21世纪高等学校规划教材
21 Shiji Gaodeng Xuexiao Guihua Jiaocai

C语言程序设计

陈宝明　骆红波　许巨定　主　编
张培君　孟　园　张亚琴　副主编

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校规划教材

C语言程序设计

陈宝明　骆红波　许巨定　主　编
张培君　孟　园　张亚琴　副主编

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计 / 陈宝明，骆红波，许巨定主编. —北京：人民邮电出版社，2009.2
21世纪高等学校规划教材
ISBN 978-7-115-19019-2

Ⅰ. C… Ⅱ. ①陈…②骆…③许… 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（2008）第197846号

内 容 提 要

C 语言作为一种结构化程序设计语言，在当今软件开发领域中有着十分广泛的应用，也是高等学校计算机语言类课程的首选语言。本书共分 11 章，主要内容包括 C 语言的基本语法和概念、数据类型及应用技巧、C 语言程序结构、数组、函数、指针、文件等，并系统阐述了各种程序设计的方法。全书案例丰富，阐述清晰，层次分明，讲述力求理论联系实际、循序渐进，注重培养读者分析问题和程序设计的能力，注重培养良好的程序设计风格和习惯。

本书可作为高等学校理工类专业 C 语言程序设计课程的教学用书，也可作为学习 C 语言的入门教材，还可作为计算机二级考试的辅导教材。为配合教学，本书配有 PPT 教学课件，并有配套的《C 语言程序设计实验指导》，供读者参考。

21 世纪高等学校规划教材

C 语言程序设计

◆ 主　　编　陈宝明　骆红波　许巨定
副 主 编　张培君　孟　园　张亚琴
责任编辑　蒋　亮

◆ 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
北京铭成印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：17.5
字数：459 千字　　2009 年 2 月第 1 版
印数：1－3 000 册　　2009 年 2 月北京第 1 次印刷

ISBN 978-7-115-19019-2/TP

定价：30.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

C 语言以其数据类型丰富、使用灵活、可移植性好、表达能力强、功能强大等优势，迅速在程序设计领域普及。它既具有高级语言的功能，又具有低级语言的特点，既可以用来编写系统软件，又可用于编写应用软件，还广泛应用于嵌入式系统的开发，是国内外软件开发领域广泛使用的一种计算机语言，也是高等学校计算机语言类课程的首选语言。

全书以程序设计为主线，以算法思维方法的训练为切入点，以编程应用为驱动，通过案例和问题引入内容，重点讲解程序设计的思想和方法。主要内容包括 C 语言的基本语法和概念、数据类型及应用技巧、C 语言程序结构、数组、函数、指针、文件等，并系统阐述了各种程序设计的方法。

本书共分 11 章。第 1 章介绍 C 语言的基本概念及 C 语言的开发及运行环境。第 2 章介绍 C 语言的各种数据类型、运算符及表达式、数据运算规则、基本输入/输出函数。第 3 章通过大量的例题分别讲解程序设计的 3 种基本结构，即顺序结构、分支结构、循环结构，以及程序设计的基本方法和技巧。第 4 章系统阐述 C 语言数组的概念及使用方法。第 5 章介绍函数的概念及用函数进行模块化程序设计的基本方法，并描述了各程序模块间数据传递的方法。第 6 章介绍指针访问数据的基本方法，并详细描述了指针在数组中的应用。第 7 章介绍宏的概念及编译与预处理的方法。第 8 章介绍结构体和共用体的概念、结构体变量、结构体数组、结构体指针、结构体函数，并描述了链表的动态数据结构。第 9 章介绍位运算及位操作应用与设计技巧。第 10 章介绍 C 语言程序文件的使用及文件中数据处理的基本方法。第 11 章以 C 语言综合程序设计应用实例，详细阐述用 C 语言开发应用软件的基本步骤和方法。

本书在讲解过程中，既注重语法结构的介绍，更注重算法的分析和训练，同时兼顾计算机二级考试的要求，所有案例力求做到与计算机二级考试题型相结合，为读者参加计算机二级考试提供方便。

本书由陈宝明、骆红波、许巨定任主编，张培君、孟园、张亚琴任副主编。第 1 章由孟园编写，第 2 章由许巨定编写，第 3 章由陈宝明编写，第 4 章由张丽华编写，第 5 章由潘云燕编写，第 6 章由骆红波编写，第 7 章由段亚伟编写，第 8 章由梁田编写，第 9 章由刘锦萍编写，第 10 章由贾小军编写，第 11 章由张培君编写，附录由张亚琴编写。

本书是嘉兴学院计算机基础部多年从事 C 语言程序设计教学及程序设计实践的教师们教学经验的结晶和共同努力的结果。在此向他们表示衷心的敬意和感谢。

由于时间仓促和作者水平有限，书中难免存在不足或者遗漏之处，恳请广大读者提出修改建议。

编　者

2008 年 10 月

目 录

第 1 章 C 语言概述

计算机程序通俗地讲，就是人们在所设计的、能控制计算机正确解决给定问题的算法的基础上，进一步用某种选定的计算机语言，把算法翻译成计算机可接受、读懂和执行的“计算机语言文章”。计算机语言已经历了 4 代（即机器语言、汇编语言、高级语言、非过程语言）的发展历程，现在正向第 5 代（智能语言）积极迈进。

