

计算机程序设计学习指导丛书

C语言程序设计 实验学习指导

● 主 编 陈丽芳 周 黎
副主编 周阳花 魏 敏 王 骏



苏州大学出版社

计算机程序设计学习指导丛书

C 语言程序设计实验学习指导

主 编 陈丽芳 周 黎
副主编 周阳花 魏 敏 王 骏

苏州大学出版社

图书在版编目(CIP)数据

C 语言程序设计实验学习指导/陈丽芳,周黎主编.
苏州:苏州大学出版社,2006.8
ISBN 7-81090-717-4

I. C… II. ①陈…②周… III. C 语言—程序设计
—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 095757 号

内 容 简 介

本书是根据国家教育部考试中心最新发布的《全国计算机等级考试大纲》和教育部制定的面向 21 世纪高校教材计算机基础课程的基本要求,结合了历年考试试题以及在校任课教师的实际经验编著而成的,适合作为普通高校理工科学生《C 语言程序设计》学习的辅助教材。全书为考生提供了一个从学习、练习到动手调试程序的过程,使考生了解学习要求、学习重点,培养学生独立思考问题和编程的能力。本书共分为 13 章,每章包括学习目标和要求、示例分析、同步强化练习、上机练习、常见错误和算法分析等内容;并配有 4 套二级考试模拟试卷。

本书在编排上注意完整性和独立性,紧扣考试大纲,叙述通俗易懂,既可以作为各高等院校和培训班学生学习《C 语言程序设计》的参考书,也可作为 C 语言自学者或准备参加二级考试者的参考用书。

C 语言程序设计实验学习指导

陈丽芳 周 黎 主编

责任编辑 苏 秦

苏州大学出版社出版发行

(地址:苏州市干将东路 200 号 邮编:215021)

常熟高专印刷有限公司印装

(地址:常熟市元和路 98 号 邮编:215500)

开本 787mm×1092mm 1/16·印张 13.5 字数 337 千

2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷

ISBN 7-81090-717-4/TP·46 定价:26.00 元

苏州大学版图书若有印装错误,本社负责调换
苏州大学出版社营销部 电话:0512-67258835

前 言

C 语言是许多计算机课程中描述算法和进行程序设计的主要语言之一。C 语言具有语言简练、数据类型丰富、表达能力强、使用灵活方便、易于理解、结构化程度高等特点,一直是各类院校理工科专业广泛开设的计算机语言课程,是编写应用程序和系统程序的常用工具,同时也是进一步学习 C++、VC++ 等语言的基础。C 语言程序设计是一门实践性很强的课程,学生不仅要掌握 C 语言的语法规则,更重要的是掌握一些常用算法,并能利用 C 语言作为工具来解决一些实际问题,这就需要一个学习实践的过程。为此我们根据国家教育部考试中心最新发布的《全国计算机等级考试大纲》和教育部制定的面向 21 世纪高校教材计算机基础课程的基本要求以及编者多年的教学经验编写该辅助教材,希望通过例题解析、上机实验、错误分析、算法分析等,帮助学生更好地掌握 C 语言的基础知识和基本概念,逐步建立用计算机处理问题的逻辑思维方式,培养学生独立思考问题和编程的能力。

本书共分为两大部分:

第一部分包括了 13 个章节的内容,各个章节包括以下几部分内容:

- (1) 学习目标和要求。列出 C 语言学习中的知识重点及解析。
- (2) 示例分析。对学习中的易忽略和出错以及考试中常出现的题目进行分析和解答,并且在点评部分给出重要的提示,以期收到举一反三的效果。
- (3) 同步强化练习。针对基础知识进行测试,分为选择题、填空题两个部分。在对最近几年考题深入研究的基础上,结合学生易出错和不易掌握的知识点,题目从易到难,灵活多样,避免重复和意思含糊的题目,从深度和广度上反映该课程学习中的重点和难点。
- (4) 实验题。体现知识的综合应用能力,有完善程序题和编程题两种形式。
- (5) 常见错误。提供学生调试程序时常出现的错误以及解决的方法。
- (6) 算法分析。介绍一些常用的算法。
- (7) 同步强化练习答案。

