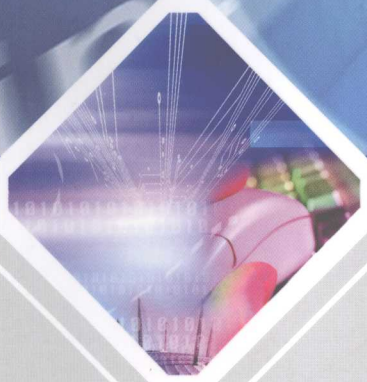

C语言程序设计（等级考试版）

■ 高等院校教材

CYu Yan CHENGXUSHEJI

李勇智 杨静宇 主 编
王士同 赵明生 郭沛仪 副主编
周 宇 李 广 主 审

清华大学出版社

C语言程序设计(等级考试版)

李勇智　杨静宇　主　编
王士同　赵明生　郭沛仪　副主编
周　宇　李　广　主　审

清华大学出版社
北　京

内 容 简 介

本书是参照教育部《非计算机专业计算机基础课程教学基本要求》和教育部考试中心《全国计算机等级考试大纲(2008 年版)》在计算机 C 语言程序设计方面的基本要求进行编写的。本书作为 C 语言程序的入门与应用教材，共分为 12 章和 5 个附录，主要内容包括程序设计基本概念、C 程序设计初步知识、顺序结构、选择结构、循环结构、字符型数据、函数、指针、数组、字符串、用户标识符的作用域、存储类、编译预处理、动态存储分配、结构体、共用体、用户自定义类型、位运算、文件、上机指导等。附录中给出了全国计算机等级考试最新大纲、两套全国计算机等级考试笔试试题及参考答案，以及各章习题参考答案。

本书以“基础理论—实用技术—实训”为主线组织编写，同时兼顾等级考试的需要，书中贯穿了大量考试真题作为示例进行分析。本书易教易学、学以致用、注重能力培养，对易混淆和实用性强的内容进行了重点提示和讲解。本书可作为非计算机专业大学本科(或重点高等职业院校专科)计算机程序设计基础教材，也可作为成人教育应用型专业以及等级考试培训教材。

本书配有电子教案，以方便教学。

图书在版编目(CIP)数据

C 语言程序设计(等级考试版)/李勇智，杨静宇主编；王士同，赵明生，郭沛仪副主编；周宇，李广主审. —北京：清华大学出版社，2008.3
(高等学校应用型特色规划教材)
ISBN 978-7-302- 17039-6

Ⅰ. C…　Ⅱ. ①李…　②杨…　③王…　④赵…　⑤周…　⑥李…　⑦郭…　Ⅲ.C 语言—程序设计—水平考试—自学参考资料　Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2008)第 018169 号

责任编辑：章忆文　宋延清
封面设计：杨玉兰
版式设计：北京东方人华科技有限公司
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c—service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京密云胶印厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：27.5　**字　数**：644 千字
版　　次：2008 年 3 月第 1 版　　**印　次**：2008 年 3 月第1 次印刷
印　　数：1～4000
定　　价：38.00 元

前　言

C 语言是高等院校以及高等职业学校开设范围最广的一门程序设计语言课程，同时也是教育部考试中心指定的二级考试科目之一。全国计算机等级考试是全国范围内参考人数最多的计算机类水平考试，考生群体大。学生学完 C 语言课程后，往往要求参加全国计算机等级考试二级 C 语言科目的考试，很多学校把通过计算机等级考试作为评价学生学士学位合格的条件之一。因此，编写一本既适合高校教学需要又能方便学生复习迎考全国计算机等级考试的 C 语言程序设计教材是一件非常有意义的工作。本书以“基础理论—实用技术—实训”为主线组织编写，同时兼顾等级考试的需要，是一本标准的应用与应试型教材。

为了增强本书的实用性，帮助读者学以致用，更快地掌握程序设计与应用系统的开发方法，本书在每一章末尾均安排典型例题解析和上机实训，这些案例不仅有助于读者对该章内容进行综合运用，实现融会贯通，也有助于培养读者的实际应用能力。

