第1章 概　　述

伴随着计算机的高速发展和普及，来自不同专业背景的各行各业的人们越来越感受到计算机对于工作生活产生的巨大影响。作为无生命的计算机本身，是无法与人直接进行交流沟通的，人们只有通过编写程序，命令计算机执行程序，才能完成相应的任务。程序可以通过各种语言进行编写，最早出现的低级语言对计算机硬件过于依赖，并且编写的程序在可读性和可移植性方面都比较差，为解决这一问题，人们创造了许多高级语言。在各种各样的高级语言中，C 语言以其强大的功能和优良的特点，成为国际上公认的、重要的少数几种通用程序设计语言之一。C++语言进一步扩充和完善了 C 语言，C 语言是 C++的基础，二者在很多方面是兼容的。因此，掌握了 C 语言的编程规范，再进一步学习 C++语言就更容易和便利，并能达到事半功倍的效果。

本章简要介绍 C 语言和 C++语言的发展历史和特点，举例介绍两种语言编写的简单程序格式以及程序的运行环境和运行方法。

1.1 语言的发展及特点

1.1.1 语言的发展

C 语言的出现比许多程序设计语言，如 BASIC、FORTRAN、PASCAL、COBOL 等都要晚，但相较而言它却是最具生命力的。C 语言最早的原型是 Algol 60。1963 年，剑桥大学将其发展成为 CPL(Combined Programming Language)。1967 年，剑桥大学的 Matin Richards 对 CPL 语言进行了简化，产生了 BCPL 语言。1970 年，美国贝尔实验室(Bell Labs)的 Ken Thompson 对 BCPL 进行了修改，并取名为 B 语言，意思是提取 CPL 的精华，并用 B 语言编写了第一个 UNIX 系统，但 B 语言过于简单，功能有限。1973 年，AT&T 贝尔实验室的 Dennis Ritchie(D.M.Ritchie)在 BCPL 和 B 语言的基础上设计出了一种新的语言，取 BCPL 中的第二个字母为名，这就是大名鼎鼎的 C 语言。随后不久，UNIX 的内核(Kernel)和应用程序全部用 C 语言改写。从此，C 语言成为 UNIX 环境下使用最广泛的主流编程语言，后来经过多次改进，但主要应用在贝尔实验室内部。直到 1975 年，随着 UNIX 操作系统第 6 版的问世，C 语言的突出优点才引起人们的普遍关注。可以看出，起初 C 语言的发展和 UNIX 操作系统是密切相关的，1978 年之后，它已先后移植到大型、中型、小型及微型机上，并独立于 UNIX。同年，B.W.Kernighan 和 D.M.Ritchie 合著了 *The C Programming Language* 一书，该书对 C 语言的语法进行了规范化描述。1983 年，美国国家标准协会(ANSI)为 C 语言制定了一套 ANSI 标准，统一了 C 语言的各种版本。1987 年，ANSI 再次公布了新标准——87 ANSI C。1990 年国际标准化组织(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，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。

随着 C 语言的应用推广，C 语言存在的一些缺陷和不足也显现出来，比如缺少支持代码重用的

结构；随着软件工程规模的扩大，难以适应开发特大型程序等。20 世纪 80 年代初，贝尔实验室的 Bjarne Stroustrup 开发了 C++语言，对 C 语言的功能进行了改进和扩充。C++语言对于 C 语言是兼容的，这对于继承和开发当前已广泛使用的软件是非常重要的，可节省大量的人力和物力。

1.1.2 语言的特点

每种语言都有其特点，都会有优缺点，其中某些优缺点必然是并存的，只是立足的观察点不同。

C 语言的特点如下：

(1) 语言简洁，结构紧凑，使用方便、灵活。C 语言一共有 32 个关键字和 9 条控制语句，以易读易写的小写字母为基础，源程序书写规整紧凑。

(2) 数据类型丰富。C 语言除了标准数据类型(整型、实型和字符型)，还有多种构造类型(数组、结构体和共同体等)以及指针类型，有助于构造复杂的数据结构，从而提高程序的运行效率。

(3) 运算符丰富。C 语言提供了 34 种标准的运算符。C 语言的运算符不仅具有优先级的概念，还有结合性的概念。因此，灵活使用各种运算符和表达式，不仅可以简化程序，还可以实现在其他语言中难以实现的运算。

