高职高专“十一五”规划教材

实用C语言程序设计教程

王冬艳　赛煜　主编

化学工业出版社

高职高专“十一五”规划教材

实用C语言程序设计教程

王冬艳　赛煜　主编
柏淑红　副主编

化学工业出版社
·北京·

C 语言是电子类专业的一门重要基础课程，其任务是培养学生建立程序设计概念和学会程序设计的基本方法，使学生具有应用计算机解决实际问题的初步能力，为进一步学习和熟练使用计算机编制高级程序打下基础。

本书主要介绍 C 语言基础、数据类型及其运算、C 语言的控制结构、数组、函数、指针、结构体、共用体与文件、C 语言应用等内容，还附有 C 语言库函数介绍、常用字符与 ASCII 代码对照表、C 语言语句的语法格式（常用）、C 语言运算符的优先次序、C 程序常见的出错信息浅析、C 语言程序中常见的语法错误浅析等内容。内容翔实，实例丰富，方便读者轻松学习、快速上手。

本书可作为高等职业院校电类、计算机类相关专业教材，也可供有关读者参考。

图书在版编目（CIP）数据

实用 C 语言程序设计教程/王冬艳，赛煜主编.—北京：化学工业出版社，2009.2
高职高专“十一五”规划教材
ISBN 978-7-122-04380-1

Ⅰ. 实… Ⅱ.①王…②赛… Ⅲ.C 语言-程序设计-高等学校：技术学院-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2008）第 206489 号

责任编辑：廉 静 张建茹 金 杰　　文字编辑：王 洋
责任校对：吴 静　　装帧设计：韩 飞

出版发行：化学工业出版社（北京市东城区青年湖南街 13 号 邮政编码 100011）
印 装：三河市延风印装厂
787mm×1092mm 1/16 印张 16½ 字数 442 千字 2009 年 3 月北京第 1 版第 1 次印刷

购书咨询：010-64518888（传真：010-64519686） 售后服务：010-64518899
网 址：http://www.cip.com.cn
凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定 价：28.00 元

前　言

本书结合C语言程序设计的教学经验，吸收了近年来国内教学发展的先进理论、方法和技术，立足培养面向工程实践的应用型人才而编写的。

本教材对C语言程序设计所涉及的基础知识和应用技术作了较为全面和系统的论述。全书共分为九章，介绍了C语言基础、数据类型及其运算、C语言的控制结构、数组、函数、指针、结构体、共同体与文件及C语言应用。

本书在编写过程中力求做到基础知识与应用技术并重，突出重点、层次分明、语言易懂，并列举了大量的例题和习题，便于读者自学和练习。

本书第三章、第八章和第九章由王冬艳编写，第一章、第二章由柏淑红编写，第四章由蔡琴编写，第五章、第六章由赛煜编写，第七章由周俊编写，附录由李俊杰和王冬艳共同编写。王冬艳负责全书的统稿。天津中德职业技术学院的孙锋副教授仔细审阅了全书并提出了许多宝贵的意见。在编写过程中，成都电子机械高等专科学校电气工程系的领导和同事给予了大力帮助，在此表示诚挚的感谢。

限于水平，本书中难免存在缺点和不足，恳请读者批评指正。

编者

2008.12

目 录

第一章　C 语言基础

本章主要介绍 C 语言和 C 程序设计的初步知识，学习完这个章节后，将会对 C 程序有一个初步的了解。在本章的学习中，要求熟悉 C 程序的基本结构和书写风格；掌握 C 语言关键字和标识符的命名方法；了解 C 编译系统提供的标题文件的功能；掌握 Turbo C 集成环境的使用。

第一节　C 语言简史及特点

一、C 语言的发展简史

C 语言是一种面向过程的通用程序设计语言，它以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征。C 语言不仅是一种高级语言，还兼有低级语言的功能。因此可用于编写系统程序，也可用于编写不同领域的应用程序。

C 语言的祖先是 ALGOL60（ALGOLrithm Language）。ALGOL60 具有可读性和可移植性好的特点，但它不能对硬件进行操作，不宜用来编写系统程序。人们开始考虑一种集高级语言和低级语言功能于一身的语言，以便可以用它编写可读性和可移植性都比较好的系统程序。

1963 年，英国的剑桥大学和伦敦大学首先将 ALGOL60 发展成 CPL。1967 年，剑桥大学的 Martin Richard 将 CPL 改制成 BCPL。1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 修改成 B 语言，并用 B 语言开发了第一个高级语言——UNIX 操作系统，在 DEC 公司的 PDP-7 小型机上运行。

