

★ 高等院校 21 世纪新视野教材 ★

C 语言程序设计教程

(修订版)

张陵山 陈国平 杨子华 谭骏珊 主编



湖南大学出版社

12C
6:1

TP312C
Z136:1

★高等院校 21 世纪新视野教材★

C 语言程序设计教程

主 编	张陵山	陈国平	杨子华	谭骏珊	
副主编	周光宇	周铁山	杨泽凡	丁敬忠	胡晶晶
编 委	张陵山	杨子华	罗庆云	陈国平	
	胡晶晶	周光宇	周铁山	杨泽凡	
	丁敬忠	陈世清	戴日光	黎 昂	
	高为民	罗 方	刘 伟	熊国华	
	黄天强	胡 斌			
策 划	周铁军	徐建波			
主 审	郭 印				

湖南大学出版社

2004 年·长沙

图书在版编目(CIP)数据

C 语言程序设计教程/张陵山,陈国平,杨子华,谭骏珊主编.

长沙:湖南大学出版社,2003.1

ISBN 7-81053-593-5

I. C... II. ①张...②陈...③杨...④谭... III. C 语言
—程序设计—高等学校—材料 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 004613 号

C 语言程序设计教程

Cyuyan Chengxusheji Jiaocheng

张陵山 陈国平 杨子华 谭骏珊 主编

- | | |
|-------------------------------|------------------------------|
| ☐ 责任编辑 | 李 由 |
| ☐ 特约编辑 | 福 继 |
| ☐ 封面设计 | 张 毅 |
| ☐ 出版发行 | 湖南大学出版社 |
| | 社址 长沙市岳麓山 邮编 410082 |
| | 电话 0731-8821691 0731-8821593 |
| ☐ 经 销 | 湖南省新华书店 |
| ☐ 印 装 | 长沙环境保护学校印刷厂 |

- | | | | | | |
|-----------------------------|--------------------------|-----------------------------|-------------------|-----------------------------|-------|
| ☐ 开本 | 787×1092 16 开 | ☐ 印张 | 15.75 | ☐ 字数 | 364 千 |
| ☐ 版次 | 2003 年 1 月第 1 版 | ☐ | 2004 年 1 月第 2 次印刷 | | |
| ☐ 印数 | 5 001—11 000 册 | | | | |
| ☐ 书号 | ISBN 7-81053-593-5/TP·28 | | | | |
| ☐ 定价 | 30.00 元 | | | | |

(湖南大学版图书凡有印装差错,请向承印厂调换)

目 次

第一章 C语言简介

1.1 C语言与 Turbo C	(1)
1.2 Turbo C 的特点和配置要求	(2)
1.2.1 Turbo C 的发展及特点	
1.2.2 Turbo C 2.0 基本配置要求	
1.3 简单 C 程序介绍	(3)
1.4 C 语言可执行程序的产生过程	(4)
1.4.1 编辑程序	
1.4.2 编译程序	
1.4.3 库函数和连接	
1.4.4 运行程序	
1.4.5 简单的 C 语言程序的运行过程	
习题	(9)

第二章 基本数据类型、运算符及表达式

2.1 基本数据类型	(10)
2.2 标识符、常量、变量及其初始化	(11)
2.2.1 标识符和关键字	
2.2.2 常量与变量	
2.2.3 整型数据	
2.2.4 实型数据	
2.2.5 字符型数据	
2.2.6 变量赋初值与变量类型转换	
2.3 算术运算符及表达式	(18)
2.3.1 运算符的种类、优先级和结合性	
2.3.2 算术运算符和算术表达式	
2.4 赋值运算符及表达式	(21)
2.4.1 简单赋值运算符和赋值表达式	
2.4.2 复合赋值运算符及表达式	
2.5 类型转换运算符	(22)
2.6 逗号运算符和逗号表达式	(22)