(4) 提供了功能齐全的函数库。C 语言标准函数库提供了功能极强的各类函数库，如高等数学运算、字符串处理、标准输入/输出等，只需要调用库函数即可实现，为编程者提供了方便，极大地降低了编程工作量。

(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)。C 程序由具有独立功能的函数构成，用函数作为程序的模块单位，便于实现程序的模块化，其基本思想是将一个大的程序按功能分割成一些模块，使每个模块都成为功能单一、结构清晰、容易理解的函数。函数定义是平行、独立的，但函数调用可以有嵌套调用和递归调用，通过函数调用可以实现复杂的程序功能。另外，对于复杂的源程序文件，可以分割成多个较小的源文件，分别编译和调试，最后组装，连接得到可执行的程序文件。

(6) 可移植性好。所谓可移植性是指从一个系统环境下不加或稍加改动就可搬到另一个完全不同的系统环境中运行。C 语言编译程序的大部分代码都是公共的，基本上不做任何修改就能运用于各种不同型号的计算机和各种操作系统环境中。

(7) 语言生成的代码质量高。对一个应用程序来说，如果生成的目标代码(可执行程序)质量低，则系统开销大，无实用性。C 语言生成目标代码的效率与汇编语言相比，一般低 10%～20%。C 语言也可以像汇编语言一样对位、字节和地址，甚至对硬件进行直接操作。换句话说，C 语言既具有汇编语言的强大功能，又具有较强的可读性，特别适合做底层开发。

当然，C 语言也存在缺陷，比如语法限制不太严谨；运算符的优先级和结合性比较复杂，不容易记忆；对数据类型缺乏一致性的检测；对数组的使用不进行越界检查，容易导致数据出错；在程序设计方面灵活性有余，而安全性和可靠性不足。

C++语言是建立在 C 语言基础之上，但它并不是对 C 语言的功能做简单的改进和扩充，它具备了“面向对象程序设计”(Object-Oriented Programming，OOP)的能力。OOP 是一种功能非常强大的编程技术，可以使得程序的各个模块的独立性更强、程序的可读性和可理解性更好、程序代码的结构性更加合理。这对于设计和调试一些规模大的软件是非常重要的。本书将从第 10 章开始详细介绍 C++语言的核心内容。

1.2　简单的语言程序介绍

1.2.1　简单的 C 程序分析

下面通过两个最基本的 C 程序来说明 C 程序的基本结构特性。

【例 1.1】 在屏幕显示“How are you ?”。

```
/*  第一个 C 程序
在屏幕上显示 How are you */
#include <stdio.h>                    //头文件
void main()                           //主函数
{
    printf("How are you? \n");        //调用库函数 printf()显示字符串
}
```

屏幕显示运行结果：

```
How are you?
```

程序说明：

(1)程序中出现的用“/*……*/”和以“//”开头的文字说明为注释，用于解释程序，并不是程序的组成部分，不参加程序的执行。程序员编写程序时在程序中添加必要的注释，有助于增加程序的可读性。“/*……*/”和“//”在使用上有所区别，“/*……*/”可以包含多行文字，常用于说明一个程序或一个函数的整体功能作用等；而“//”只能包含一行文字，常用于说明局部语句的作用。

(2)#include 为编译预处理命令。若使用系统库函数，如本例中使用的 printf 标准输出函数，则需要在程序的开头包含相应的系统头文件，如本例中的 stdio.h。本书第 5 章将详细介绍编译预处理命令。

(3)main 是函数名，并且表示主函数。C 语言编写的程序是由函数构成的，一个程序可以包含多个函数，至少有而且只能有一个主函数。换言之，一个 C 语言程序由一个主函数(main 函数)和若干个其他函数组成。并且，程序是从主函数开始执行的，其他的函数可以通过函数调用来执行，本书将在第 4 章详细介绍函数。

(4)花括号“{ }”中编写用于实现功能的语句，如本例中的 printf 语句，实现将字符串“How are you？”输出到标准输出设备(通常为显示屏)上。“\n”是转义字符，表示输出结束后光标转到下一行的行首。C 语言规定以“;”作为语句的结束标志。