1972 年，Ken Thompson 与在 UNIX 操作系统上的亲密合作者 Dennis M.Richie 将 B 语言修改设计成 C 语言。C 语言既保持了 BCPL 和 B 语言的精炼和接近于硬件的特点，也克服了它们过于简单、数据无类型等缺点。1973 年，Ken Thompson 和 Dennis M.Richie 又合作，将 UNIX 操作系统改用 C 语言编写，其中 C 语言代码占 90%以上，只保留了少量汇编语言代码。这样就使得 UNIX 操作系统向其他类型的机器上移植变得相当简单。到 70 年代中期，UNIX 操作系统和 C 语言作为软件设计师的良好工具开始风靡世界。

1978 年，以 UNIX 第 7 版中的 C 编译程序为基础，Brain W.Kernighan 和 Dennis M.Richie 合著了影响深远的《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，称为 K&R C 语言。其后的十几年中，适用于不同机种和不同操作系统的 C 语言编译系统相继问世，从而把 C 语言的应用推向了更加广泛、普及的阶段。1983 年，美国国家标准局（ANSI）制定了 C 语言标准。这个标准不断完善，并从 1987 年开始实施，称为 ANSI C。1988 年，Brain W.Kernighan 和 Dennis M.Richie 修改了他们的经典著作《The C Programming Language》，按 ANSI C 标准重新编写了该书。现在一般称 ANSI C 为新标准或现代 C，K&R C 为旧标准或传统 C。此后陆续出现的各种 C 语言版本（目前流行的 C 语言

编译系统大多是以 ANSI C 为基础进行开发的），如 Microsoft C 5.0、Microsoft C 9.0、Turbo C 2.0、Turbo C 3.0、Quick C 等都是与 ANSI C 兼容的版本。它们的语法和语句功能是一致的，差异表现在各自的标准库函数中收纳的函数种类、格式和功能上，尤其是图形函数库的差异更大一些。因此应了解所用的 C 语言编译系统的特点(可以参阅有关手册)。本书的叙述基本上以 ANSI C 为基础。

二、C 语言的特点

一种语言之所以能存在和发展，并具有较强的生命力，总是有其不同于（或优于）其他语言的特点。C 语言的主要特点如下。

① C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 语言既具有高级语言的功能，又具有低级语言的许多功能，可用来编写系统软件。C 语言的这种双重性使它既是成功的系统描述语言，又是通用的程序设计语言。因此有人把 C 语言称为“高级语言中的低级语言”或“中级语言”，意为兼有高级和低级语言的特点，但一般仍习惯将 C 语言称为高级语言。因为 C 语言程序也要通过编译、链接才能得到可执行的目标程序，这是和其他高级语言相同的。

② 数据类型丰富，具有现代语言的各种数据结构。C 语言提供的数据类型有：整型、实型、字符型、数组类型、指针类型、结构体类型、共同体类型等，能用来实现各种复杂的数据结构（如链表、树、栈等）的运算，尤其是指针类型数据，使用十分灵活和多样化。

③ 运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

④ 语言简洁、紧凑，使用方便、灵活。C 语言一共有 32 个关键字、9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一切不必要的成分。C 语言程序比其他许多高级语言简练、源程序短，因此输入程序时工作量少。

⑤ 具有结构化的控制语句（如 if…else 语句、while 语句、do…while 语句、switch 语句、for 语句）。用函数作为程序的模块单位，便于实现程序的模块化。C 语言是完全模块化和结构化的语言。

⑥ 语法限制不太严格，程序设计自由度大，例如，对数组下标越界不做检查，由程序编写者自己保证程序的正确。对变量的类型使用比较灵活，例如，整型数据、字符型数据以及逻辑型数据可以通用。一般的高级语言语法检查比较严，能检查出几乎所有的语法错误，而 C 语言允许程序编写者有较大的自由度，因此放宽了语法检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 语言编译程序去查错。“限制”与“灵活”是一对矛盾体。限制严格，就会失去灵活性；而强调灵活，就必然放松限制。一个不熟练的人员编一个正确的 C 语言程序可能会比编一个其他高级语言程序难一些。也就是说，对用 C 语言的人，要求程序设计更熟练一些。

⑦ 用 C 语言编写的程序可移植性好（与汇编语言比）。基本上不做修改就能用于各种型号的计算机和各种操作系统。

⑧ 生成目标代码质量高，程序执行效率高。C 语言一般只比汇编程序生成的目标代码效率低 10%～20%。

C 语言的这些优点，使它应用面很广。许多大的软件都用 C 语言编写，这主要是由于 C 语言的可移植性好、硬件控制能力强、表达和运算能力强。许多以前只能用汇编语言处理的

问题，现在可以改用 C 语言处理了。

第二节　C 语言的基本符号与关键字

C 程序的语句总是由 C 语言的基本符号和关键字构成的。

一、C 语言的基本符号

（1）标识符　标识符是用来表示变量名、数组名、函数名、指针名、联合名、枚举常数名、用户定义的数据类型名及语句标号等用途的字符序列，可由 1～32 个字符组成，第一个字符可以是字母或下划线，后面的字符可以是字母、数字或下划线，例如 BC、Bc、ab、AB、A_B、_ab、ab_、s2d、W_length 都是合法的标识符，而 A+b、A'b、a.b、2sdc、d%都是错误的标识符。