C 语言正是在汇编语言向高级语言过渡过程中出现的一种语言。本章将介绍 C 语言的发展和特点，并从简单的 C 语言程序入手，介绍其运行步骤和开发环境。

1.1 C 语言简介

C 语言是国际上广泛流行的计算机高级语言，从它诞生之日起就迅速发展成为最受欢迎的语言之一，其主要原因是它具有强大的功能，既可以用来编写系统软件，也可以用来编写应用软件。许多著名的系统软件，如 dBASE Ⅲ PLUS、dBASE Ⅳ 都是由 C 语言编写的。C 语言的发展与操作系统的发展密不可分。

1.1.1 C 语言的发展及其标准

从 20 世纪 60 年代开始，贝尔实验室的 Ken Thompson 就开始着手开发 UNIX 操作系统。1970 年，Ken Thompson 以当时专门用来描述操作系统的语言 BCPL 为基础，设计出了更简洁、更接近硬件的 B 语言，并编写了第一个 UNIX 操作系统。此时的 B 语言比较简单，功能有限。随后几年，Ken Thompson 的同事 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 第 2 个字母），它既保持了 BCPL 和 B 语言精练、接近硬件等优点，又克服了它们过于简单、功能有限等缺点。1973 年，Ken Thompson 和 D.M.Ritchie 合作把 UNIX 的 90%以上的程序用 C 语言成功进行改写。此后，C 语言又进行了多次改进，直到 1977 年实现了不依赖于具体机器的 C 语言编译器，使得 C 语言可以很容易地移植到其他机器，这也使得 UNIX 操作系统迅速在各种机器上实现。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广并很快风靡世界，成为世界上应用最广泛的几种计算机语言之一。随着时间的推移，又有多种程序设计语言在 C 语言的基础上产生，如 C++、Visual C++、Java、C#等。

随着各种 C 语言版本的推出，为 C 语言制定标准成为必需。以 1978 年发布的 UNIX（第 7 版）中的 C 语言编译程序为基础出版的《The Programming Language》成为后来使用的 C 语言版本的基础，因而被称为标准的 C。1983 年，美国国家标准化协会（ANSI）为 C 语言制定了标准，称为 83 ANSI C。1987 年，ANSI 又公布了新标准——87 ANSI C。1990 年，国际标准化组织

（ISO）接受 87 ANSI C 为 ISO C 的标准。目前流行的 C 编译系统都以此为基础。

1.1.2 C 语言的特点

C 语言直到今天仍能存在和发展，和它兼有高级语言和低级语言的优点是分不开的。其主要特点如下。

（1）语法简洁、功能强大。C 语言的语法简洁，比其他高级语言简练、源程序短，且功能强大，对操作系统以及需要对硬件进行操作的系统而言，C 语言明显优越于其他高级语言。

（2）移植性强。用 C 语言编写的程序可移植性好，只需稍作修改即可在其他的计算机系统中运行。

（3）具有丰富的运算符和数据类型。C 语言提供了丰富的运算符和数据类型，表达式类型多样化，可以满足各种程序设计的需要。

（4）结构化的程序设计语言。C 语言具有结构化的控制语句，用函数作为程序的基本单位，是完全模块化和结构化的语言。

（5）允许直接访问物理地址。C 语言可以直接对硬件进行操作，兼有高级语言和低级语言的特点。

（6）程序设计自由。C 语言的语法限制不太严格，使用灵活、方便。

（7）生成目标代码质量高，程序执行效率高。

1.2 简单的 C 语言程序介绍

下面介绍几个简单的 C 语言程序的例子，从中可以了解 C 语言的基本构成。

【例 1.1】 输出字符串。

```
# include<stdio.h>
void main( )
{
     printf("Welcome to Beijing!\n");          /*输出一串字符*/
}
```

这是一个非常简单的 C 语言程序，功能是在屏幕上输出信息：

```
Welcome to Beijing!
```

程序说明如下。

① #include<stdio.h>为编译预处理命令中的文件包含命令，其作用是在编译之前把所需的有关 printf()函数的一些信息文件 stdio.h 包含进来，为输入和输出提供支持。在程序中用到系统提供的输入/输出函数时，应在程序的开头写下这一行。

② void main( ) 中 void 是空的意思，表示不返回任何类型的值，main 是函数的名称，每一个 C 语言都必须有一个 main 函数，也可写成 void main(void)，在有的编译系统中可直接写成 main()。

③ 一对花括号组成的部分称为函数体，函数体由语句组成，如 printf("Welcome to Beijing!\n"); 就是一条输出语句，表示在输出“Welcome to Beijing!”后回车换行，注意语句必须以分号结束。printf 函数是 C 编译系统提供的标准函数库中的输出函数，它的具体用法在后续章节中具体介绍。