【例 1.2】 用户从键盘输入两个整数，计算机在屏幕上输出两个数的和。

```
#include <stdio.h>
/*函数名 add；函数有返回值,为整型(int)；函数有两个整型形参 x 和 y */
int add(int x,int y)
{
    int z;        //定义一个整型变量 z
    z=x+y;        //将 x 和 y 的和赋值给变量 z
    return(z);    //将 z 的值返回给将要调用 add 函数的主调函数
}
```

```
/*主函数*/
void main()
{
    int x,y,z;                          //定义变量
    scanf("%d,%d",&x,&y);               //标准输入函数，属于系统库函数
    z=add(x,y);                         //调用 add 函数
    printf("The result is %d",z);       //输出结果
}
```

执行程序后，输入：

```
1, 5↙
```

运行结果：

```
The result is 6
```

程序说明：

(1)程序中包含了一个主函数(main 函数)和一个普通函数(add 函数)，程序从主函数开始执行，当发现出现函数调用时就转去执行被调函数，执行完被调函数后再转回调用点继续执行主调函数。本例中，main 函数即为主调函数，add 函数为被调函数，调用语句为“z=add(x,y);”。

(2)scanf 为系统提供的标准输入函数，和 printf 一样，只要包含了相应的头文件 stdio.h 就可以直接使用。通过执行该语句，允许程序的使用者按格式要求从标准输入设备(通常为键盘)输入值。比如，此程序执行后，在输入两个数时需要用逗号分开。关于标准化的输入/输出语句将在第 2 章进行详细介绍。

1.2.2 简单的 C++程序分析

下面用 C++语言编写程序实现例 1.1：

```
#include <iostream>
using namespace std;
void main()
{
    cout<< "How are you? \n";        //输出
}
```

程序说明：

(1)“#include <iostream>”是以“#”开头的预编译指令行，#include <文件名>通常用于将其他文件内容包含到当前程序中。本例中，指令“#include <iostream>”告诉预处理程序把 iostream 标准文件包含进来。这个特殊的文件(iostream)中包含有 C++标准输入/输出的功能，本例中使用它的 cout 来实现输出功能。

(2)“using namespace std;”告诉编译器包含 std 名称空间中的功能。C++标准库中的所有功能都被包含在 std 名称空间中。因此，为了使用这些功能(如#include<iostream>引入的输出功能)，在使用标准库的 C++程序里这一行是经常出现的。有些书中将这两条语句合为一条语句“#include <iostream.h>;”。

(3)“cout << “Hello world!\n”;”的功能是向标准输出设备(通常为显示器)输出一行字符。cout 表示 C++中的标准输出流，双引号内的字符串被原样输出。

下面再用 C++语言来改写例 1.2 的程序：

```
#include <iostream.h>
int add(int x,int y)
{
    int z;
    z=x+y;
    return(z);
}

void main()
{
    int x,y,z;
    cin>>x>>y;                    //标准输入函数 cin，属于系统库函数
    z=add(x,y);
    cout<<"The result is "<<z;   //标准输出函数 cout，属于系统库函数
}
```

执行程序后，输入：

```
1  5↙
```

运行结果：

```
The result is 6
```

程序说明：

对比两种语言编写的程序，除了包含的头文件不同之外，最大的区别在于标准输入/输出函数的使用上，C 语言使用的标准输入/输出函数分别是 scanf 函数和 printf 函数，C++语言使用的则是 cin 函数和 cout 函数。

1.3　程序上机环境、步骤方法简介

在 Windows 操作系统中，Visual C++ 6.0(以下简写为 VC++6.0)是比较流行的程序开发平台。VC++6.0 不仅支持 C 程序的编译调试，而且支持 C++语言，为用户开发 C 或 C++程序提供了一个集成环境。这个集成环境包括实现源程序的输入、编辑和修改，源程序文件的保存与打开，源程序的编译和链接，程序运行期间的调试与跟踪，工程项目的自动管理，提供应用程序的开发工具，多窗口管理，提供联机帮助等。该集成环境功能齐全、环境复杂，只有经过较长时间的上机实践，逐步理解和体会，才能熟练运用集成环境中的各种工具。