第二部分为附录,包括了 4 套二级考试的模拟试卷。帮助考生全面了解考试内容,适应考试形式。

本书第一章、第五章、第八章由周阳花编写,第二章、第十二章、第十三章由周黎编写,第四章、第九章、第十章、第十一章和模拟试卷一、二、三由魏敏编写,第三章由王骏编写,第六章、第七章和模拟试卷四由陈丽芳编写,并由陈丽芳统稿。

本书在编排上注意完整性和独立性,紧扣考试大纲,叙述通俗易懂,既可以作为高等职业院校、高等本专科院校、成人高考及培训班学生学习《C 语言程序设计》的辅助教材,也可作为自学者或准备参加二级考试者的参考用书。

本书在编写过程中得到了江南大学信息工程学院诸多教师的支持和帮助,在此表示衷心的感谢。由于编者水平有限,缺点和错误在所难免,敬请广大读者指正。

本书中实验题没有提供答案,如果需要答案请发 E-mail 与我们联系索取,我们的 E-mail 地址是: cyuyan_2006@126.com 或 cyuyan@126.com。

编 者

2006 年 6 月

目 录

第1章 绪论

1.1 学习目标和要求	(1)
1.2 Turbo C 环境介绍	(1)
1.3 Turbo C 的出错信息和调试	(6)
1.4 Visual C ++ 环境介绍	(9)

第2章 算法

2.1 学习目标和要求	(13)
2.2 示例分析	(13)
2.3 练习题	(16)
2.4 实验题	(16)
2.5 参考答案	(17)

第3章 数据类型、运算符与表达式

3.1 学习目标和要求	(18)
3.2 示例分析	(18)
3.3 练习题	(22)
3.4 实验题	(25)
3.5 常见错误	(27)
3.6 参考答案	(28)

第4章 最简单的 C 程序设计——顺序程序设计

4.1 学习目标和要求	(29)
4.2 示例分析	(29)
4.3 练习题	(31)
4.4 实验题	(35)
4.5 常见错误	(35)
4.6 常用算法	(37)
4.7 参考答案	(37)

第5章 选择结构程序设计

5.1 学习目标和要求	(38)
-------------------	------



5.2	示例分析	(39)
5.3	练习题	(42)
5.4	实验题	(46)
5.5	常见错误	(47)
5.6	常用算法	(49)
5.7	参考答案	(50)
第6章 循环结构		
6.1	学习目标和要求	(51)
6.2	示例分析	(52)
6.3	练习题	(55)
6.4	实验题	(63)
6.5	常见错误	(64)
6.6	常用算法	(65)
6.7	参考答案	(68)
第7章 数组		
7.1	学习目标和要求	(69)
7.2	示例分析	(70)
7.3	练习题	(73)
7.4	实验题	(79)
7.5	常见错误	(80)
7.6	常用算法	(82)
7.7	参考答案	(85)
第8章 函数		
8.1	学习目标和要求	(87)
8.2	示例分析	(88)
8.3	练习题	(95)
8.4	实验题	(100)
8.5	常见错误	(101)
8.6	常用算法	(102)
8.7	参考答案	(106)
第9章 预处理命令		
9.1	学习目标和要求	(107)
9.2	示例分析	(107)
9.3	练习题	(109)
9.4	实验题	(113)

9.5 常见错误	(113)
9.6 参考答案	(114)

第10章 指针

10.1 学习目标和要求	(115)
10.2 示例分析	(116)
10.3 练习题	(120)
10.4 实验题	(127)
10.5 常见错误	(128)
10.6 常用算法	(132)
10.7 参考答案	(134)

第11章 结构体和共用体

11.1 学习目标和要求	(135)
11.2 示例分析	(136)
11.3 练习题	(140)
11.4 实验题	(148)
11.5 常见错误	(150)
11.6 参考答案	(153)

第12章 位运算

12.1 学习目标和要求	(154)
12.2 示例分析	(154)
12.3 练习题	(156)
12.4 实验题	(158)
12.5 常见错误	(158)
12.6 参考答案	(159)

第13章 文件

13.1 学习目标和要求	(161)
13.2 示例分析	(161)
13.3 练习题	(163)
13.4 实验题	(166)
13.5 常见错误	(167)
13.6 参考答案	(168)

附录

模拟试卷一	(171)
模拟试卷二	(181)