④ /* …… */之间的内容为注释，用以说明和解释，不被编译，也不被执行。

【例 1.2】　判断一个非 0 整数是正数还是负数。

```
# include<stdio.h>
void main()
{
     int a;                                          /*定义变量*/
     printf("please input a : \n");                  /*屏幕上输出提示信息*/
     scanf("%d", &a);                                /*从键盘给变量 a 赋值*/
     if(a>0)printf("it is a positive number!\n");
     else
        printf("it is a negative number!\n ");       /*判断 a 的正负并将判断的结果输出*/
}
```

本程序的功能是从键盘输入一个非 0 的整数，将其赋给变量 a，然后进行判断。如果 a > 0，就在屏幕输出 a 大于 0 的结论，否则，输出 a 小于 0 的结论。注意，程序中的 scanf 函数的作用是从键盘输入值赋给变量，其中，“&a”前的“&”不能少，所输入的值存放到变量 a 中。scanf 函数是 C 编译系统提供的标准函数库中的输入函数，它的具体用法在后续章节中具体介绍。

【例 1.3】　通过调用子函数，求两个数中的较大者。

```
# include<stdio.h>
int Getmax(int x, int y);    /*对被调用的函数 Getmax 的声明*/
void main( )                 /*主函数*/
{
     int a, b, max;
     printf("please input a and b: \n");      /*屏幕上输出提示信息*/
     scanf("%d%d", &a,&b);                    /*从键盘给变量 a 和 b 赋值*/
     c=Getmax(a,b);                           /*调用子函数 Getmax*/
     printf("max is %d",c);                   /*输出较大值*/
}
int Getmax(int x, int y)                      /*子函数*/
{
     int z;
     if(x>y) z=x;                             /*比较两个数的大小并将大数赋给变量 z*/
     else
         z=y;
     return z;                                /*返回变量 z 的值*/
}
```

本程序由两个函数组成，主函数 main 和子函数 Getmax，主函数 main 负责数据的输入和输出，子函数 Getmax 的作用是将 x 和 y 中的较大者赋给变量 z，return 语句将 z 的值带回到函数调用的位置上并赋给变量 c，最后在 main 函数中输出较大数的值也即 c 的值。

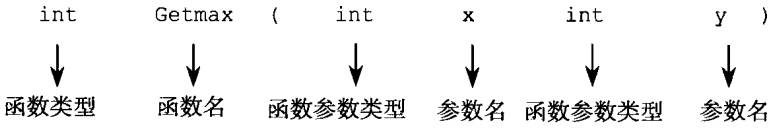
通过以上几个程序，我们可以了解 C 语言程序的基本构成如下：

① C 语言程序是由函数构成的。每一个 C 程序必须有且只能有一个主函数，即 main 函数，可以包含若干个其他的子函数。C 程序的执行是从 main 函数开始，到 main 函数中结束。

② main 函数既可以调用编译系统标准函数库中提供的库函数（如 printf 函数和 scanf 函数），也可以调用用户自己定义的子函数（如例 1.3 中的 Getmax 函数）。若调用库函数，则在程序开始前加上相应的文件包含命令，若调用自己定义的子函数，则加上对子函数的原型声明。

③ 从函数的角度来看，一个函数可以分为函数的首部和函数体两个部分。函数首部依次包括函数类型、函数名、函数的参数类型及参数名。

例如，例 1.3 中 Getmax 函数的首部是：



函数首部下大括号括起的部分称为函数体。函数体一般分为函数声明部分和执行部分，声明部分对函数中要用到的变量进行定义，如例 1.2 中的“int a;”，其余部分称为函数体的执行部分，完成具体的操作。

④ 函数体内不管是声明部分还是执行部分，都是由语句组成，C 语言规定语句必须以分号“;”结束。C 程序的语句书写格式比较自由，一行可以写多条语句。

⑤ 注意 C 语言程序是严格区分大小写的，如例 1.3 中的函数名“Getmax”若写成“getmax”则编译就会出现错误。变量名也同样区分大小写，读者在引用变量时要特别注意其定义时是大写还是小写。

1.3 C 语言程序的运行步骤和开发环境

任何程序设计都需要基于一定的开发步骤和特定的开发环境。本节在概述 C 程序运行步骤的基础上，主要介绍两种 C 语言程序的开发运行环境的简单用法：Visual C++6.0 和 Turbo C2.0。

1.3.1 C 语言程序的运行步骤

按照 C 语言语法规则而编写的 C 程序称为源程序。源程序由字母、数字及其他符号等构成，在计算机内部用相应的 ASCII 表示，并保存在扩展名为“.C”的文件中。源程序是无法直接被计算机运行的，因为计算机的 CPU 只能执行二进制的机器指令。这就需要把 ASCII 的源程序先翻译成机器指令，然后计算机的 CPU 才能运行翻译好的程序。

