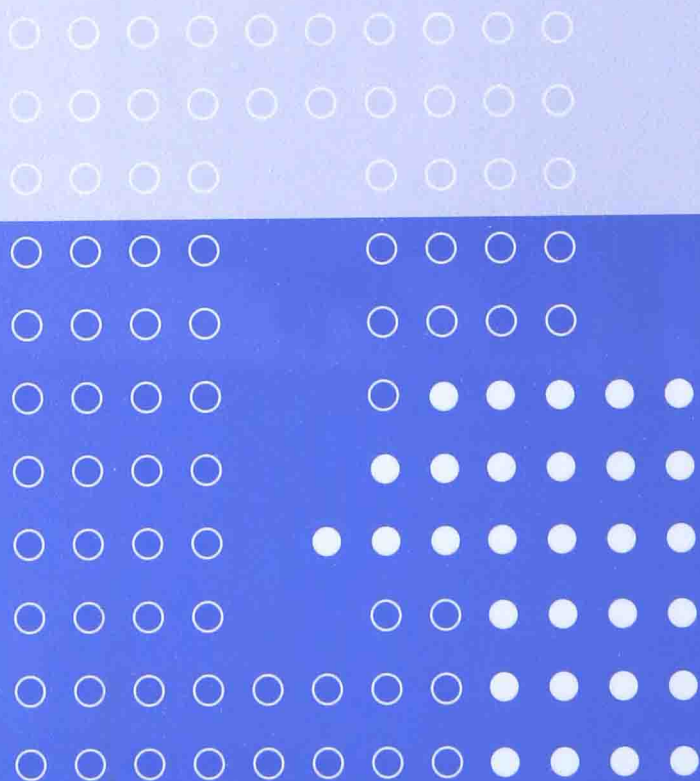




普通高等教育“十一五”国家级规划教材 计算机系列教材

C程序设计 实验指导与习题集 (第2版)



衣治安 刘华莹 主 编
时贵英 刘金月 副主编



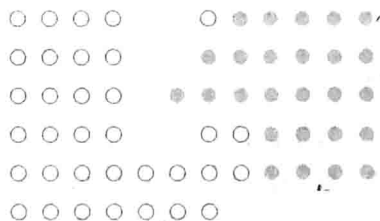
清华大学出版社



普通高等教育“十一五”国家级规划教材 计

衣治安 刘华莹 主 编
时贵英 刘金月 副主编

C程序设计 实验指导与习题集 (第2版)



清华大学出版社
北京

内 容 简 介

本书是与《C 程序设计教程(第 2 版)》(马瑞民、衣治安主编,清华大学出版社出版)配套的辅助教材。全书分为三个部分,第一部分是实验指导,包括 20 个实验,除了实验 1 和实验 20 是介绍运行环境外,其他的每个实验都提供了调试程序题和编程题,覆盖了相关章节的知识点。每个调试程序后面的分析讨论都能够帮助学生举一反三。有些实验还提供了选做题,该部分的实验都不提供参考答案,以锻炼学生独立学习的能力。第二部分是习题集,通过选择、填空、编程等题型,让学生掌握 C 语言的基本语法,提高编写程序的能力。第三部分是习题解答,提供第二部分全部习题的参考答案。

本书适合作为高等学校各专业本科生的实验教材和教学辅导教材,建议实验学时为 30~40 学时。本书也可以作为软件技术人员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 程序设计实验指导与习题集/衣治安,刘华莹主编. —2 版. —北京:清华大学出版社,2015
(2015.4 重印)

计算机系列教材

ISBN 978-7-302-38983-5

I. ①C… II. ①衣… ②刘… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312
中国版本图书馆 CIP 数据核字(2015)第 005697 号

责任编辑:张瑞庆

封面设计:常雪影

责任校对:李建庄

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:保定市中国画美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:13 字 数:323 千字

版 次:2011 年 2 月第 1 版 2015 年 2 月第 2 版 印 次:2015 年 4 月第 2 次印刷

印 数:2501~4000

定 价:25.00 元

产品编号:062125-01

本书是与马瑞民教授等编写的《C 程序设计教程(第 2 版)》(主教材)配套的实验指导书与习题集,其中的内容和章节的编排顺序也是与主教材一致的。

根据中国高等学校计算机基础教育课程体系(CFC2014)的要求,本着提高学生计算机应用能力和注重实践教学的精神,书中实验指导部分占据较大的比重,大幅度压缩了习题部分的篇幅。实验指导部分配有大量的分析讨论问题,希望学生在完成每道题时能够举一反三,收获更多。实验指导部分无论是改错、填空还是编程,都没有给出答案,一方面可以通过上机来完成,另一方面也可以给学生更多的思考空间。全书共设计了 20 个实验,其中的所有程序都在 Microsoft Visual C++ 6.0 环境下调试通过。为了兼顾仍然使用 Turbo C 环境的读者,在实验 20 中给出了 Turbo C 运行 C 程序的基本步骤。本书的习题集部分都给出了参考答案。

