第一部分　C 语言程序设计实验指导

实验 1　Visual C++ 6.0 集成开发环境介绍

1.1　实 验 目 的

通过本实验内容，实现如下学习目标：

- 熟悉 Visual C++ 6.0 集成开发环境的界面布局和各功能部件。
- 掌握在 Visual C++ 6.0 集成开发环境下编辑 C/C++源程序的方法和步骤。
- 掌握在 Visual C++ 6.0 集成开发环境下对 C/C++源程序进行编译、调试和运行的方法和步骤。

1.2　Visual C++ 6.0 的界面布局和各功能部件

Visual C++6.0 是 Microsoft 公司开发的在 Windows 环境下进行应用程序开发的 C/C++集成开发环境。它是 Microsoft 公司的 Visual Studio 开发工具箱中的一个 C++程序开发包。Visual Studio 提供了一整套开发 Internet 和 Windows 应用程序的工具，包括 Visual C++、Visual Basic、VisualFoxPro、Visual InterDev、Visual J++以及其他辅助工具。Visual C++包中除包括开发程序所必需的编辑器、C++编译器、连接程序、调试程序外，还包括许多为开发 Windows 系统下的 C++程序而设计的各种各样的工具。Visual C++6.0 中的集成开发环境称为 Developer Studio。图 1-1 展示了 Visual C++6.0 界面的布局情况。

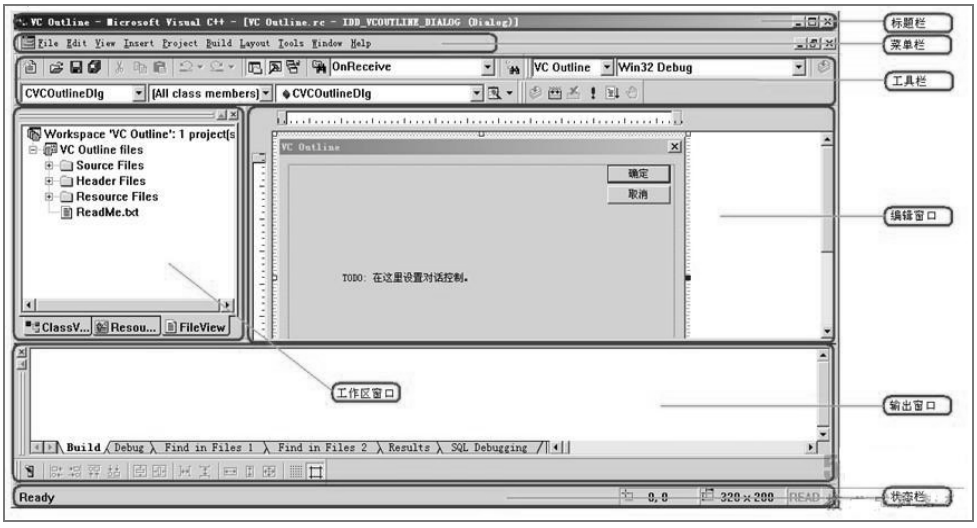

图 1-1　Visual C++6.0 界面的布局

Visual C++6.0 的功能部件总体上由菜单、工具栏、窗口三个主要部分构成，下面分别介绍这些部件的常用部分：

1.2.1　Visual C++6.0 的常用菜单命令项

(1) File 菜单

- New：打开 new 对话框，以便创建新的文件、工程或工作区。
- Close Workspace：关闭与工作区相关的所有窗口。
- Exit：退出 Visual C++6.0 环境，将提示保存窗口内容等。