一个程序的开发包含四个步骤：①编辑：在程序开发平台上编写源程序代码，如果使用 C 语言进行编写，则创建的源程序文件扩展名为“.c”；如果使用 C++语言进行编写，则源程序文件扩展名为“.cpp”。②编译：利用开发平台提供的编译系统对编写的源程序文件进行语法错误的检测，形成目标文件，目标文件的扩展名为“.obj”，若没有错误，则可继续进行下一

步，否则必须修改。③链接：链接目标文件形成可执行文件，可执行文件的扩展名为“.exe”。④执行：运行可执行文件。

下面将结合 VC++6.0，详细叙述一个 C 程序的开发过程。

1．创建项目

在 VC++6.0 中，以工作空间(Workspace)来管理项目(Project)。一个工作空间中可以包含一到多个相互关联的项目，一个项目是多个相互关联的源文件以及其他资源的集合。通常一个项目的代码文件放在同一个物理目录下。

创建项目的操作步骤如下。

(1)运行 VC++6.0，出现如图 1-1 所示界面。

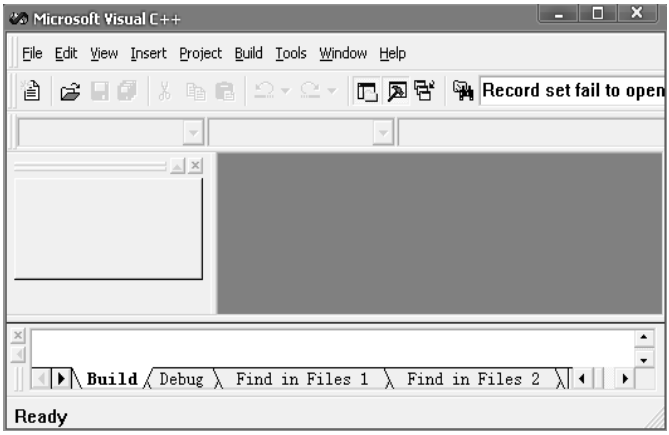

图 1-1　VC++6.0 运行界面

(2)选择 File | New 命令，进入创建新项目的向导对话框，首先要确定项目的类型、名称和位置等信息，如图 1-2 所示。

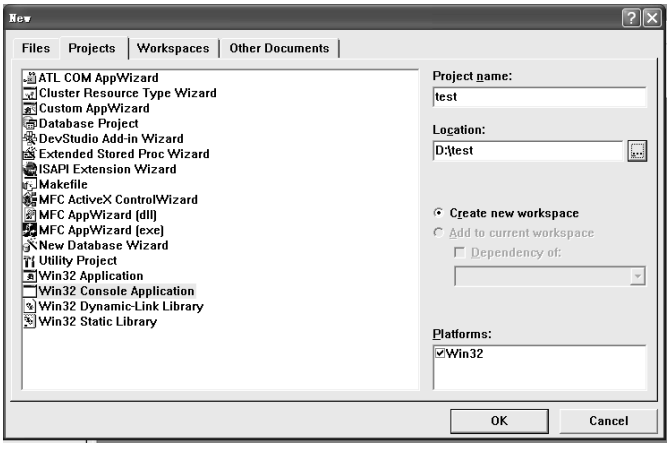

图 1-2　新建项目对话框

① 在 Projects 选项卡中，选中项目类型“Win32 Console Application”。

② 在 Project name 文本框中输入项目名，如项目名为“test”

③ 在 Location 文本框中输入项目的位置，如保存在 D 盘下，填写“D:\test”。

④ VC++6.0 会在 Location 指定的目录下创建一个与项目同名的子目录，本项目的所有代码默认都放到该目录下。

⑤ 选中 Create new workspace 单选按钮，表示要创建新的工作空间。

(3) 单击 OK 按钮，弹出如图 1-3 所示的向导对话框，用来确定框架代码的组成。这里创建一个空的项目：选中 An empty project 单选按钮，单击 Finish 按钮。

