

1.1 C 语言的出现、发展及特点

在 20 世纪 70 年代早期,C 语言作为最初的 UNIX 操作系统的系统实现语言诞生了,它以无类型的 B 语言为基础。1963 年,剑桥大学将 ALGOL60 语言发展成为 CPL(Combined Programming Language)语言。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改,并为它起了一个有趣的名字——“B 语言”,并且他还用 B 语言写了第一个 UNIX 操作系统,在 PDP-7 上实现。1973 年,美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出一种新的语言,他取了 BCPL 的第二字母作为这种语言的名字,这就是 C 语言。

1978 年,B. W. Kernighan 和 D. M. Ritchie 出版了名著 *The C Programming Language*,从而使 C 语言成为目前世界上应用最广泛的高级程序设计语言。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准协会(ANSI)于 1983 年为 C 语言制定了一套 ANSI 标准,称为 ANSI C。1987 年,ANSI 又公布了新标准——87 ANSI C。1990 年,国际标准化组织(International Standard Organization,ISO)接受 87 ANSI C 为 ISO C 的标准(ISO 9899—1999)。目前流行的 C 编译系统都是以它为基础的。

由于 C 语言既具有低级语言的特性,又具有一般高级语言的特性,因此受到广大编程爱好者的青睐。它正以其强大的生命力在不断地发展着。

C 语言从出现、发展,到标准的制定,再到目前的备受青睐,在短短的几十年间,C 语言的特点和超越其他编程语言的优越性无不展示着它强劲的生命力。C 语言功能强大、使用方便、可移植性好等特点,使其成为备受欢迎的编程语言之一。从使用范围、功能效率等方面归纳 C 语言的特点,主要有以下几点。

1. C 语言功能强大、适用范围广

C 语言具有各种各样的数据类型,包括整型、字符型、数组类型、结构体类型、共用体类型等,并引入了指针概念,使得程序运行效率更高。

C 语言包含很广泛的运算符,共有 34 个。C 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 C 语言的运算类型极其丰富、表达式类型多样化,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。

另外,C 语言也具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以实现决策目的等。

2. C 语言可直接操作硬件

C 语言既具有高级语言的功能,又具有低级语言的许多特性,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,能够像汇编语言一样对位、字节和地址进行操作,从而实现对硬件的直接操作。在 C 语言出现之前,能对计算机硬件直接操作的是诸如汇编语言等低级语言,这使得程序的可读性和可移植都比较差。由于 C 语言具有这种双重特性,所以它既可以是通用的程序设计语言,又可以是硬件直接操作语言。

3. C 语言是结构式语言

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、维护以及调试。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,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

4. C 语言可移植性好

C 语言的可移植性较好,能适合于多种操作系统。除了能在操作系统运行外,在目前日趋流行的或者操作系统上也能不加修改地运行。C 语言的这种较好的可移植性,为开发跨平台程序提供了有力的支持。

1.2 典型的 C 语言开发环境

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。

1.2.1 C 语言程序的开发过程

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 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。C 语言程序的开发要经过编辑源程序、编译源程序、连接目标代码、运行可执行文件四个过程。

在目前众多开发平台中,如 Turbo C、Visual C++ 6.0、Dev-C++、C++ Builder (BCB)、Source Insight、Eclipse for C/C++、Code::Blocks,等等,均提供了一个全屏幕编辑器编辑 C 源程序,使用非常方便。在 VC++ 6.0 环境中的源程序文件,其扩展名为 .cpp,而在 Turbo C 2.0 和 Win TC 环境中的源程序文件的扩展名为 .c。

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个机器指令来实现,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,才可由计算机运行,最终得到结果。

编译过程主要进行词法分析和语法分析,并将分析结果提示给用户。例如,如果发现错误,错误信息将显示在屏幕上通知用户,经过编译后的目标文件的扩展名为 .obj。

连接过程是将编译过程中生成的目标代码进行连接处理,进而生成可执行程序文件的过程。在连接过程中,时常还要加入一些系统提供的库文件代码。经过连接后生成的可执行文件的扩展名为 .exe。

C 语言程序的调试、运行步骤可以用如图 1-1 所示。

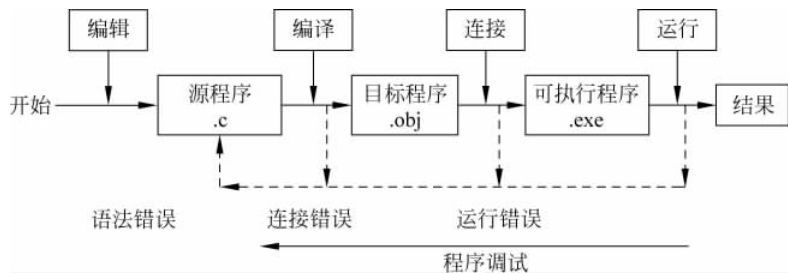


图 1-1 C 语言程序的调试、运行步骤

在图 1-1 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。

除了较简单的程序,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以分为三类:

