 语言的一些主要特点。C 语言内部的其他特点,将会在以后各章中进一步介绍。

正是由于上述特点,目前 C 语言愈来愈得到人们的广泛重视,受到世界上软件专家们的高度赞赏。目前我国各理工院校几乎所有的专业教学以及工程技术人员的编程中, C 语言愈来愈得到广泛的使用,并被专家们认为是软件的三种基本技能(一是高级语言, C 一门数据库程序设计语言、一门图形程序设计语言)之一。

当然, C 语言也有些不足之处,例如运算符优先级太多,不使记忆,有些还与其他语言有些不同;类型检验较弱;转换比较随便,所以不太安全。尽管如此, C 语言还不失为一个实用的通用程序设计语言。使用者更可以发挥 C 语言格式自由、限制较少、语言简洁等特点,编制自己的程序。

1.3 C 程序入门与一般形式

用 C 语言编写的程序称为 C 语言源程序,或简称 C 程序。

C 语言的源程序是什么形式呢?下面通过几个简单的例程来分析一下 C 源程序的特性。

例 1.1 打印一串字符的 C 程序

```
main()  
{  
    printf("Hello, world!");
```

C 程序一般是由一个或者多个函数组成,而这些函数可以保存在一个或多个源程序文件中,这些文件都以 .C 为扩展名。在一个 C 程序中必须有且只有一个名为 main 的函数,称为主函数。在运行 C 程序时总是从 main 函数开始执行。函数名后一对圆括号是函数的标志。其中 printf("Hello,world!")语句将在显示屏幕上打印输出括号中双引号内的内容:

Hello,world!

例 1.2 打印一串字符的 C 程序

```
main()
{
    printf(" * * * * * \n");
    printf("Welcome to C! \n");
    printf(" * * * * * \n");
}
```

本程序的作用是在终端屏幕上输出如下三行字符:

```
* * * * *
Welcome to C!
* * * * *
```

其中,main()表示“主函数”,每一个 C 程序必须有一个而且仅有一个 main()函数。大括号“{”和“}”括起来的部分是函数的体。本例中主函数体由三个输出语句组成,其中 printf 是 C 语言中的输出函数(也称为库函数),双引号内的字符串原样输出。而“\n”是换行符。当遇到“\n”时就换行。每条语句最后都有一个“;”号,不可缺少。

例 1.3 可将例 1.2 的程序改写成:

```
main() /* 打印星号的主函数 */
{
    void stars(void);
    stars();
    printf(" Welcome to C!");
    stars();
}

/* 注: void 是不返回任何值的函数说明 */
void stars() /* 打印星号的函数 */
{printf(" * * * * * \n");}
```

该程序的运行结果与例 1.2 相同。但它打印一行星号是由函数 stars()来实现的。其中 /* */表示注释部分。注释的部分仅供人阅读,在编译和运行时都不加理睬。注释可以放在程序中的任何位置。为了便于理解,我们用汉语进行注释,在不支持中文的纯英文版本下则只可以用英文或汉语拼音进行注释。

例 1.4 求两个整数之和。

```
main() /* 求两整数的和 */
{
```

```

int a,b,sum;
a=12;b=13;
sum=a+b;
printf("sum is %d \n",sum);
}

```

程序的第二行是一变量定义语句,说明 a,b 和 sum 分别为整型(int)变量。第三行包含两个赋值语句,分别将 a 和 b 赋以 12 和 13。第四行将 a+b 的结果赋给 sum。第五行是通过调用输出库函数 printf 在标准输出设备(屏幕)上输出结果。printf 中第一个参数是由一对双引号括起来的“格式控制字符串”,其中 %d 表示十进制整数格式说明。在执行输出时,%d 所处的位置将被代之以 printf 的第二个参数的值。因为 sum 的值为 25(即 12+13 的结果),所以输出一行信息为:

```
sum is 25
```

例 1.5 用子函数求两个数中的较小者

```

main() /* 主函数 */
{
    int a,b,c,min(); /* 定义变量 a,b,c 及函数 min() 的类型 */
    scanf("%d%d",&a,&b); /* 输入变量 a 和 b 的值 */
    c=min(a,b); /* 调用函数 min,结果返回并送给 c */
    printf("min=%d",c); /* 输出 c 的值 */
}

int min(x,y); /* 定义函数 min,函数带形式参数 x 和 y,且值为整型 */
int x,y; /* 对形参 x 和 y 作类型定义 */
{
    int z; /* 对函数 min 中使用的变量 z 加以定义 */
    if (x<y) /* if ..... else 为条件语句 */
        z=x;
    else
        z=y;
    return(z); /* 将 z 的值作为函数 min() 的值返回 min() 的被调用处 */
}