注意：标识符不能与 C 关键字相同，而且应区分大小写，例如 BC 和 bC 是两个不同的标识符；ELSE 可以作为标识符，它不会与 C 关键字 else 混淆。

顺便指出，在 C 编译系统的库函数中，经常使用以下划线“_”打头的函数名和变量名，所以程序中也应尽量避免使用以“_”打头的标识符，以免与库函数冲突。

ANSI C 标准没有规定标识符的长度（字符个数），但各个 C 编译系统都有自己的规定。有的系统（如 IBM-PC 的 MS C）取 8 个字符，假如程序中出现的变量名长度都大于 8 个字符，则只有前面 8 个字符有效，后面的不被识别，例如，有两个变量：student-name 和 student_number，由于两者的前 8 个字符相同，系统认为这两个变量是一回事而不加区别。可以将它们改为 stud-name 和 stud-num，以使之区别。Turbo C 则允许变量名有 32 个字符，因此，在写程序时应了解所用系统对标识符长度的规定，以免出现上面的错误，这种错误并不反映在编译过程中（即语法无错误），但运行结果不对。为了程序的可移植性（即在甲计算机上运行的程序基本上可以不加修改，就能移到乙计算机上运行）以及阅读程序的方便，建议变量名的长度最好不要超过 8 个字符。

（2）操作符　操作符包括各种运算符（如+、－、*、/等）及有特定意义的标点符号（如花括号、方括号、圆括号、逗号）等。

（3）分隔符　分隔符用来分隔相邻的标识符、关键字和常数。最常用的分隔符是空格，此外还可以用制表符、换行符、换页符等作为分隔符。

二、C 语言的关键字

关键字是 C 语言中有特定意义和用途且不得作为它用的字符序列，ANSI C 标准规定的关键字有 32 个，如表 1-1 所示。

表 1-1　ANSI C 关键字

Auto	Break	case	char	const	continue
default	Do	double	else	enum	extern
float	For	goto	if	int	long
register	Return	signed	short	sizeof	static
struct	Switch	typedef	union	unsigned	void
volatile	While				

注：所有 C 关键字都必须小写

第三节 C 语言程序的基本结构

C 程序的基本结构指的是一个 C 程序的基本组成成分。本节通过一个程序实例来说明 C 程序的基本结构。

一、一个简单的 C 程序

下面是一个简单的 C 程序，它只有一个 main()函数。

【例 1-1】 根据给定的半径编制程序，计算圆的周长 c 和面积 a。

先定义三个变量分别表示半径、周长和面积，然后指定半径 r 的值，由计算公式计算圆的周长和面积，并存放在变量 c 和 a 中，最后输出圆的半径 r、面积 a 和周长 c 的值。程序如下：

```
#include <stdio.h>
#define PI 3.14159                    /* 宏定义 */
    void main ( )                     /* 函数名 */
    {                                 /* 函数体开始 */
        float r，c，a;                    /* r：半径；c：周长a：面积 */
        r=1.5;                        /* 给定r的值 */
        c=2*PI*r;                     /* 计算周长 */
        a=PI*r*r;                     /* 计算面积 */
        printf("r=%f，c=%f，a=%f"，r，c，a); /* 输出r，c，a */
    }                                 /* 函数结束 */
```

程序第 1 行用#define 定义符号常数 PI，它代表圆周率；第 2 行是函数名 main ( )，C 语言规定：一个程序由若干个函数组成，至少有一个用 main( )命名的主函数；第 3 行至第 9 行为函数体，是程序的变量定义和执行部分，其中左花括号表示函数体的开始，右花括号表示函数体的结束。每行右边“/*”和“*/”之间的文字称为注释，它对程序的执行没有影响，只对程序进行必要的说明，以帮助阅读程序。程序运行结果如下：

```
r=1.500000，c=9.424770，a=7.068577
```

【例 1-2】 求两个数中较大者。

```
#include <stdio.h>
void main( )                          /* 主函数 */
{
        int max(int x，int y);            /* 对被调用函数max的声明 */
        int a，b，c;                      /* 定义变量a、b、c的值 */
        scanf("%d，%d"，&a，&b);          /* 输入变量a和b值 */
        c=max(a，b);                      /* 调用max函数，将得到的值赋给c */
        printf("max=%d\n"，c);            /* 输出c的值 */
}
int max(int x，int y)  /*定义max函数，函数值为整型，形式参数x、y为整型*/
{
        int z;          /*max函数中的声明部分，定义本函数中用到的变量z为整型*/
        if(x>y) z=x;
        else z=y;
        return(z);      /*将z的值返回，通过max带回到调用函数的位置*/
}
```