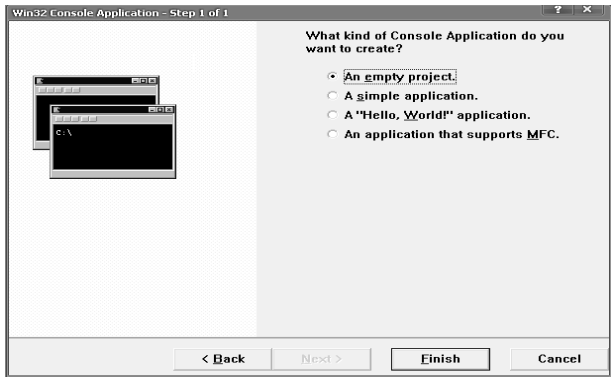

图 1-3　控制台应用程序

(4) 弹出一个工程信息对话框，如图 1-4 所示，显示创建了一个空的控制台应用程序，并且没有文件被创建或者添加到工程中。单击 OK 按钮。

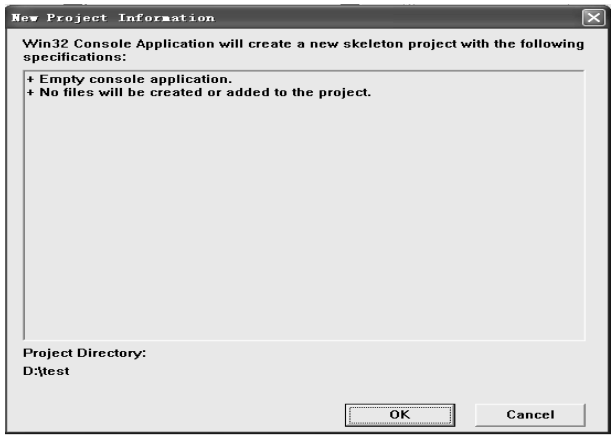

图 1-4　工程信息对话框

(5) 回到 VC++6.0 的开发界面，此时，窗口左侧的工作空间栏中出现了几个文件夹层次图，如图 1-5 所示。

2．编辑

完成以上操作之后，便可以进行代码的编写了。VC++6.0 自带的文本编辑器功能非常强大，可以方便地进行代码的编辑。在编写过程中，关键字会用不同的颜色进行标识。

首先新建一个源文件。选择 File | New 命令，然后选择 C++ Source File 选项，输入文件名，单击 OK 按钮，如图 1-6 所示。命名的源文件就被添加到刚刚新建的项目中，并处于打开状态。由于 VC++6.0 支持 C 语言和 C++语言，因此在输入源文件时要指出文件是何种语言书写的，以通过扩展名来区分，比如采用 C 语言编写，源程序取名为 1.c，如图 1-6 所示。