习题	(23)
----------	------

第三章 结构化程序设计

3.1 结构化程序设计方法	(25)
3.1.1 算法及其实现	
3.1.2 结构化程序设计方法	
3.2 顺序结构	(28)
3.2.1 概述	
3.2.2 赋值语句	
3.2.3 数据输入与输出	
3.2.4 顺序结构程序设计举例	
3.3 选择结构	(39)
3.3.1 关系运算符和关系表达式	
3.3.2 逻辑运算符和逻辑表达式	
3.3.3 if 语句	
3.3.4 条件运算符和条件表达式	
3.3.5 switch 语句	
3.3.6 选择结构程序举例	
3.4 循环结构	(48)
3.4.1 goto 语句和 if 语句构成循环	
3.4.2 while 语句	
3.4.3 do-while 语句	
3.4.4 for 语句	
3.4.5 循环的嵌套	
3.4.6 break 和 continue 语句	
3.4.7 程序举例	
习题	(54)

第四章 函 数

4.1 函数的定义	(57)
4.1.1 函数类型说明	
4.1.2 函数的定义	
4.2 函数的调用形式	(60)
4.2.1 函数的返回值	
4.2.2 函数参数传递	
4.3 数据的存储类别	(62)
4.3.1 变量的说明	
4.3.2 局部变量与全局变量	

4.3.3 静态存储变量与动态存储变量	
4.4 函数的递归调用	(68)
4.4.1 递归说明	
4.4.2 递归实例	
4.5 函数举例	(72)
习题	(73)

第五章 预处理命令

5.1 宏定义	(75)
5.1.1 简单宏定义	
5.1.2 带形式参数的宏定义	
5.1.3 宏定义的取消	
5.2 文件包含	(77)
5.3 条件编译	(79)
5.3.1 常量表达式作为编译控制条件	
5.3.2 宏名作为编译控制条件	
习题	(81)

第六章 构造数据类型

6.1 枚举类型	(83)
6.2 数组	(86)
6.2.1 一维数组	
6.2.2 字符数组和字符串	
6.2.3 多维数组	
6.2.4 数组作为函数参数	
6.3 结构体	(103)
6.3.1 结构体类型定义与变量说明	
6.3.2 结构体变量的初始化与引用	
6.3.3 结构体数组	
6.3.4 结构体与函数	
6.4 位域与位运算	(110)
6.4.1 位域	
6.4.2 位运算符	
6.5 共用体	(114)
6.6 typedef	(116)
习题	(118)

第七章 指 针

7.1 指针	(121)
7.1.1 指针的概念	
7.1.2 指针变量的说明与初始化	
7.2 指针变量的运算性质	(124)
7.2.1 与指针变量有关的运算符 & 与 *	
7.2.2 指针变量的运算	
7.3 指针与一维数组	(130)
7.3.1 一维数组的指针变量的定义和说明	
7.3.2 指向一维数组元素的指针的定义和说明	
7.3.3 通过指针引用一维数组元素	
7.4 指针与多维数组	(133)
7.4.1 指向二维数组的指针	
7.4.2 指向二维数组的指针变量	
7.5 指针数组	(137)
7.5.1 指针数组的概念	
7.5.2 指针数组的说明	
7.5.3 用指针数组处理多维数组	
7.6 数组的动态分配	(139)
7.6.1 动态分配	
7.6.2 动态数组	
7.6.3 数组的动态分配	
7.7 指针和字符串	(143)
7.7.1 字符串的表示形式	
7.7.2 字符串的引用	
7.8 指向指针的指针	(147)
7.9 函数指针	(149)
7.10 返回指针的函数	(150)
7.11 指针参数与函数参数值的传递方法	(152)
7.11.1 指针变量作函数参数	
7.11.2 数组指针作函数参数	
7.11.3 字符串指针作函数参数	
7.11.4 指针数组作指针型函数的参数	
7.11.5 指针变量作 main 函数的参数	
7.12 指针与结构	(161)
7.12.1 指向结构体变量的指针	
7.12.2 指向结构体数组的指针	