本程序包括两个函数：主函数 main 和被调用的函数 max。max 函数的作用是将 x 和 y 中

较大者的值赋给变量 z。return 语句将 z 的值返回给主函数 main。返回值通过函数名 max 带回到 main 函数中的调用 max 函数，而 max 函数的位置在 main 函数之后，为了使编译系统能够正确识别和调用 max 函数，必须在调用 max 函数之前对 max 函数进行声明。有关函数的声明详见后面章节，在此只初步了解即可。

main（）函数中 scanf 是输入函数的名字（scanf 和 printf 都是 C 的标准输入输出函数）。本程序中，scanf 函数的作用是输入 a 和 b 的值。

在程序第 7 行中调用 max 函数，在调用时将实际参数 a 和 b 的值分别传送给 max 函数中的参数 x 和 y（称为形式参数）。经过 max 函数得到一个返回值（即 max 函数中变量 x 的值），这个值返回到调用 max 函数的位置，即程序第 7 行“＝”的右侧，然后把这个值赋给变量 c。第 8 行输出变量 c 的值。在执行 printf 函数时，对双撇号括起来的部分（max=%d\n）是这样处理的：将“max=”原样输出，“%d”将由 c 的值取代，“\n”执行换行。程序运行情况如下：

```
6，2↙          （输入6和2，将其赋给a和b）
max=6          (输出c的值)
```

为了在分析运行情况时便于区别输入和输出的信息，本书在输入的信息下加了下划线，如上述运行情况的第 1 行表示：从键盘输入 6 和 2，然后按 Enter 键。这样就把 6 和 2 分别送到计算机内存中（本例是送给变量 a 和 b）。第 2 行则是从计算机输出的信息，显示在屏幕上。

本例用到了函数调用、实际参数和形式参数等概念，在此只做很简单的解释。如对此不大理解，可以先不予以深究，在学到以后相关章节时问题自然迎刃而解。介绍此例的目的，无非是使读者对 C 程序的组成和形式有一个初步的了解。

二、C 程序的基本结构

通过上一节介绍的两个例子可以看出：C 程序的基本结构是函数，一个 C 程序由一个或多个函数组成，一个 C 函数由若干条 C 语句构成，一条 C 语句由若干基本单词形成。

1．C 函数

C 函数是完成某个整体功能的基本单位，它是相对独立的模块。一个简单的 C 程序可能只有一个 main（）函数，而复杂的 C 程序则可能包含一个 main（）函数和若干个其他函数。所有的 C 函数都具有相同的结构：函数名、形式参数和函数体。

一个 C 函数由两部分组成。

（1）函数的首部　函数的第 1 行，包括函数名、函数类型、函数属性、函数参数（形式参数）名及函数参数类型。

例如，【例 1-2】中的 max 函数的首部为：

```
int      max      (int          x,          int          y)
 ↓        ↓        ↓           ↓           ↓           ↓
函数类型  函数名  函数参数类型  函数参数名  函数参数类型  函数参数名
```

一个函数名后面必须跟一对圆括号，括号内写函数参数名及其类型。函数也可以没有参数，如 main（）。

（2）函数体　函数首部下面的花括号内的部分。如果一个函数内有多个花括号，则最外层的一对花括号为函数体的范围。

函数体一般包括以下两部分。

① 声明部分。在这部分中定义所用到的变量并对所调用的函数声明，如【例 1-1】中 main( )函数中对变量的定义“int a，b，c;”和对所调用的函数声明“int max(int x，int y);”。

② 执行部分。由若干个语句组成。

当然，在某些情况下也可以没有声明部分，甚至可以既无声明部分也无执行部分，如：

```
void dump( )
{
    }
```

它是一个空函数，什么也不做，但这是合法的。

图 1-1 描述了 C 程序的一般格式，除了 main( )函数外，函数 fun1( )到 funn( )均为用户自行命名的函数。程序执行时，无论函数的书写位置如何，总是先执行 main( )函数，由 main( )函数调用其他函数，最后终止于 main( )函数。

```
  全局变量说明
main( )
{ 局部变量说明
      语句序列
}
fun1(形式参数表)
{ 局部变量说明
      语句序列
}
…
funn(形式参数表)
{ 局部变量说明
      语句序列
}
```

图 1-1 C 程序的一般格式

2．C 语句

C 语句是完成某种程序功能（如赋值、输入、输出等）的最小单位，所有的 C 语句都以分号结尾。C 语句可以分为表达式语句、复合语句和空语句三大类。

（1）表达式语句 任何一个 C 表达式的末尾加上分号后，就构成表达式语句，如

```
a=1;
printf("this is a C program");
```