```

本程序由两个函数组成,函数 main() 为主函数,函数 min() 为非主函数。函数 min() 的功能是首先求出 x 和 y 中的较小者,放在 z 中保存,然后通过 return 语句将 z 的值作为函数 min() 的值带回给主调用函数 main(),返回值是通过函数名 min 带回到 main 函数的参数调用处的。main() 函数中的 scanf 是“扫描输入函数”的名字,属 C 语言的标准输入库函数。在这里 scanf 的作用是输入 a 和 b 的值。scanf 的第一个参数“%d%d”是输入控制用的格式字符串,第二和第三个参数为 &a 和 &b。其中“&”表示“取地址”,这是 scanf 函数所特定需要的,不能缺。该语句执行的结果是将两个数值以十进制整数(由两个 %d 所组成的格式“%d%d”规定)数值输入到变量 a 和 b 的地址所标志的单元中,即输入变量 a 和 b。

main() 函数的第四行为调用函数 min。在调用时,将实际参数 a 和 b 的值分别传送给

min 函数中的形式参数 x 和 y, 也称形(参)实(参)结合。min() 函数执行后得到一个返回值 (即 return(z) 中的 z 的值), 将返回值赋给变量 c, 然后输出 c 的值。printf 函数中双引号内的内容 "min=%d" 将在输出时把 %d 的位置用 c 的值代替之后输出。程序运行结果如下:

10 20 (输入 10 和 20 给 a 和 b, 两数之间以空格隔开)

min=10 (输出 c 的值)

本例中用到了函数调用、形参和实参结合的概念, 我们在这里仅给出了直观简明的解释。随着对 C 语言学习的不断深入, 读者会更进一步明确。在此先可不必深究。

通过上述五个例子, 我们可以看到, C 源程序的结构和书写形式为:

(1) 一个 C 程序由一个主函数和若干个其他 (可以是零个, 此时, 仅有一个主函数 main() 函数) 组成。主函数的名字一定是 main, 而其他函数的名字由编程者命名 (须符合标识符的有关规定, 详见标识符的定义)。函数是 C 程序的基本构造单位。被调用的函数可以是系统提供的库函数 (例如 printf 和 scanf 函数), 也可以是用户自己定义的函数。函数名后紧跟一对圆括号, 用来放参数。函数可以不带参数, 如上述各例中的 main(), 但括号必须保留, 不能省略, 因为它们是函数的标志。实际上 main 也可以带参量, 这里暂不介绍, 后面会介绍的。C 的非主函数相当于其他程序设计语言中的子程序或过程。用函数实现特定的功能, 使 C 程序代码变得紧凑短小, 功能结构分明。C 程序的全部工作都由函数来完成, 因此 C 语言通常被人们称为是一种函数式语言。C 语言的函数库十分丰富, 标准 C 提供一百多个库函数, 而 Turbo C 和 MS C 4.0 则提供了三百多个库函数。C 的这种特点使得容易实现程序的模块化设计。

(2) 一个 C 程序总是从主函数 main() 开始执行, 而不论 main 函数处在整个程序的什么位置。因此, main() 函数可以放在程序的最前头 (即所有其他函数之前), 也可以放在程序的最后 (即所有其他函数之后), 或者在一些函数之前, 另一些函数之后。

(3) C 程序中的一个函数总是由两部分组成: 函数的说明部分和函数体。函数的说明部分处于函数的头部, 包括函数类型、函数名、函数参数名 (形参) 及函数形参类型说明。例如, 例 1.5 中的 min() 函数的说明部分为:

int	min	(x,y)
↑	↑	↑
函数类型	函数名	函数参数
int	x,y	
↑	↑	
形参类型	形参	

当一个函数带多个参数时, 参数之间用逗号 “,” 隔开。

函数体紧接在函数说明部分下面。花括号 { 和 } 是函数体的界限。它们也可以用作语句括号形成语句块。面在一个函数的最外层的一对括号内的内容是函数体。函数体内通常有说明部分和语句部分, 说明部分对变量及被调用的函数进行定义。如例 1.3 中的语句 “void stars(void);” 例 1.4 中的 “int a,b,sum;”。例 1.5 中 main 函数内的 “int a,b,c, min();” 和 min 函数内的 “int z;”, 而例 1.1、例 1.2 中没有任何变量说明, 因而无语句说明部分, 仅由语句部分组成。语句部分由若干语句组成。有时可以没有语句说明部分, 甚至没有语句部分, 如:

empty()

