

# C 语言上机操作指导

程序设计是实践性很强的过程，任何程序最终都必须在计算机上运行，以检验程序的正确与否。因此在学习程序设计中，一定要重视上机实践环节，通过上机可以加深理解 C 语言的有关概念，以巩固理论知识，另一方面也可以培养程序调试的能力与技巧。

## 1 . C语言程序的上机步骤

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、数字及其它符号等构成，在计算机内部用相应的 ASCII 码表示，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 码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、运行步骤可以用图 A-1 表示：

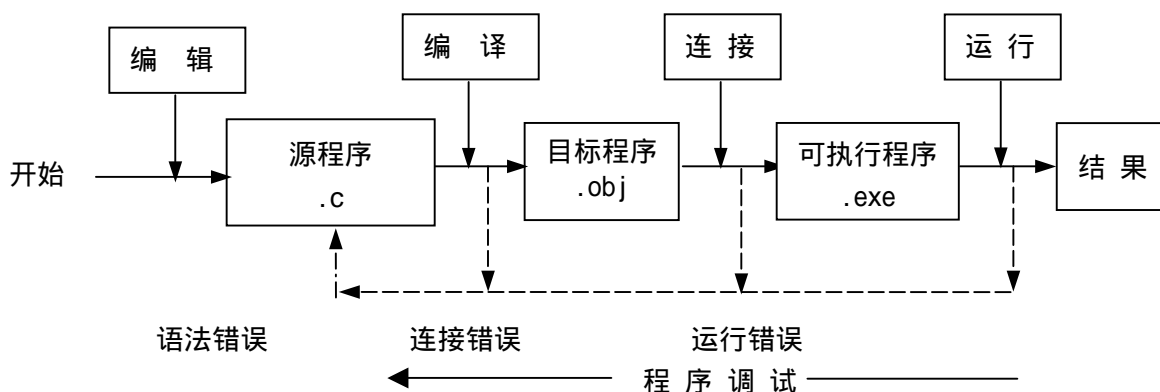


图 A-1 C 语言程序的调试、运行步骤

图 A-1 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，还是运行结果不对（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。

除了较简单的情况，一般的程序很难一次就能做到完全正确。在上机过程中，根据出错现象找出错误并改正称为程序调试。我们要在学习程序设计过程中，逐步培养调试程序

的能力，它不可能靠几句话讲清楚，要靠自己在上机中不断摸索总结，它可以说是一种经验积累。

程序中的错误大致可分为三类：

- ◇ 程序编译时检查出来的语法错误；
- ◇ 连接时出现的错误；
- ◇ 程序执行过程中的错误。

编译错误通常是编程者违反了 C 语言的语法规则，如保留字输入错误、大括号不匹配、语句少分号等等。连接错误一般由未定义或未指明要连接的函数，或者函数调用不匹配等因素引起，对系统函数的调用必须要通过“include”说明。

对于编译连接错误，C 语言系统会提供出错信息，包括出错位置（行号）、出错提示信息。编程者可以根据这些信息，找出相应错误所在。有时系统提示的一大串错误信息，并不表示真的有这么多个错误，往往是因为前面的一两个错误带来的。所以当你纠正了几个错误后，不妨再编译连接一次，然后根据最新的出错信息继续纠正。

有些程序通过了编译连接，并能够在计算机上运行，但得到的结果不正确，这类在程序执行过程中的错误往往最难改正。错误的原因一部分是程序书写错误带来的，例如应该使用变量 x 的地方写成了变量 y，虽然没有语法错误，但意思完全错了；另一部分可能是程序的算法不正确，解题思路不对。还有一些程序有时计算结果正确，有时不正确，这往往是编程时，对各种情况考虑不周所致。解决运行错误的首要步骤就是错误定位，即找到出错的位置，才能予以纠正。通常我们先设法确定错误的大致位置，然后通过 C 语言提供的调试工具找出真正的错误。

为了确定错误的大致位置，可以先把程序分成几大块，并在每一块的结束位置，手工计算一个或几个阶段性结果，然后用调试方式运行程序，到每一块结束时，检查程序运行的实际结果与手工计算是否一致，通过这些阶段性结果来确定各块是否正确。对于出错的程序块，可逐条仔细检查各语句，找出错误所在。如果出错块程序较长，难以一下子找出错误，可以进一步把该块细分层更小的块，按照上述步骤进一步检查。在确定了大致出错位置后，如果无法直接看出错误，可以通过单步运行相关位置的几条语句，逐条检查，一定能找出错误的语句。

当程序出现计算结果有时正确有时不正确的情况时，其原因一般是算法对各种数据处理情况考虑不全面。解决办法最好多选几组典型的输入数据进行测试，除了普通的数据外，还应包含一些边界数据和错误的输入数据。比如确定正常的输入数据范围后，分别以最小值、最大值、比最小值小的值和比最大值大的值，多方面运行检查自己的程序。

下面我们分别以 TC2.0 和 VC++6.0 为上机平台，对 C 程序编译、连接和调试作简单介绍。建议一开始学习上机时，把注意力放在程序的编译、连接和运行，已能运行为目标，而把调试部分放到学习了第 5 章后再看，只有具有一定的程序语句量，调试才有作用。

## 2. Turbo C 语言集成环境

Turbo C 是一个常用的、最基本的 C 语言工具，一般简称 TC。它为 C 语言开发提供了操作便利的集成环境。源程序的输入、修改、调试及运行都可以在 TC 集成环境下完成，

非常方便有效。TC2.0 系统非常小巧，但功能齐全。它主要支持 DOS 环境，因此在操作中无法使用鼠标，更多的需要通过键盘操纵菜单或快捷键完成。这也成为使用 TC 的一大遗憾。

## 2.1 TC 启动

由于 TC 支持 DOS 环境，TC 系统的安装十分方便。如果有安装盘，可以按照提示一步步完成安装；如果没有安装盘，可以从其他机器直接拷贝已安装好的系统。

由于 TC 是在 DOS 环境下工作的，无法象 Windows 应用程序那样，通过“开始”菜单的程序项运行。首先要找到 TC 系统的安装（复制）目录，双击其上的 tc.exe 应用程序，可以启动 TC 系统，其界面如图 A-2 所示。



图 A-2 TC 界面

图 A-2 中，菜单包含了所有操作的功能；编辑窗口是用于输入、修改程序的区域；信息窗口将显示程序编译、连接和运行过程中的错误信息或有关提示信息；快捷热键提示将给出常用操作的快捷键提示信息，以方便用户的操作。