等。

（2）复合语句 复合语句将一组 C 语句用花括号括起来，它在语法上被视为一条语句，如

```
while(i<=100)
{  sum=sum+i;
   i++;
}
```

复合语句通常用在条件分支或循环语句中。有时为了达到数据隐藏的目的，用复合语句形成一个代码块，块中定义的局部变量不会对程序的其他部分产生副作用。

（3）空语句 空语句指的是只有一个分号的语句，如

```
for(i=0;i<=100;i++)
;
```

它由一条 for 语句（循环语句）和一条空语句组成。空语句用作循环语句的循环体，表示什么也不做。事实上这个循环的功能是延迟一小段时间。有时空语句被用作转向点。

三、C 程序的书写风格

C 语言采用自由的书写风格。

① 每个函数在整个程序文件中的位置任意。主函数不一定出现在程序的开始处，但不管主函数位于何处，程序运行总是从主函数开始。

② 每个程序中的语句数量任意。既允许一行内写几条语句，也允许一条语句分写在多行上，但每条语句都必须以“；”结束。有时还可以在程序的适当地方（如两个函数之间）加进一个或多个空行，使程序结构更加清晰。

③ 注释的位置任意。注释可以出现在程序的任何地方，既可以独占一行或几行；也可以出现在某语句的开头和结尾处。如果注释占有几行，则每一行都要以“/*”开始，以“*/”结束，“*”和“/”之间不能有空格。注释不是 C 语句，它对程序的编译和运行没有影响，使用注释的唯一目的是增加程序的可读性。

④ 尽管 C 程序的书写几乎没有限制，但为使程序清晰易读，通常每行写一条语句。不同结构层次的语句从不同的位置开始，即按缩进格式书写成阶梯形状，可以用 Tab 键或空格键调整各行的起始位置。

第四节　C 语言程序的编辑及运行

C 语言是一种编译型程序设计语言，一个 C 程序要经过编辑、编译、连接和运行四个步骤才能得到正确结果。

1．编辑

编辑是指 C 语言源程序的输入和修改，最后以文本文件的形式存放在磁盘上，文件名由用户自行定义，扩展名一般为“.c”、例如 a.c、sample.c、eg.c 等。

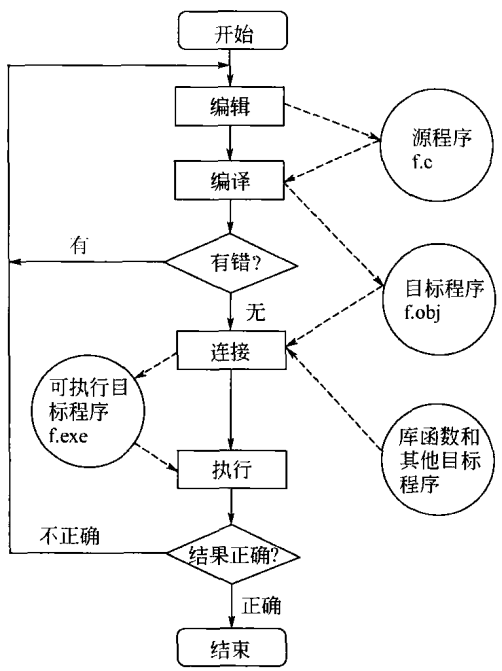

图 1-2　运行 C 程序的流程图

2．编译

编译是把 C 语言程序翻译成可重定位的二进制目标程序。编译过程由编译程序完成。编译程序自动对源程序进行句法和语法检查，当发现这类错误时，就将错误的类型和在程序中的位置显示出来，以帮助用户修改源程序的错误。如果未发现句法和语法错误，就自动形成目标代码并对目标代码进行优化后生成目标文件。目标文件的名称由编译系统自动规定，也可以由用户指定。

3．连接

连接也称链接和装配，是用连接程序将编译过的目标程序和程序中用到的库函数连接装配在一起，形成可执行的目标程序。可执行目标文件的扩展名（.exe）由系统自动确定。

4．运行

将可执行的目标文件投入运行由计算机执行后，可获取程序的运行结果。通常，在 DOS 操作系统提示符下直接键入可执行文件名，或在 Windows 操作系统下用鼠标双击可执行文件名即可。

上述四个步骤示于图 1-2。图中带箭头的实线表示操作流程，带箭头的虚线表示操作所需要的条件和产生的结果，例如，编辑后得到一个源程序文件 f.c，然后在进行编译时再将源程序文件 f.c 输入，经过编译后得到目标文件 f.obj，再将目标程序 f.obj 输入内存，与系统提供的库函数等连接，得到可执行的目标程序 f.exe，最后把 f.exe 调入内存并执行。