本书大幅度增加了实验的题量,大量增加了常用算法的实验,希望学生通过实验能够更好地掌握课堂所学内容。在实验部分,本书覆盖面比较广泛,以满足不同学生的学习要求以及教师的实验安排。程序题按照先易后难的原则,首先给出程序,让学生在“上机练习”的过程中发现题中的错误,或者根据分析讨论中的提示进行完善和补充,使学生掌握程序设计思路的多样性、程序设计方法的灵活性,同时避免以往学生常犯的错误。

本书涉及的题量比较大,也比较全面,教师可以根据自己学校的实际授课情况选择其中的部分实验内容,根据学生的具体情况进行整合,以满足不同层次的学生不同进度的学习需要。学生在应用本书实验指导部分时,一定要多练习、多实践,努力在“做中学”,在知识的积累过程中获得快乐的感受;而在应用习题集部分时,一定要多读、多分析,不要着急参看习题答案以及上机操作。这部分内容重点考核学生的“走程序”能力和对基础知识的掌握情况,希望能够更好地帮助学生提高读程序和编程序的能力。

本书由衣治安、刘华莹主编,时贵英、刘金月副主编。衣治安编写了习题集及习题解答,刘华莹编写了实验 2~7,时贵英编写了实验 1、实验 8~13,刘金月编写了实验 14~20。在本书编写及以往应用过程中,东北石油大学计算机基础教育系的教师们提出了许多中肯的意见,在此表示衷心感谢。由于作者水平有限,书中一定存在许多不足,恳请读者提出宝贵意见和建议。

作 者

2015 年 1 月

第一部分 C 程序设计实验指导 /1

- 实验 1 用 Visual C++ 运行 C 程序 /2
- 实验 2 简单的 C 程序 /12
- 实验 3 顺序结构程序设计 /16
- 实验 4 选择结构程序设计 /21
- 实验 5 while、do-while 语句构成的循环 /26
- 实验 6 for 语句构成的循环 /31
- 实验 7 多重循环程序设计 /36
- 实验 8 一维数组 /42
- 实验 9 二维数组 /52
- 实验 10 字符型数据 /60
- 实验 11 变量作函数参数 /67
- 实验 12 数组作函数参数 /74
- 实验 13 递归函数 /84
- 实验 14 指针变量的基本使用 /88
- 实验 15 用指针变量处理数组 /95
- 实验 16 指针和函数 /101
- 实验 17 结构体 /108
- 实验 18 文件 /114
- 实验 19 C 语言的其他应用 /120
- 实验 20 用 Turbo C 环境运行 C 程序 /130

第二部分 C 程序设计习题集 /139

- 第 1 章 概述 /140
- 第 2 章 顺序结构程序设计 /142
- 第 3 章 选择结构程序设计 /145
- 第 4 章 循环结构程序设计 /149
- 第 5 章 数组 /153
- 第 6 章 函数 /156
- 第 7 章 指针 /160
- 第 8 章 结构体 /166
- 第 9 章 文件 /171

第 10 章 C 语言涉及的其他知识 /173

第三部分 C 程序设计习题解答 /181

第 1 章 概述 /182

第 2 章 顺序结构程序设计 /183

第 3 章 选择结构程序设计 /184

第 4 章 循环结构程序设计 /186

第 5 章 数组 /189

第 6 章 函数 /192

第 7 章 指针 /195

第 8 章 结构体 /197

第 9 章 文件 /198

第 10 章 C 语言涉及的其他知识 /199

参考文献 /200

第一部分

C 程序设计实验指导

实验 1 用 Visual C++ 运行 C 程序

一、实验目的

- 掌握在 Visual C++ 环境下编辑、编译、连接以及运行程序的方法。
- 掌握在 Visual C++ 环境下调试程序的方法。

二、实验内容

为了运行一个 C 程序,一般要经过编辑、编译、连接和运行 4 个步骤。所谓的编辑,就是把用户编写的 C 程序代码利用某种编辑软件录入计算机,并保存为扩展名是 C 的源程序的过程。C 源程序是一个纯文本文件,经过编译后可以得到二进制的目标代码,文件扩展名是 obj,这个目标文件不能直接执行。目标代码经过连接操作后得到一个扩展名是 exe 的可执行文件,执行 exe 文件才能得到相应的结果。

运行 C 程序的环境有很多,目前比较常用的有 Turbo C(简称 TC)和 Visual C++(简称 VC)环境,前者运行在 DOS 系统下,后者运行在 Windows 系统下。本教材的所有程序都是在 VC 6.0 英文版环境下调试通过的。本实验介绍在此环境下编辑、编译、运行和调试 C 程序的方法。另外,考虑到读者的编程环境可能会与本教程使用的不完全一致,文中对英文界面描述必要的地方加了中文标注,相信不会影响读者对文字叙述的理解。