本书以循序渐进的方式，系统、全面地介绍 C 语言的语法结构。全书共分 12 章。

第 1 章主要介绍 C 语言开发环境的安装和使用，并介绍 C 语言的组成和运行。

第 2 章主要介绍 C 语言的数据类型、运算符、表达式和数据间的类型转换。

第 3 章主要介绍 C 语言的语句类型和数据的输入输出以及顺序程序设计。

第 4 章主要介绍用于选择程序设计的关系运算符和关系表达式、逻辑运算符和逻辑表达式、选择结构语句，并对选择结构程序设计进行举例。

第 5 章主要介绍循环结构程序设计，其中包括 while 循环、do-while 循环、for 循环和控制语句，并对循环结构程序设计进行举例。

第 6 章主要介绍一维数组、二维数组与多维数组、字符数组与字符串。并给出一些数组应用的例子。

第 7 章主要介绍函数的组成、调用和参数传递的方式。本章还介绍变量的作用域、存储类型以及函数的作用域。

第 8 章主要介绍 C 语言中的指针，其中包括指针的定义、指针与数组的关系，指针与函数的关系、指针与字符串的关系以及字符串指针，最后还介绍指向指针的指针，并给出一些实例。

第 9 章主要介绍编译预处理，其中包括文件包含、宏定义、条件编译。

第 10 章主要介绍结构体、链表、共用体、枚举类型和类型定义 typedef 等复杂的数据类型。

第 11 章主要介绍文件的相关操作，其中包括文件的打开与关闭、文件的读写、文件的定义与出错检测。

第 12 章主要介绍位运算符的含义及其运算功能。

本书注重理论联系实际，由浅入深、概念明确、条理清晰，适合作为非计算机类大学本科计算机程序设计基础教材，也可作为高职高专计算机专业教材以及等级考试培训教材和自学参考书。

本书由李勇智、杨静宇任主编，王士同、赵明生任副主编，周宇、李广任主审，全书框架结构由何光明拟定，由郭沛仪等具体编写。李勇智教授在全书的组织、统稿、内容审核等方面付出了大量心血，并参与编写了部分章节，另外对本书做出重要贡献的还有李亚非、姚昌顺、葛武滇、宋正虹、夏霖、严云洋、李胜、许勇、杨明、杨萍、赵传审、史国川、李海、周松、赵明、陈智等同志，在此一并表示谢意。

本书配有电子教案，以方便读者自学，请到 www.wenyuan. com.cn 下载。

限于作者水平，书中难免存在不当之处，恳请广大读者批评指正。如有批评和建议请发至：Book21Press@126.com。

本书编委会

目　录

第 1 章　C 语言概述

1.1　C 语言的发展及特点

C 语言是一种面向过程的通用程序设计语言。它以表达方式简明、使用灵活、结构化流程控制完善、拥有丰富的数据结构和操作集合、具备良好的程序可移植性和可获得较高效率的目标代码为特征。

C 语言不仅具有高级语言的要素，还兼有低级语言的功能，因此既可用于编写系统程序，也可用于编写不同领域的应用程序。本节主要介绍 C 语言的发展及特点。

1.1.1　C 语言的发展

C 语言的前身是 ALGOL 60(ALGOrithm Language)。

ALGOL 60 是 1960 年由国际计算机委员会设计的一种面向过程的结构化程序设计语言，用它编写的程序具有可读性和可移植性好的特点。但是，它不能直接对硬件进行操作，不宜用来编写系统程序。系统程序主要用汇编语言编写，而汇编语言是面向机器的程序语言，用它编写的程序可读性和可移植性都比较差。为此，人们开始考虑设计一种集高级语言和低级语言功能于一身的语言，以便用它来编写可读性和可移植性都比较好的系统程序。

1963 年，英国剑桥大学和伦敦大学将 ALGOL 60 发展成 CPL(Combined Programming Language)。该语言已比较接近于硬件，但规模较大，实用性不强。