第五节　实训　Turbo C 系统的基本操作方法

为了编译、连接和运行 C 程序，必须要有相应的 C 编译系统。目前使用的大多数 C 编译系统都是集成环境（IDE）的，把程序的编辑、编译、连接和运行等操作全部集中在一个界面上运行，功能丰富、使用方便、直观易用。

可以用不同的编译系统对 C 程序进行操作，常用的有 Turbo C2.0、Turbo C++ 3.0、Visual C++等。前一段时间，Turbo C2.0 用的比较多，但 Turbo C2.0 是用于 DOS 环境的，在进入 Turbo C 环境后，不能用鼠标进行操作，主要通过键盘选菜单，不大方便。近年来，不少人使用 Turbo C++ 3.0。Turbo C++ 3.0 也是一个集成环境，把程序的编辑、编译、连接和运行等操作全部集中在一个界面上进行，它具有方便、直观和易用的界面。虽然它也是 DOS 环境下的集成环境，但是可以把启动 Turbo C++ 3.0 集成环境的 DOS 执行文件 tc.exe 生成快捷方式，并以图标形式放在 Windows 桌面上，只要双击该图标，就能进入 Turbo C++ 3.0 集成环境。此外，它可以用鼠标形式操作菜单，因此在 Windows 环境下使用也十分方便。也可以用 Visual C++对 C 程序进行编译，由于目前学习 C++的人多数使用 Visual C++ 6.0，因此在学习时使用 Visual C++集成环境也有利于今后进一步学习 C++语言。

学习本课程的目的主要是掌握 C 语言并利用它编制程序，写出源程序后可以用任何一种编译系统对程序进行编译和连接工作，只要用户感到方便、有效即可。不应当只会使用一种编译系统，而对其他的一无所知。无论用哪一种编译系统 ，都应当能举一反三，在需要时会用其他编译系统进行工作。

本节主要介绍在 Turbo C++ 3.0 中怎样编辑、编译、连接和运行 C 程序。Turbo C++ 3.0 是 Borland 公司为 C++程序的编辑、编译、连接和运行而研制的集成环境。由于 C++是从 C 语言发展而来的，C++对于 C 程序是兼容的，因此可以用 C++的编译系统对 C 程序进行编译，或者说一个 C 程序可以在 C++集成环境中进行调试和运行。

为了能使用 Turbo C++ 3.0，必须先将 Turbo C++ 3.0 编译程序装入磁盘的某一目录下，

例如放在 C 盘根目录下一级 TC3.0 子目录下。上机运行 C 程序有以下六个步骤。

1．进入 Turbo C++ 3.0 集成环境

可以通过两种方法进入 Turbo C++ 3.0 集成环境。

① 在 DOS 环境下。如果用户的当前目录是 Turbo C++ 3.0 编译程序所在的子目录（例如 C：\TC3.0），可以在 DOS 环境下用键盘输入 DOS 命令“tc”：

C：\TC3.0>tc↙

这时就执行 C：\TC3.0 子目录中的 tc.exe 文件，屏幕上出现 Turbo C++ 3.0 集成环境，如图 1-3 所示。

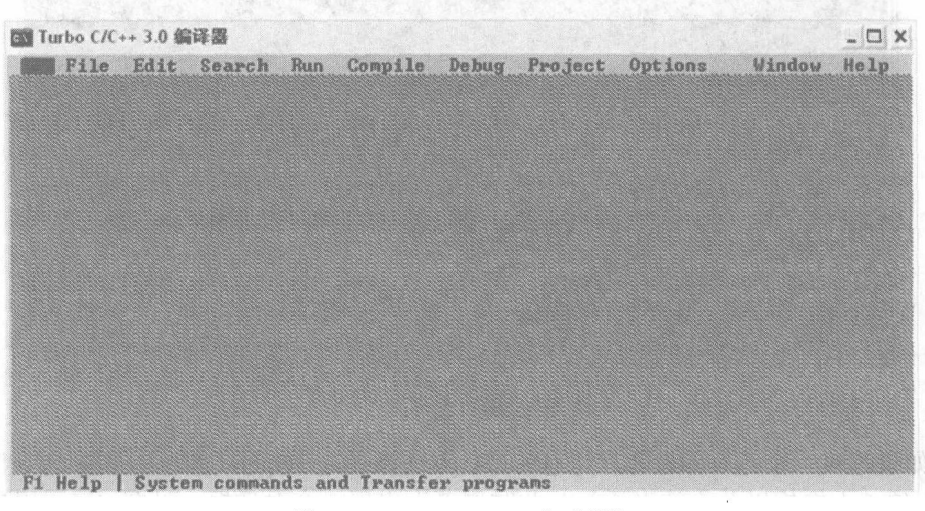

图 1-3　Turbo C++ 3.0 集成环境

② 在 Windows 环境下。先通过浏览找到 Turbo C++ 3.0 集成环境所在的子目录（如 C:\TC3.0），从中找到可执行文件 tc.exe，创建其快捷方式，并拖拽到 Windows 桌面上，用一个图标表示。双击该图标，就可以打开如图 1-4 所示的 Turbo C++ 3.0 集成环境。

