

21世纪高等学校计算机基础教育系列教材

21 SHIJI GAODENG XUEXIAO JISUANJI JICHU JIAOYU XILIE JIAOCAI

C语言 程序设计

案例 教程

■ 杨旭 主编

■ 张晓蕾 蒋凌云

魏雪英 王浩轩 等 编著



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目(CIP)数据

C 语言程序设计案例教程/杨旭主编; 张晓蕾等编著. —北京: 人民邮电出版社, 2005.1
(21 世纪高等学校计算机基础教育系列教材)

ISBN 7-115-13004-3

I. C... II. ①杨... ②张... 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 (2005) 第 001410 号

内容简介

本书采用任务驱动方式进行讲解,以程序实例为主导,将知识点融入实例,以实例带动知识点的学习。在按实例进行讲解时,充分注意保证知识的相对完整性和系统性,通过学习实例掌握软件的操作方法和操作技巧。

全书共分 11 章,具有较大的知识信息量,从程序设计的基础知识、流程控制到最后的文件操作与图形绘制,通过近 80 个实例的分析讲解,再利用 100 余道习题的练习与巩固,由浅至深,层层引导,让学生能够快速掌握 C 语言,提高编程能力。程序实例有详细的讲解,容易看懂、便于教学。本书内容丰富、结构清晰、图文并茂,易于教学与个人自学。

本书可以作为高等学校本科非计算机专业、专科计算机专业的教材,还可作为初学者的自学用书。

21 世纪高等学校计算机基础教育系列教材

C 语言程序设计案例教程

- ◆ 主 编 杨 旭
编 著 张晓蕾 蒋凌云 魏雪英 王浩轩 等
责任编辑 邹文波
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67129259
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 20.25
字数: 488 千字 2005 年 1 月第 1 版
印数: 1—5 000 册 2005 年 1 月北京第 1 次印刷

ISBN 7-115-13004-3/TP · 4398

定价: 27.00 元

本书如有印装质量问题,请与本社联系 电话: (010) 67129223

编 者 的 话

C 语言由于其强大的功能,在计算机的各个领域内得到了广泛的应用,操作系统从 DOS、UNIX 到 Windows 都是使用 C 语言设计的。虽然 C 语言是高级语言,但它也可以完成许多只有低级语言才能完成的、面向机器的底层工作,因此它也被称为“中间语言”。正是由于 C 语言的这些特性,决定了它成为一种重要的程序设计语言。

本书内容丰富、结构清晰、图文并茂,易于教师进行教学与读者自学。全书共分 11 章,具有较大的知识信息量,从程序设计的基础知识、流程控制到最后的文件操作与图形绘制,通过近 110 个实例的分析讲解,再利用 100 余道习题的练习与巩固,由浅至深,层层引导,使学生能够快速掌握 C 语言,提高编程能力。程序实例有详细的讲解,容易看懂、便于教学。

全书采用任务驱动方式进行讲解,以程序实例为主导,将知识点融入实例,以实例带动知识点的学习。在按实例进行讲解时,充分注意保证知识的相对完整性和系统性,使读者通过学习实例掌握 C 语言的程序设计方法和程序设计技巧。

本书的作者有丰富的教学和实践经验,并总结了一套任务驱动式的教学方法。采用这种方法学习的学生,十分容易掌握 C 语言的程序设计方法和程序设计技巧。

在本书的编写过程中,作者在内容选取、概念引入和文字叙述等各方面,都力求遵循面向实际应用、重视实践、便于自学的原则,注重理论联系实际,强调对分析问题、解决问题能力的训练与培养。

本书主编为杨旭,策划主审为沈大林。参加本书编写工作的主要人员有:杨旭、蒋凌云、魏雪英、王浩轩、曲彭生、姜树昕和张晓蕾等。

本书可以作为高等院校非计算机专业和高职高专计算机专业的教材,也适于作为初学者的自学用书。本书配套的电子教案及相关资料请到人民邮电出版社的网站(www.ptpress.com.cn)上下载。