- 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。连接错误通常由于未定义或未指明要连接的函数,或者函数调用不匹配等,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的错误,往往是由前面的一两个错误带来的。所以当纠正了几个错误后,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,解题思路就不对。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。对于出错的程序段,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为更小的块,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会面。最好的解决办法是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不正确的数据。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。

C 语言的开发工具众多,下面分别以 TC 2.0、Visual C++ 6.0 和 Dev-C++为上机平台,对程序编译、连接和调试做简单介绍。

1.2.2 Turbo C 集成环境

美国 Borland 公司在 1987 年首次推出了 Turbo C 1.0 产品,其中使用了全新的集成开发环境,即使用了一系列下拉式菜单,将文本编辑、程序编译、连接以及程序运行一体化,即源程序的输入、修改、调试及运行都可以在 TC 集成环境下完成,大大方便了程序的开发。1989 年该公司又推出了 Turbo C 2.0,在原来集成开发环境的基础上增加了查错功能,并可以在 Tiny 模式下直接生成 .COM(数据、代码、堆栈处在同一 64KB 内存中)文件。此外, Turbo C 2.0 还可对数学协处理器(支持 8087/80287/80387 等)进行仿真。Borland 公司后来推出的 Turbo C++ 版本包含了面向对象的基本思想和设计方法。1991 年为了适用 Microsoft 公司的 Windows 3.0 版本,Borland 公司又对 Turbo C++做了更新,即 Turbo C 的新一代产品 Borland C++问世了。

Turbo C 是一个常用的、最基本的 C 语言工具,一般简称 TC。TC 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,更多地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。

1. TC 启动

由于 TC 支持 DOS 环境,TC 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复制已安装好的系统。

由于 TC 是在 DOS 环境下工作的,无法像 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 1-2 所示。



图 1-2 Turbo C 集成环境界面

在图 1-2 中,菜单包含了所有操作的功能,8 个菜单项分别代表文件操作、编辑、运行、编译、项目文件、选项、调试、中断/观察等功能。按键盘上的左、右方向键可以进行菜单间的切换,被选中的菜单项将以变亮形式显示。此时按 Enter 键,就会出现一个下拉菜单,通过上、下方向键可以选择所需要的命令;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. 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) 编辑源文件

在编辑状态下可以根据需要输入或修改源程序。

在编辑窗口中,直接输入程序。如果要进行修改,可以使用 ↑、↓、←和→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。当按 F9 只进行编译连接,出现如图 1-3 所示的界面时,表示编译连接通过,程序没有语法错误,可以运行了(按 Ctrl+F9 键)。



图 1-3 编译连接通过界面

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 或 Ctrl+c 键,强制结束。

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,输入一个空格或回车,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 键才出现输出窗口。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 键运行,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 键就能看到结果。

5) 运行环境设置

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

通过上述步骤,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,下面的内容是必须了解的。

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,下面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 F9 键,然后按←或→键选择相应菜单位置,再按回车键。
- 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,完成菜单功能操作。按 Esc 键可以取消菜单选择(不是撤销菜单功能)。

TC 中提供了完善的帮助信息,按 F1 键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 表示同时按下 Ctrl 和 F9 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: 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 菜单(编辑操作)

它没有下拉菜单,按回车键直接进入编辑窗口,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 1-4 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。
- Ctrl+q+a: 替换字符串。它上面查找的过程中,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。

要在“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,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。

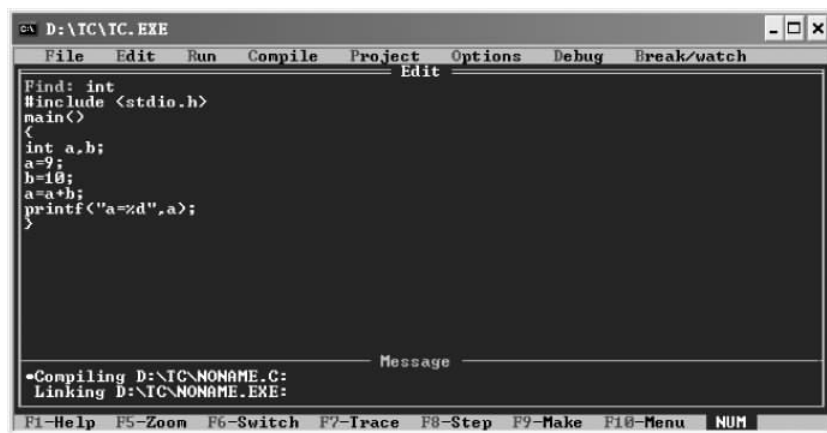


图 1-4 字符查找

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 1-5 所示。



图 1-5 TC 工作目录设置

- 命令行参数输入：在学习了“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 键隐去菜单，然后可按 Ctrl+F9 键运行程序，参数便能被主函数接受。