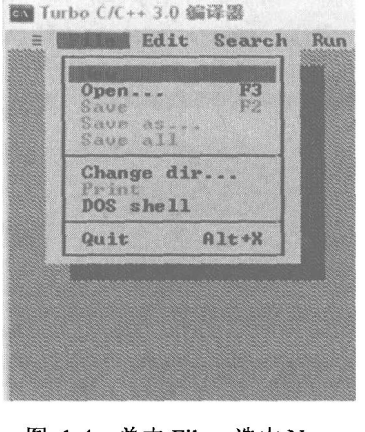

图 1-4　单击 File，选中 New

从图 1-3 可以看到：在集成环境的上部有一行“主菜单”，其中包括 10 个菜单项：File、Edit、Search、Run、Compile、Debug、Project、Options、Window 和 Help。用户可以通过以上菜单来选择使用集成环境所提供的 Turbo C++ 3.0 的各项主要功能。以上 10 个菜单项分别代表文件操作、编辑、寻找、运行、编译、调试、项目文件、选项、窗口和帮助。用鼠标可

以选择菜单条中所需要的菜单项，单击此菜单项就会出现一个下拉菜单。

2．编辑源文件

① 如果新建立一个源程序，可以用鼠标单击 File 菜单，然后在其下拉菜单选择 New（见图 1-4），表示要建立一个新的 C 源程序。屏幕上出现如图 1-5 所示的界面，上部是编辑窗口，供用户输入源程序。

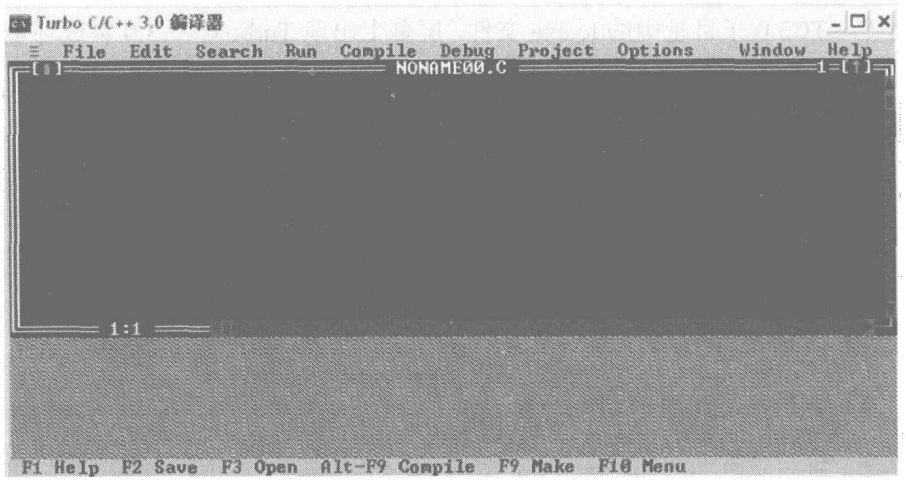

图 1-5 编辑源程序窗口

② 如果想对一个已有的源程序进行修改，应选择 File→Open（即单击 File 的下拉菜单中的 Open 项，下面均以此方法表示），此时屏幕上出现一个对话框，见图 1-6。用户可以在 Name 下面输入指定的文件路径和文件名（也可以只输入指定的文件路径并单击 Open 按钮，下面的文件列表中找到所需的文件名），然后单击 Open 按钮。系统会将此文件调入内存并显示在图 1-5 所示的编辑窗口中。此时集成环境自动设为编辑（Edit）状态。