由于作者水平有限,加上编著、出版时间仓促,书中难免有不妥之处,恳请广大读者批评指正。

编者

目 录

第 1 章 C 语言概述	1
1.1 程序与 C 语言	1
1.1.1 什么是程序	1
1.1.2 C 语言能够做什么	1
1.1.3 怎样学习 C 语言	2
1.2 第一个 C 语言程序	2
1.2.1 在 Turbo C 中编程	2
1.2.2 在 Visual C++ 中编程	6
1.2.3 程序的调试	10
习题	13
第 2 章 C 语言程序基础	15
2.1 主函数的基本结构	15
2.1.1 C 语言程序的入口与结束	15
2.1.2 主函数结构	17
2.2 C 语言程序的基本元素	18
2.2.1 标识符与关键字	18
2.2.2 常量与变量	19
2.2.3 数据类型	21
2.3 程序中的运算	23
2.3.1 运算符与表达式	24
2.3.2 类型转换	31
2.4 综合实例	34
习题	35
第 3 章 数据的输入/输出	38
3.1 数据的输入	38
3.1.1 格式化输入	38
3.1.2 控制输入的格式	40
3.2 数据的输出	43
3.2.1 格式化输出	43
3.2.2 控制输出的格式	45
3.3 综合实例	48

习题	49
第 4 章 程序控制结构	51
4.1 算法与程序控制结构	51
4.1.1 算法和控制结构的概念	51
4.1.2 顺序结构	56
4.2 选择结构	56
4.2.1 if 语句	57
4.2.2 switch 语句	63
4.2.3 选择结构的嵌套	67
4.3 循环结构	70
4.3.1 while 语句	70
4.3.2 do-while 语句	71
4.3.3 for 语句	73
4.3.4 循环嵌套	77
4.4 中断、接续、转向及返回语句	79
4.4.1 中断语句 break	79
4.4.2 接续语句 continue	80
4.4.3 转向语句 goto	82
4.4.4 返回语句 return	86
4.5 综合实例	86
习题	89
第 5 章 函数	92
5.1 函数的定义与调用	92
5.1.1 函数的定义	92
5.1.2 函数的说明	96
5.1.3 函数的调用	97
5.2 函数参数的传递	100
5.2.1 赋值调用	100
5.2.2 传址调用	101
5.3 函数的嵌套与递归	105
5.3.1 函数嵌套	105
5.3.2 函数的递归调用	106
5.4 变量的作用域	111
5.4.1 局部变量	111
5.4.2 全局变量	112
5.4.3 变量的存储类型	114
5.5 main() 函数	118

5.5.1	main()函数的参数	118
5.5.2	带参数 main()函数的应用	120
5.6	标准函数	121
5.6.1	标准输入/输出函数	122
5.6.2	字符函数与字符串处理函数	122
5.6.3	数学函数	123
5.6.4	其他函数	125
5.7	程序实例	125
	习题	128
第 6 章	数组与字符串	130
6.1	一维数组	130
6.1.1	数组	130
6.1.2	一维数组的定义	131
6.1.3	一维数组的引用	132
6.2	多维数组	135
6.2.1	多维数组的定义	135
6.2.2	多维数组的引用	137
6.3	字符串与字符串数组	142
6.3.1	字符数组与字符串	142
6.3.2	字符处理函数	147
6.4	程序实例	150
	习题	153
第 7 章	指针	155
7.1	指针的定义与应用	155
7.1.1	变量在内存中的地址	155
7.1.2	指针的定义	156
7.1.3	指针的引用	157
7.2	指针与数组	160
7.2.1	指针的运算	160
7.2.2	指针与数组	161
7.3	指针与字符串	164
7.3.1	字符串指针	165
7.3.2	字符指针与字符数组	166
7.3.3	指针数组	168
7.4	程序实例	169
	习题	172

第 8 章 复杂数据类型	174
8.1 结构体	174
8.1.1 结构体的定义	174
8.1.2 结构体变量的引用	178
8.1.3 结构体数组	182
8.1.4 自定义数据类型	184
8.1.5 链表	187
8.2 共用体	191
8.2.1 共用体的定义	191
8.2.2 共用体的引用	192
8.2.3 共用体与结构体的异同	194
8.3 位运算	194
8.3.1 位运算	195
8.3.2 位段	200
8.4 枚举	203
8.5 程序实例	206
习题	213
第 9 章 编译预处理	215
9.1 宏定义	215
9.1.1 不带参数的宏定义	215
9.1.2 带参数的宏定义	216
9.1.3 宏的作用范围	218
9.1.4 宏定义与函数的区别	218
9.2 文件包含	218
9.3 条件编译	220
9.3.1 条件编译的意义	220
9.3.2 条件编译语句	221
习题	223
第 10 章 文件	225
10.1 数据文件概念	225
10.1.1 数据文件	225
10.1.2 文件分类	225
10.1.3 文件与指针	226
10.1.4 文件系统	227
10.2 文件的打开和关闭	228
10.2.1 文件的打开	228