## 2.2 运行程序

要运行一个 C 程序，必须经过：输入源程序、修改错误、编译连接和运行几个步骤。

### (1) 编辑

在编辑窗口中，直接输入程序。如果要进行修改，可以使用【↑】、【↓】、【←】、【→】四个方向键，移动光标到所需位置，然后删除错误，输入正确的内容。

### (2) 编译、连接和运行

同时按下【Ctrl】+【F9】键，将对编辑窗口中的程序，完成编译、连接和运行三个步骤。

- 如果程序没有错误，将直接运行程序；

- 如果存在编译错误，信息窗口中将显示错误信息，并终止连接与执行步骤；
- 如果编译通过，但存在连接错误，信息窗口中将显示错误信息，并终止执行步骤。

任何错误都必须纠正后，重新按【Ctrl】+【F9】键运行。如果还有错，继续修改，直到能正确执行为止。

即使通过了编译和连接，并不说明程序就没有错误了，解题思路错误或语句的错误使用（语句格式没有错），都会导致无法得到正确的结果，甚至程序无法正确执行。如果程序在执行过程中无法结束（死机），可以按【Ctrl】+【break】强制结束。（【break】位于键盘最右上角——功能键）。

### （3）输入数据

如果程序中有 scanf() 语句，则屏幕将出现一个黑底的输入窗口，等待输入数据，通常是输入一个数据，打一个空格或回车，然后输入下一个数据。直到输入完所有的数据。如果 scanf() 语句中格式规定是逗号分隔，则各数据之间要输入逗号。

### （4）查看结果

输入完数据，系统将自动关闭输入窗口，运行程序后，回到编辑窗口。但运行结果并未在屏幕上显示，若要查看运算结果，需要按【Alt】+【F5】才出现输出窗口。看完后，再按任意键将关闭输出窗口。

对于没有输入要求的程序，按下【Ctrl】+【F9】运行，屏幕上不会有什么变化，但这并不表示程序未被运行，按【Alt】+【F5】就能看到结果。

通过上述步骤，读者可以快速掌握 C 语言的上机过程，运行自己的程序。但还只能处理一些简单的问题，如果想更有效地调试运行 C 程序，下面的内容是必不可少的。

### （5）运行环境设置

第一次上机时，如果一个简单正确的程序却无法运行，需要检查一下运行环境是否设置正确。具体参看下面“TurboC 菜单”中介绍的“Option”菜单项。

## 2.3 Turbo C 菜单

前面介绍的是上机过程中的最基本的一些操作。为了对 TC 的功能有一个全面的了解，我们将对各主要菜单项分别予以介绍。

由于 TC 环境下不能使用鼠标，对打开菜单栏有两种方法：

- ◇ 按【F10】，然后按【 】或【 】选择相应菜单位置，再按回车键【Enter】。
- ◇ 按【Alt】+【菜单上红色大写字母】

当选中一个菜单后，会弹出下拉菜单项，通过【 】或【 】选择相应菜单项，再按回车键【Enter】，完成菜单功能操作。按【Esc】可以取消菜单选择（不是撤销菜单功能）。

TC 中提供了完善的帮助信息，按【F1】屏幕上将弹出帮助窗口。但其中的信息都是英文的。

TC 的快捷键通常是组合键，如【Ctrl】+【F9】表示要同时按下两个键。对于三键的组合键，其前两个键要同时按下，松手后再按第三个键，如【Ctrl】+【k】+【b】表示先同时按下【Ctrl】键和【k】键，松手后再按【b】键。

#### （1）File 菜单（文件操作）

- Load (【F3】): 调入一个已存在的程序文件（扩展名为 c）。【F3】是快捷键。

- Pick (【Alt】+【F3】): 从最近曾经调入过的文件中选择一个调入。
- New: 清除编辑窗口中的程序, 供输入一个新程序。
- Save (【F2】): 把编辑窗口中的程序保存到文件中。如果该程序已经保存过, 该操作将更新文件内容; 如果该程序是新输入的, 需要进一步输入文件名称或路径。
- Write to: 把当前程序写到另外命名的文件上, 相当于 Windows 文件菜单中的“另存为”功能。
- Quit (【Alt】+【x】): 退出 TC 系统。

## (2) Edit (编辑操作)

它没有下拉菜单, 按回车键【Enter】直接进入编辑窗口, 在编辑过程中, 一些常用的编辑功能依靠快捷键实现。

- 【Ctrl】+【y】: 删除光标所在行的整行信息。
- 【Ctrl】+【k】+【b】: 把光标所在位置定义为块信息的头部。
- 【Ctrl】+【k】+【k】: 把光标所在位置定义为块信息的尾部。块头部定义与块尾部定义要按顺序配合起来使用, 所定义的块变成白底蓝字。
- 【Ctrl】+【k】+【h】: 取消所定义的块信息。若重新定义新块也会取消原定义块。
- 【Ctrl】+【k】+【c】: 把定义的块信息复制到光标所在位置。
- 【Ctrl】+【k】+【v】: 把定义的块信息移动到光标所在位置。
- 【Ctrl】+【q】+【f】: 查找特定字符, 它会在屏幕上部提示输入查找字符, 如图 A-3 所示, 当输入了“main”后, 屏幕上出现 Option 信息, 要求输入查找方式:

- ✧ G 对整个文件进行查找
- ✧ 回车 从光标当前位置向后查找

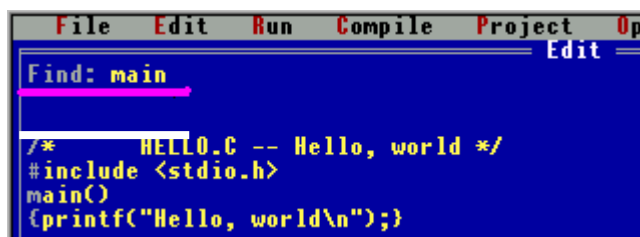


图 A-3 字符查找

- 【Ctrl】+【q】+【a】: 替换字符串。它在上一步查找的过程中, 再输入替换字符 Replace。
  - 【Ctrl】+【L】: 重复上一次查找或替换。
  - 【Ctrl】+【q】+【[】: 对光标所处位置的“(、[ 或 {”定位相应的“}、] 或 )”。
- 这在程序中检查三种括号是否匹配十分有用。

在“记事本”中选中一块文本, 使用鼠标能很方便地实现。而在 TC 中由于无法使用鼠标, 只能通过键盘进行, 我们需要定义文本块的开始位置与结束位置, 当选中后, 文字变成白底蓝字, 然后才能实现文本块的复制和移动, 步骤与“记事本”中相似。因此上述 5 个操作需要配合使用。