模拟试卷三	(190)
模拟试卷四	(199)



第 1 章

绪 论

1.1 学习目标和要求

1. 熟悉 Turbo C 环境, 熟练掌握在 Turbo C 窗口中进行程序编辑、编译、连接和运行等操作。
2. 学会分析程序出错的原因, 掌握 Turbo C 中的调试工具和调试技巧。
3. 熟悉 Visual C++ 环境, 熟练掌握在 Visual C++ 窗口中进行程序编辑、编译、连接、调试和运行等操作。

1.2 Turbo C 环境介绍

C 语言是一种编译性语言, 它的编译器和连接器都是程序, 这些程序负责把 C 语言编写的源代码转换成可执行代码。常用的 C 编译器在 DOS 环境下支持 Turbo C。

Turbo C 是美国 Borland 公司的产品, 该公司先后推出了 Turbo C 1.0、2.0 版本, 而后的 Turbo C/C++ 3.0 版本能够在 Windows 环境下使用, 并完全支持鼠标操作。由于我们普遍使用的教学环境是 Turbo C 2.0 版本, 所以下面的介绍以该版本为例。

Turbo C 只要求很小的系统资源, 它仅占用 384KB 内存, 其系统文件占用的磁盘空间不到 3MB。可运行于 IBM-PC 系列微机, 包括 XT、AT 及 IBM 兼容机。要求 DOS 2.0 或更高版本支持, 并至少需要 448K 的 RAM, 可在任何彩色、单色显示器上运行。

Turbo C 向用户提供了一个集成开发环境 (IDE), 把程序的编辑、编译、连接和运行等操作全部集中在一个界面上, 使用非常方便。在使用之前, 必须先将 Turbo C 系统安装好, 用户可指定安装路径, 其系统默认的路径为 C:\TC。该路径下主要有 tc.exe、tcc.exe 等可执行文件和 include、lib (分别包含各种头文件和库函数文件) 两个子目录。

1. 启动 Turbo C。

在 TC 主目录中有 tc.exe 可执行文件, 可以在 DOS 平台下启动, 也可以在 Windows 平台下启动。

(1) 由 DOS 平台进入 Turbo C。

在命令提示符下, 使用下面的 DOS 命令进入 Turbo C。

C:\>cd tc (将当前目录改为 C:\>tc)



C:\tc > tc (执行 tc. exe 可执行文件)