(2) Edit 菜单

- Cut：快捷键 Ctrl+X。将选定内容复制到剪贴板，然后再从当前活动窗口中删除所选内容。与 Paste 联合使用可以移动选定的内容。
- Copy：快捷键 Ctrl+C。将选定内容复制到剪贴板，但不从当前活动窗口中删除所选内容。与 Paste 联合使用可以复制选定的内容。
- Paste：快捷键 Ctrl+V。将剪贴板中的内容插入(粘贴)到当前鼠标指针所在的位置。注意，必须先使用 Cut 或 Copy 使剪贴板中具有准备粘贴的内容。
- Find：快捷键 Ctrl+F。在当前文件中查找指定的字符串。顺便指出，可按快捷键 F3 寻找下一个匹配的字符串。
- Find in Files：在指定的多个文件中查找指定的字符串。
- Replace：快捷键 Ctrl+H。替换指定的字符串(用某一个串替换另一个串)。
- Go To: 快捷键 Ctrl+G。将光标移到指定行上。
- Breakpoints：快捷键 Alt+F9。弹出对话框，用于设置、删除或查看程序中的所有断点。断点将告诉调试器应该在何时何地暂停程序的执行，以便查看当时的变量取值等现场情况。

(3) View 菜单

- Workspace：如果工作区窗口没显示出来，选择执行该项后将显示出工作区窗口。
- Output：如果输出窗口没显示出来，则选择执行该项后将显示出输出窗口。输出窗口中将随时显示有关的提示信息或出错警告信息等。

(4) Project 菜单

- Add To Project：选择该项将弹出子菜单，用于添加文件或数据链接等到工程之中去。例如，子菜单中的 New 选项可用于添加 C++ Source File 或 C/C++ Header File；而子菜单中的 Files 选项则用于插入已有的文件到工程中。
- Settings：为工程进行各种不同的设置。当选择其中的 Debug 标签(选项卡)，并通过在 Program arguments 文本框中填入以空格分割的各命令行参数后，可以为带参数的 main 函数提供相应参数(呼应 void main(int argc,char* argv[ ]){…}形式的 main 函数中所需各 argv 数组的各字符串参数值)。注意，在执行带参数的 main 函数之前，必须进行该设置，当 Program arguments 文本框中为空时，意味着无命令行参数。

(5) Build 菜单

- Compile：快捷键 Ctrl+F7。编译当前处于源代码窗口中的源程序文件，以便检查是否有语法错误或警告，如果有则将显示在 Output 输出窗口中。
- Build：快捷键 F7。对当前工程中的有关文件进行连接，若出现错误则将显示在 Output 输出窗口中。
- Execute：快捷键 Ctrl+F5。运行(执行)已经编译、连接成功的可执行程序(文件)。
- Start Debug：选择该项将弹出子菜单，其中含有用于启动调试器运行的几个选项。例

如，其中的 Go 选项用于从当前语句开始执行程序，直到遇到断点或遇到程序结束；Step Into 选项开始单步执行程序，并在遇到函数调用时进入函数内部再从头单步执行；Run to Cursor 选项使程序运行到当前鼠标光标所在行时暂停其执行(注意，使用该选项前，要先将鼠标光标设置到某个希望暂停的程序行处)。执行该菜单的选择项后，就启动了调试器，此时菜单栏中将出现 Debug 菜单(而取代了 Build 菜单)。