1967 年，剑桥大学的 Martin Richards 将 CPL 改成 BCPL(Basic Combined Programming Language)。BCPL 与 CPL 相比大为简化，既具有结构化程序设计语言的特点，也能直接处理与硬件相关的数据，被软件人员用作系统程序的描述语言。

1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 修改为 B 语言，并用 B 语言开发了第一个高级语言的 UNIX 操作系统，在 DEC 公司的 PDP-7 小型机上运行。

1972 年，Dennis M.Ritchie 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.Ritchie 又合作将 1969 年用汇编语言编写的 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。到了 20 世纪 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传遍了贝尔实验室，接着也传遍了几乎所有的美国大学校园。随着西欧和日本相继宣布加入 UNIX 和 C 的行列，UNIX 和 C 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.Ritchie 合著了影响深远的名著 *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 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，Kernighan 和 Ritchie 修改了经典著作 *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。1990 年 ISO 通过了 C 程序设计语言的国际标准，称之为标准 C。此后陆续出现的各种 C 语言版本，如 Microsoft C 5.0/9.0、Turbo C 2.0/3.0、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函数库中收纳的函数种类、格式和功能上，尤其是图形函数库的差异更大一些。

在 C 语言的基础上，1983 年贝尔实验室又推出了 C++语言。C++语言进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C 语言是 C++的基础，C++语言与 C 语言在很多方面是兼容的。因此，掌握了 C 语言，再进一步学习 C++语言时，会感到更加容易和便利，并能够达到事半功倍的效果。

1.1.2 C 语言的特点

C 语言是近年来较流行的高级程序设计语言之一，许多大型软件均是用 C 语言编写的，如前面介绍过的 UNIX 操作系统。C 语言同时具有汇编语言和高级语言的双重特性。具体来说，C 语言具有以下特点。

(1) 简洁紧凑，灵活方便

C 语言一共只有 32 个关键字，9 种控制语句，程序书写自由，主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

(2) 运算符丰富

C 有 34 个运算符，该语言把括号、赋值、强制类型转换等都作为运算符处理，从而使运算类型极其丰富，表达式类型十分多样化，通过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。

(3) 数据结构丰富

C 的数据类型有整型、实型、字符型、数组型、指针型、结构体型、共用体型等，能用来实现各种复杂的数据类型运算。指针概念的引入，使程序效率更高。

(4) 是结构式语言

结构式语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用；C 语言具有多种循环语句、条件语句，可用来控制程序的流向。借助于函数和流程控制，可以使程序完全结构化。

(5) 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严，能够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。

(6) 允许直接访问物理地址，可以直接对硬件操作

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，能够像汇编语言一样对位、字节和地址进行操作，可以用来编写系统软件。

(7) 程序生成代码质量高，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%。

(8) 适用范围大，可移植性好

C 语言突出的优点就是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

1.2 简单的 C 语言程序

我们先来看几个 C 语言的简单程序。

例 1.1 输出“Simple C program”：

```
#include <stdio.h>      /*头文件*/
main()
{
   printf("Simple C program\n");
}
```

本程序的运行结果为：

```
Simple C program
```

程序第一行的#include <stdio.h>是头文件包含，stdio.h 是 C 语言的标准输入输出库的头文件，因为在程序中用到了 printf 输出函数，因此需要采用#include 语句来包含标准输入输出库，否则程序无法运行。/* ... */之间的内容是注释，通过注释可以让我们更好地了解程序的功能。程序的后几行是一个 main 函数，在这个函数内只有一条输出语句：

```
printf("Simple C program\n");
```

该语句的功能是在屏幕上显示“Simple C program”，并且换行。“\n”是转义字符，意思是回车换行。

需要注意的是，main 函数是 C 语言程序的入口点，每个 C 程序都有且仅有一个 main 函数。C 语言中的函数体由一对花括号包围起来。

例 1.2 求两数之和：

```
#include <stdio.h>
main()
{
   int a, b, sum;              /*定义变量*/
   a = 123; b = 456;           /*赋值语句*/
   sum = a + b;                /*将 a，b 的和赋值给 sum*/
   printf("sum = %d\n", sum);
}
```