以上带有下划线的部分是需要用户输入的命令。

注意：若用户将程序安装在自己指定的目录，则进入指定的目录后，再运行 tc. exe 即可。

(2) 由 Windows 平台进入 Turbo C。

① 在桌面上单击“开始”→“程序”→“MS-DOS”，进入 DOS 方式，键入上述 DOS 命令，进入 Turbo C 环境。

② 在“资源管理器”中找到相应的文件夹 TC，双击 tc. exe，即可进入 Turbo C 环境。

③ 在桌面上创建 tc. exe 文件的快捷方式，以后每次进入时，只需双击桌面上的“tc. exe”图标即可。

2. Turbo C 的界面。

启动 Turbo C 后，屏幕正中会显示 Turbo C 的版本信息框(图 1-1)，标明其版本号、生产日期和公司名称。用户只需按任意键，此版本信息框就会关闭。

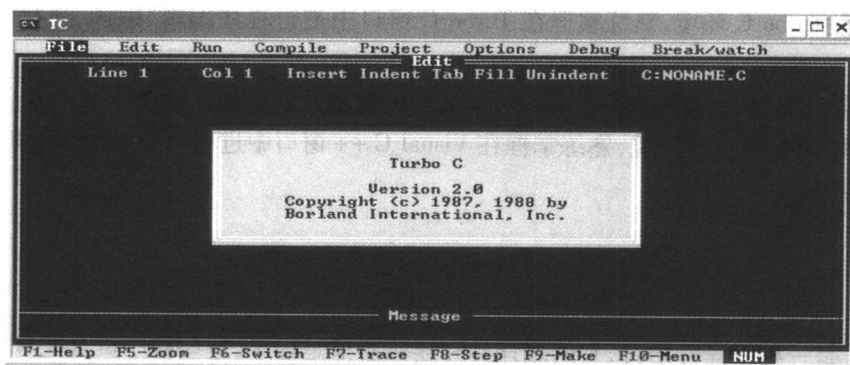


图 1-1

启动后的 Turbo C 窗口如图 1-2 所示。

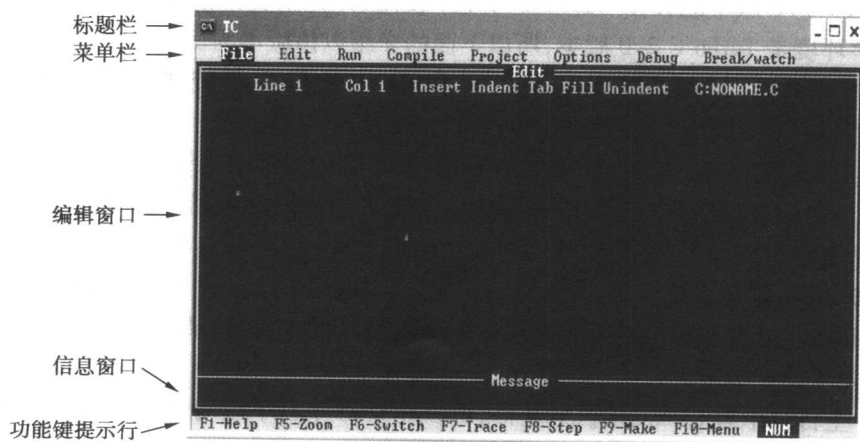


图 1-2

Turbo C 窗口包括如下内容：

(1) 菜单栏。

包括 File(文件)、Edit(编辑)、Run(运行)、Compile(编译)、Project(项目)、Option(选项)、Debug(调试)、Break/watch(断点/监视)主菜单,每一个主菜单还有对应的子菜单,通过菜单可实现相应的操作。

(2) 编辑窗口。

正上方有 Edit 字符作为标志。编辑窗口的作用是对 Turbo C 源代码进行输入和编辑。源代码程序在这个窗口显示。该窗口上方有一行说明性标志,如 Line 1 和 Col 1,表示当前光标的位置。在该行的最右边显示当前正在编辑的源程序文件名(如默认的 NONAME.C)。

(3) 信息窗口。

正上方有 Message 字符作为标志,用来显示编译和连接时的有关信息。

(4) 功能键提示行。

在信息窗口下面,显示了一些功能键的作用,如:

- ① F1-Help(帮助): 显示帮助信息。
- ② F5-Zoom(分区控制): 将当前激活窗口(编辑窗口或信息窗口)满屏显示。
- ③ F6-Switch(转换激活状态): 将当前激活窗口在编辑窗口和信息窗口中进行转换。
- ④ F7-Trace(跟踪命令): 用于跟踪程序的运行情况。
- ⑤ F8-Step(单步执行): 按一次 F8 键执行一个语句。
- ⑥ F9-Make(进行编译和连接): 生成 .obj 文件和 .exe 文件,但不运行程序。
- ⑦ F10-Menu(菜单): 回到主菜单。

(5) 编辑命令。

常用的编辑命令简介如下:

PageUp	向前翻页
PageDn	向后翻页
Home	将光标移到所在行的开始
End	将光标移到所在行的结尾
Ctrl + Y	删除光标所在的一行
Ctrl + N	插入一空行
Ctrl + T	删除光标所在处的一个词
Ctrl + KB	设置块开始
Ctrl + KK	设置块结尾
Ctrl + KV	块移动
Ctrl + KC	块拷贝
Ctrl + KY	块删除
Ctrl + KR	读文件
Ctrl + KW	存文件
Ctrl + KP	块文件打印
Ctrl + F1	如果光标所在处为 Turbo C 2.0 库函数,则获得有关该函数的帮助信息

3. Turbo C 的使用。

(1) 编辑一个新文件。

- ① 按 F10 键激活“File”主菜单,按回车键(或 Alt + F 组合键)弹出子菜单,选择子菜单

“New”，回车后进入编辑状态。光标定位在第一行第一列。

② 编辑过程中，可用 Ins 键转换插入/改写状态，用 Del 键删除字符。

③ 编辑过程中，灵活使用编辑命令可以加快编辑速度。

(2) 保存文件。

按 F10 键激活“File”主菜单，并按回车键，选择子菜单“Save”，按回车键后，TC 会弹出一个对话框，如图 1-3 所示。用户可指定文件名并保存在用户工作目录中。如果用户指定目录路径，则文件存入指定目录中。