1. VC 编程环境的使用

(1) 打开 VC 窗口

首先在“开始”菜单的“程序”项下找到 Microsoft Visual C++ 6.0 选项,单击该选项,打开如图 1-1-1 所示的 VC 主窗口。

为了以后应用的统一,本教材中使用的所有 C 源程序都保存在 E 盘已经提前建好的 VCLIST 文件夹下。读者可以在自己的计算机上建立一个专门用来存放 C 源程序的文件夹,在建立文件夹后再进行后面的操作。

现在,开始准备编辑一个 C 源程序。在 VC 主窗口的 File(文件)菜单中单击 New(新建)命令,此时弹出一个 New(新建)对话框,如图 1-1-2 所示。该对话框有 4 个选项卡,一般情况下默认显示的是 Projects(工程)选项卡,这时单击 Files(文件)选项卡,然后在下面显示的文件选项中选择其中的 C++ Source File 列表项(图 1-1-2 中反白显示的项),表明要编辑的是 C(或 C++)源程序。此时,在 New(新建)对话框右下侧的 Location(位置)文本框中显示的 C:\windows\system32,表示的是源程序目前默认的存放位置。要想改变文件的存放位置,需要单击其后的按钮 ,弹出如图 1-1-3 所示的 Choose Directory(选择目录)对话框,先在下部的 Drives(驱动器)中选择驱动器(例如 e:),再在上部的 Directory name(目录名称)中选择文件夹(例如 VCLIST),然后两次单击 OK(确定)按钮,接着在