4. 窗口操作

在如图 1-2 所示的 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 键就像一个电源开关。如果编辑窗口扩大到整屏，而又想看一下信息窗口，可使用 F6 键进行窗口切换。如果在程序执行时又开了观察窗口，F6 键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 键又会恢复较小窗口。

5. 程序调试

TC 提供了一些调试的手段和工具，下面按照使用过程给以介绍。

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。

(1) 使光标移动到需暂停的行上。

(2) 按 F4 键或执行菜单 Run 中 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 键，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 键执行，当程序执行到该行将会暂停。设置断点的步骤为：

(1) 把光标移动到需暂停的行上。

(2) 按 Ctrl+F8 键或执行菜单 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 键或执行菜单 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 1-6 所示，这里输入了变量 i 进行查看。

对于图 1-6 中的例子，先把光标移动到第六行，然后按 F4 键执行，程序到第六行暂停，如图 1-7 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 键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 键，使查看窗口成为操作窗口，然后按回车键，可以改变查看变量，按 Delete 键可以删除查看变量。这些菜单功能都包括在 Break/watch 菜单中。



图 1-6 输入查看结果

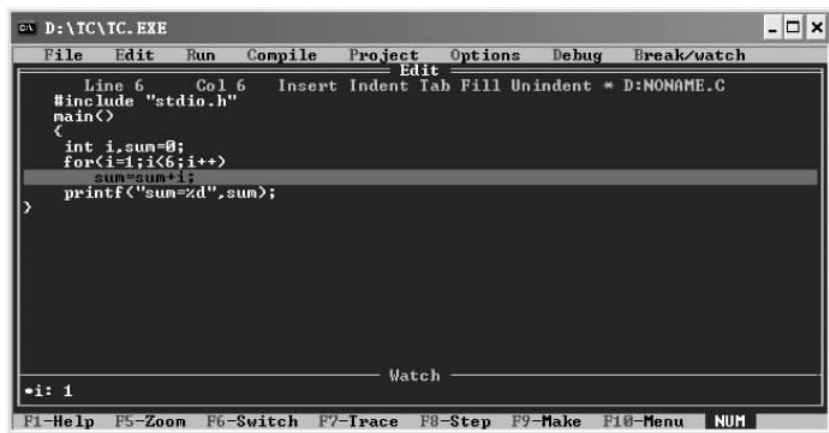


图 1-7 查看中间结果

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 键或执行菜单 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 键或执行菜单 Run 中的 Trace into 操作。对不是函数调用的语句来说,F7 键或 F8 键作用相同。但一般对系统函数不要使用 F7 键。

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。因此调试程序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 键或执行菜单 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 键(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 键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 键或执行菜单 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 1-8 所示。程序中把 for(...) 分成三行,以便观察执行流程。先把光标移动到第五行,然后按 F4 键或执行菜单 Run 中的 Go to Cursor 操作,按 Ctrl+F7 键输入查看变量 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 键单步执行,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,可以充分反映出 for 语句的执行流程。

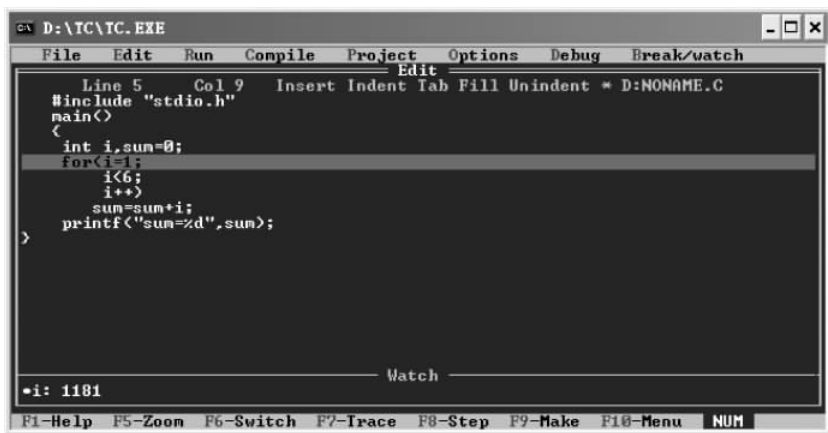


图 1-8 循环调试实例

通过这个例子,可以举一反三调试自己的程序了。

上面只对 TC 中主要的功能做了介绍,对于其他的操作可以自己试验,或参考有关 Turbo C 手册。

为了尽量减少错误的发生,除了程序本身的正确,书写格式也非常重要,它体现了编程者的自身素养。每一个人都应培养一个良好的程序设计风格,以下几点应该做到:

- 用阶梯形式书写程序,充分体现循环、分支程序结构的位置与层次,便于程序的调试,减少程序的书写错误。
- 左右花括号尽量对齐,函数间增加空行,以示分隔。
- 变量名、函数名的取名采用与其所表达的意义相同或相近的英文单词,使得在阅读程序时容易理解。
- 在程序中加入注解,尤其在关键步骤上,函数的开始处应对函数的功能、参数的意义及返回值予以说明。

6. 常用快捷键小结

F2: 保存程序。

F3: 调入程序文件。

Ctrl+F2: 结束程序调试运行。

Alt+F3: 调入最近曾经用过的文件。

F4: 程序运行到光标处暂停。

Alt+F5: 查看运行结果。

F7: 单步执行(可进入函数)。

F8: 单步执行(不能进入函数)。

F9: 编译、连接程序。

Ctrl+F9: 编译、连接、执行程序。

F1: 帮助。

F5: 放大/缩小窗口。

F6: 窗口切换。

Ctrl+F7: 增加查看变量。

Ctrl+F8: 把光标所在行设为断点或取消断点。

Alt+F9: 编译程序。

F10: 菜单选择。

Alt+x: 退出系统。

1.2.3 Visual C++ 6.0 语言集成环境

C++语言是在C语言的基础上发展而来的,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++ 6.0(简称VC++ 6.0)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。Visual C++ 6.0广泛地用于各种编程,使用面很广。这里简要介绍如何在Visual C++ 6.0下运行C语言程序。

在使用Visual C++ 6.0调试程序之前,必须先了解项目和项目工作区两个概念。项目(project)是一组相互关联的文件,是将一个应用程序中需要的所有文件组成一个有机的整体,一般包括源文件、头文件、资源文件等。项目文件必须在某个项目工作区(Workspace)的管理之下,一个项目工作区可以包含一个以上的项目。

使用Visual C++ 6.0调试程序的大致步骤是:

(1) 创建一个项目工作区(Workspace)。

(2) 在项目工作区中新建一个项目(project)。

也可以将上面两步合并,在建立新项目的同时创建项目工作区。

(3) 在项目中建立源文件、头文件等(或将这些文件添加到项目中)。

(4) 编辑项目中的源文件、头文件。

(5) 连接、编译应用程序。

(6) 运行应用程序。

1. 启动 VC++

Visual C++ 6.0 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。通过执行“开始”→“程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令,可启动 Visual C++ 6.0,屏幕上将显示如图 1-9 所示的窗口。