### (3) Compile 菜单 (编译连接操作)

- Compile (【Alt】+【F9】): 把编辑窗口中的程序编译成目标文件。
- Link: 把编辑窗口中的程序连接成可执行文件。
- Make (【F9】): 把编辑窗口中的程序经编译、连接, 生成可执行文件。

### (4) Run 菜单 (运行操作)

- Run (【Ctrl】+【F9】): 执行编辑窗口中的程序。如果该程序最近未编译连接过, 将先自动编译连接, 然后再执行。

### (5) Option 菜单 (建立工作环境)

- 工作环境目录的设置: 如果使用安装盘安装, 该工作环境目录会自动设置好。如果是通过系统复制的, 则需要对“Option”菜单的“Directories”菜单项进行设置。假设 TC 所在目录为: “D:\TC”, 在打开的目录窗口中应填入:

Include directories: D:\TC\INCLUDE

Library directories: D:\TC\LIB

以确保程序连接时能从这两个位置找到系统包含文件和系统库文件。如图 A-4 所示。



图 A-4 TC 工作目录设置

- 命令行参数输入: 在学习了第 8 章“指针”后, 会用到命令行参数。可以执行“Option”菜单的“Arguments”菜单项, 输入命令行参数 (不包括可执行文件名, 各参数用空格分隔), 回车结束输入, 按【Esc】隐去菜单, 然后可按【Ctrl】+【F9】运行程序, 参数便能被主函数接受。

## 2.4 窗口操作

在图 A-2 的 TC 界面下, 屏幕上半部分是编辑窗口, 下半部分是信息窗口。如果想把编辑窗口扩大到整屏, 可按【F5】键, 这时信息窗口将被遮住, 再按【F5】键又可以恢复成上下两个窗口。因此【F5】键就象一个电源开关。如果编辑窗口被扩大到整屏, 而又想看一下信息窗口, 可使用【F6】进行窗口切换。如果在程序执行时又开了观察窗口, 【F6】可以对三个窗口进行切换, 切换过程是按一个方向循环。

在 Windows 中运行 TC, 所打开的窗口往往较小, 边框线也不对。按【Alt】+【Enter】

键可以使窗口最大化，成为仿真 Dos 界面。再按【Alt】+【Enter】又会恢复较小窗口。

## 2.5 程序调试

TC 提供了必要的调试手段和工具，下面按照使用过程予以介绍。

### (1) 让程序执行到中途暂停以便观察阶段性结果：

方法一：使程序执行到光标所在的那一行暂停。

把光标移动到需暂停的行上；

按【F4】或执行菜单 Run 中的“Go to Cursor”操作。当程序执行到该行将会暂停。如果把光标移动到后面的某个位置，再按【F4】，程序将从当前的暂停点继续执行到新的光标位置，第二次暂停。

方法二：把光标所在的那一行设置成断点，然后按【Ctrl】+【F9】执行，当程序执行到该行将会暂停。设置断点的步骤为：

把光标移动到需暂停的行上；

按【Ctrl】+【F8】或执行菜单“Break/watch”中的“Toggle breakpoint”操作。

不管是通过光标位置还是断点设置，其所在的程序行必须是程序执行的必经之路，亦即不应该是分支结构中的语句，因为该语句在程序执行中受到条件判断的限制，有可能因条件的不满足而不被执行。这时程序将一直执行到结束位置或下一个断点位置。

### (2) 设置需观察的结果变量

按照上面的操作，可使程序执行到指定位置时暂停，其目的是为了查看有关的中间结果。按【Ctrl】+【F7】或菜单“Break/watch”中的“Add watch”操作，屏幕上将会弹出小窗口供输入查看变量，如图 A-5 所示，我们输入了变量 i 进行查看。

对于图 A-5 中的例子，我们先把光标移动到第五行，然后按【F4】执行，程序到第五行暂停，如图 A-6 所示，查看 (Watch) 窗口中就会显示查看变量 i 的当前值。绿色光条表示当前将被执行的程序位置 (或暂停位置)。

多次使用【Ctrl】+【F8】可增加多个新的查看变量，如果想改变查看变量的名字或删除查看变量，可以按【F6】，使查看窗口成为操作窗口，然后按回车键【Enter】，可以改变查看变量，按删除键【Delete】可以删除查看变量。这些菜单功能分别在“Break/watch”中。



图 A-5 输入查看变量

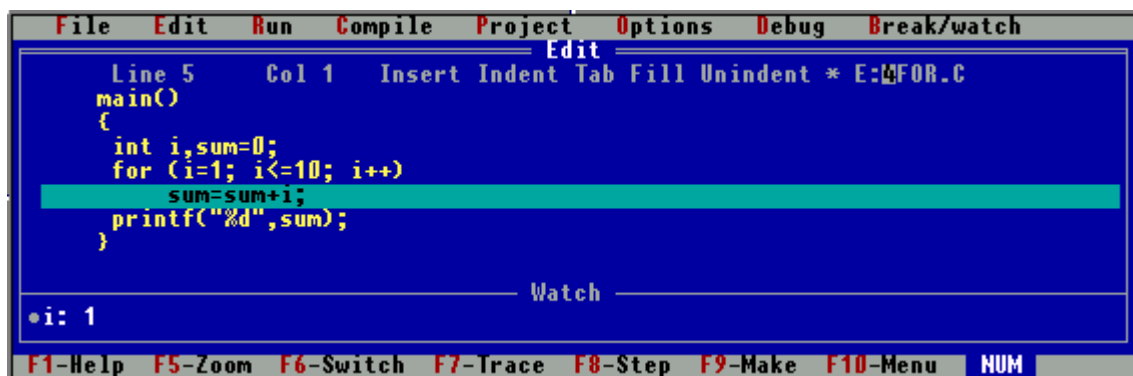


图 A-6 查看中间结果

### (3) 单步执行

当程序执行到某个位置时发现结果已经不正确了，说明在此之前肯定有错误存在。如果能确定一小段程序可能有错，先按上面步骤暂停在该小段程序的头一行，再输入若干个查看变量，然后单步执行，即一次执行一行语句，逐行检查下来，看看到到底是哪一行造成结果出现错误，从而能确定错误的语句并予以纠正。

单步执行按【F8】或执行菜单 Run 中的“Step over”操作。如果遇到自定义函数调用，想进入函数进行单步执行，可按【F7】或执行菜单 Run 中的“Trace into”操作。对不是函数调用的语句来说，【F7】与【F8】作用相同。但一般对系统函数不要使用【F7】。