7.12.3 指向结构体的指针作函数参数

习题.....	(165)
---------	-------

第八章 文 件

8.1 文件的概念	(169)
8.1.1 文件(file)的定义	
8.1.2 C语言对文件的处理	
8.1.3 文件的分类	
8.2 标准 I/O	(170)
8.3 缓冲文件的输入、输出	(171)
8.3.1 文件类型指针	
8.3.2 文件的打开与关闭	
8.3.3 文件的读写	
8.4 非缓冲文件系统	(177)
8.4.1 open()——打开非缓冲文件	
8.4.2 close()——关闭已打开的非缓冲文件	
8.4.3 creat()——新建一个文件	
8.4.4 read()——从指定的文件中读入信息	
8.4.5 write()——把信息写入到指定的文件	
8.4.6 lseek()——移动文件位置指针	
8.5 文件的随机存储	(179)
8.5.1 rewind()函数——位置指针复位	
8.5.2 fseek()函数和随机读写	
8.5.3 ftell()函数——获得文件当前位置	
习题	(181)

第九章 综合编程

9.1 需求分析	(182)
9.2 实用编程技术	(182)
9.3 面向对象编程简介	(183)
9.3.1 从C到C++	
9.3.2 面向对象程序设计的基本概念	
9.4 应用举例	(186)
习题	(194)

第十章 应用调试技术

10.1 Turbo C 2.0 常见错误提示	(197)
10.2 常用调试技术	(206)

- 10.2.1 Turbo C 2.0 的安装和启动
- 10.2.2 集成开发环境
- 10.2.3 Turbo C 程序设计初步
- 10.2.4 Turbo C 程序设计的调试
- 10.2.5 Turbo C 编译、连接和运行时的常见错误
- 10.2.6 C 语言常见错误分析

习题..... (225)

模拟试卷

C 语言程序设计模拟试卷一..... (226)

C 语言程序设计模拟试卷二..... (229)

附录 1 C 语言常用编辑命令表..... (232)

附录 2 C 语言运算符的优先级别表..... (234)

附录 3 C 语言库函数..... (236)

参考文献..... (241)

后记..... (243)

第一章 C 语言简介

导语:本章概要地介绍 C 语言,包括它的发展史、特点以及 C 语言程序的格式和生成可执行程序的执行过程;而且介绍了 C 语言程序的运行环境 Turbo C 的发展、配置要求及对 C 程序的一些操作命令。

1.1 C 语言与 Turbo C

C 语言是 1972 年由美国 Dennis Ritchie 设计,并首次在一台使用 UNIX 操作系统的 DEC PDP-11 的计算机上实现的。它由一种早期的编程语言 BCPL 发展演变而来,BCPL 语言目前仍在使用,但主要是在欧洲。Martin Richards 改进了 BCPL 语言,从而促进了 Ken Thompson 所设计的 B 语言的发展,最终导致了 70 年代 C 语言的问世。

C 语言的特点是简洁、灵活、表达能力强,产生的目标代码质量高,可移植性好。归纳起来,它具有以下 6 个特点:

1. C 语言兼容了其他计算机语言的一些优点

其程序结构紧凑、简洁、规范,表达式简练、灵活、实用。用 C 语言编写的程序可读性强,编译效率高。

2. C 语言具有丰富的数据类型

在系统软件中,特别是操作系统,对计算机的所有软、硬件资源要实施管理和调度,这就要求有相应的数据结构作为操作基础。C 语言具有四种基本的数据类型即(字符型、整型、浮点单精度型、浮点双精度型)和多种构造数据类型(数组、结构、联合、枚举)以及复杂的导出类型。C 语言还提供了与地址密切相关的指针及其运算符。指针可以指向各种类型的简单变量、数组、结构和联合,乃至函数等。C 语言还允许用户自己定义数据类型。

3. C 语言具有丰富的运算符,多达 44 种

丰富的数据类型与丰富的运算符相结合,使 C 语言具有表达灵活和效率高等特点。

4. C 语言是一种结构化程序设计语言,特别适合于大型程序的模块化设计

C 语言具有编写结构化程序所必须的基本流程控制语句。C 语言程序是由函数集合构成的,函数各自独立,并且作为模块化设计的基本单位。C 语言的源文件,可以分割成多个源程序,分别进行编译,然后连接起来构成可执行的目标文件,为开发大型软件提供了极大的方便。C 语言还提供了多种存储属性,使数据可以按其需要在相应的作用域起作用,从而提高了程序的可靠性。