(6) Debug 菜单

启动调试器后才出现该 Debug 菜单(而不再出现 Build 菜单)，菜单项如下所示。

- Go：快捷键 F5。从当前语句启动继续运行程序，直到遇到断点或遇到程序结束而停止(与 Build→Start Debug→Go 选项的功能相同)。
- Restart：快捷键 Ctrl+Shift+F5。重新从头开始对程序进行调试执行(注意，当对程序做过某些修改后往往需要这样做！)。选择该项后，系统将重新装载程序到内存，并放弃所有变量的当前值(而重新开始)。
- Stop Debugging：快捷键 Shift+F5。中断当前的调试过程并返回正常的编辑状态(注意，系统将自动关闭调试器，并重新使用 Build 菜单来取代 Debug 菜单)。
- Step Into：快捷键 F11。单步执行程序，并在遇到函数调用语句时，进入该函数内部，并从头单步执行(与 Build→Start Debug→Step Into 选项的功能相同)。
- Step Over：快捷键 F10。单步执行程序，但当执行到函数调用语句时，不进入该函数内部，而是一步直接执行完该函数后，接着再执行函数调用语句后面的语句。
- Step Out：快捷键 Shift+F11。与 Step Into 配合使用，当执行进入到函数内部，单步执行若干步之后，若发现不再需要进行单步调试，则通过该选项可以从函数内部返回(到函数调用语句的下一语句处停止)。
- Run to Cursor：快捷键 Ctrl+F10。使程序运行到当前鼠标光标所在行时暂停其执行(注意，使用该选项前，要先将鼠标光标设置到某个希望暂停的程序行处)。事实上，相当于设置了一个临时断点，与 Build→Start Debug→Run to Cursor 选项的功能相同。
- Insert/Remove Breakpoint：快捷键 F9。本菜单项并未出现在 Debug 菜单上(在工具栏和程序文档的上下文关联菜单上)，列在此处是为了方便大家掌握程序调试的手段，其功能是设置或取消固定断点——程序行前有一个圆形的黑点标志，表示已经该行设置了固定断点。另外，与固定断点相关的还有 Alt+F9(管理程序中的所有断点)和 Ctrl+F9(禁用/使能当前断点)。

(7) Help 菜单

通过该菜单来查看 Visual C++6.0 的各种联机帮助信息。

(8) 上下文关联菜单

除了主菜单和工具栏外，Visual C++6.0 开发环境还提供了大量的上下文关联菜单，右击窗口中很多地方都会弹出一个关联菜单，里面包含有与被单击项目相关的各种命令。

在程序运行时，Visual C++ 6.0 的菜单栏可以动态改变，还会显示[Layout]和[Debug]。如果是在调试状态下，则[Build]会显示为[Debug]。

1.2.2 Visual C++6.0 的常用工具栏

工具栏由多个操作按钮组成，这些操作一般都与某个菜单项对应。主要工具栏如下。

- Standard：提供最基本的功能，如文件操作、编辑、查找等。

- Build：工程的编译、连接、修改活动配置、运行调试程序。
- Resource：添加各种类型的资源。
- Edit：剪切、复制和粘贴等功能。
- Debug：用于调试状态的若干操作
- Browse：源程序浏览操作
- Database：跟数据库有关的操作。

1.2.3　Visual C++6.0 的常用窗口

(1) Workspace(工作空间)窗口

工作空间(workspace)是一个包含用户的所有相关项目和配置的实体。项目(project)被定义为一个配置和一组文件，用以生成最终的程序或二进制文件。一个工作空间可以包含多个项目，这些项目既可以是同一类型的项目，也可以是由不同类型的项目(如 Visual C++和 Visual J++项目)。Workspace 窗口显示了项目各个方面的信息。在窗口底端选择相应的选项卡来按不同视图显示项目的列表。

- ClassView：列出项目中的类和成员函数。双击列表中的类或函数，即可在 Visual C++文本编辑器中打开该类的源文件。
- ResourceView：列出项目的资源数据。双击列表中的数据项会打开合适的编辑器并加载资源。
- FileView：列出项目的源文件、头文件。

(2) Output(输出)窗口

输出窗口输出一些用户操作后的反馈信息，它由一些页面组成，每个页面输出一种信息，输出的信息种类主要有以下几种。

- 编译信息：在编译时输出，主要是编译时的错误和警告。
- 调试信息：在对程序进行调试时输出，主要是程序当前的运行状况。
- 查找结果：在用户从多个文件中查找某个字符串时产生，显示查找结果的位置。