10.2.2 文件的关闭	228
10.2.3 文件检测函数	229
10.3 文件的输入/输出	230
10.3.1 文件的字符输入/输出函数	230
10.3.2 文件的字符串输入/输出函数	231
10.3.3 文件的格式化输入/输出函数	234
10.3.4 文件的数据块输入/输出函数	236
10.4 文件的定位操作	238
10.4.1 随机读写	238
10.4.2 文件指针的重置	239
10.4.3 取文件指针的当前位置	240
10.5 综合实例	240
习题	244
第 11 章 C 语言的图形	248
11.1 图形处理基础	248
11.1.1 图形处理与显示适配器	248
11.1.2 文本模式	249
11.1.3 图形模式	250
11.2 文本模式下的图形处理	250
11.2.1 文本模式设置	250
11.2.2 文本颜色的设置	251
11.2.3 文本窗口与文本输出	252
11.2.4 文本图形的绘制	255
11.3 图形模式下的图形处理	261
11.3.1 图形系统的初始化	261
11.3.2 基本图形绘制	264
11.3.3 封闭图形	270
11.3.4 屏幕操作	275
11.4 图形方式下的文本输出	277
11.4.1 文本的输出	278
11.4.2 输出文本的格式设置	279
11.5 图形程序的独立运行	282
11.5.1 驱动程序 BGI 的使用	282
11.6 综合实例	284
习题	296
附录 1 ASCII 码对照表	298

附录 2 运算符的优先级与结合性	300
附录 3 标准库函数.....	302
附录 4 TC 2.0 的菜单命令	309

第 1 章

C 语言概述

1.1 程序与 C 语言

1.1.1 什么是程序

就硬件而言,计算机只是一台设备,它本身不能做任何工作,是各种软件让计算机能够完成不同的工作。例如,安装了办公软件,计算机可以完成打字排版、绘制表格的工作;安装了网络软件,人们就可以在网上查询资料、娱乐、聊天。如果把计算机比作一个能干的工人,那么计算机的硬件就是它的躯体,而软件就是它的大脑,由大脑指挥躯体来完成各种不同的工作。

软件由程序和程序的相关文档(如说明书、源代码等)组成,程序是软件中最重要的部分,计算机的工作都是在程序的控制下完成的。

程序由一系列的指令所组成,存储在计算机中。当执行程序时,将自动按一定的顺序逐条地调用指令来完成工作。

1.1.2 C 语言能够做什么

计算机不懂得人类的语言,它只能理解由 0 和 1 组成的二进制代码指令,而这样的指令要人来理解则是相当困难的。为了便于学习和操作,人们使用了接近自然语言的程序语言来完成程序设计,这种语言被称为“高级语言”,而二进制的语言则被称为“机器语言”。

C 语言是近年来在国内外迅速推广应用的计算机语言。虽然可以进行程序设计的高级语言有很多种,如 Basic、Pascal 等,但是功能最强大、被大多数的程序员所认可的,还是 C 语言。

C 语言虽然是高级语言,但也可以完成许多只有低级语言才能完成的、面向机器的底层工作,因此也被称为“中间语言”。

C 语言功能丰富,表达能力强,使用相当自由和灵活。正是由于这些特性,决定了它成为一种重要的程序设计语言,我们日常所使用的程序中,大多是由 C 语言编写而成,例如,使用最多的个人电脑操作系统 Windows 98/2000 就有相当多的部分是由 C 语言编写的。

1.1.3 怎样学习 C 语言

C 语言是众多后继课程的基本编程工具，特别是与 Windows 编程有关的课程。因此，与计算机相关的专业都把 C 语言程序设计列为基础课程之一。

C 语言的灵活性能给熟练的用户带来方便，同时也给初学者带来了许多麻烦。在 Windows 操作系统取得了巨大成功之后，计算机相关专业的课程发生了很大变化（例如，Windows 编程已逐渐变得轻松、愉快），并且将会继续产生更多、更深刻的变化，这些变化必然会反映到 C 语言的教学中来。C 语言的教学如何适应这种趋势，如何快速地掌握 C 语言；教学内容如何做到少而精，而又不失 C 语言的精华，这都是当前面临的问题。

在学习过程中，读者要注意把握 C 语言教学内容的重点，掌握 C 语言基本知识和基本程序结构。阅读程序是学习 C 语言的重要手段。根据学到的有关基本知识，阅读和分析一些典型实例程序有利于检验和提高对基本知识的理解，同时也为学习正确编写程序打好基础。因此初学者必须善于阅读和分析程序。