本程序的运行结果是：

```
sum = 579
```

在main函数体中，第一行是定义变量的语句，将a、b、sum定义为整型(int)，第二行是两个赋值语句，这两个语句将123赋值给a，将456赋值给b，第三行将a与b的和赋值给sum，即将123与456相加赋值给sum。最后一个语句是将sum的值输出。其中的“%d”是输入输出的“格式字符串”，用来指定输入输出时的数据类型和格式，“%d”表示十进制整数类型。printf函数中括弧内最右端sum是要输出的变量。

例1.3 求两数之中最大的一个数：

```
#include <stdio.h>
int max(int x, int y)          /*定义max函数，求出x与y之间的最大的一个*/
{
    int z;                     /*定义局部变量z，将x、y中最大的一值存于其中*/
    if(x > y)
        z = x;
    else
        z = y;
    return z;                  /*将z值返回*/
}

main()                         /*主函数*/
{
    int a, b, c;
    scanf("%d, %d", &a, &b);
    c = max(a, b);             /*调用max函数，将得到的值赋给c*/
    printf("max = %d\n", c);
}
```

本程序包含两个函数，一个是max的函数，它有两个整型参数x和y。这个函数中包含了if ... else等语句。if ... else是一个选择语句，这个语句起到了判断功能，当if后的条件满足x>y时，执行z=x，否则执行z=y。整个函数的作用是将x和y中较大的值赋给变量z，并通过return语句将z的值带回到主函数main中的调用处，赋给变量c。

以main开头的是主函数，在花括号中的函数体中先定义变量a、b和c，再调用scanf函数将从键盘输入的两个数赋给变量a和b。语句“c=max(a, b);”的作用是先调用函数max，并将a和b的值传递给max函数的变量x和y，调用结束时，通过max函数的变量z将返回值赋给变量c，最后通过printf语句输出c的值。

当程序运行时，输入20和36两个数并按Enter键后，输出结果是：

```
max = 36
```

通过上面三个例子的学习，我们对C语言程序的结构、格式有了一定的了解，下面来总结一下C语言的构成及格式。

1.3　C 语言程序的构成及格式

C 程序的基本结构是函数，一个或多个 C 函数组成一个 C 程序，并且每一个 C 程序由若干条语句组成。所有的 C 语句都必须以分号“;”结束。例如，在例 1.2 中：

```
a = 123;
b = 456;
```

注意： ① 所有的 C 语句都必须以分号“;”结束。单独的一个分号而没有前面的语句体，称为空语句，在 C 语言中是合法的语句。

② 语句中大写字母和小写字母代表不同的含义。例如变量 A 和变量 a 代表不同的变量。

③ 为了增加程序的可读性，应该避免在一行中连续书写多条语句。

④ 提倡按照程序的逻辑结构使用缩进的书写形式，以明确地表示程序的层次性和逻辑性。

C 语言中所使用的函数可以分为两类：一类是系统定义的函数，如 printf 和 scanf 函数等，称为标准库函数，可以直接在程序中使用；另一类是用户自己定义的函数，如 main 和 max 函数等，对于这些函数我们需要自己编写代码。

根据例 1.1~1.3，我们可以得出函数的一般格式如下：

```
函数类型 函数名(参数类型 参数名)   /*自定义函数*/
{
    函数体;
}
main()
{
    变量定义部分;
    语句执行部分;          /*语句执行部分包括调用自己定义的函数*/
}
```

在编写 C 程序时，我们需要注意下面的几点。

(1) 在整个程序文件中，函数可以出现在任意位置。主函数不一定出现在程序的开始处，但不管主函数位于程序中的何处，程序运行时总是从主函数开始。

(2) 每个程序行中的语句数量是任意的，既允许一行内写几条语句，也允许一条语句分几行书写，但每条语句都必须以分号结束。有时也可以在程序中的适当位置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或结尾处。如果注释占有几行，则每一行都要以“/*”开头，以“*/”结尾，“*”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。