(3) 编辑窗口

编辑窗口为开发者提供了编辑文件和资源的手段。通过编辑窗口，开发者可以编辑和修改源程序和各种类型的资源。

(4) 调试窗口

调试窗口包括一组窗口，在调试程序时分别显示各种信息，主要包括以下几个：

- 变量查看窗口(Watch)；
- 过程调用查看窗口(Call Stack)；
- 内存查看窗口(Memory)；
- 寄存器查看窗口(Register)。

1.3　C/C++应用程序的开发步骤

一个程序的开发包含以下四个步骤：

(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。

下面具体介绍在 Visual C++6.0 环境中开发一个 C 程序的过程和步骤。

在开始编程之前，必须先了解工程 project(也称项目，或工程项目)的概念。这是 Visual C++系列开发工具的特色。工程又称项目，它具有两种含义，一种是指最终生成的应用程序；另一种则是为了创建这个应用程序所需的全部文件的集合，包括各种源程序、资源文件和文档等。绝大多数较新的开发工具都利用工程来对软件开发过程进行管理。用 Visual C++6.0 编写并处理的任何程序都与工程有关(都要创建一个与其相关的工程)，而每个工程又总与一个工作区相关联。工作区是对工程概念的扩展。一个工程的目标是生成一个应用程序，但很多大型软件往往需要同时开发数个应用程序，Visual C++6.0 开发环境允许用户在一个工作区内添加数个工程，其中有一个是活动的(默认的)，每个工程都可以独立进行编译、连接和调试。

1．创建项目

在 Visual C++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个或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 Visual C++6.0，出现如图 1-2 所示界面。

(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 1-3 所示。

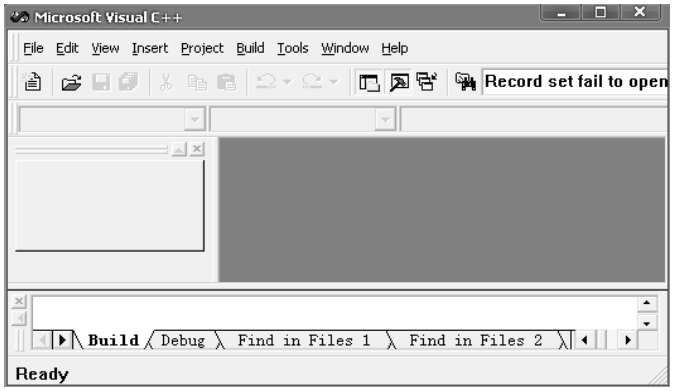

图 1-2　Visual C++6.0 运行界面

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

在 Project name 文本框中输入项目名，如“test”。

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

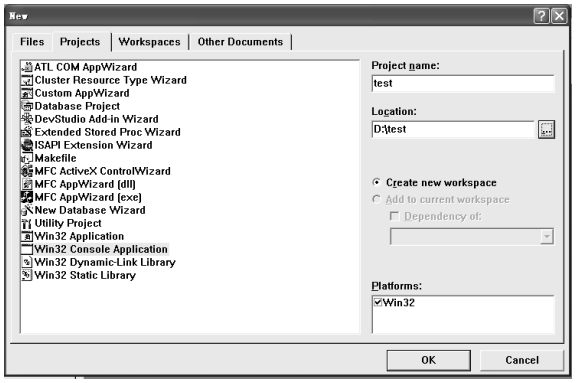

图 1-3　新建项目对话框

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

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 1-4 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。

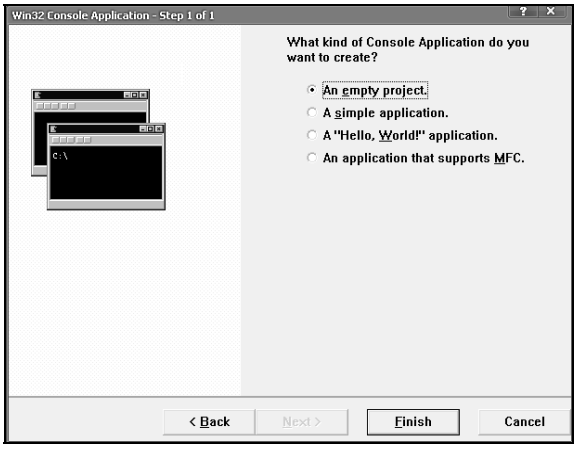

图 1-4　控制台应用程序

(4) 弹出一个工程信息对话框，如图 1-5 所示。该对话框中显示，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。