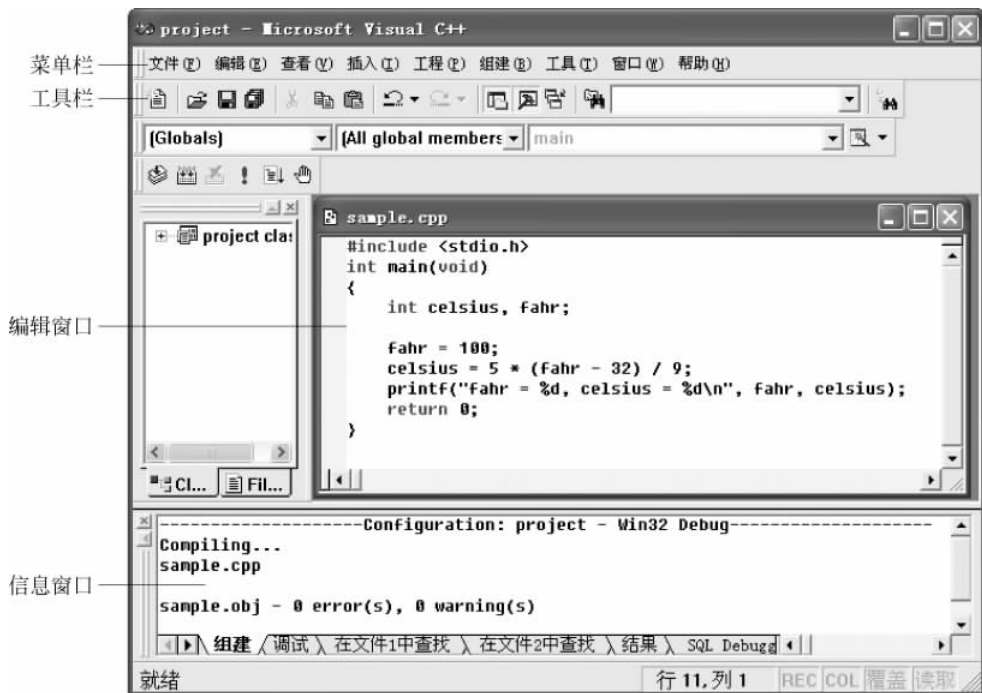


图 1-9 Visual C++ 窗口

2. 新建/打开 C 程序文件

选择“文件”菜单的“新建”命令,单击如图 1-10 所示的“文件”标签,选中 C++ Source File 选项,单击“确定”按钮,然后在编辑窗口中输入程序。

如果程序已经输入过,可选择“文件”菜单的“打开”命令,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. 程序保存

在打开的 VC++ 6.0 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。程序保存对话框如图 1-11 所示。

4. 执行程序

首先要生成可执行文件。使用 VC++ 6.0“组建”菜单中的“组建”命令,如图 1-12 所示,也可使用快捷键 F7。

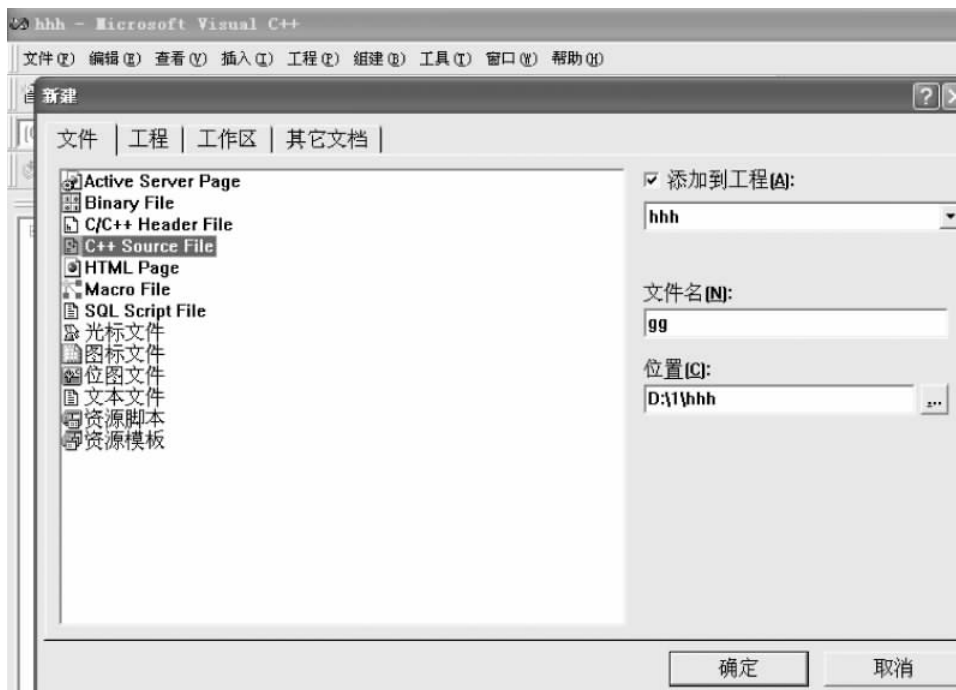


图 1-10 新建文件

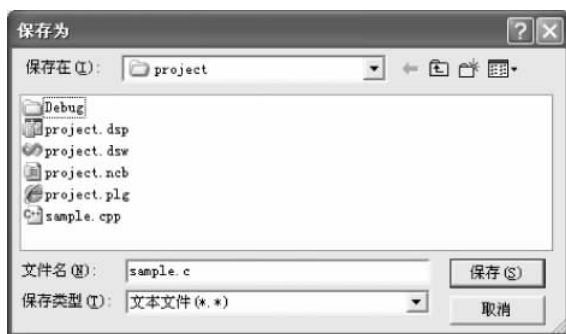


图 1-11 程序保存对话框



图 1-12 编译连接菜单

在编译连接过程中 VC++ 6.0 将保存该新输入的程序,并生成一个同名的工作区。保存文件时需填入文件名,如“prime.c”。假如不指定扩展名.c,VC++ 6.0 会把扩展名自动定义为.cpp,即 C++ 程序。如果程序没有错误,将在如图 1-13 所示的信息窗口中显示内容:

0 error(s),0 warning(s)

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 1-14 所示,双击某行出错信息,程序窗口中会指示对应出错位置,可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”命令(或快捷键 Ctrl+F5)执行程序。

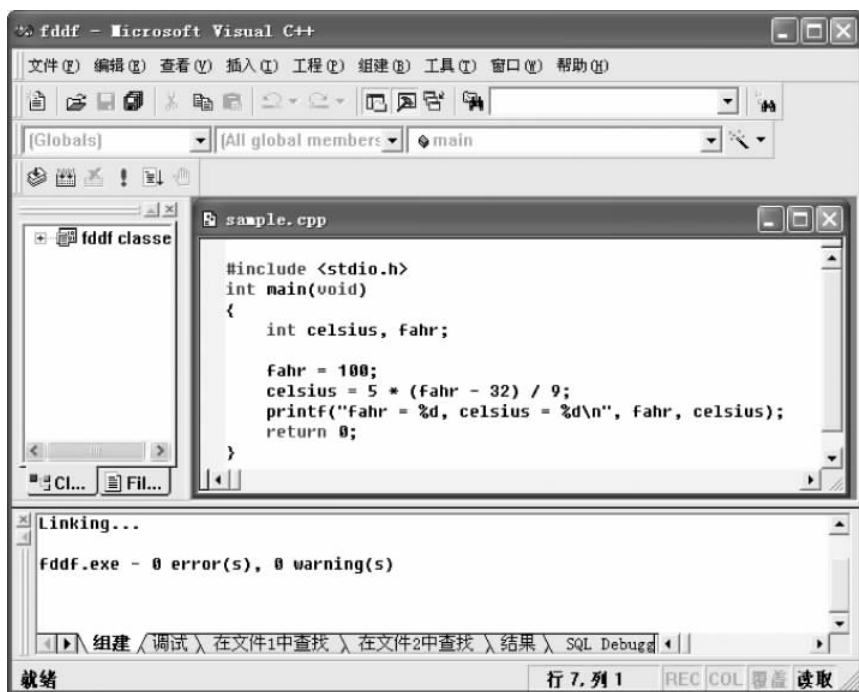


图 1-13 编译连接正确

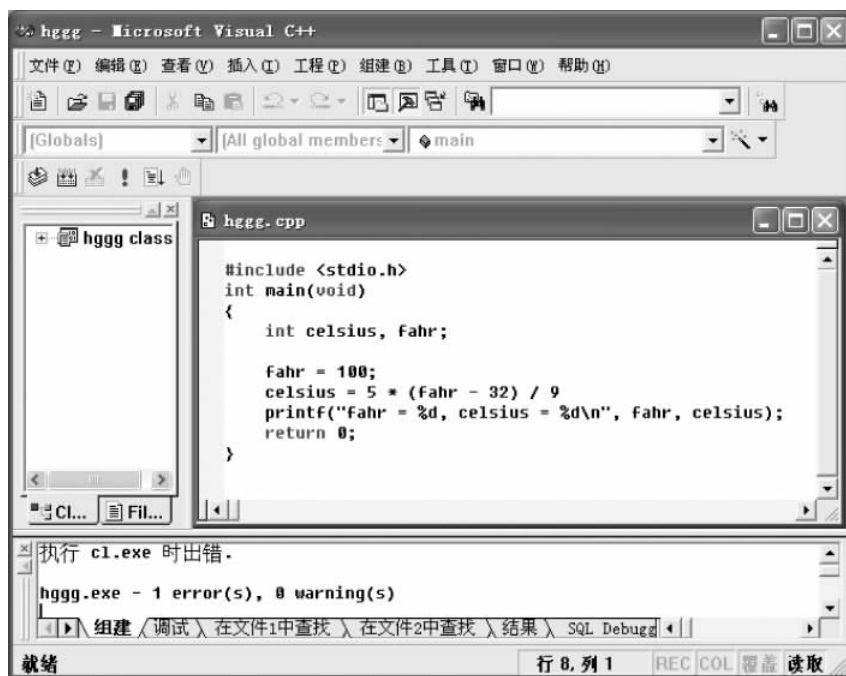


图 1-14 编译连接出错

当运行 C 程序后, VC++ 6.0 将自动弹出程序运行(数据输入输出)窗口, 如图 1-15 所示, 按任意键将关闭窗口。

对于编译、连接、执行操作,VC++ 6.0 还提供了一组工具按钮,如图 1-16 所示。

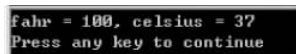


图 1-15 程序运行窗口



图 1-16 编译连接执行工具按钮组

5. 关闭程序工作区(也叫工作空间)