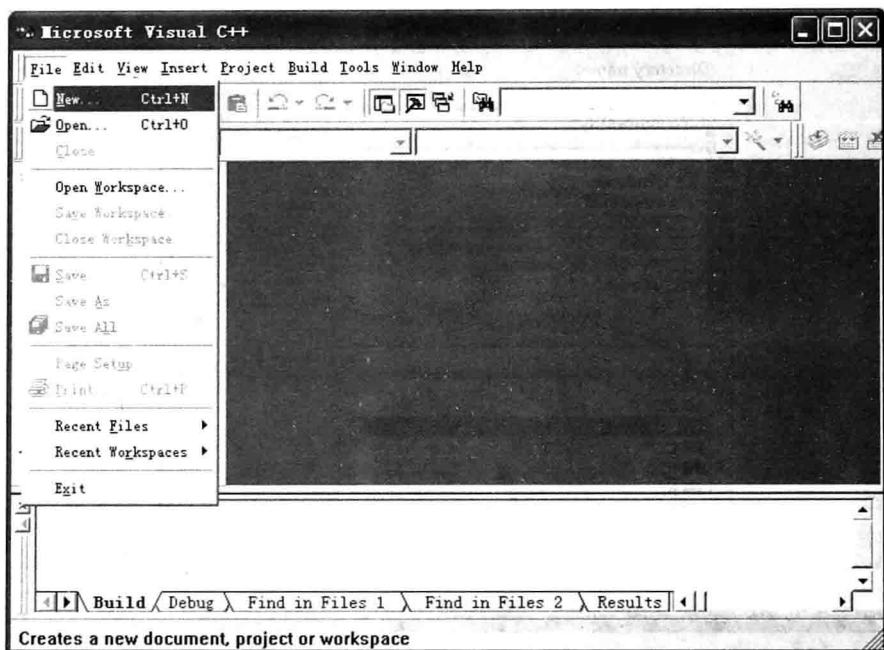


图 1-1-1 Microsoft Visual C++ 主窗口

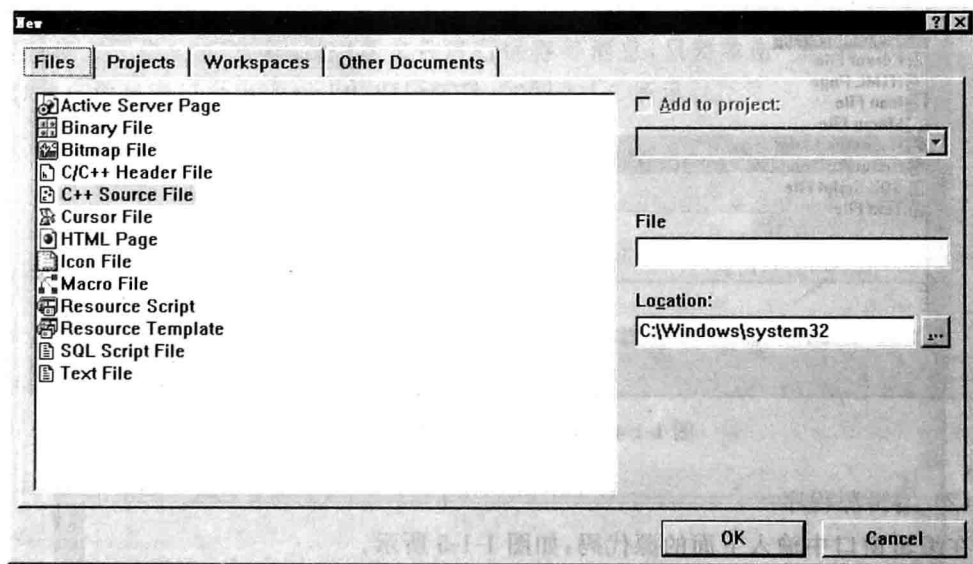


图 1-1-2 New 对话框的 Files 选项卡

New(新建)对话框中右侧的 File(文件名)下的文本框中输入文件名(此处输入的是 1-1.c),得到如图 1-1-4 所示的界面。此时单击 OK(确定)按钮,进入如图 1-1-5 所示的 VC 编辑窗口。

注意: 在输入文件名时一定要输入扩展名 C,以防止不必要的麻烦,因为如果读者不给源程序指定扩展名,系统将默认指定扩展名是 cpp,按 C++ 程序处理。

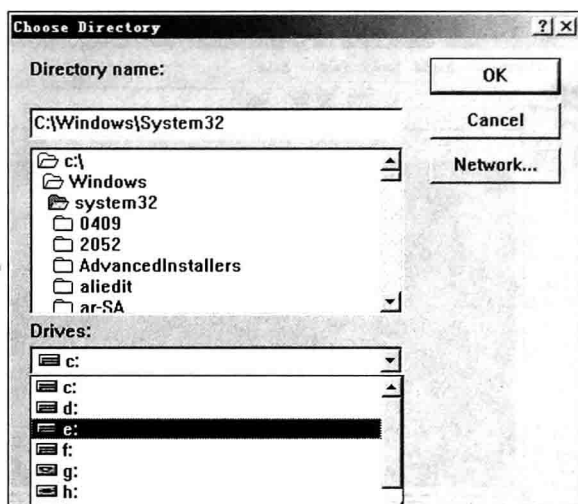


图 1-1-3 “选择目录”对话框

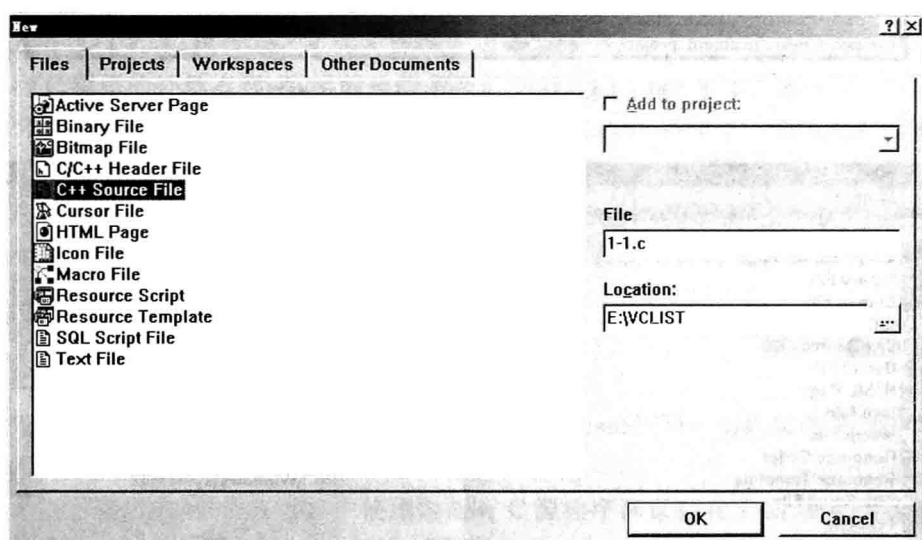


图 1-1-4 全部设置完成的 New 对话框

(2) 编辑源程序

在编辑窗口中输入下面的源代码,如图 1-1-5 所示。

```
#include "stdio.h"
main()
{
    printf("*****\n");
    printf("Hello World!\n");
    printf("*****\n");
}
```