C 语言是一门实践性课程，上机实验是必不可少的教学环节。学习任何高级语言编程的“秘诀”就是多实践。通过上机实践不仅可以深化和巩固讲授的理论知识，而且能够“真正”学会使用它们来编写并调试程序。因此，读者必须十分重视上机实验。

1.2 第一个 C 语言程序

1.2.1 在 Turbo C 中编程

“工欲善其事，必先利其器”，要学习 C 语言程序设计，必须先学会使用 C 语言的编程工具。

我们使用 C 语言编写的程序称为程序的“源代码”。由于计算机只能识别二进制代码指令，所以必须对源代码进行编译，将其转换为二进制指令，才能被计算机所识别和执行。

可以用于 C 语言程序设计的工具很多，其中最著名、使用最多的就是 Borland 公司的 Turbo C 系列。

Turbo C（以下简称 TC）以其编译的速度快、代码执行效率高而著称，是 C 程序员最乐于使用的工具，其中广泛使用的是 TC 2.0 和 TC 3.0。在本书中，我们将学习如何使用它们进行 C 语言程序的设计。

1. 启动 TC 2.0

TC 2.0 是在 DOS 下运行的程序，TC 2.0 的安装极为简单，只要插入安装盘，在 DOS 提示符下输入 Install，再按回车键，接下来按提示信息进行操作就能够完成。

在 TC 2.0 安装完成后，在 DOS 提示符下输入 TC，按下回车键就可以启动。启动后的程序界面如图 1.1 所示。

注释：关于 TC 2.0 集成环境菜单命令的使用，请查阅本书附录 4 “TC 2.0 的菜单命令”。

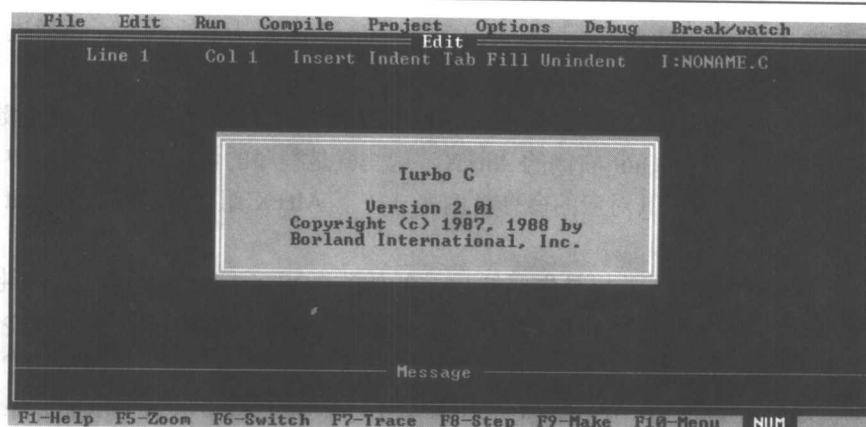


图 1.1 TC 2.0 启动界面

此时，连续按两下 Esc 键，可以看到屏幕中有黄色的光标在闪烁，此时就可以编辑程序了。

在屏幕窗口中，最上方的是程序下拉菜单，其下是程序编辑区域，也称为编辑窗口，在编辑窗口上方，是所编辑程序文件的相关信息，例如，Line 表示正在编辑的行号，Col 表示正在编辑的列号，Insert 表示现在是在插入状态下编辑，“I: NONAME.C”表示正在编辑的程序文件名。

在编辑窗口下方的是 Message（信息）窗口，在程序编译时，该窗口显示相关的编译信息。窗口的最下方是编辑时最常用的快捷键，例如，按 F1 键显示帮助、按 F10 键激活菜单等。

TC 2.0 的下拉菜单和 Windows 下的菜单一样，不过由于 TC 2.0 是 DOS 下的程序，基本上操作都是用键盘来完成，这里先学习如何使用键盘来操作菜单。

可以用两种方法来进行菜单的操作。

(1) 在屏幕窗口上方的下拉菜单中，都有一个字母是红色的，例如，File 中的 F、Edit 中的 E、Run 中的 R，这些字母称为菜单的热键。要选择命令时，可以通过同时按住 Alt 键和热键来打开菜单，再通过方向键上下或左右移动高亮的亮条来选择命令，再按下回车键来执行菜单命令，如图 1.2 所示。