5. C 语言是处于汇编语言和高级语言之间的一种中间型程序设计语言,常被称为中级语言

C 语言把高级语言的基本结构和汇编语言的高效率结合起来。因此,它既具有高级语言面向用户、可读性强、容易编程和维护等特点,又具有汇编语言面向硬件系统,可以直

接访问硬件的功能,C 语言的运行效率可以与汇编语言媲美。

6. C 语言具有较高的可移植性

在 C 语言的语句中,没有依赖于硬件的输入输出语句,程序的输入输出功能是通过调用输入输出函数实现的,而这些函数是由系统提供的独立于 C 语言的程序模块,从而便于硬件结构不同的计算机之间实现程序的移植。

由于 C 语言具有上述诸多特点,因此近年来 C 语言迅速得到了广泛的普及和应用。特别是在微处理机和微型计算机的软件开发以及各种软件工具的研制中,使用 C 语言的趋势日益增强,呈现出 C 语言可能取代汇编语言的发展趋势。

随着微型计算机的日益普及,大量的 C 语言工具得以问世,这些工具程序高度地兼容。然而,由于没有统一的标准,这些工具之间也存在着不一致的地方。为了改变这种情况,ANSI(美国国家标准化协会)于 1983 年初专门成立了一个委员会,为 C 语言制定了 ANSI 标准。Turbo C 完全是按照 ANSI 的 C 语言标准实施的。它是一种快速的、高效的编译程序。Turbo C 不仅提供了一个功能强劲的综合开发环境,同时还按传统方式提供了一个命令行编译程序版本,以满足不同用户的需求。

1.2 Turbo C 的特点和配置要求

1.2.1 Turbo C 的发展及特点

Turbo C 是美国 Borland 公司的产品。该公司在 1987 年首次推出 Turbo C 1.0 产品,其中使用了全然一新的集成开发环境,即使用了一系列下拉式菜单,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。1988 年,Borland 公司又推出 Turbo C 1.5 版本,增加了图形库和文本窗口函数库等,而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能,并可以在 Tiny 模式下直接生成 .COM (数据、代码、堆栈处在同一 64K 内存中)文件。还可对数学协处理器(支持 8087/80287/80387)等进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++,它继承发展 Turbo C 2.0 的集成开发环境,并包含了面向对象的基本思想和设计方法。

1991 年为了适用 Microsoft 公司的 Windows 3.0 版本,Borland 公司又将 Turbo C++ 作了更新,即 Turbo C 的新一代产品 Borland C++问世。

1.2.2 Turbo C 2.0 基本配置要求

Turbo C 2.0 可运行于 IBM-PC 系列微机,包括 XT,AT 及 IBM 兼容机。此时要求 DOS 2.0 或更高版本支持,并至少需要 448K 的 RAM,可在任何彩、单色 80 列监视器上运行。支持数学协处理器芯片,也可进行浮点仿真,这将加快程序的执行。

Turbo C 2.0 有六张低密软盘(或两张高密软盘)。下面对 Turbo C 2.0 的主要文件作一简单介绍:

INSTALL.EXE 安装程序
 TC.EXE 集成编译
 TCINST.EXE 集成开发环境的配置设置程序
 TCHELP.TCH 帮助文件
 THELP.COM 读取 TCHELP.TCH 的驻留程序
 README 关于 Turbo C 的信息文件
 TCCONFIG.EXE 配置文件转换程序
 MAKE.EXE 项目管理工具
 TCC.EXE 命令行编译
 TLINK.EXE Turbo C 系列连接器
 TLIB.EXE Turbo C 系列库管理工具
 C0?.OBJ 不同模式启动代码
 C?.LIB 不同模式运行库
 GRAPHICS.LIB 图形库
 EMU.LIB 8087 仿真库
 FP87.LIB 8087 库
 *.H Turbo C 头文件
 *.BGI 不同显示器图形驱动程序
 *.C Turbo C 例行程序(源文件)