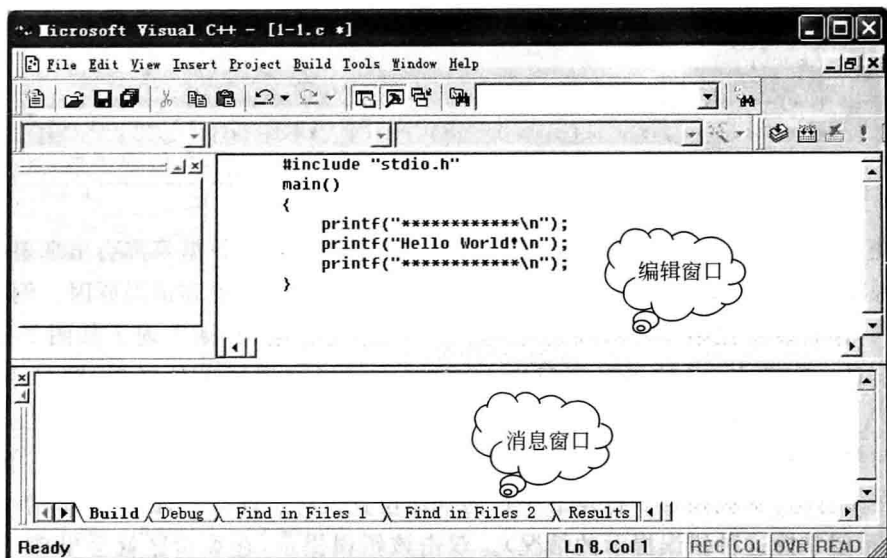


图 1-1-5 在 VC 中编辑 C 程序

(3) 编译、调试程序

在 Build(组建)菜单中单击 Compile(编译)命令(如图 1-1-6 所示),或者单击工具栏上如图 1-1-6 中箭头指向的编译按钮就可以进行编译了。在此过程中,一般系统要提示需要建立活动工作空间以及对源文件进行保存等消息,只要单击“是”按钮即可。编译完成后,在消息窗口会出现相应的编译信息,如图 1-1-7 所示。

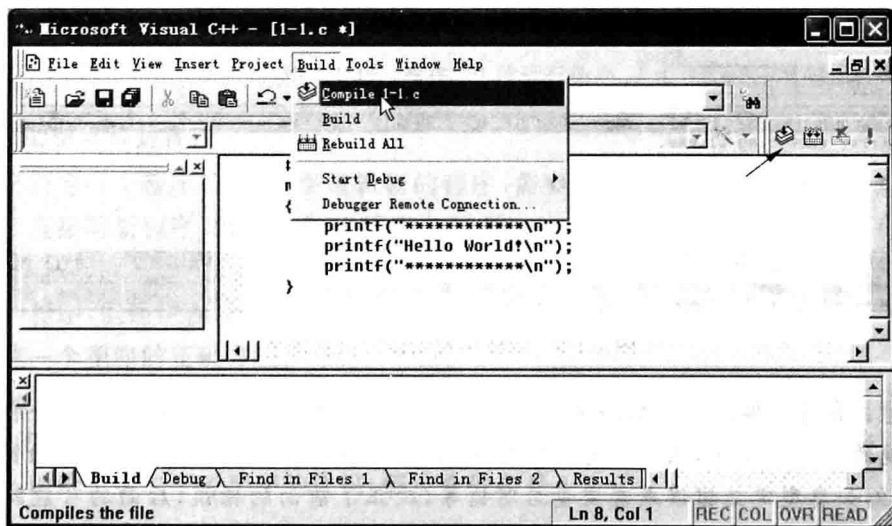


图 1-1-6 VC 的“编译”命令

从图 1-1-7 中可以看出,本次编译的是 1-1.c 源文件,编译生成的是 1-1.obj 文件,编译中出现 0 error(错误)和 0 warning(警告)。编译显示 0 error 表明编译通过,程序中没有语法错误。如果在编译时出现 warning 信息,一般不影响进一步的操作。例如,程序中

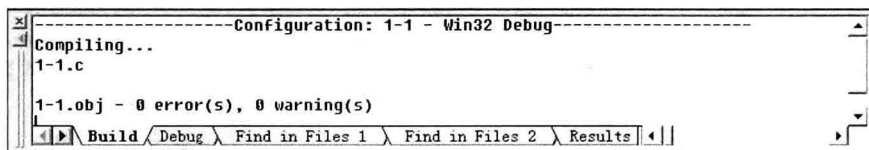


图 1-1-7 编译信息

定义了变量但是没有使用,或者把一个浮点型变量赋值为整型数据等都会出现警告信息。

如果程序中含有语法错误,在编译时系统会指出错误的位置和错误原因。例如,在编辑的时候,作者故意把第 5 行后面的分号去掉,然后进行编译,则出现了如图 1-1-8 所示的编译信息。在图 1-1-8 的消息窗口中,系统提示编译出现了 1 处错误,窗口中显示的 e:\vclist\1-1.c(6),提示错误出现在 e:\vclist\1-1.c 文件的第 6 行,冒号后面显示的是错误类型和错误原因,此次提示的是在标识符 printf 之前缺少了分号(这是由于一条 C 语句允许写在多行上,系统在第 5 行及下一行的合适位置没有发现分号,因此提示错误产生在第 6 行,希望掌握这种错误提示的情况)。双击该错误提示,在编辑区就会出现一个指向错误行的符号 ➡,这样可帮助读者快速定位错误行。改正错误后一定要重新编译程序。