### (4) 断点的使用

使用断点也可以使程序暂停。但一旦设置了断点，不管你是否还需要调试程序，每次执行程序都会在断点上暂停。因此调试结束后应取消所定义的断点。方法是先把光标定位在断点所在行，再按【Ctrl】+【F8】或执行菜单“Break/watch”中的“Toggle breakpoint”操作，该操作是一个开关，第一次按是设置，第二次按是取消设置。被设置成断点的行将呈红色背景。如果有多个断点想一下全部取消，可执行菜单“Break/watch”中的“Clear all breakpoints”操作。



断点通常用于调试较长的程序，可以避免使用【F4】（运行程序到光标处暂停）功能时，需要经常把光标定位到不同的地方。而对于长度为上百行的程序，要寻找某位置并不太方便。

如果一个程序设置了多个断点，按一次【Ctrl】+【F9】会暂停在第一个断点，再按一次【Ctrl】+【F9】会继续执行到第二个断点暂停，依次执行下去。

#### （5）结束调试

TC 中通过“结束程序运行”（Program reset）来结束程序调试。按【Ctrl】+【F2】或执行菜单 Run 中的“Program reset”操作实现。

#### （6）循环的调试实例

下面用单步执行功能来看一下 for 语句的执行流程。如图 A-7 所示。程序中把 for(.....) 分成三行，以便观察执行流程。先把光标移动到第四行，然后按【F4】或执行菜单 Run 中的“Go to Cursor”操作，按【Ctrl】+【F7】输入查看变量 i，由于此时 i 未赋过值，所以显示的是一个随机数。再连续按【F8】单步执行，可以观察绿色光条的位置变动和查看变量 i 的变化，绿色光条的位置变动就是程序执行的过程，学员可以充分体会到 for 语句的执行流程。

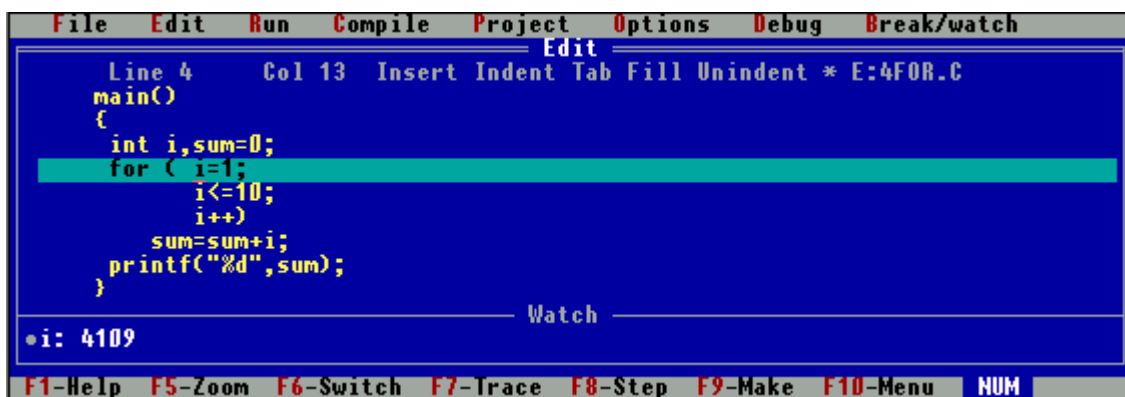


图 A-7 循环调试实例

通过这个例子，读者可以举一反三调试自己的程序了。

上面我们只对 TC 中主要的功能作了介绍，对于其他的操作读者可以自己试验，或参考有关 Turbo C 手册。

## 2.6 常用快捷键小结

【F2】：保存程序	【Ctrl】+【F2】：结束程序调试运行
【F3】：调入程序文件	【Alt】+【F3】：调入最近曾经用过的文件
【F4】：程序运行到光标处暂停	【F5】：放大/缩小窗口
【Alt】+【F5】：查看运行结果	【F6】：窗口切换
【F7】：单步执行（可进入函数）	【Ctrl】+【F7】：增加查看变量
【F8】：单步执行（不能进入函数）	【Ctrl】+【F8】：把光标所在行设为断点或取消断点

【F9】：编译、连接程序

【Alt】+【F9】：编译程序

【Ctrl】+【F9】：编译、连接、执行程序

【F10】：菜单选择

【F1】：帮助

【Alt】+【x】：退出系统

### 3 Visual C 语言集成环境

C++语言是在 C 语言的基础上发展而来，它增加了面向对象的编程，成为当今最流行的一种程序设计语言。Visual C++是微软公司开发的，面向 Windows 编程的 C++语言工具。它不仅支持 C++语言的编程，也兼容 C 语言的编程。由于 VC++被广泛地用于各种编程，使用面很广。这里简要地介绍如何在 VC++下运行 C 语言程序。

#### 3.1 启动 VC++

VC++是一个庞大的语言集成工具，经安装后将占用几百兆磁盘空间。从“开始”——“程序”——“Microsoft Visual Studio 6.0”——“Microsoft Visual C++ 6.0”，可启动 VC++，屏幕上将显示图 A-8 所示的窗口。