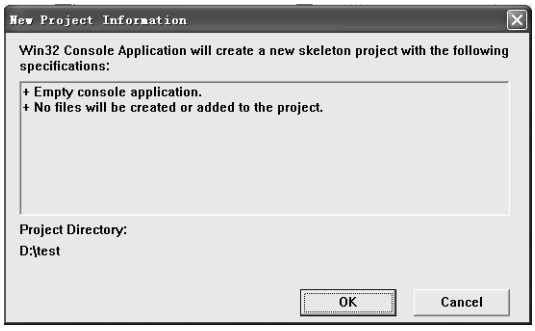

图 1-5　工程信息对话框

(5)回到 Visual C++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 1-6 所示。

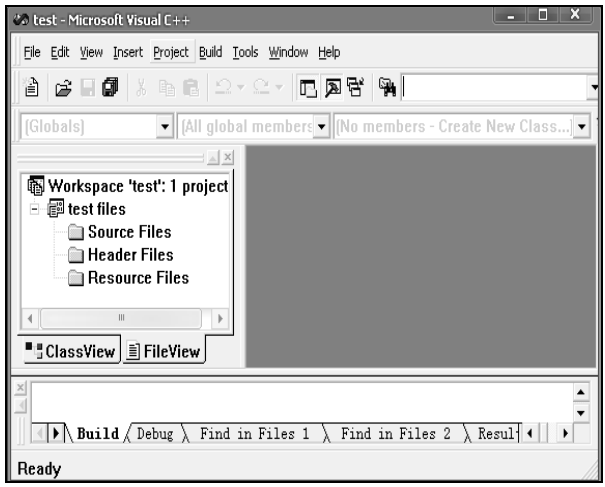

图 1-6　工程创建完成

2．编辑

完成以上操作之后，便可以进行代码的编写了。Visual C++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 1-7 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 Visual C++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时何种语言书写的，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 1-7 所示，

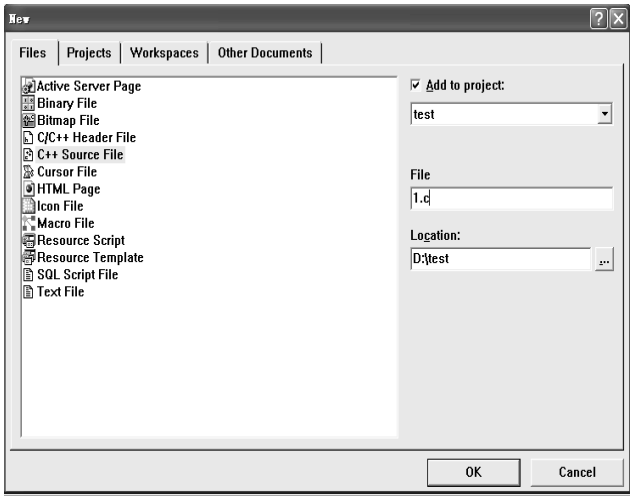

图 1-7　添加 C 源文件

此时，就可以直接在代码编辑区进行程序设计了，如图 1-8 所示，输入了一个简单程序的代码，其功能是要输出字符串“How are you？”。

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。

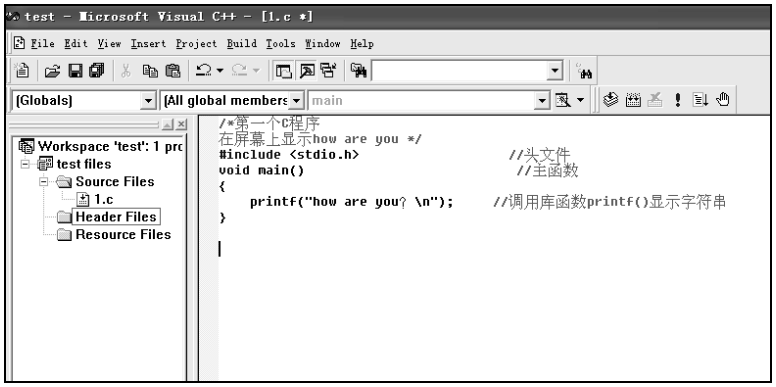

图 1-8　编辑状态的 Visual C++6.0

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)使用快捷键 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，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 1-9 所示。