源程序翻译过程由两个步骤实现：编译与连接。首先对源程序进行编译处理，即把每一条语句用若干条机器指令来实现，以生成由机器指令组成的目标程序。但目标程序还不能马上交计算机直接运行，因为在源程序中，输入、输出以及常用函数运算并不是用户自己编写的，而是直接调用系统函数库中的库函数。因此，必须把“库函数”的处理过程连接到经编译生成的目标程序中，生成可执行程序，并经机器指令的地址重定位，便可由计算机运行，最终得到结果。

C 语言程序的调试和运行步骤如图 1-1 所示。

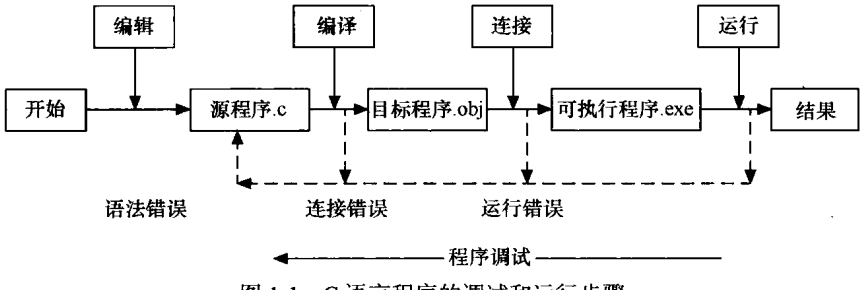


图 1-1 C 语言程序的调试和运行步骤

图中虚线表示当某一步骤出现错误时的修改路线。运行时，无论是出现编译错误、连接错误，还是运行结果不正确（源程序中有语法错误或逻辑错误），都需要修改源程序，并对它重新编译、

连接和运行，直至将程序调试正确为止。

1.3.2　Visual C++6.0 简介

Visual C++6.0 是 Microsoft 公司开发的面向 C++语言的开发环境，它不仅支持 C++语言，也支持 C 语言。由于 Visual C++的安装比较简单，读者只需将安装盘插入光驱，根据安装向导提示即可进行快速安装，在此不再详述。

1. Visual C++6.0 的主界面

系统中如果成功安装了 Visual C++，可以直接在桌面双击 Visual C++6.0 图标启动程序，也可以选择“开始”→“Microsoft Visual C++6.0”命令启动。Visual C++启动后，其主界面如图 1-2 所示。

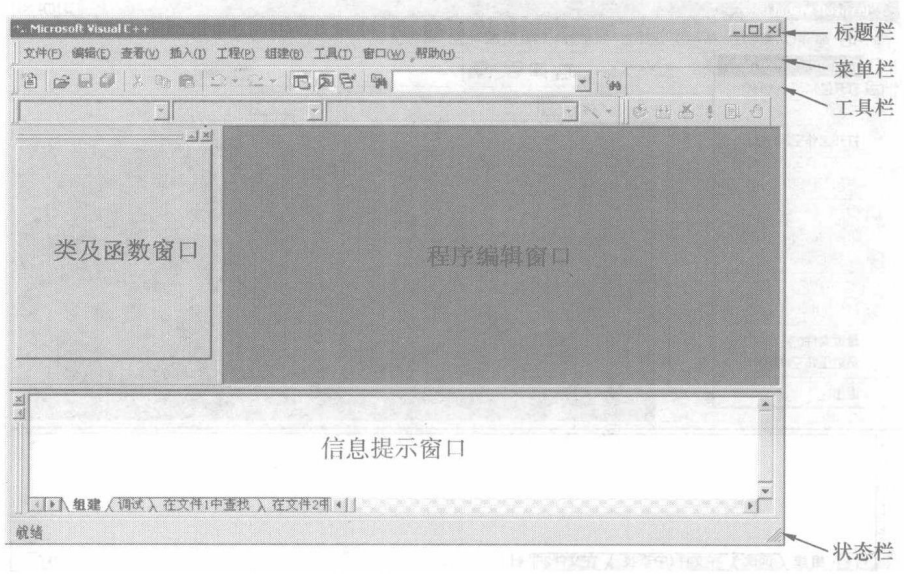


图 1-2　Visual C++主窗口

主窗口从上至下依次为：标题栏、菜单栏、工具栏、类及函数窗口、程序编辑窗口、信息提示窗口和状态栏。

2. Visual C++的菜单

如图 1-2 所示，Visual C++包含如下菜单。

- “文件”菜单：主要用来对文件、工程和工作区进行管理，包括创建新的文件、工程和工作区，关闭当前的文件、工程和工作区，打开已有的文件、工程、工作区等。
- “编辑”菜单：主要对文件进行各种编辑操作，如剪切、粘贴、插入、删除、复制、全部选择、查找、定位等。
- “查看”菜单：主要用于改变窗口和工具栏的显示方式。
- “插入”菜单：主要用于创建、添加项目和资源。
- “工程”菜单：主要用于设置与工程相关的源文件、库函数，如添加工程、设置活动空间、插入工程到工作区等。
- “组建”菜单：主要用于对源程序进行编译、连接和运行。
- “工具”菜单：主要用于选择或制定开发环境中的某些功能，以激活各个调试窗口，改变各个窗口的显示模式等。