1.4 C 语言程序上机调试步骤和方法

目前，在电脑上广泛使用的 C 语言开发环境有 Microsoft C、Turbo C、Borland C、Visual C++(简称 VC++)等，它们的开发环境基本相同，只有一些细微差异，但这影响并不是很大。本书的上机环境采用 VC++ 6.0 集成开发环境，它的启动界面如图 1.1 所示。

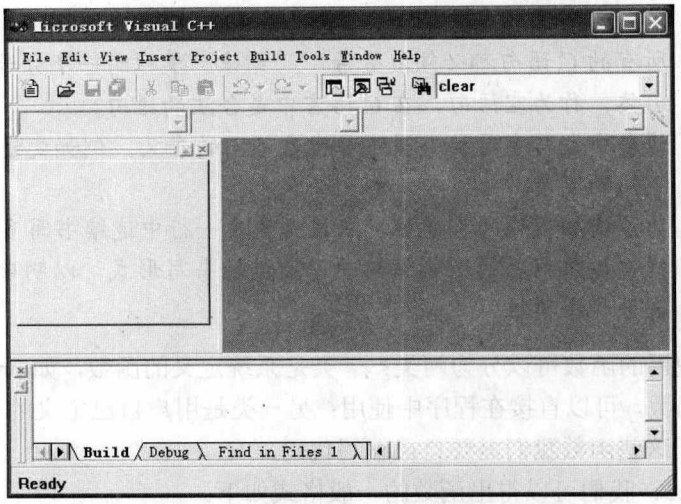

图 1.1 VC++ 6.0 的启动界面

1.4.1 新建 C 程序

如果想要编写 C 语言程序，可从菜单栏中选择“File”→“New”命令，出现如图 1.2 所示的新建文件界面。

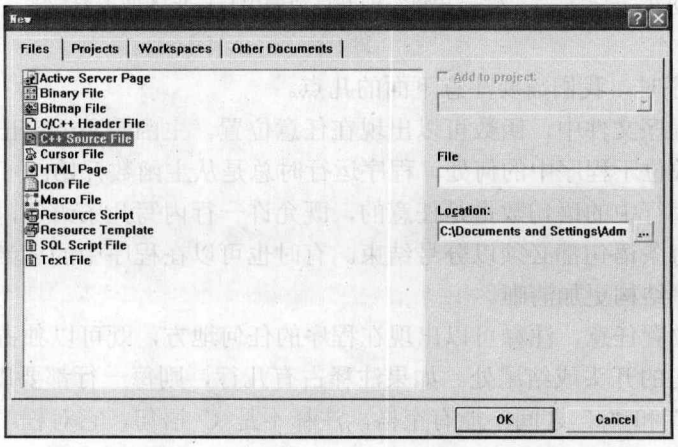

图 1.2 VC++ 6.0 的新建文件界面

在 Files 选项卡左侧的列表框中选择 C++ Source File 选项，并在对话框右边的 File 文本框中输入文件名(例如“hello”)，加上.c 扩展名(一定不能少)，然后在 Location 文本框中选择存储路径(单击[...]按钮)，如图 1.2 所示，最后单击 OK 按钮。

这时我们又回到了启动界面，如图 1.3 所示，好像 VC++并没有做些什么事，其实不然，大家可以仔细地观察一下，启动界面中的右侧区域变成了白色，可以在这片白色区域内编写代码。