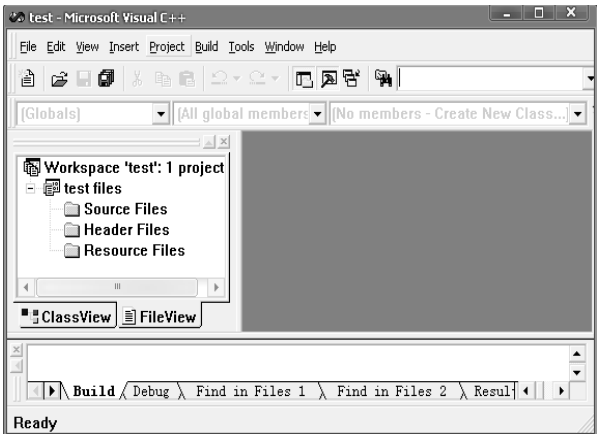

图 1-5　工程创建完成

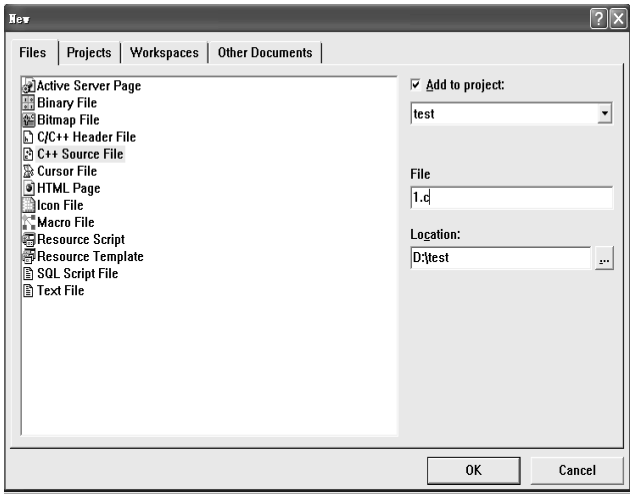

图 1-6　添加 C 源文件

此时，就可以直接在代码编辑区进行程序设计了，如图 1-7 所示，输入了一个简单程序的代码，其功能是输出字符串“How are you？”。

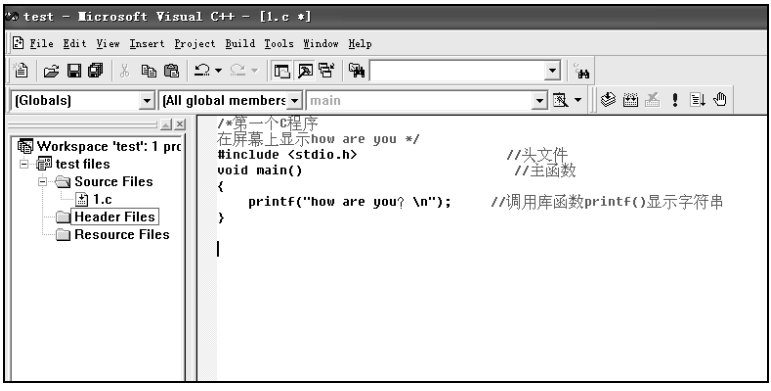

图 1-7　编辑状态的 VC++6.0

通常注释的颜色为绿色，关键字的颜色为蓝色，普通字符串及运算符的颜色为黑色。

3．编译

编译操作可以用以下三种方法来完成。

① 在 Build 菜单中选择 Compile filename.cpp 命令。

② 使用组合键 Ctrl + F7 进行编译。

③ 单击工具栏中的编译按钮。

这三种方法执行的结果都是生成对应源文件的目标文件(文件扩展名为“.obj”)。编译完成之后，如果有错误，则输出区会显示错误以及警告个数，双击错误项可以定位到错误代码所在的行，如图 1-8 所示。

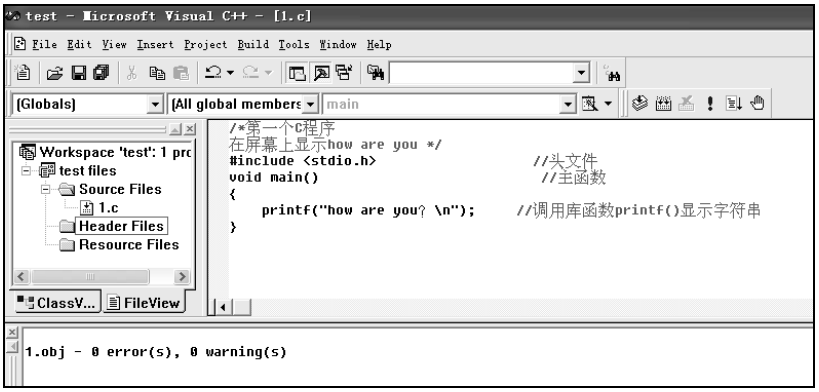

图 1-8　编译状态的 VC++6.0

4．链接

链接操作是将编译生成的目标文件，与运行程序必需的库文件合并生成一个可执行文件。VC++6.0 中没有单独的链接操作，但是在 Build 菜单下有一个 Build filename.exe 命令。选择该命令可以一次性执行编译和链接操作。也可以使用快捷键 F7，或者单击工具栏中的按钮。执行完链接之后，如果没有错误，则可执行文件(文件扩展名为“.exe”)就被存放在了工程目录下的 Debug 文件夹中，如图 1-9 所示。