- “窗口”菜单：主要用于对窗口进行各种设置。

3. Visual C++环境中 C 程序的运行步骤

C 程序的编辑、编译、连接、运行等开发和运行步骤，在 Visual C++开发环境中的具体操作如下：

（1）源程序的编辑

在 Visual C++主窗口中单击“文件”菜单，在其下拉菜单中单击“新建”命令，如图 1-3 所示，屏幕上出现一个新建对话框，选择“文件”选项卡，单击“C++ Source File”选项，在右边的“文件名”文本框中输入文件名称（如 1-1.c），注意系统会默认源文件为 C++程序，自动加上后缀“.cpp”，因此这里需将后缀改为“.c”，然后在“位置”文本框中选择或输入源文件将要存储的路径（如 D：\），如图 1-4 所示。

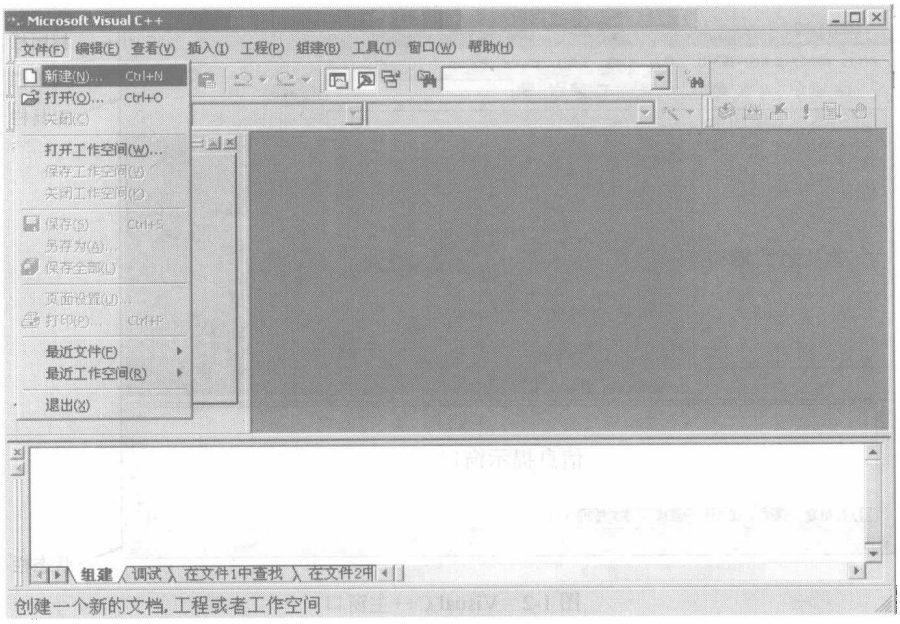


图 1-3　新建源文件

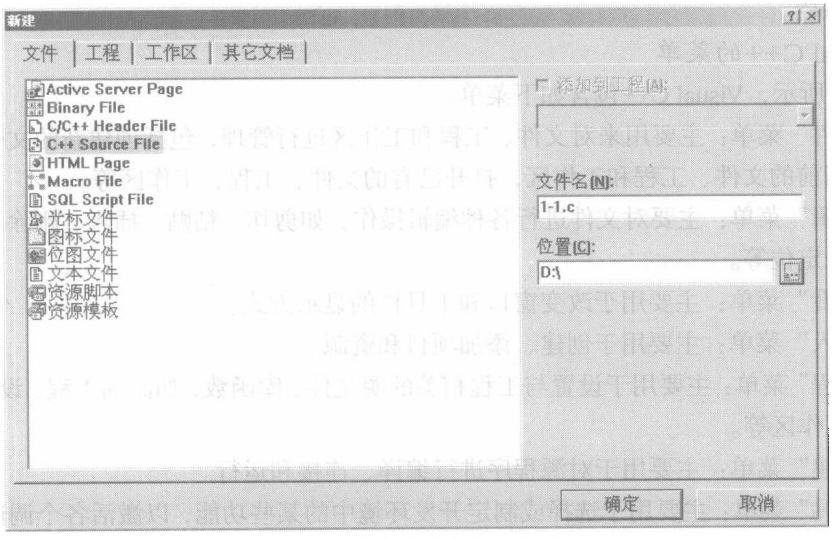


图 1-4　命名和选择保存路径

（2）源程序的编译

单击“组建”→“Compile 1-1.c”菜单命令，编译源程序。屏幕上会出现一个对话框，询问你是否同意建立一个默认的项目工作区，单击“是”按钮，开始进行编译。若编译有错，系统会在编译窗口显示错误信息，包括错误数量、错误位置、错误类型等。也可直接按 Ctrl+F7 组合键来完成编译，如图 1-5 所示。