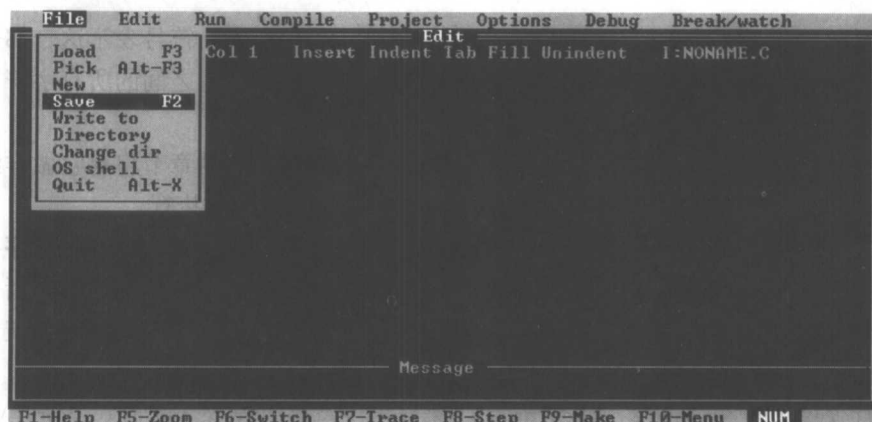


图 1.2 使用菜单的热键进行菜单操作

也可以先按 F10 键激活菜单,此时可以看到“File”菜单为高亮状态,再通过方向键移动高亮的亮条来选择命令。

(2) 在如图 1.2 所示的屏幕中,可以看到部分的命令右侧有菜单项的名称或按键的组合,如图中的 Load 右侧的 F3, Quit 右侧的 Alt+X, 这些就是程序的快捷键,在程序中,按下该键(或组合键)就可以执行其对应的命令,例如,按下 Alt+X 组合键将执行“Quit”命令,退出程序。

TC 中也可以输入中文,前提是先要打开一个中文 DOS 系统平台,如果是在纯 DOS 环境下可运行 UC DOS 等中文系统,如果是在 Windows 95 的 DOS 环境中则输入命令 PDOS95 再回车,就可以启动 Windows 95 中所附带的中文环境。启动中文系统后再启动 TC,就可以在其中输入中文了。因此,在 Windows 95 中文版下,启动 TC 2.0 并且要求能够显示汉字的完整操作步骤如下。

- 从 Windows 的“开始”菜单,进入 MS-DOS 命令提示符环境。
 - 键入 PDOS95 并回车: C:\Windows>PDOS95 ✓
- 这时屏幕右下角显示“Windows 95 中文 DOS 方式”,表示已进入汉字显示状态。
- 使用 CD 命令转到 TC 2.0 的安装目录,键入 TC 并回车,装载 TC.EXE。

2. 编辑与保存代码

启动 TC 2.0 后,按 Alt+F 组合键激活“File”菜单,使用方向键移动亮条到“New”,按回车键创建新文件,此时就可以在编辑窗口中编辑程序的代码了。

在编辑窗口中键入如下的代码。

```
/* TC1-1.c */
/* 第一个 C 程序 */

#include "stdio.h"
void main()
{
    printf("Welcome to C World!\n"); /* 显示信息 */
}
```

代码的前三行为文件头,对程序进行相关说明。“/* */”符号表示这一部分为注释性文字,不作为程序代码运行。“/*”与“*/”必须成对出现,两者之间的所有字符(可以是多行)均为注释文字。

下面的一行以#include 开始,这是程序的预处理命令(在以后的章节将对它做详细解释),引用了 C 语言的标准库“stdio.h”,使下面的 printf()语句得以顺利执行。

main()是 C 语言程序的主函数, C 语言源程序的基本单位是函数。每个 C 程序有且仅有一个主函数,所有的 C 程序都是从这里开始执行。下一行的左花括号“{”与最后一行的右花括号“}”之间是函数的主体,它们必须成对出现。main()函数前面的“void”表示主函数没有返回值。

第七行使用了 C 语言的输出函数 printf()来输出字符串“Welcome to C World!\n”,在 C 语言中的字符串都必须用半角双引号(“”)括起来。其中的“\n”是转义字符,表示换行。这

行语句的最后是分号(;)。注意：所有的C语句都必须以分号结束。该行最后用中文注释该函数的功能。

接下来对程序进行保存，按F10键激活菜单，选择执行“File”菜单下的“Save”命令，或者按下F2键，此时将弹出窗口提示输入文件的名称，默认的文件名为TC系统目录下的NONAME.C，如图1.3所示。