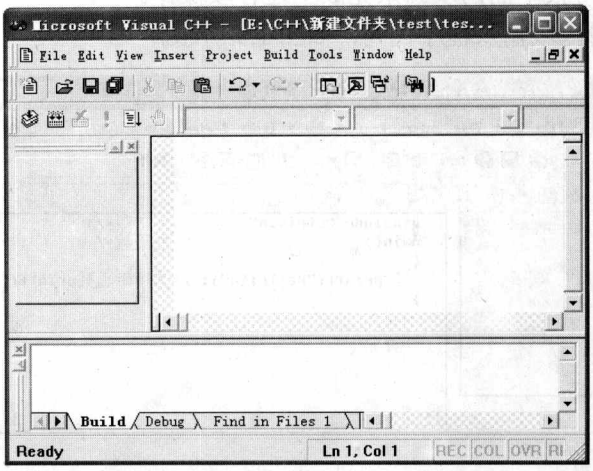

图 1.3　VC++ 6.0 的编辑界面

1.4.2　运行 C 程序

C 语言是一种编译型的程序设计语言。一个 C 程序要经过编辑、编译、连接和运行 4 个步骤，才能得到运行结果。

1. 编辑

编辑是用户在编辑界面中书写和修改 C 语言源程序的过程。

2. 编译

编译是把 C 源程序翻译成可重定位的二进制目标程序的过程。编译过程由编译程序来完成，编译程序自动对源程序进行语法检查。

当发现错误时，就将错误类型以及错误在程序中的位置显示出来，以帮助用户修改源程序中的错误。如果未发现错误，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。目标文件的名称与源程序名称的前一部分相同，但扩展名为“.obj”。

3. 连接

连接也称链接或装配，是用连接程序将编译过的目标程序与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。它是一个与源文件名称前一部分相同、但扩展名为“.exe”的可执行文件。

4. 运行

运行是将可执行的目标文件投入运行，以获取程序的运行结果。在操作系统平台中，可以直接执行扩展名为“.exe”的文件。如果执行后没有得到预定的结果，说明程序中还存在逻辑错误或算法错误，此时必须重复前面的步骤，对源程序进行修改，重新编译连接，直到得出正确的运行结果。

在 VC++ 6.0 中可按下面的步骤来完成上述各项操作。

(1) 在代码编辑区域输入程序代码，如图 1.4 所示。

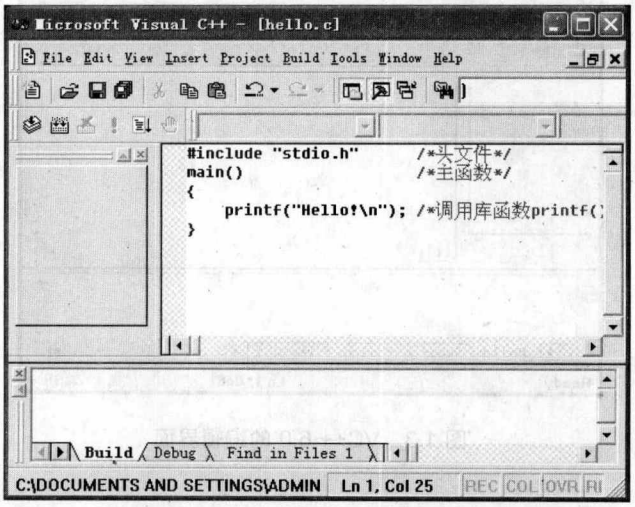

图 1.4 编写 C 语言程序

(2) 输入完代码后，单击工具栏上的按钮或者从菜单栏中选择“Build”→“Compile”命令进行编译。这时，出现如图 1.5 左图所示的对话框，单击【是】按钮。这时图 1.4 的下方窗口中会出现编译的整个过程。如图 1.5 右图所示。

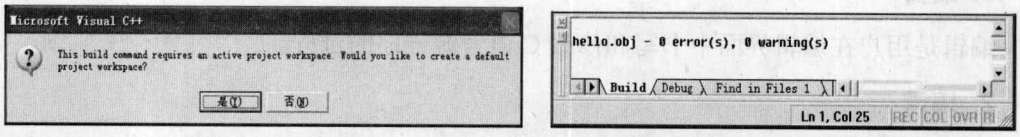

图 1.5 编译过程

从图 1.5 中可以看到，编译过程中生成了一个 hello.obj 的文件，它有 0 个错误(error)，以及 0 个警告(warning)。

注意： error 代表程序中含有的语法错误，error 必须 0，程序才能运行起来。
warning 代表程序中含有警告，存在警告时并不影响程序的运行。

(3) 编译完成以后，单击工具栏上的按钮或者从菜单栏中选择“Build”→“Rebuild all”命令，进行连接，经过连接后，就从源程序生成了可执行文件，它是计算机能够直接运行的文件，其文件扩展名为.exe。如图 1.6 所示。