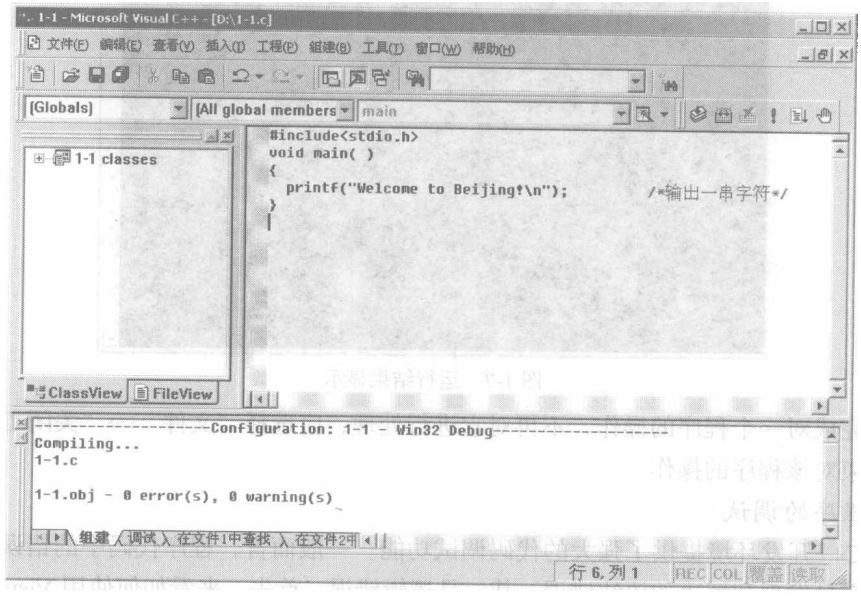


图 1-5　源程序编译后

（3）连接

编译得到目标程序 1-1.obj 后，下面要进行连接，以生成可执行文件。选择“组建”→“组建 1-1.exe”菜单命令，完成连接后，在信息提示窗口显示连接时的信息，此时生成了一个可执行文件 1-1.exe，如图 1-6 所示。也可直接按 F7 键一次完成编译和连接操作。

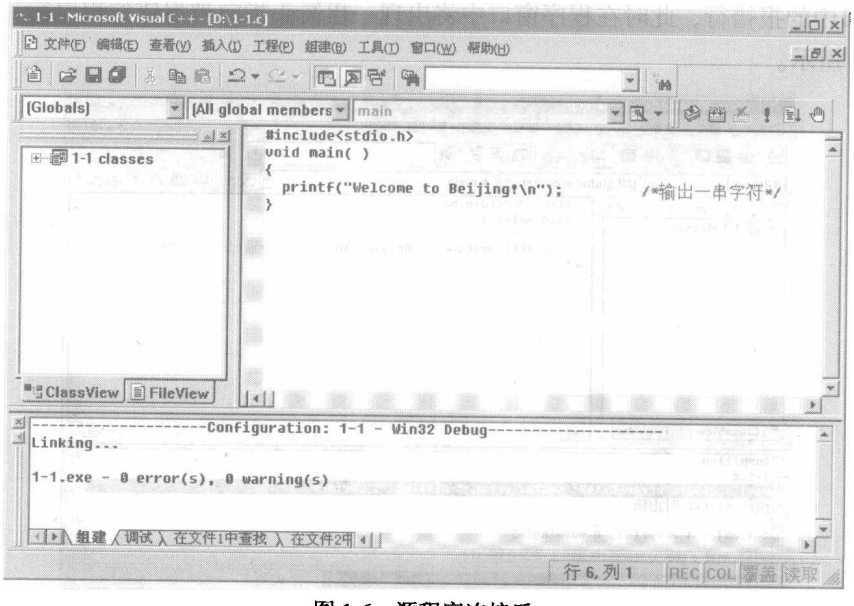


图 1-6　源程序连接后

（4）运行

选择“组建”→“执行 1-1.exe”菜单命令，可以看到一个黑色窗口输出程序中的运行结果，如图 1-7 所示。按任意键后回到 Visual C++的主窗口，继续进行操作。

图 1-7　运行结果显示

如果已完成对一个程序的操作，不再对它进行处理，应选择“文件”→“关闭工作区”菜单命令，以结束对该程序的操作。

4. 源程序的调试

Visual C++开发环境提供了强大的代码调试功能。一般而言，程序代码中的错误主要分为两种，其一是容易发现和解决的语法错误，其二是逻辑错误。首先，来看如何使用 Visual C++6.0 来解决第一类问题。

（1）使用 Visual C++的错误提示信息

语法错误是指程序员所输入的指令违反了 C 语言的语法规定，如下面的语句：

```
printf("Welcome to Beijing! \n")
```

显然，语句的末尾必须以分号结束。在编译时，Visual C++会在信息提示窗口提示错误信息，双击信息窗口中的报错行，此时在程序窗口中将出现一粗箭头指向被报错的程序行，提示改错位置，如图 1-8 所示。