#### 3.2 新建/打开 C 程序文件

选择“文件”菜单的“新建”菜单项，单击如图 A-9 所示的“文件”标签，选中“C++ Source File”，按“确定”。然后在编辑窗口中输入程序。

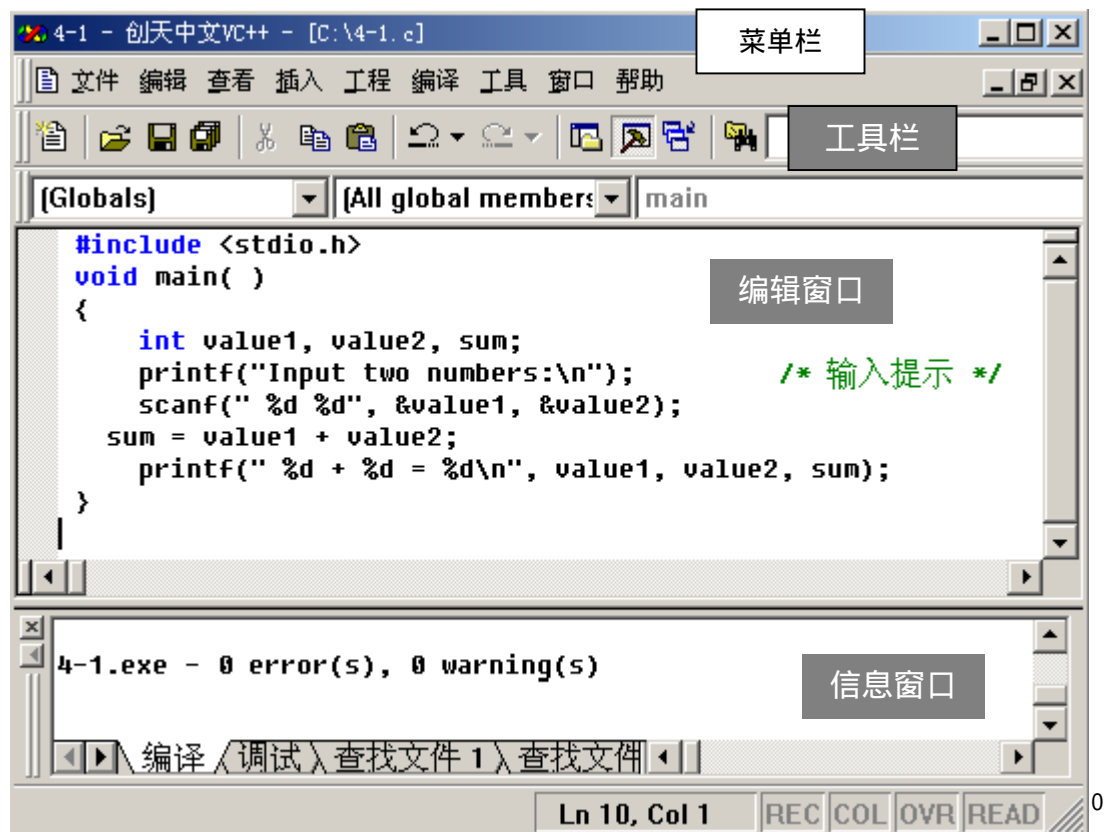


图 A-8 VC++窗口

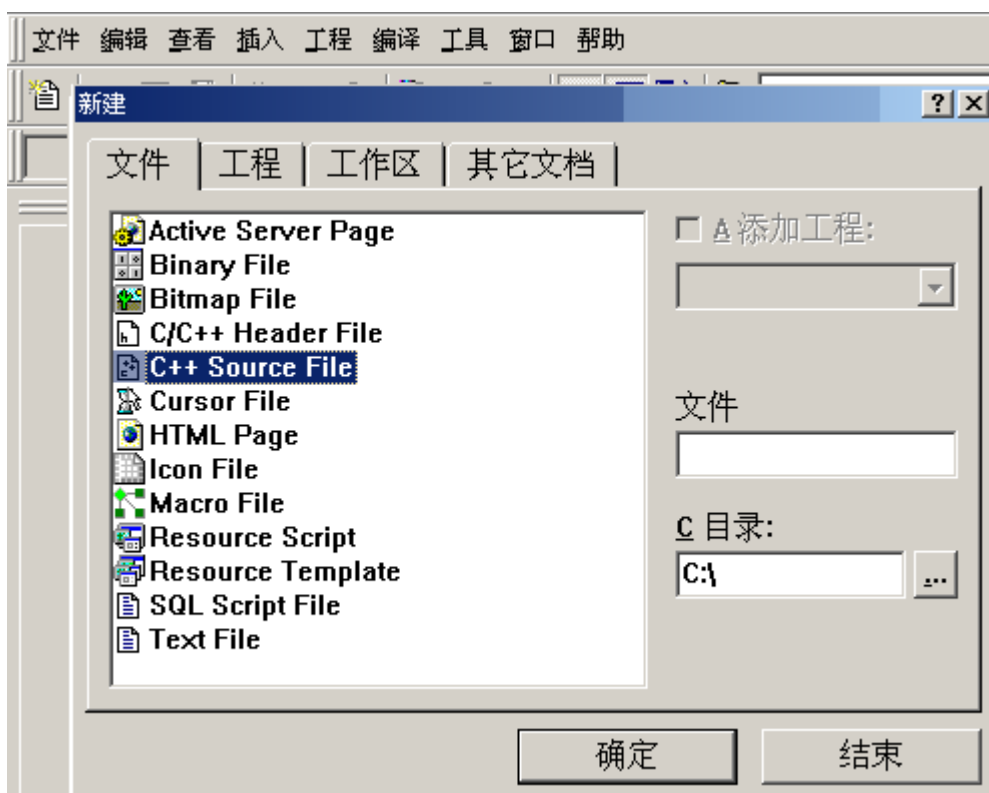


图 A-9 新建文件

如果程序已经输入过，可选择“文件”菜单的“打开”菜单项，并在查找范围中找到正确的文件夹，调入指定的程序文件。

### 3.3 程序保存

在打开的 VC++ 界面上，可直接在编辑窗口输入程序，由于完全是 Windows 界面，输入及修改可借助鼠标和菜单进行，十分方便。当输入结束后，保存文件时，因指定扩展名“.C”，否则系统将按 C++ 扩展名“.CPP”保存。如图 A-10 所示。

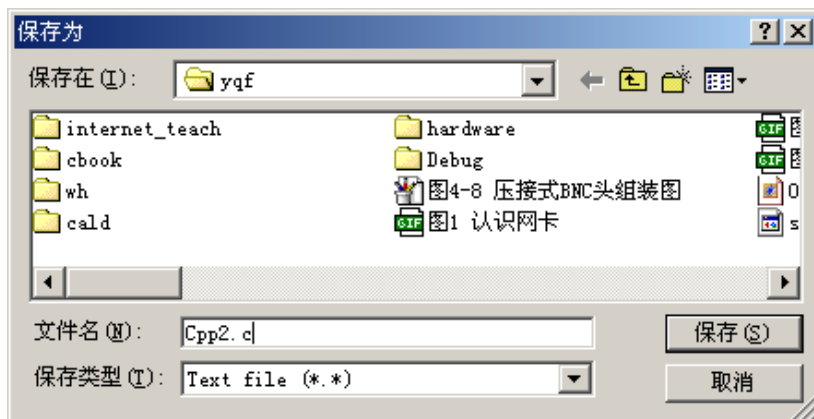


图 A-10 指定保存文件名

### 3.4 执行程序

首先要生成可执行文件。使用 VC++ “编译” 菜单中的“构件”菜单项，如图 A-11 所示，也可使用快捷键【F7】。在编译连接过程中 VC++ 将保存该新输入的程序，并生成一个同