其中：上面的“?”分别为：

T Tiny(微型模式)
 S Small(小模式)
 C Compact(紧凑模式)
 M Medium(中型模式)
 L Large(大模式)
 H Huge(巨大模式)

1.3 简单 C 程序介绍

任何一种程序设计语言都具有特定的语法规则和一定的表示形式。按照一定的格式和构成规则书写程序,不仅可以使程序设计人员和使用程序的人员易于理解,更重要的是,把程序输入到计算机,计算机能够识别,并且执行它。

在分析 C 语言格式前,先来看一个用 C 语言编写的求半径为 R 的圆面积的程序。

```

#include<stdio.h>
#define PI 3.1416
main()
{
    int r;

```

```
float s;  
scanf("%d",&r);  
s=PI*r*r;  
printf("s=%f",s);  
}
```

由上面程序可以看出,C 语言的格式具有以下特点:

其一,C 语言程序习惯上使用英文小写字母书写,当然也可以用大写字母,但是大写字母在 C 语言中通常作为常量或其他特殊用途来使用。应该注意的是,C 语言对大小写是区分的,因此 NUM,num,Num 在 C 语言中代表着不同的变量。

其二,C 语言程序是由一条条语句组成的,每个语句都具有规定的语法格式和特定的功能。上面程序中,scanf()语句是输入变量数值的函数调用语句;s=PI*r*r 是赋值语句;printf()是输出数值的函数调用语句。

其三,C 语言程序不使用行号,这一点和 BASIC 语言有区别。

其四,C 语言程序使用分号“;”号作为语句的结束符。

其五,C 语言程序中,一个语句可以占用多行,一行也可以有多个语句。不过为了使程序易于阅读,一行上最好只写一个语句。

其六,C 语言程序中使用大括号对“{”和“}”来表示程序的结构层次范围。应该注意的是,左大括号“{”和右大括号“}”应该对应使用,当一个程序中层次多的时候,往往会产生某一左大括号“{”没有右大括号“}”与它对应,或者某一右大括号“}”没有左大括号“{”与它对应。这一点应引起每位编程者的注意。

其七,为了增加程序的可读性,可以使用适量的空格和空行。但是,变量名、函数名和 C 语言保留字中间不能加入空格。除此之外的空格和空行可以任意设置,C 语言编译系统是不会理会这些空格和空行的。

另外应该指出的是,C 语言程序的书写格式自由度较高,灵活性很强,有较大的随意性。但是为了避免程序层次混乱不清,还是建议用户在平时编程中,养成良好的编程习惯,使自己编写的程序既便于别人阅读和理解,也便于自己今后的修改。

1.4 C 语言可执行程序的产生过程

使用程序解决一个问题,首先要理清解题思路,也就是说要写出解题的算法;然后根据算法编写程序,人们把编写程序这一过程称为编辑源程序;第三步是编译程序,也就是把用高级语言编写的程序翻译成计算机能直接识别的二进制程序;第四步是连接库函数;最后一步是运行程序,程序通过编译及连接后产生出一个可执行程序,可执行程序的程序名的后缀为 .exe,现一一加以介绍。

1.4.1 编辑程序

根据算法写出源程序。选择适当的编辑程序,将 C 语言源程序通过键盘输入到计算机中,并以文件的形式存入到磁盘。在 DOS 系统下,经过编辑后得到的源程序文件都是

以 .C 为其文件扩展名。下面以例 1 为例来说明在 Turbo C 2.0 环境下编辑程序。

在 Turbo C 2.0 环境下编辑 C 语言程序,首先要进入 Turbo C 2.0 编辑环境。如果计算机安装的是 Windows 操作系统,则进入 Turbo C 2.0 编辑环境有两种方法,即一是进入 DOS 环境,如图 1-1 所示,找到 TC 的路径,在 TC 目录下直接输入 edit 文件名即可,如图 1-2 所示,注意文件名的扩展名是 .C;二是找到 Turbo C 2.0 编辑程序所在的文件夹后打开文件 TC,如图 1-3 所示。