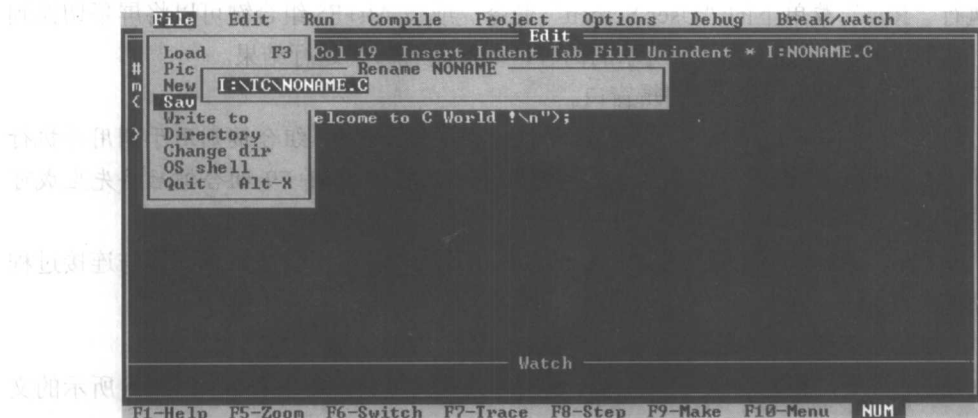


图 1.3 保存编辑好的程序

输入文件名 TC1-1.C 后按回车键，程序的保存完成。此时编辑窗口右上角的文件名由NONAME.C 改变为 TC1-1.C。

注意：在进行程序的编译执行前最好先保存程序，以免由于错误引起的程序崩溃而丢失文件。

3. 程序编译与连接

编辑完成后，按F10键激活菜单，选择“Compile”菜单下的“Make EXE File”命令，先将其编译为目标文件，然后连接为可执行文件，此时会出现如图1.4所示的信息窗口，提示编译成功。以上步骤也可通过按F9键来完成。

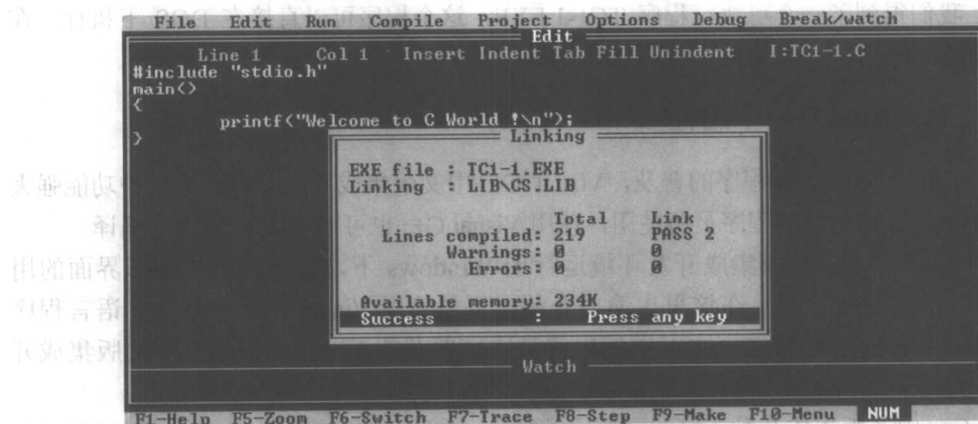


图 1.4 将源文件编译为目标文件

如果程序有错误，此时将弹出提示编译出错的信息窗口，并在屏幕下方的 Message 窗口

中显示相关的错误信息。

4. 运行程序与查看结果

要运行编辑完成的程序，执行“Run”菜单下的“Run”命令，或按 Ctrl+F9 组合键，可以看到屏幕闪了一下，此时程序已执行完成。

选择执行“Run”菜单下的“User Screen”命令，或按 Alt+F5 组合键可以将屏幕切换到用户屏幕。这时，可以看到如图 1.5 所示的内容，这就是程序的运行结果。

在用户屏幕按任意键可以回到编辑窗口。

说明：F9 键用于将源代码编译、连接为可执行程序；Ctrl+F9 组合键则用于调用并执行可执行程序，如果当前源代码并没有生成可执行程序，则按下 Ctrl+F9 组合键后将先生成可执行程序再执行。

现在，我们的第一个 C 语言程序设计完成了，接下来看看在代码编辑和编译与连接过程中生成了哪些文件。

执行“File”菜单下的“Quit”命令，或按 Alt+X 组合键，退出 TC 2.0。

在 DOS 提示符下，输入 DIR TC1-1.*命令，并回车执行它，可以看到如图 1.6 所示的文件列表。

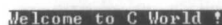


图 1.5 程序的运行结果



I:\tc 的目录		
2004-02-28	22:12	69 TC1-1.C
2004-02-28	22:34	464 TC1-1.OBJ
2004-02-28	22:34	8,358 TC1-1.EXE