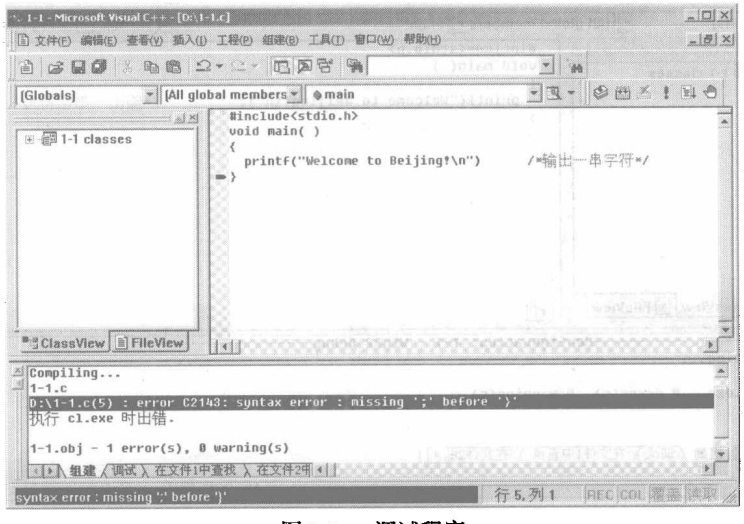


图 1-8　调试程序

由于 C 语言允许将一条语句写成几行，因此检查完第 4 行末尾无分号时还不能判定该语句有错，必须再检查下一行，直到发现第 5 行的“}”前没有分号，才判定出错。

除了上面介绍的这种明显的语法错误之外，还有一些初学者容易出现的语法错误，如变量未定义，双引号写成单引号等，都可以通过这种方式来解决。

（2）寻找逻辑错误

逻辑错误是指在语法上没有错误，但从程序的功能上来看，代码却没有正确完成其功能。同样，读者也可以在 Visual C++环境下来寻找逻辑错误。在调试模式下运行程序，Visual C++并非仅仅给出最后的结果，还保留了程序中所有的中间结果，即 Visual C++知道代码每一行都发生了什么。程序员可以通过跟踪这些中间结果，来发现错误到底藏在哪里。为了便于介绍，以下给出一个含有逻辑错误的例子。

【例 1.4】 在屏幕上输出 10 次字符串“加油!”。

```
#include <stdio.h>
void main( )
{
  int i ;          /*定义循环变量*/
  for(i=0;i<=10;i++)          /*i 循环 10 次，使 printf 语句中的字符串输出 10 次*/
      printf("加油! ");
}
```

运行程序，结果输出了 11 次。原因很明显，for 表达式中的“i<=10”应该改为“i<10”。在实际的开发中，逻辑错误往往比较难于发现。通过这个例子，我们来分析如何通过调试把错误找出来。

① 单步执行程序。使用单步执行程序，可以跟踪代码的每一步运行，想要单步调试，可以使用快捷键 F10 或 F11，或单击菜单命令“调试”→“step over”或“step into”。两者的区别在于：单步执行时，可以选择是否跳过一行代码中所调用的方法，如果是，则使用快捷键 F10；如果想要进入过程，则使用快捷键 F11。当程序暂停以后，Visual C++的监视窗口便可以显示当前的变量值情况。监视窗口有两列，分别显示想要监视的变量名称和变量的值，当变量的值发生变化时，就会用红色显示，也可以在右下角的变量窗口新建要监视的变量，如图 1-9 所示。

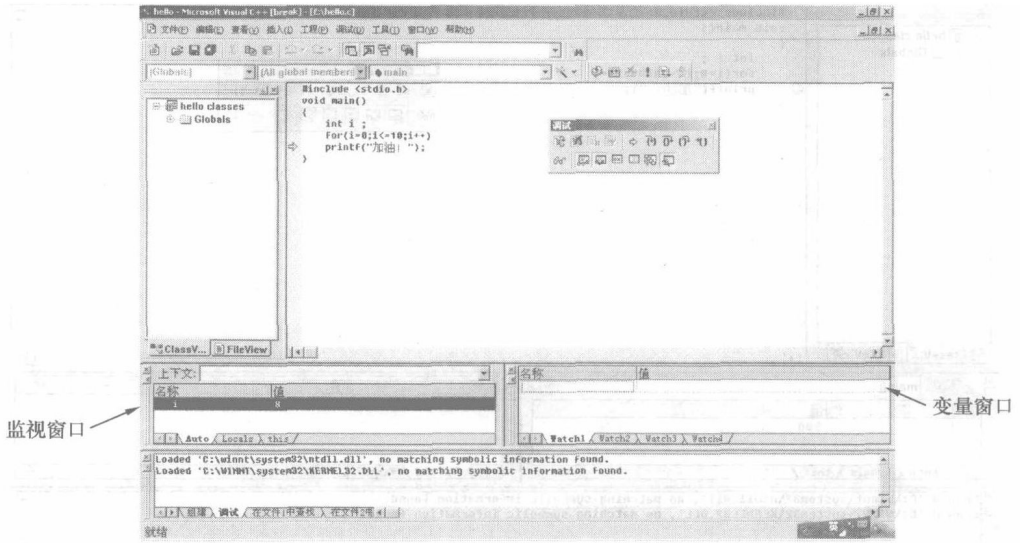


图 1-9　单步调试

连续按 F10 键 11 次，仔细观察监视窗口内 i 的值，在最后一次的时候，将发现 i 的值为 10，

这时便可发现问题所在，最后按 Shift+F5 组合键取消调试。

② 设置断点。对于单步执行，调试较大规模的程序将耗费大量时间，如例 1.4，若循环条件为 i < 10000，单步调试就很难办了。这时，可以通过设置断点的方式，使代码暂停在程序员想要的地方。