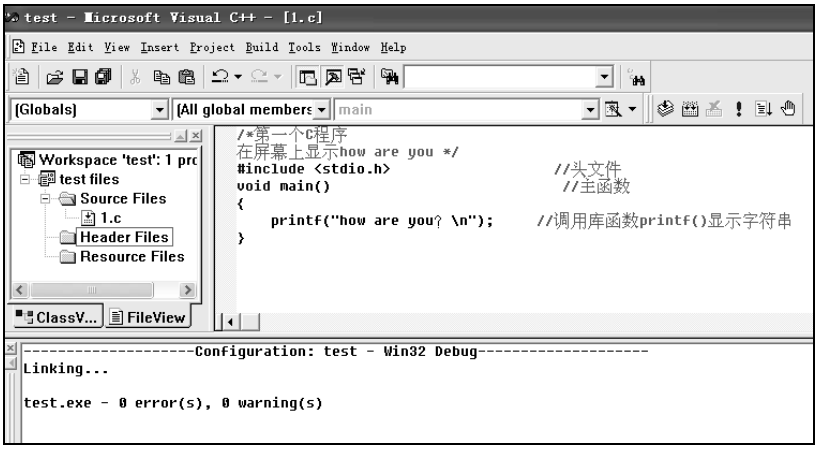

图 1-9　链接状态的 VC++6.0

5．运行和调试

在编译和链接完成后，可以选择 Build 菜单下的 Excute filename.exe 命令或使用组合键 Ctrl+F5 来运行程序，也可以单击工具栏中的 ! 按钮。VC++6.0 会自动将链接生成的可执行文件加载到内存中运行。若 VC++6.0 发现上次编译和链接操作后源文件被修改过，则自动编译被修改的源文件，并重新链接所有的目标文件，生成可执行文件，最后运行。成功运行后，结果将显示在屏幕上，如图 1-10 所示。

以上是一个简单 C 语言程序的开发过程，利用 C++语言进行开发时，只有创建源程序文件时需要标明文件扩展名为“.cpp”，其他步骤都相同。但需要指出的是，利用开发平台在编译时只能找出程序在语法上的错误，编译通过并不意味着程序最终的运行结果一定是正确的，程序中可能存在语句逻辑上的错误，如果运行结果不正确，我们可以利用 VC++6.0 提供的调试功能帮助查找错误。下面，我们先来认识一下 VC++ 6.0 的调试菜单，然后将介绍常用的调试操作，即设置断点和分步执行。

1)调试菜单

调试菜单对编写程序非常重要，它可以完成单步执行功能，帮助我们找到程序的错误。从菜单栏中选择“Build”→“Start Debug”命令，即可打开调试菜单，如图 1-11 所示。

调试菜单中包含了一些与程序调试有关的命令。

图 1-10　运行结果

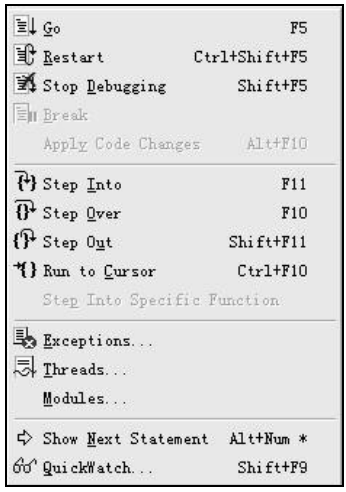

图 1-11　调试菜单

下面具体说明一些主要的调试命令。

- Go：开始调试。
- Restart：重新进行调试。
- Stop Debugging：结束调试。
- Step Into：单步执行，并跟踪进入调用函数内部。
- Step Over：单步执行，不跟踪进入调用函数内部。
- Step Out：单步执行，并从所调用函数内部跳出。
- Run to Cursor：运行程序至光标所在行。