当一个程序编译连接后,VC++ 6.0 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。否则,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 1-17(a)所示,执行“关闭工作区”命令,然后在如图 1-17(b)所示的对话框中单击“否”按钮。如果单击“是”按钮,将同时关闭源程序窗口。



图 1-17 关闭程序工作区

6. 命令行参数处理

VC++ 6.0 是一个基于窗口操作的 C++ 系统,没有提供命令行参数功能,需要在 Windows 的“MS-DOS 方式”窗口里以命令方式实现。具体参考如下步骤:

- (1) 正确编译连接,生成可执行文件;
- (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 test.c);
- (3) 进入 debug 文件夹(它包含 test.c 程序的可执行文件 test.exe);
- (4) 执行“开始”菜单的“运行”命令,填入 command,然后单击“确定”按钮;
- (5) 在打开的“MS-DOS”方式窗口中输入: test 参数 1 参数 2...,带参数运行程序。

7. 程序调试

VC++ 6.0 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。

(1) 在需暂停的行上单击鼠标,定位光标;

(2) 如图 1-18 所示,分别单击菜单“组建”→“开始调试”→Run to Cursor 命令,或按 Ctrl+F10 键,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 键,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 1-18 执行到光标所在行暂停

方法二: 在需暂停的行上设置断点。

- (1) 在需设置断点的行上单击鼠标,定位光标;
- (2) 单击“编译微型条”中最右面的按钮,如图 1-19 所示,或按 F9 键。

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。



图 1-19 设置断点

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 1-20 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 4、5,而变量 i、sum 的值是不正确的,因为它们还未赋值。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“名称”框中填入相应变量名。

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,则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。

单步执行单击“调试”工具栏中 Step Over 按钮  或按 F8 键,如图 1-21 所示。如果遇到自定义函数调用,想进入函数进行单步执行,可单击 Step Into 按钮  或按 F11 键。当想结束函数的单步执行,可单击 Step Out 按钮  或按 Shift+F11 键。对不是函数调用的语句来说,F11 与 F8 键作用相同。但一般对系统函数不要使用 F11 键。

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,不管用户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,再单击“编译微型条”中最右面的按钮  或按 F9 键,该操作是一个开关,单

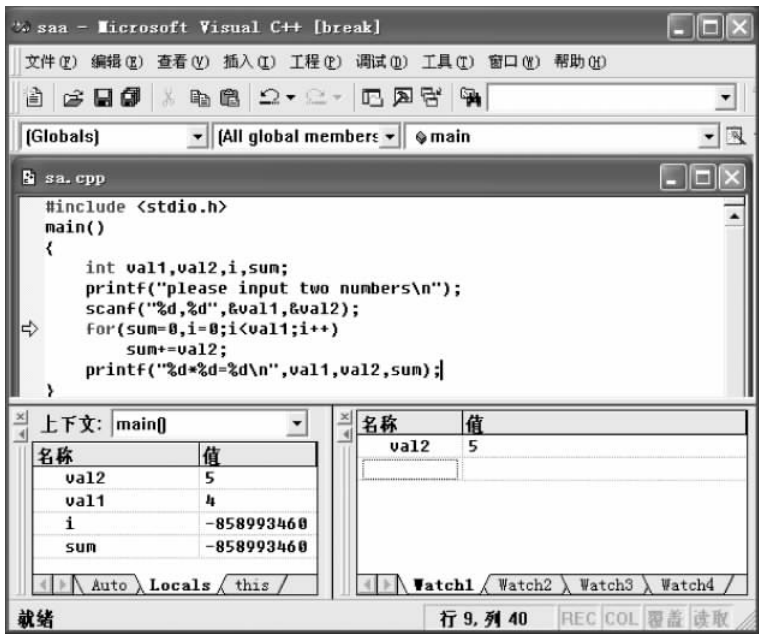


图 1-20 调试过程中观察变量的值

击一次是设置,单击两次是取消设置。如果有多个断点想全部取消,可执行“编辑”菜单中“断点”命令,屏幕上会显示 Breakpoints 窗口,如图 1-22 所示,窗口下方列出了所有断点,按“全部移除”按钮,将取消所有断点。