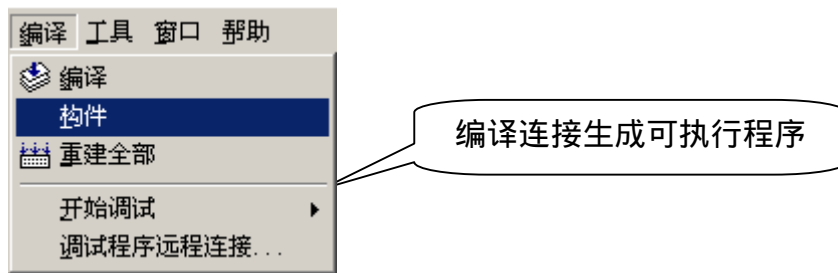


图 A-11 编译连接菜单

名的工作区。保存文件时须填入文件名，如“4-1.C”。假如不指定扩展名.C，VC++ 会把扩展名定义为.CPP，即 C++ 程序。如果程序没有错误，将在图 A-12 信息窗口中显示内容：

0 error(s) 0 warning(s)

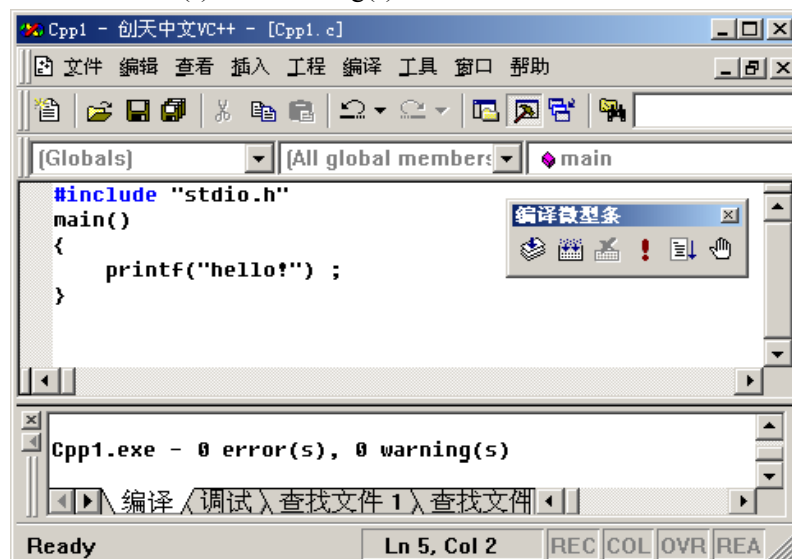


图 A-12 编译连接正确

表示没有任何错误。有时出现几个警告性信息(warning)，不影响程序执行。假如有致命性错误 (error)，如图 A-13 所示，双击某行出错信息，程序窗口中会指示对应出错位置，根据信息窗口的提示分别予以纠正。然后用“编译”菜单中的“执行”菜单项(或快捷键【Ctrl】+【F5】)执行程序。

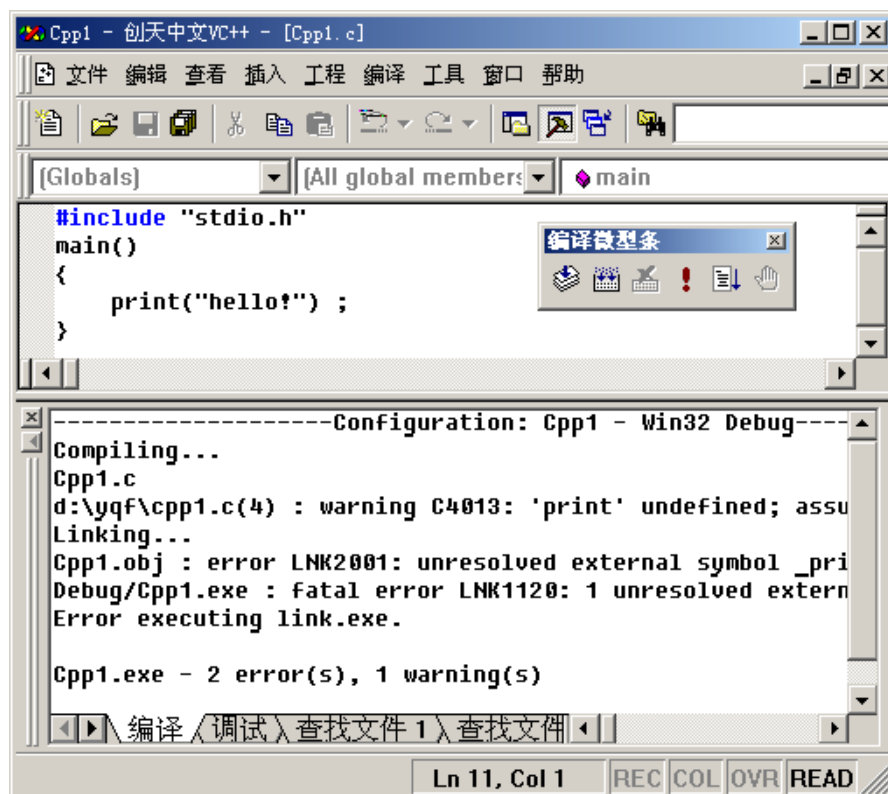


图 A-13 编译连接出错

当运行 C 程序后，VC++ 将自动弹出数据输入输出窗口，如图 A-14 所示。按任意键将关闭该窗口

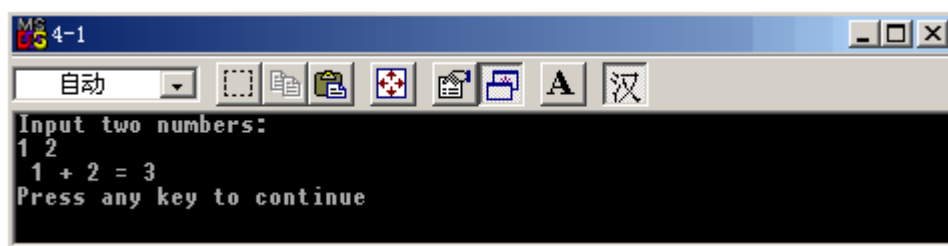


图 A-14 数据输入输出窗口

对于编译连接执行操作，VC++ 还提供了一组工具按钮，如图 A-15 所示。



图 A-15 编译连接执行工具按钮组

### 3.5 关闭程序工作区

当一个程序编译连接后，VC++系统自动产生相应的工作区，以完成程序的运行和调试。若想执行第二个程序时，必须关闭前一个程序的工作区，然后通过新的编译连接，产生第二个程序的工作区。否则的话运行的将一直是前一个程序。

“文件”菜单提供关闭程序工作区功能，如图 A-16(a)，执行“关闭工作区”菜单功能，然后在如图 A-16(b)对话框中选择“否”。如果选择“是”将同时关闭源程序窗口。