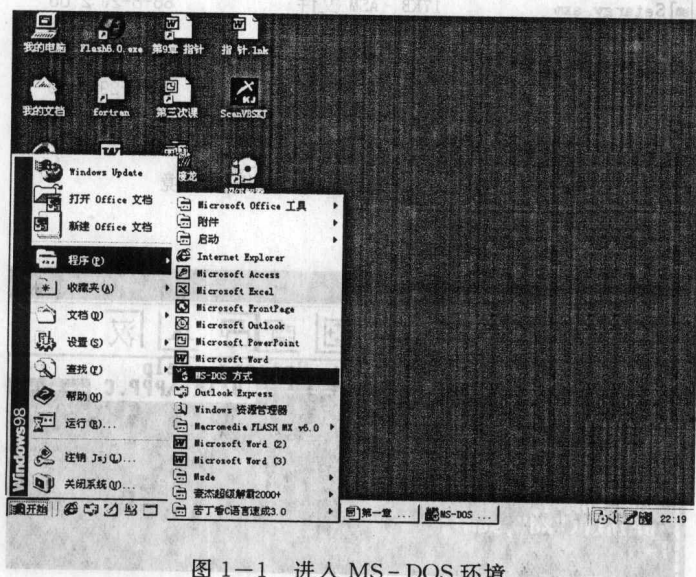


图 1-1 进入 MS-DOS 环境

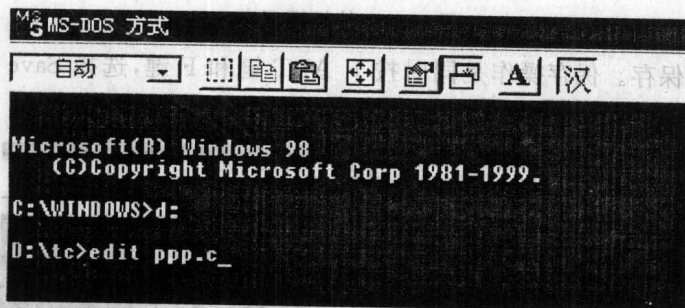


图 1-2 进入 Turbo C 2.0 编辑环境

进入 Turbo C 2.0 编辑环境后,直接输入源程序,源程序为:

```
main()
{
int i,sum=1;
for(i=2;i<=6;i++)
sum=sum+i;
printf("%d",sum);
}
```