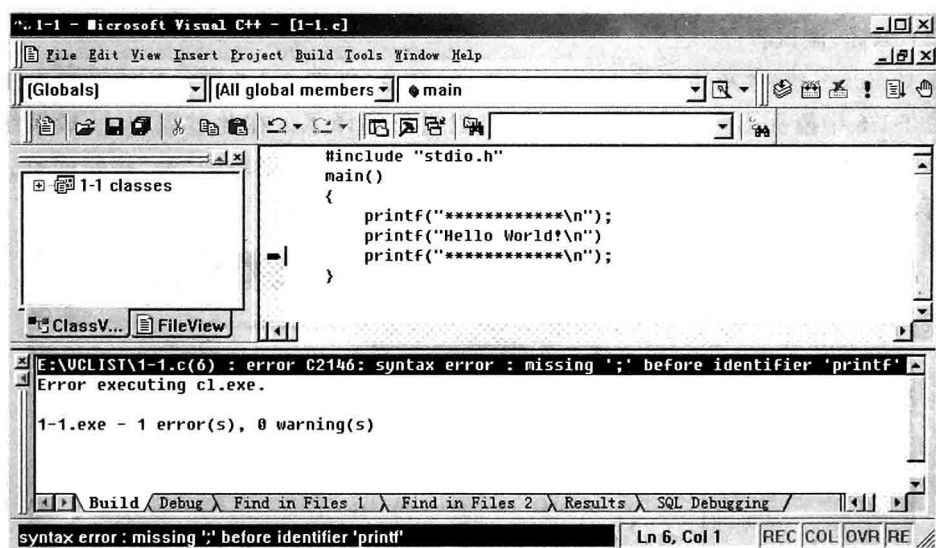


图 1-1-8 编译出现错误的信息提示

注意: 有时在编译一个程序时会出现很多错误提示信息,这时不要着急,一般来说先改正发现的第一条错误,然后试着再编译一次,会发现错误少了很多。因为有的错误是连带错误,例如类型定义错误或者变量名写错等,改正了前面的错误,后面的可能就不再是错误了。

(4) 运行程序

编译成功后,Build(构建)菜单的命令数目会发生变化,如图 1-1-9 所示,读者可以与图 1-1-6 中的 Build 菜单进行对照。在 VC 系统中,可以通过 Build 菜单的 Build 命令(或工具栏上的构建按钮)进行连接来得到可执行文件,然后再单击图 1-1-9 中鼠标指向的

Execute 命令(或工具栏上红色的执行程序按钮)来执行程序。

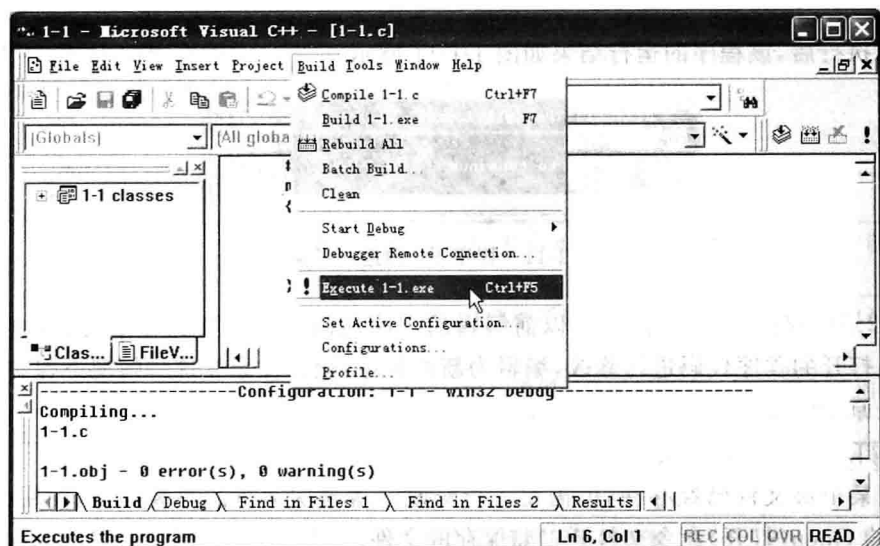


图 1-1-9 VC 的“执行”命令

程序运行后的输出结果如图 1-1-10 所示。

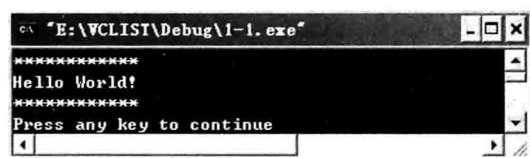


图 1-1-10 程序 1-1 的运行结果

(5) 编辑新的程序

在运行完一个程序后,如果要编辑新的程序,需要特别注意:第一,不能直接在本程序的下面直接写新程序,因为一个 C 程序中只能有一个 main 函数;第二,不能在已经打开了一个 C 程序的前提下直接用 New 命令新建一个源程序,因为在这种情况下虽然可以编译新的程序,但是在程序连接时会出现问题。