图 1-21 单步调试

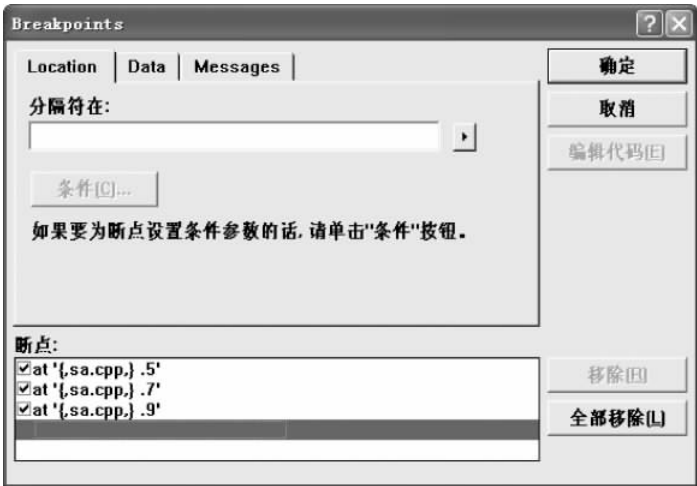


图 1-22 取消所有断点

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,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+F5 键会暂停在第一个断点,再按一次 Ctrl+F5 键会继续执行到第二个断点暂停,依次执行下去。

5) 停止调试

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

上面只对 Visual C++ 6.0 中主要的功能做了介绍,对于其他的操作读者可以自己试验,或参考有关 Visual C++ 6.0 手册。

1.2.4 Dev-C++ 集成环境

Dev-C++ 是一个 Windows 下的 C/C++ 集成开发环境 (IDE)。Dev-C++ 是一款自由软件,使用 GCC 编译器,遵循 C/C++ 标准,当然也支持更高版本的 MinGW (Windows 版的 GCC)。Dev-C++ 开发环境包括多页面窗口、工程编辑器以及调试器等,在工程编辑器中集合了编辑器、编译器、连接程序和执行程序,提供高亮度语法显示的,以减少编辑错误,还有完善的调试功能。多国语言版中包含简体和繁体中文语言界面及技巧提示,还有英语、俄语、法语、德语、意大利语等二十多个国家和地区的语言供选择。

与 Visual C++ 相比,Dev-C++ 没有提供针对 Windows 编程的类库,使得开发环境比较小巧,特别适合初学者使用。而且,多国语言版中包含简繁体中文语言界面及技巧提示,还有英语、俄语、法语、德语、意大利语等二十多个国家和地区语言提供选择,大大方便了国内用户的使用。

1. Dev-C++ 开发环境简介

Dev-C++ 是一个共享软件,用户可以免费从互联网上载使用。Dev-C++ 的开发环境如图 1-23 所示,各界面元素的功能分别简介如下。



图 1-23 Dev-C++ 开发环境

- 菜单栏：这是 Windows 应用程序的标准元素，主要提供了项目操作、编辑操作、编译调试及帮助等各种常用操作。所有的操作基本上都可以通过菜单命令实现。
- 工具栏：这是 Windows 应用程序的标准元素，将常用操作集中到工具栏中，方便用户操作。
- 工程管理：用于对创建包含多个源文件的应用程序进程管理。
- 源代码：主要用于输入和编辑 C/C++ 语言程序源代码。
- 报告窗口：主要用于显示编译、链接等过程中的输出信息，包括编译进展情况、是否成功、出错位置及原因等信息。
- 状态栏：显示开发环境中的一些状态，如是否修改、当前插入点所在位置、代码的总行数等。

2. Dev-C++ 开发环境的设置

在使用 Dev-C++ 开发 C 语言程序时，需对开发环境进行设置。这个主要是通过 Dev-C++ 开发环境的编译器选项、环境选项和编辑器选项等来实现，具体设置步骤如下：

(1) 在 Dev-C++ 开发环境中，选择“工具”→“编译器选项”命令，打开“编译器选项”对话框，如图 1-24 所示。这里编译器主要的设置工作都在这个对话框中进行。其中，在“编译器”选项卡中可以设置不同的编译器配置文件。可通过右侧的三个按钮添加、删除、重命名编译器配置。



图 1-24 “编译器选项”对话框

(2) 选择“编译器选项”对话框中的“代码生成/优化”选项卡，设置“支持所有 ANSI C”标准为“Yes”，如图 1-25 所示。

(3) 选择“编译器选项”对话框中的“目录”选项卡，在其中可分别设置 Dev-C++ 中二进制文件、库文件、C 包含文件等的具体位置，如图 1-26 所示。

(4) 选择“编译器选项”对话框中的“程序”选项卡，在其中可设置不同的编译器、链接程序的名称，如图 1-27 所示。