图 1-1 图

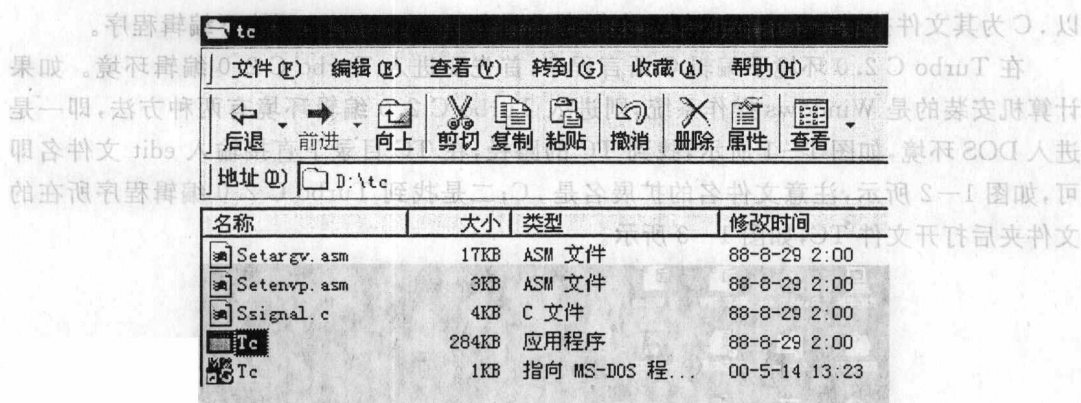


图 1-3 进入 Turbo C 2.0 编辑环境

如图 1-4 所示：

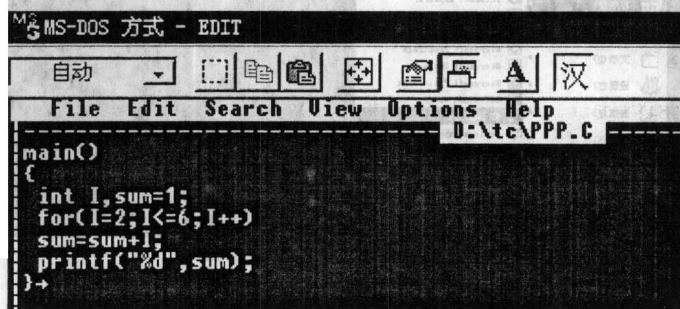


图 1-4 编辑源程序

输入完毕后保存。保存操作是同时按下 ALT 键和 F 键,选择 Save 保存源程序,如图 1-5 所示。

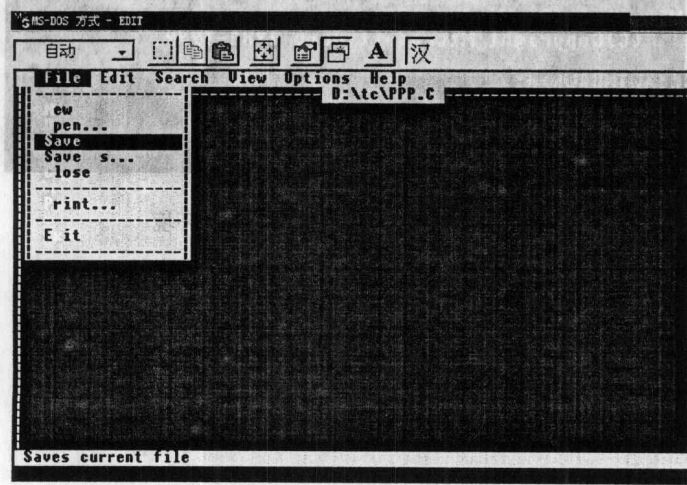


图 1-5 保存源程序

1.4.2 编译程序

通过编辑程序将源程序输入到计算机后,需要经过 C 语言编译器将其生成目标程序。编译程序的过程可以通过在编辑窗口中同时按下 ALT 键和 F9 键来完成。在对源程序的编译过程中,可能会发现程序中的一些语法错误,这时就需要重新利用编辑程序来修改源程序,然后再重新编译。在 DOS 系统下,经过编译后得到的目标文件都是以 .OBJ 为其文件扩展名。

1. 编译程序和解释程序

编译程序与解释程序这两个术语是实现程序运行的途径。从理论上来说,任何编程语言既可以编译,也可以解释,但很多语言通常只通过编译或只通过解释来运行。如, BASIC 语言常通过解释来运行,而 C 语言常通过编译实现运行。

解释程序一次只读一行源程序,并且执行该行源程序所指定的操作。而编译程序则读进整个源程序并将其转换为目标代码。目标代码是按照计算机能够直接运行的形式而实现的对源程序的一种翻译,也叫二进制码或机器码。

当使用解释程序时,每次运行用户程序时都必须有解释程序存在。例如,在 BASIC 中,必须先运行 BASIC 解释程序,然后才能装入用户自己的程序,每运行一次都要键入“RUN”,然后由 BASIC 解释程序逐行检查程序是否正确,若正确则执行。与此不同的是,C 语言的编译程序将源程序转换为目标代码,然后由计算机直接运行。编译过程本身占用了一些额外的时间,但当用户运行程序时,这一点很容易得到补偿。经过编译的程序比在解释环境下运行的程序快得多。惟一的例外是程序特别短,不足 50 行,且没有循环语句。