新建一个程序的正确方法如下。

① 在 File(文件)菜单中选择 Close Workspace(关闭工作空间)命令,关闭当前的工作空间,然后再使用 File(文件)菜单的 New(新建)命令,按照第一个程序的方法新建文件即可。下面给出第二个源程序的代码(假设文件名是 1-2.c):

```
#include "stdio.h"
main()
{
    int i,sum=0;
    for(i=1;i<=100;i++)
        sum=sum+i;
```

```
printf("sum=%d",sum);  
}
```

正常执行后,该程序的运行结果如图 1-1-11 所示。

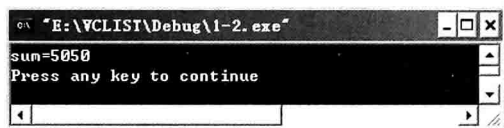


图 1-1-11 程序 1-2 的运行结果

② 如果仅仅是用来练习,对于以前编辑的程序代码并不需要保存的时候,也可以对当前已经打开的程序代码进行修改,编辑为新的程序,但要注意在这种情况下保存文件时用的还是原来的文件名,只是原来的程序代码被覆盖。

(6) 打开已经存在的文件

如果某个源文件已经存在,但需要进行修改或者再次运行的时候,可以使用 File(文件)菜单的 Open(打开)命令来打开以前保存的文件。

注意: 在打开一个新的文件的时候,同样需要关闭刚刚运行程序的工作空间,否则一样会出现可以编译但是不能连接运行的情况。

VC 的功能很强大,尤其是在编辑 C 程序时能给读者提供很大的方便。例如,对于关键字或者特殊的预定义标识符都采用蓝色字体显示,在回车的时候一般能够自动缩进等,这些功能可以帮助编程人员大大降低编辑错误。本实验中只是非常简单地介绍了编辑 C 程序时用到的最初级的功能,希望读者在应用中逐步体会其他的功能。

2. VC 对程序的跟踪调试方法

通常情况下,程序一般不会运行一次就能成功,几乎总是包含一些错误。程序的错误类型大致可以分为两种,语法错误和逻辑错误。其中,语法错误可以通过编译器的出错提示信息得到纠正;而对于语义上的逻辑错误则很难发现,因此常常出现程序在编译时没有任何错误,但是运行结果却并不正确甚至产生不了结果的情况,这就需要对程序进行跟踪调试来分析和发现其中的错误。下面对 VC 环境下的调试方法进行简单介绍。

(1) 动态调试的主要方法——设置断点

断点(breakpoint)是调试器设置的一个代码位置。在调试程序过程中,只要运行到断点处,系统就会自动停下。

设置断点的方法: 在程序代码中,把光标移动到需要设置断点的那一行上,按 F9 键(或工具栏上的按钮) ,可以看到在代码行的左端出现了一个红色的圆点——这就是 VC 中断点的标志,以后程序在调试过程中,每次执行到这里都会停下,以方便用户观察当前变量的状态。

清除断点的方法: 清除一个断点时,与设置断点的命令相同,在已设置断点的地方,按一次 F9 键(或工具栏上的按钮) ,红色的断点标志就会消失,断点便被清除了。当要清除程序中的所有断点时,可以使用快捷键 Ctrl+Shift+F9,也可以使用 Edit(编辑)菜

单下的 Breakpoints(断点)命令,打开 Breakpoints 对话框,单击 Remove All(全部移除)按钮即可。

(2) VC 常用的调试命令

VC 常用的调试命令如表 1-1-1 所示。

表 1-1-1 VC 常用的调试命令

命 令	快捷键	功 能
Go	F5	系统将进行编译,连接,并自动运行程序,但是会在程序中设置了断点的地方停下
Step Into	F11	单步执行每条语句,在遇到函数的时候系统将进入函数,单步执行其中的语句
Step Over	F10	单步执行每条语句,但在遇到函数时候会自动执行其中的内容,而不进入函数内部单步执行
Step Out	Shift+F11	单步执行到子函数内部时,用 Step Out 可以执行完子函数中剩下的语句,即跳出子函数
Run to Cursor	Ctrl+F10	系统将自动执行到用户光标所指的位置
Stop Debugging	Shift+F5	用来终止动态调试过程

(3) 程序的调试过程——举例

首先,在 E 盘上的 VCLIST 文件夹下新建一个文件 1-3.c,然后在编辑区输入以下代码:

```
#include "stdio.h"
main()
{
    int s,t,i,j;
    s=0;t=1;
    for(i=1;i<=5;i++)
    {
        for(j=1;j<=i;j++)
            t=t*j;
        s=s+t;
    }
    printf("%d\n",s);
}
```