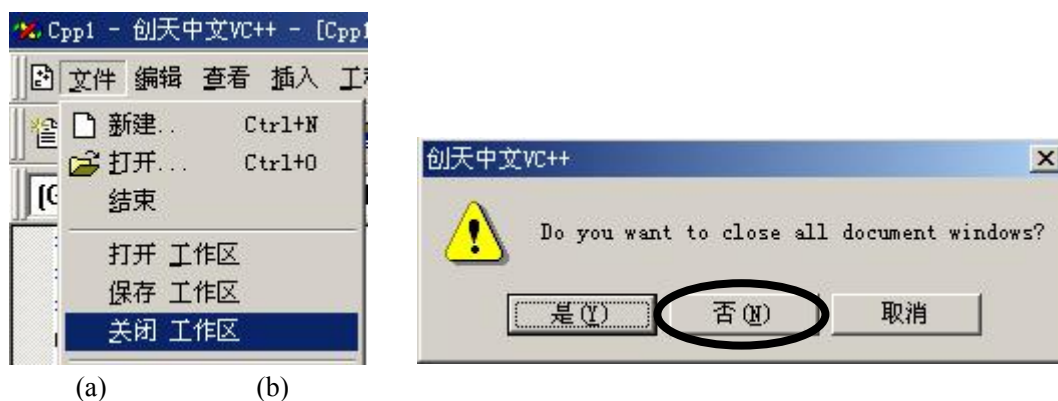


图 A-16 关闭程序工作区

### 3.6 命令行参数处理

VC++是一个基于窗口操作的 C++系统，没有提供命令行参数功能。我们需要在 Windows 的“MS-DOS 方式”窗口里以命令方式实现。具体步骤参考如下：

- (1) 正确编译连接，生成可执行程序；
- (2) 通过“我的电脑”或“资源管理器”找到所运行的 C 源程序（设为 a.c）；
- (3) 进入 debug 文件夹（它包含 a.c 程序的可执行文件 a.exe）；
- (4) 执行“开始”菜单的“运行”，填入 command，然后“确定”；
- (5) 在打开的“MS-DOS 方式”窗口中输入：a 参数 1 参数 2 .....，带参数运行程序。

### 3.7 程序调试

VC++是一个完全基于 Windows 的系统，它的调试过程通过鼠标比较容易进行。

- (1) 程序执行到中途暂停以便观察阶段性结果：

方法一：使程序执行到光标所在的那一行暂停。

在需暂停的行上单击鼠标，定位光标；

如图 A-17 所示，分别点击菜单“编译”——“开始调试”——“Run to Cursor”，或按【Ctrl】+【F10】，程序将执行到光标所在行会暂停。如果把光标移动到后面的某个位置，再按【Ctrl】+【F10】，程序将从当前的暂停点继续执行到新的光标位置，第二次暂停。

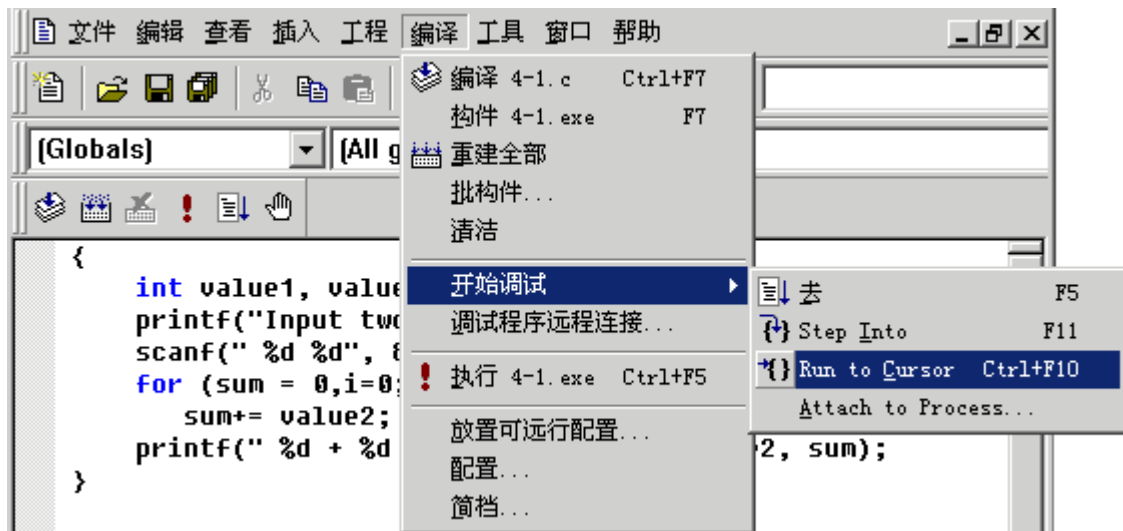


图 A-17 执行到光标所在行暂停

方法二：在需暂停的行上设置断点；  
 在需设置断点的行上单击鼠标，定位光标；  
 按“编译微型条”中最右面的按钮，如图 A-18 所示，或按【F9】。

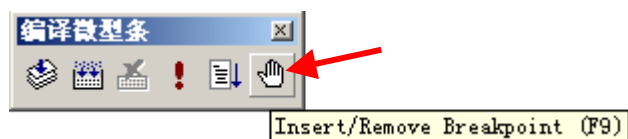


图 A-18 设置断点

被设置了断点的行前面会有一个红色圆点标志。

与 TC 一样，不管是通过光标位置还是断点设置，其所在的程序行必须是程序执行的必经之路，亦即不应该是分支结构中的语句，因为该语句在程序执行中受到条件判断的限制，有可能因条件的不满足而不被执行。这时程序将一直执行到结束或下一个断点为止。

## (2) 设置需观察的结果变量

按照上面的操作，使程序执行到指定位置时暂停，目的是为了查看有关的中间结果。在如图 A-19 中，左下角窗口中系统自动显示了有关变量的值，其中 value1 和 value2 的值分别是 3、4，而变量 i、sum 的值是不正确的，因为他们还未被赋值。图中左侧的箭头表示当前程序暂停的位置。如果还想增加观察变量，可在图中右下角的“Name”框中填入相应变量名。