2)设置断点

通过断点的设置，可以让程序在我们所需要的地方停止运行。设置断点的方法如下。

首先将光标移到需要设置断点的地方；然后单击按钮，这时就会出现一个红点，如图 1-12 所示，此红点就是断点；最后单击按钮，进行调试。

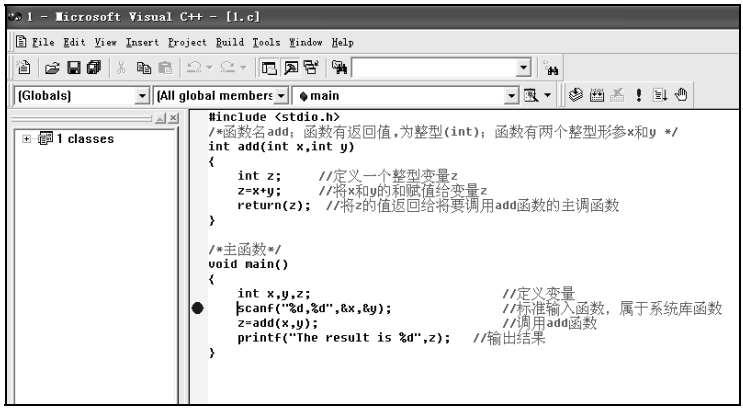

图 1-12　在某个被调试的程序中设置断点

这时程序会在断点处停下来，如图 1-13 所示，左下角的界面区域中出现变量的名称和值（Name 栏列出当前所用到的变量名，Value 栏列出这些变量的值）。通过这个区域的内容，我们可以看到每步中变量的值，以判断程序的运行状况。

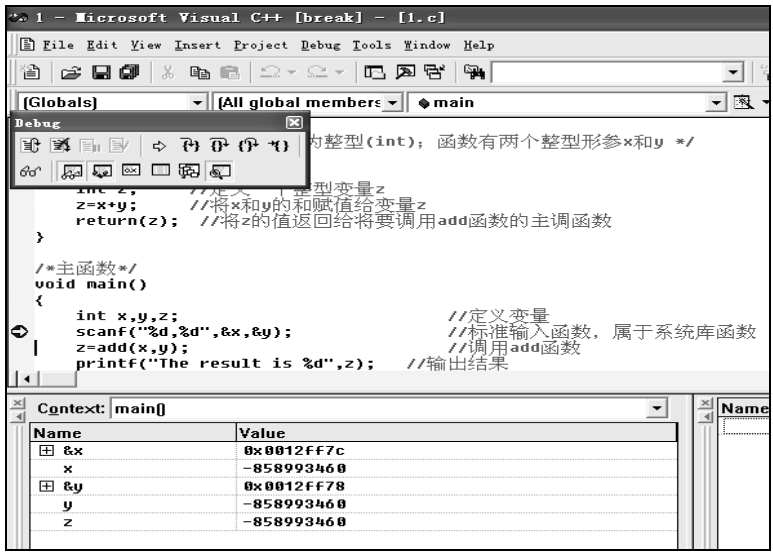

图 1-13　断点停留处

3）分步执行

按 F10 键或从菜单栏中选择“Build”→“Start Debug”→“Step Over”命令跳到下一步，如图 1-14 所示，这时指示箭头会向下移一行，代表程序运行了一步，如果这时变量的值有变化，就可以从 Name 栏和 Value 栏中反映出来。如果程序中调用了函数，我们想要看到所调用函数的运行情况，就需要按 F11 键。

4）结束调试

通过从菜单栏中选择“Debug”→“Stop Debugging”命令可以结束调试，当程序修改结

束后，需要运行前应取消断点。取消断点的方法是将光标移到断点的地方，并单击该断点，然后单击按钮，则红点消失，断点取消。

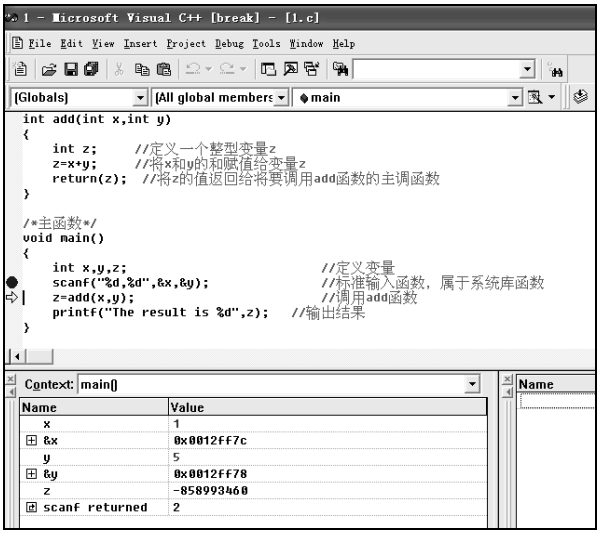

图 1-14 分步执行

习 题

1．模仿书中例子，利用 C 语言和 C++语言编写程序，用于显示下面的信息：
Welcome to programming world!

2．编写程序，从键盘输入两个数，并将这两个数的积显示出来。

3．对于习题 2 中的程序进行单步运行调试。