该段代码想要实现的功能是求 $1!+2!+3!+4!+5!$ 的值。程序中的变量 t 用来存放每个数的阶乘值, s 用来存放阶乘的累加和。这段程序编译运行后,会发现程序的输出结果是 34863,与人工计算的结果 153 相差甚远,然而编译器却并没有提示程序中存在任何错误,这就需要使用 VC 的跟踪调试功能来发现程序中隐藏的逻辑错误。

然后,在程序中使用 F9 设置两个断点,如图 1-1-12 所示。启动功能键 F5(或工具栏上的按钮)进入调试状态,图 1-1-13 为程序运行到断点时的界面,图中断点上的黄色箭头指向的是系统下一步将要执行的语句,编辑窗口的下面是监视区,左边的部分称为“自

动监视区”，右边的部分称为“手动监视区”。自动监视区显示的是系统自动跟踪的变量。手动监视区可随时加入想要监视的变量，手动加入变量的方法为将光标移到要监视的变量前面，单击鼠标，在弹出的快捷菜单中选择 QuickWatch 项，随后在弹出的 QuickWatch 对话框中，单击 Add Watch 按钮，便将选中的变量加入手动监视区中，并显示出当前该变量的取值。

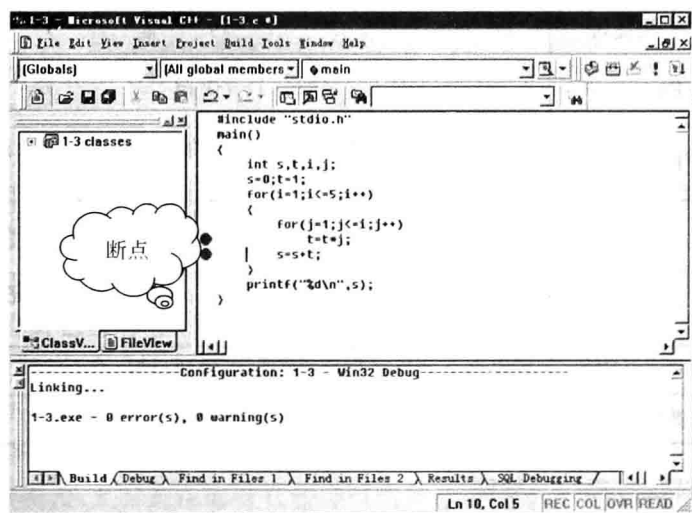


图 1-1-12 设置程序断点

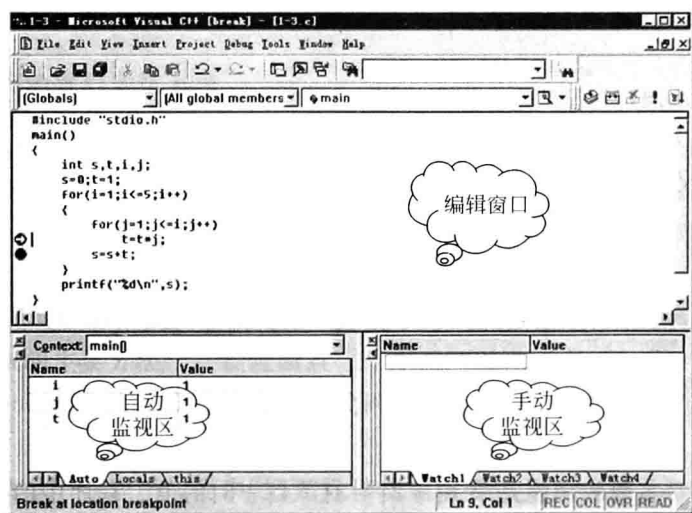


图 1-1-13 调试执行界面

接下来，依次按 F10 功能键来单步执行程序，并在监视区查看随着程序执行每个变量值的变化情况，同时比对主观和客观的程序执行结果以发现和改正错误。例如，在执行到 $i=3$ 时，可以看到 t 中存放的阶乘值为 12，而不是 3 的阶乘 6，从而分析出是构造阶乘的语句出了问题。在明确出错原因后，单击 Debug(调试)菜单的 Stop Debugging 命令

(或按 Shift+F5 快捷键)可以终止程序的调试过程。

如果在做完上面的操作后,还不知道程序错误的具体原因,这也没有关系。只要通过以上步骤,掌握程序的调试方法即可。相信读者在学完循环一章后,可以自己来改正该程序中存在的错误,然后再参照前面介绍的方法进行程序的编译、连接、运行,得到正确的结果。

说明: 由于 Run to Cursor 功能的用法比较简单,因此不再单独举例介绍。另外,考虑到本例中没有涉及子函数,因此对于 Step Info 和 Step Out 命令的使用方法在这里也不再进行详细的介绍,读者在学完了函数的知识后,可以从实践运用中掌握这两个命令的用法。