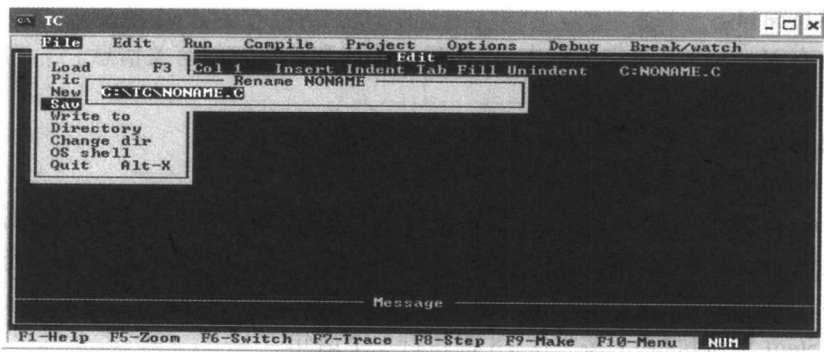


图 1-3

用户也可改变工作目录，方法是选择“File”主菜单中的“Change dir”子菜单，回车后就会弹出新目录输入框，提示用户输入自己选择的（或建立的）工作目录名。

注意：输入的目录必须已存在，否则系统会报错。

(3) 打开一个已存在的文件。

选择“File”主菜单中的“Load”子菜单，这时屏幕上会出现一个 *.c 的“装入文件”对话框，要求用户输入准备调入的文件路径和文件名。假设想打开的文件是 TC 目录下的 file 子目录中的 fl.c 文件，则输入文件名，如图 1-4 所示。

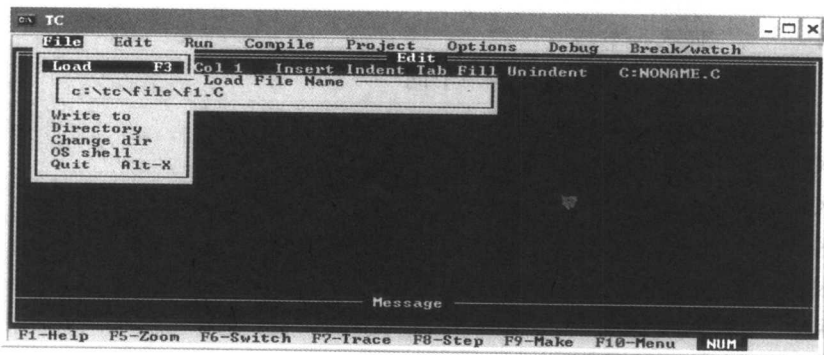


图 1-4

(4) 编译和连接。

编辑好源程序后，可用以下方法中的任一种，对源程序进行编译、连接。

① 按 F10 键，选中主菜单“Compile”后按回车键，选择“Compile to OBJ”，按回车键就可进行编译。编译完成后，再选择“Compile”中的“Link EXE file”，然后按回车键，就可执行连