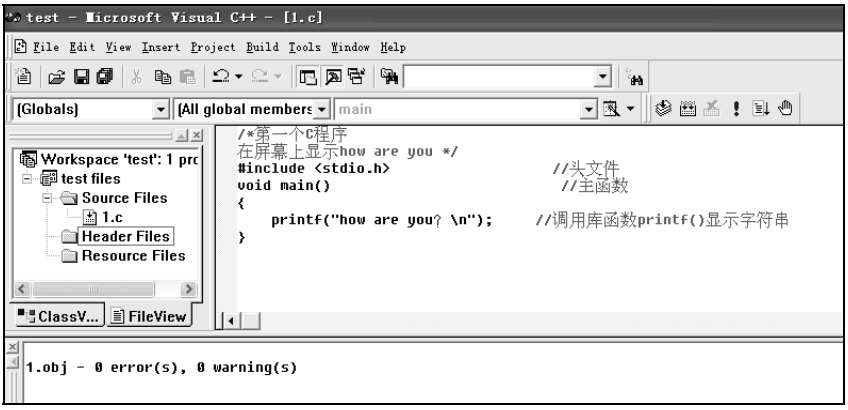

图 1-9　编译状态的 Visual C++6.0

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。Visual C++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 1-10 所示。

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 命令或按组合键

Ctrl+F5 来运行程序，也可以单击工具栏中的按钮。Visual C++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 Visual C++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 1-11 所示。

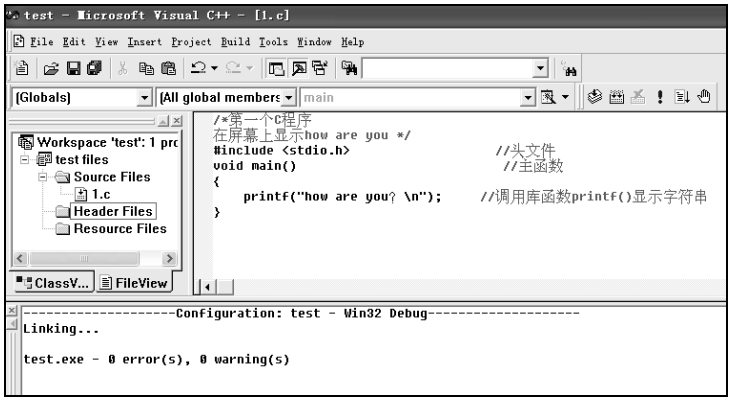

图 1-10　链接状态的 Visual C++6.0

图 1-11　运行结果

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。

1.4　C/C++应用程序的调试

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，程序中可能存在算法和逻辑上的错误，如果运行结果不符合预期或者被证明错误，我们可以利用 Visual C++6.0 提供的调试功能帮助查找错误。下面先来认识一下 Visual C++6.0 的调试菜单，然后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(见图 1-12)。

调试菜单中包含了一些与程序调试有关的命令，在前面有关菜单的内容里已经详细介绍，这里不再赘述。

(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，这时会出现一个红点，如图 1-13 所示，此红点就是断点，最后单击按钮，进行调试。