2. 分块编译

大部分较短的 C 语言程序完全可以包含在一个源文件里。然而,随着程序变长,编译时间也就长了。Turbo C 允许把程序分割成许多块,分别装入不同的文件里,对每个文件单独进行编译,然后再把它们连接起来,形成一个可以运行的目标代码文件。分块编译的优点在于修改一个源文件的代码后,只是对这一个文件进行编译,而不必对整个程序中的所有文件都编译一遍,这样可以节省大量的时间。

1.4.3 库函数和连接

Turbo C 提供了一个标准库,其中包含许多完成大部分常规功能所需的库函数。当程序中调用一个未曾定义的函数时,Turbo C 记住这个函数名,然后由连接程序将该程序与标准库中该函数的目标代码连接起来,这一过程就是连接。经过编译后生成的目标文件是不能直接执行的,它需要经过连接之后才能生成可执行的代码。在 DOS 系统下,连接后所得到的可执行文件都是以 .EXE 为其文件扩展名,连接命令为 link。通常,高级语言在编译程序的同时就进行了连接,因此使用前面的编译命令,可同时完成编译和连接这两个操作。

1.4.4 运行程序

1.4.4 运行程序

经过编译、连接之后,源程序文件就可以生成可执行的文件,这时就可以执行了。在DOS系统下,只要键入可执行程序的文件名,并按“回车”键后,就可执行此程序了。现在有许多厂家都推出一种集成环境来处理C语言程序。如Turbo C,在这种集成环境下,对程序的编辑、编译和连接等操作,都可以在一个窗口下进行,使用起来非常方便,Turbo C程序环境下,运行程序的命令是在编辑窗口中同时按下Ctrl键和F9键来完成。如果想要查看运行的结果可在编辑窗口中同时按下Ctrl键和F5键来完成。如上述例题的结果为21,如图1-6所示。

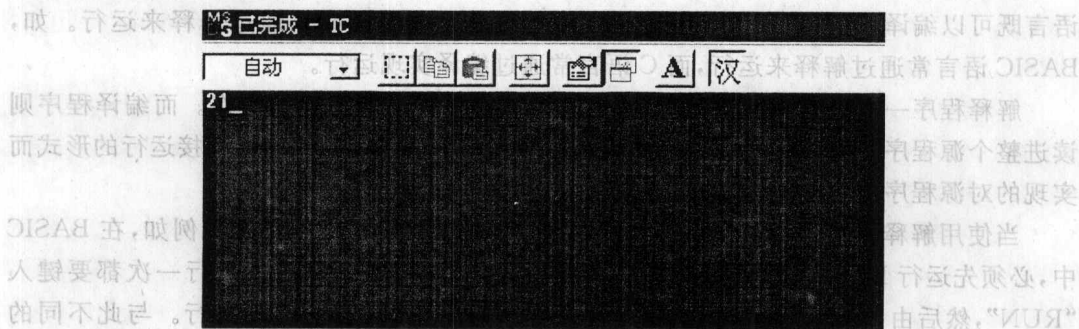


图 1-6 程序运行结果

1.4.5 简单的C语言程序的运行过程

这一节将通过介绍简单的C语言程序,并对其基本语法成分进行相应的说明,使读者对使用C语言程序来解决问题有一个概括的了解。

例1:从键盘输入两个整数,并将这两个整数之和显示出来。

```
#include "stdio.h"
main()
{
    /* 计算两个整数之和 */
    int x,y,z;
    scanf("%d%d",&x,&y); /* 读入两个整数,存入变量x和y中 */
    z=x+y;
    printf("The sum of %d and %d is %d",x,y,z);
}
```

上述程序经过编辑、编译和连接后,执行情况如下:

当程序执行到scanf语句时,将等待用户输入两个整型数据后再继续执行,假如用户输入10和20(此时,x的值为10,y的值为20),则程序将显示如下信息:

The sum of 10 and 20 is 30

下面,对上述程序进行说明:

(1) 程序中由/*和*/括起来的内容是程序的注释部分,它是为增加程序的可读