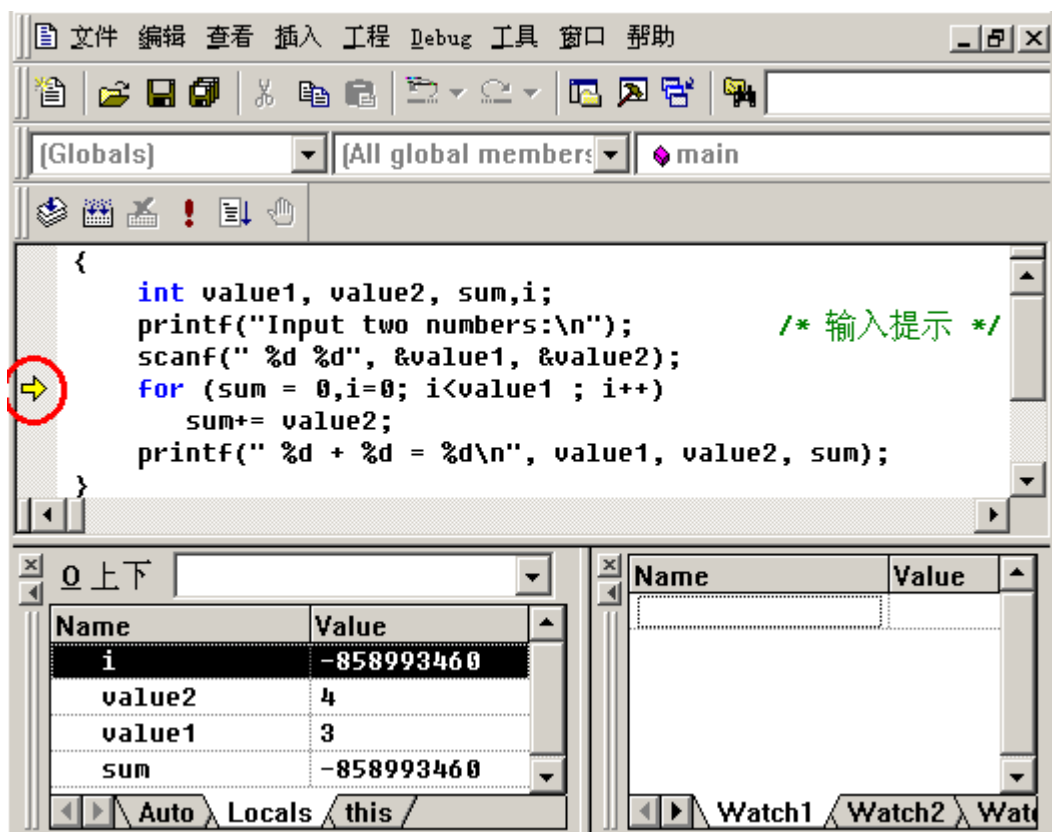





图 A-19 观察结果变量

### (3) 单步执行

当程序执行到某个位置时发现结果已经不正确了，说明在此之前肯定有错误存在。如果能确定一小段程序可能有错，先按上面步骤暂停在该小段程序的头一行，再输入若干个查看变量，然后单步执行，即一次执行一行语句，逐行检查下来，看看到底是哪一行造成结果出现错误，从而能确定错误的语句并予以纠正。

单步执行按“调试条”中“Step Over”按钮  或【F8】，如图 A-20 所示。如果遇到自定义函数调用，想进入函数进行单步执行，可按“Step Into”按钮  或【F11】。当想结束函数的单步执行，可按“Step Out”按钮  或【Shift】+【F11】。对不是函数调用的语句来说，【F11】与【F8】作用相同。但一般对系统函数不要使用【F11】。

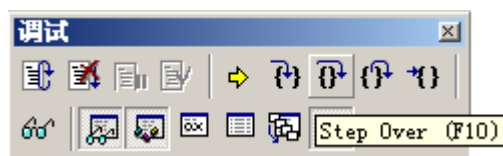



图 A-20 单步调试

### (4) 断点的使用



使用断点也可以使程序暂停。但一旦设置了断点，不管你是否还需要调试程序，每次执行程序都会在断点上暂停。因此调试结束后应取消所定义的断点。方法是先把光标定位在断点所在行，再按“编译微型条”中最右面的按钮或【F9】，该操作是一个开关，按一次是设置，按二次是取消设置。如果有多个断点想全部取消，可执行“编辑”菜单中的“断点”菜单项，屏幕上会显示“Breakpoints”窗口，如图 A-21 所示，窗口下方列出了所有断点，按“Remove All”按钮，将取消所有断点。

断点通常用于调试较长的程序，可以避免使用“Run to Cursor”(运行程序到光标处暂停)或【Ctrl】+【F10】功能时，经常要把光标定位到不同的地方。而对于长度为上百行的程序，要寻找某位置并不太方便。

如果一个程序设置了多个断点，按一次执行键【Ctrl】+【F5】会暂停在第一个断点，再按一次【Ctrl】+【F5】会继续执行到第二个断点暂停，依次执行下去。

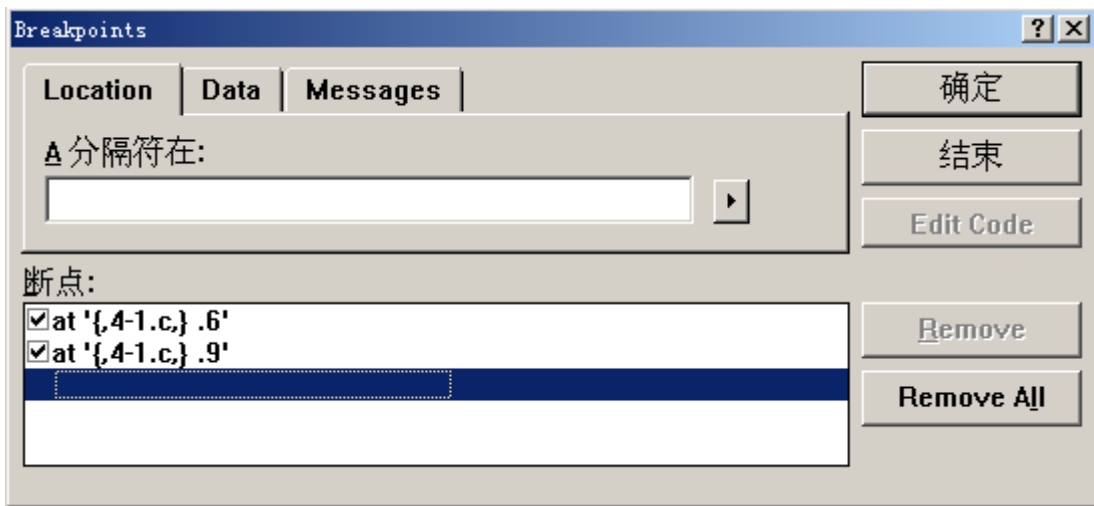


图 A-21 取消所有断点

#### (5) 停止调试

使用“Debug”菜单的“Stop Debugging”菜单项，或【Shift】+【F5】键可以结束调试，从而回到正常的运行状态。

上面我们只对 VC++ 中主要的功能作了介绍，对于其他的操作读者可以自己试验，或参考有关 Visual C++ 手册。