图 1.6 代码编辑、编译与连接过程中生成的文件

可以看到有 3 个名为 TC1-1 的文件，这 3 个文件都是我们在编程中所创建的。其中，TC1-1.C 为 C 语言程序的源文件，也就是我们前面输入的程序代码。

TC1-1.OBJ 为目标文件，它包含的是编译器所生成的二进制机器指令代码。

TC1-1.EXE 是 TC 所生成的可执行文件，它包括了 TC1-1.OBJ 中的代码以及在程序连接时从相关运行库所得到的必需的支持过程。

现在，我们得到了一个可执行程序 TC1-1.EXE，这个程序可以直接在 DOS 下执行，在 DOS 提示符后输入 TC1-1，再回车就可以执行了。

1.2.2 在 Visual C++ 中编程

近年来，由于 C++ 语言程序的普及，Visual C++ 中文版集成开发环境作为一种功能强大的程序编译器也被相当多的程序员所使用，使用 Visual C++ 也可以完成 C 语言的编译。

由于 Visual C++ 中文版集成开发环境运行于 Windows 下，对于习惯于图形界面的用户来说是比较易学的，因此，在这里也简要介绍一下如何用 Visual C++ 来完成 C 语言程序设计。Visual C++ 有多种版本，本书中使用的是比较普及的 Visual C++ 6.0 中文版集成开发环境。

Visual C++ 中文版集成开发环境的安装比较简单，在输入 SETUP 执行后，可以按照提示信息完成程序安装，在这里不再详述。

1. 启动 Visual C++ 6.0

如图 1.7 所示,单击“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”命令,就可以启动 Visual C++ 6.0 中文版集成开发环境。