在需要设置断点的代码行前，按 F9 键，将出现一个红色的圆点，可以设置断点有效的条件，如例 1.4，我们设置当“i= =500”时暂停下来。单击 “编辑”→“断点”菜单命令，将弹出断点属性对话框，如图 1-10 所示，在“location”选项卡页面中，先选择默认的断点，单击“条件”按钮，在弹出的“断点条件”对话框中，输入表达式“i= =500”，如图 1-11 所示，单击“确定”按钮后，回到程序调试窗口。

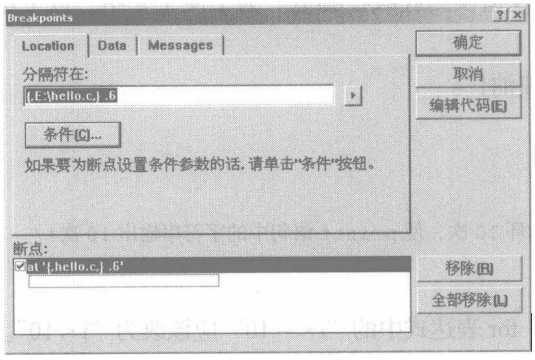


图 1-10 断点属性

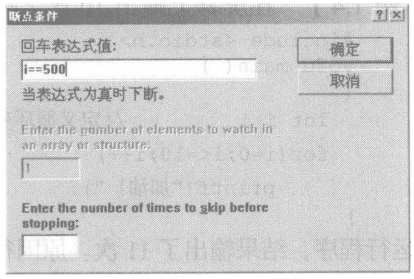


图 1-11 断点条件

然后按 F5 键开始调试，运行程序，当程序运行到指定的断点位置，并满足暂停条件时，代码就会暂停运行。这时就可以使用单步执行中同样的变量监视方法，来查看运行状态，以寻找错误的根源，如图 1-12 所示。

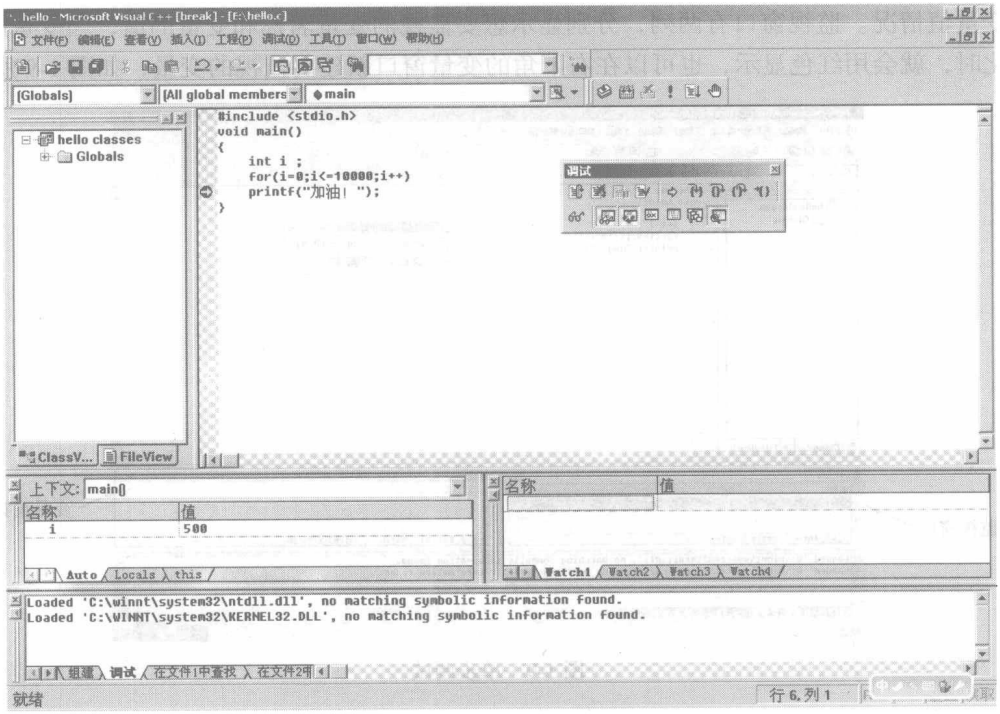


图 1-12 断点调试暂停运行