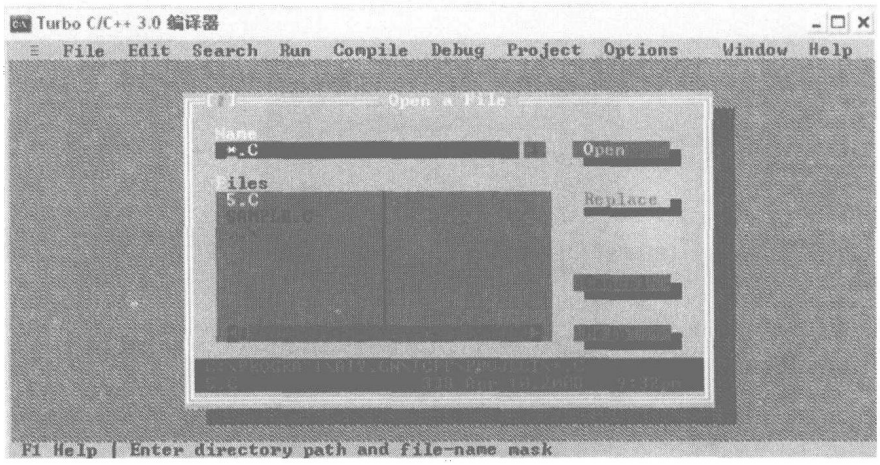

图 1-6 打开文件对话框

在编辑（Edit）状态下，光标表示当前进行编辑的位置，在此位置可以进行插入、删除或修改，直到满意为止。在完成编辑之后，应当保存源程序。如果该源程序是已有的，刚选择 File→Save 保存已修改过的源程序。如果该源程序是新输入的，刚选择 File→Save，并在弹出的 Save File As 对话框中的 Name 栏中输入文件路径和文件名，见图 1-7。

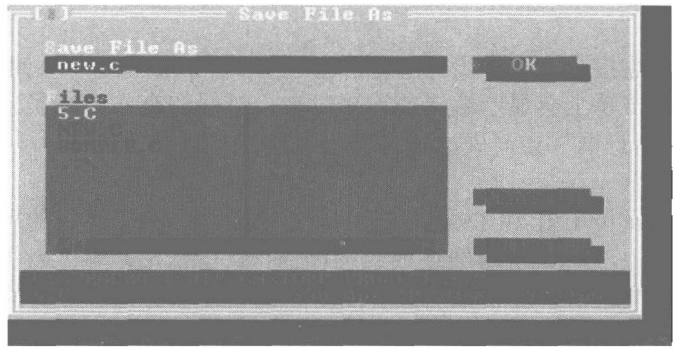

图 1-7 保存文件对话框

说明：C 程序的后缀应该是.c，如 c1.c、c2.c 等。由于现在是用 Turbo C++ 3.0 集成环境，它把源程序默认作 C++程序。如果用户在保存源程序时文件时文件名未加后缀，则系统会认为其是 C++程序，自动加上后缀.cpp，如 c1.cpp、c2.cpp 等。cpp 是“C plus plus”的缩写，意为 C++。如果在输入源程序后在保存时加上后缀 .c，在编译时系统能识别并编译以 .c 为后缀的 C 程序。也就是说，Turbo C++ 3.0 能编译以 .cpp为后缀的 C++源程序，也能编译以 .c 为后缀的 C 源程序（按照 Tubro C++的语法规定进行编译）。如果用 Visual C++ 6.0，也按同样方法处理。如果用 Turbo 2.0，则系统自动加上的后缀是 .c。

3. 对源程序进行编译

选择 Compile 菜单，并在其下拉菜单中选择 Compile（也可以按 Alt 和 F9）对源程序进行编译。屏幕上显示编译消息框，如图 1-8 所示。从图 1-8 可以看到：编译 c1.cpp 源程序，出现 1 个错误（error），0 个警告（warning）。在按任意键后，消息框消失，在集成环境的下部消息（Message）框中具体地指出在哪一行发生错误以及错误的原因，光标停留在与错误有关的行上，提醒用户改正错误。在修改程序后再进行编译，直到不出现错误和警告为止。此时得到一个后缀为 .obj 的目标程序。

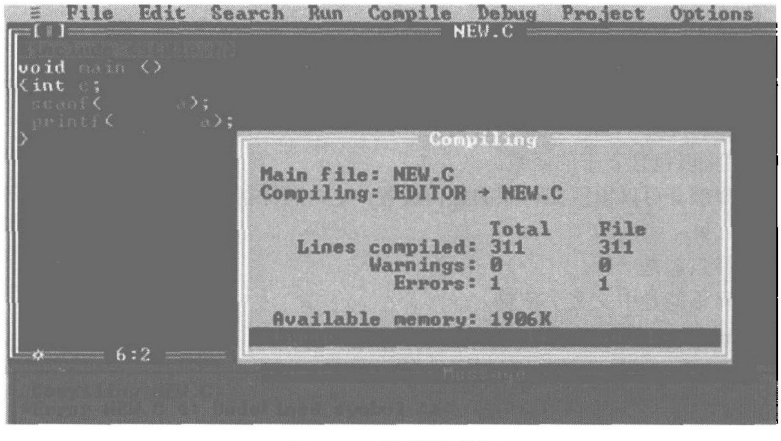

图 1-8 编译消息框

4. 将目标程序进行连接

假如一个程序包含多个文件，在分别对每个源程序进行编译并得到多个目标程序后，要把这些目标程序连接起来，同时还要和系统提供的资源（如函数库）连接成为一个整体。方法是选择菜单 Compile→Link，如果不出现错误，会得到一个后缀为 .exe 的可执行文件。应当说明的是：如果一个程序只包含一个文件，也必须进行连接，因为还要与系统提供的资源