{ }

它是一个空函数,不作任何事情,但它是合法的。

(4) C 程序中的每个语句必须以分号结尾。分号是 C 语句的必要组成部分。即使程序的最后一个语句也应包含分号,必不可少。

(5) C 程序书写格式自由,一行可以写一个或多个语句,一个语句也可以分写在几行上。但是一个标识符不可分成两行或多行书写。同时 C 程序也不象 BASIC 语言那样每行有行号,也不象 FORTRAN 或 COBOL 那样对书写格式有严格的规定(语句必须从某一列开始书写)。

(6) C 语言本身没有输入输出语句。输入输出操作是调用库函数 scanf 和 printf 等来完成的。C 对输入和输出操作实行“函数化”。scanf 和 printf 分别是“格式”输入和“格式”输出函数。

(7) 双引号中的字符序列叫做字符串或字符串常数。而反斜线“\”是转义符号。\\n 表示换行字符,应看成是一个字符。它使光标移至下一行的左端。如果例 1.2 中的 \\n 都去掉,那么输出就不分行,且运行结束时,光标也不移动到下一行的左端。

(8) /* 和 */ 分别是注释的开始和结束符号。一个好的具有可读性和使用价值的源程序都应加进 /* */ ,对程序的重要部分进行说明。

C 语言的源程序的一般形式可以直观地概括为:

全局变量的说明

main(参量表) /* 主函数 */

参量说明

{

局部变量说明

语句序列

}

f1(参量表) /* 函数 f1 */

参量说明

{

局部变量说明

语句序列

}

.....

fn(参量表) /* 函数 fn */

参量表

{

局部变量说明

语句序列

}

1.4 C 程序的上机编辑、编译、连接和执行过程

对于手工编写的 C 语言源程序,如何输入计算机,在计算机上运行执行,以达到深刻理解 C 的各种语句功能呢?下面,我们以微机上的 Turbo C 为例作一简单的介绍。

一般说来,任何一种编译型语言的源程序要在机器上运行都要经过下列几个步骤(解释型语言除外)。

(1) 编辑 用编辑程序(如微机上的 Edlin、Wordstar、CCED、WPS 等文本编辑软件)或系统自带的编辑功能(Turbo C 自带)将源程序键入计算机,经修改无误后,存入计算机文件系统。Turbo C 源程序文件一般以 .C 为文件扩展名,存放到磁盘上。

(2) 编译 调用 C 编译程序对源程序进行编译。如果在编译过程中编译程序发现源程序有语法错误,则系统输出“错误信息”(一般由错误号码和提示组成),告诉用户第几行有错误,并提请编程者应作如何修改或提出建议。修改后再作编译,直到程序编译通过为止。编译的结果是生成与源程序文件同名但扩展名不同(如 Turbo C 为 .obj)的目标程序。