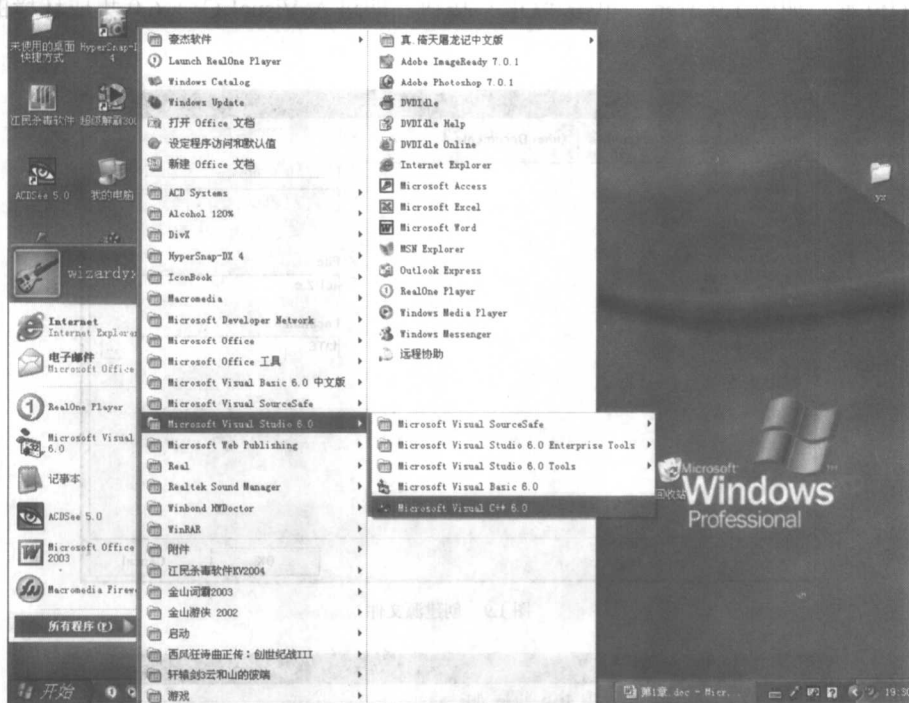


图 1.7 启动 Visual C++ 6.0 的方法

启动后的 Visual C++ 6.0 中文版集成开发环境如图 1.8 所示。

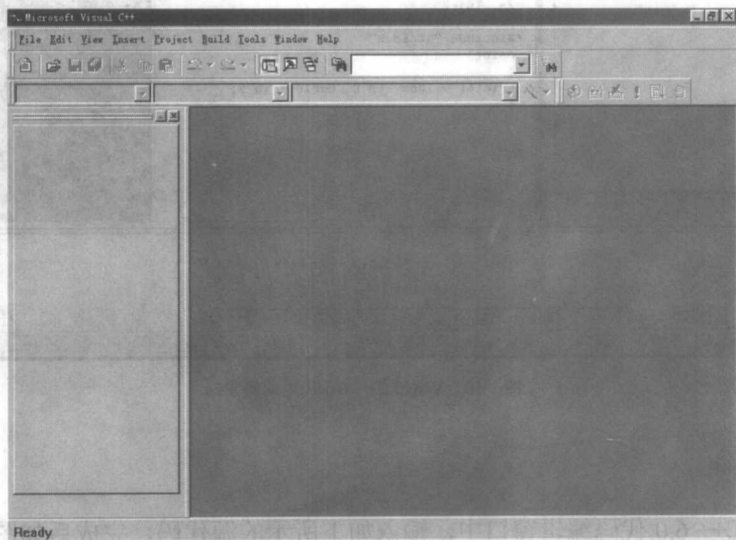


图 1.8 Visual C++ 6.0 中文版集成开发环境

2. 创建文件

单击主菜单中的“File”→“New”命令，在“New”对话框中选择“File”选项卡。在左边列出的选项中，选择“C++ Source File”；在右边相应的文本框中，输入文件名称“tc1-2.c”及保存的位置，如图 1.9 所示。单击“OK”按钮，即进入 Visual C++ 6.0 集成环境的代码编辑窗口，如图 1.10 所示。

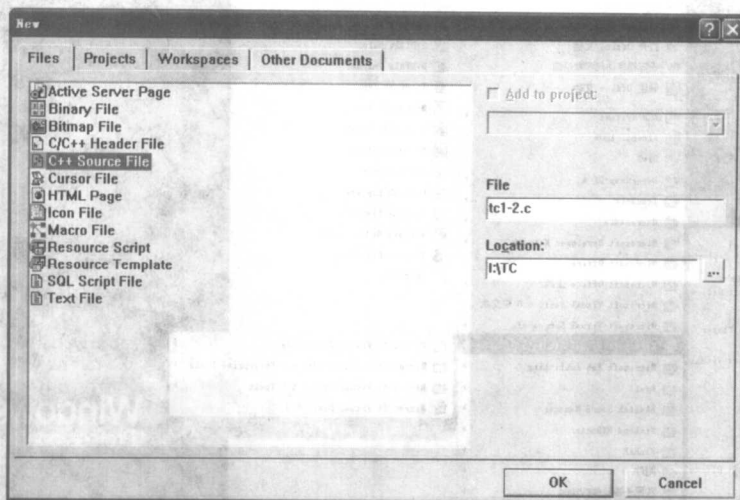


图 1.9 创建源文件

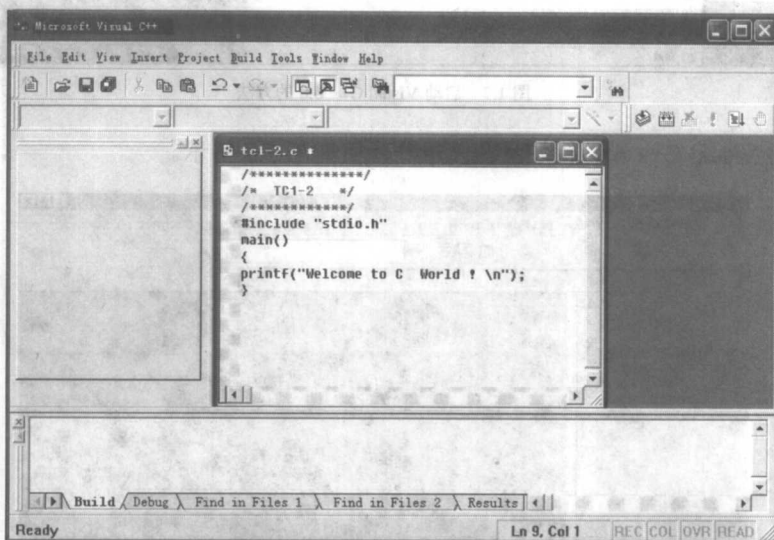


图 1.10 Visual C++ 6.0 代码编辑窗口

3. 编辑代码

在 Visual C++ 6.0 代码编辑窗口中，输入如下所示的源代码，完成后如图 1.10 所示。

```
/* TC1-2.c */
```