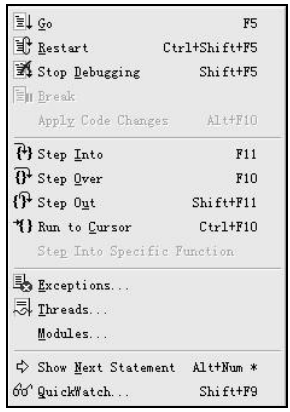

图 1-12　调试菜单

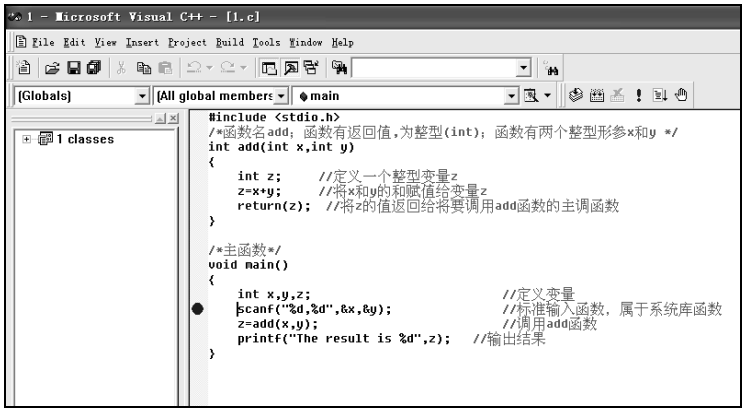

图 1-13　在某个被调试的程序中设置断点

这时程序会在断点处停下来，如图 1-14 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，我们可以看到每步变量的值，来判断程序的运行状况。

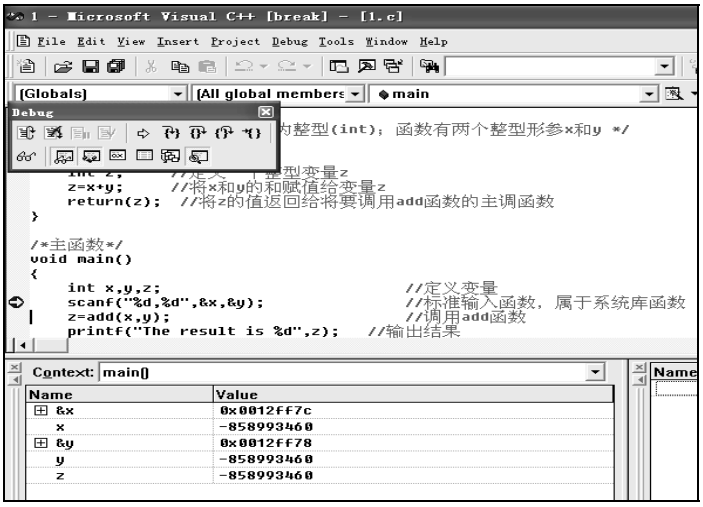

图 1-14　断点停留处

(3)分步执行

通过快捷键 F10 或从菜单栏中选择 Build | Start Debug | Step Over 命令跳到下一步，如图 1-15 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，就需要按快捷键 F11。

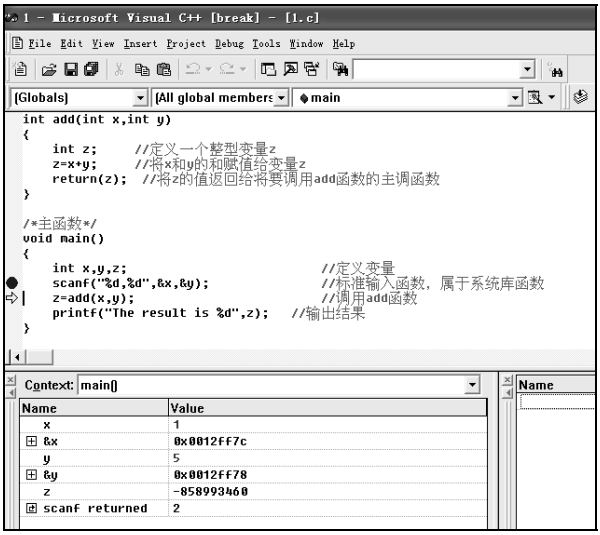

图 1-15 分步执行

(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。程序修改结束，需要取消断点。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。

1.5 实 验 内 容

1．创建一个 Win32 Console Application 类型的空白项目，项目以学号命名。编写一个 C 语言源程序添加到刚才的项目中，要求程序的运行结果是在屏幕上显示“Welcome to C Programming！”。

2．查找下面程序的错误，使之能通过编译成功运行：

```
#include "stdio.h"
void main();
{
    Printf"(hello world)";
}}
```