(3) 连接 将编译通过的目标程序和库函数或其他目标程序连接起来,形成可执行的目标程序。在 Turbo C 中可执行的目标程序的扩展名(或后缀)为 .exe。

(4) 执行 让可执行的目标程序在计算机上运行。可以在 DOS 提示符下直接键入文件名即可运行。上述过程如图 1.1 所示。

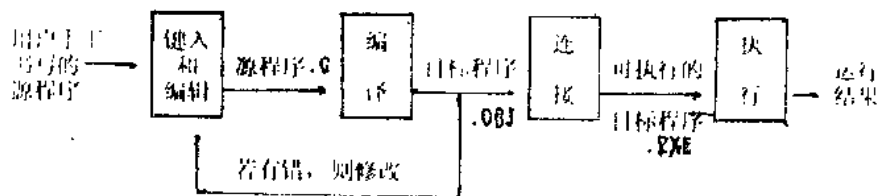


图 1.1 Turbo C 程序上机运行过程

用 Turbo C 来运行 C 程序的步骤如下。

(1) 先将 Turbo C 装入计算机的硬盘。Turbo C (2.0 版) 系统的全部文件分别存在六张低密度的软盘上(每张软磁盘的容量为 360KB),也可以是通过压缩软件压缩后存放在一张高密度软盘(1.2MB)上。为了将系统安装到硬盘上,可以运行第一张盘上的装配程序 INSTALL.BAT。该程序首先在硬盘上建立特定的子目录,然后将六张软盘上的文件按类别拷贝到各个子目录上。由于要建的子目录不止一个,配给的软盘也不止一张,故在此过程中,装配程序要求用户确定子目录名,更换软盘等。用户只管按提示逐一回答。该程序执行完毕, Turbo C 系统即已安装到硬盘上。

(2) 运行 Turbo C。转到 Turbo C 所在的子目录,在 DOS 提示符下键入 TC,并按回车键:

TC <CR>

Turbo C 即进入其集成开发环境(将编辑、编译、连接、运行集成在一起的状态)下。此时初启屏幕包括主菜单和版本信息。按任意键版本信息消失,留下如图 1.2 所示的主屏幕。主屏幕自上面下由四部分组成:主菜单、编译窗、信息窗和快速参考行。

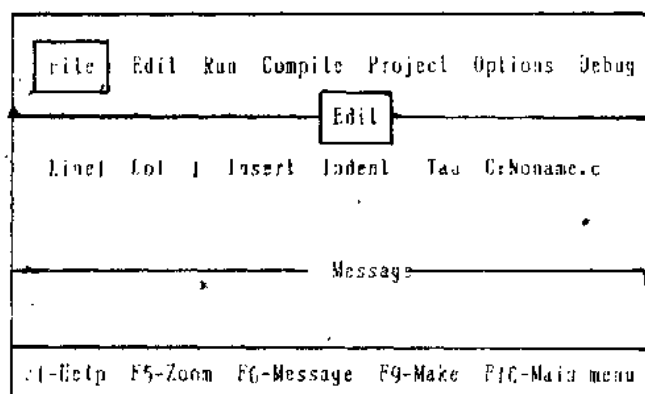


图 1.2 Turbo C 集成环境主菜单

高亮度光条指向“File”。用键盘上的“→”和“←”键可水平移动屏幕上的光标,光标指向哪一个命令,按回车就表示选择该命令执行。大多数的主菜单都有自己的子菜单,它们以下拉菜单的形式出现在主菜单的下边。有些子菜单还有自己更进一步的选择项,同时还会再显示另一个下拉菜单。从下一级菜单退回上一级菜单时请按 ESC 键。

在主菜单下选择“File”,会在其下面出现一个窗口,覆盖在原菜单上,如图 1.3 所示:

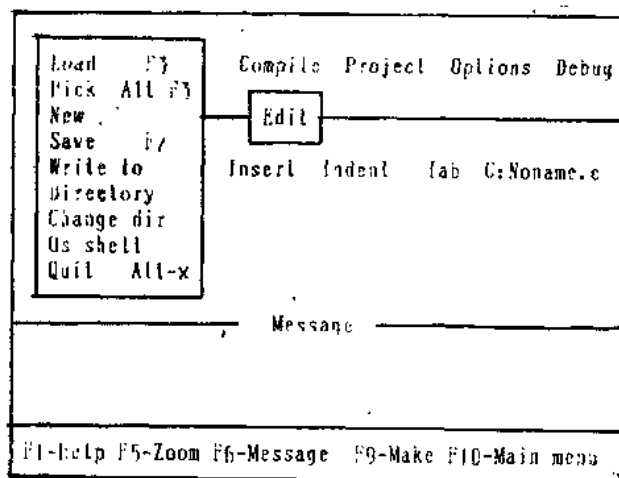


图 1.3 子菜单的选择

它是 File 的一个带多项选择的子菜单窗口。用“↑”和“↓”光标键可将光标在其上下移动。选择“Load”或“New”,屏幕上又将出现一个小的窗口(见图 1.4)。要求编程者输入要编写的源程序文件名。可键入 example 表示文件名

为 example.c。如果原来不存在此文件名,则 Turbo C 系统将建立一个新文件。如果已有此文件,则 Turbo C 系统将把该文件调入并显示在屏幕上。并自动进入编辑(Edit)状态。光标处在编辑窗口中。



图 1.4 文件名选择