接,如图 1-5 所示。

- ② 选择“Compile”中的“Make EXE file”,按回车键后一次完成编译和连接。
- ③ 按 F9 键可一次完成编译和连接。

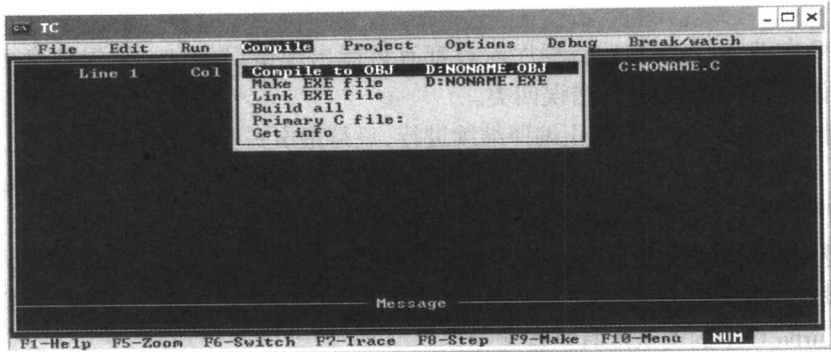


图 1-5

(5) 运行。

编译、连接好程序后,可用以下方法中的任一种,运行该程序。

- ① 选择“Run”中的“Run”,按回车键后即可运行。
- ② 按 Ctrl + F9 组合键即可运行。此方法将编译、连接和运行一起完成。

(6) 查看结果。

按 Alt + F5 组合键可观看运行结果,按任一键可返回编辑状态。

(7) 配置 TC 工作环境。

按 F10 键,选择“Options”主菜单中的“Directories”子菜单,按回车键后弹出如图 1-6 所示的窗口。

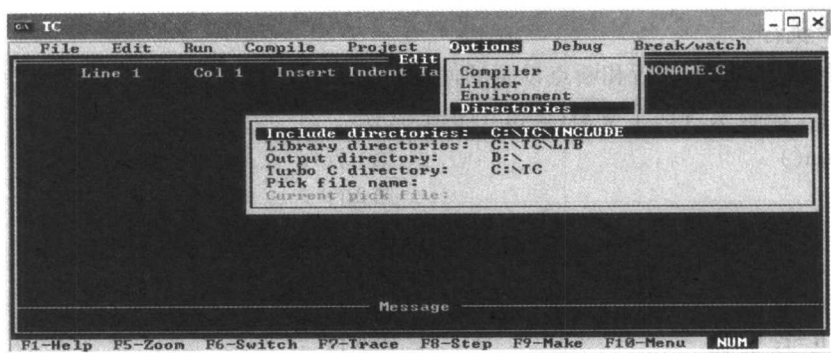


图 1-6

其中:

- | | |
|------------------------------------|-------------------------|
| Include directories: C:\TC\INCLUDE | 表示头文件的默认目录 |
| Library directories: C:\TC\LIB | 表示库文件默认目录 |
| Output directory: D:\ | 表示 *.obj 和 *.exe 文件存放目录 |
| Turbo C directory: C:\TC | 表示安装建立的默认目录 |
- 用户若修改完以上选项,还应保存这些信息。方法是选择“Options”主菜单中的“Save

options”子菜单,按回车键。以后上机不需要再重新配置。

1.3 Turbo C 的出错信息和调试

Turbo C 在编译、连接和运行程序阶段都有可能出现错误信息。错误信息可以分成警告、一般错误、严重错误和逻辑错误四类。

(1) 警告(warning)。不阻止编译继续进行,只是指出一些值得怀疑的地方。

(2) 一般错误(error)。包括语法错误、磁盘或内存存取错误、命令行错误等。

(3) 严重错误(fatal error)。通常是内部编译错误。

(4) 逻辑错误。指的是应用程序可以正常运行,但程序得不到正确的结果。这类错误是由程序设计本身存在逻辑缺陷造成的,所以查错、纠错的难度较大。这时,我们要灵活机动地使用 Turbo C 中的调试工具来排除错误。

前三种出错信息出现时,信息窗口中会显示文件名、出错行号以及出错代码和提示信息。

Turbo C 中提供的调试工具包括单步执行、设置断点、添加监视等,打开“Break/watch”(断点及监视表达式)菜单可以看到所提供的调试功能。该菜单有以下内容:

Add watch	向监视窗口中插入一监视表达式
Delete watch	从监视窗口中删除当前的监视表达式
Edit watch	在监视窗口中编辑一个监视表达式
Remove all watches	从监视窗口中删除所有的监视表达式
Toggle breakpoint	对光标所在的行设置或清除断点
Clear all breakpoints	清除所有断点
View next breakpoint	将光标移动到下一个断点处

1. 单步执行。

下面举例说明单步执行和断点设置的使用。

【题目】 编程实现两个变量(a 和 b)值的交换。

main()	第1行
{	第2行
int a,b,t;	第3行
a=1;b=3;	第4行
t=a;	第5行
b=t;	第6行
a=b;	第7行
printf("a=%d,b=%d\n",a,b);	第8行
}	第9行

分析: 这个源程序没有任何语法错误,能够顺利通过编译和连接。但是运行结果为 a=1, b=1;而正确结果应该为 a=3,b=1。这时可以采用单步执行和添加变量监视的方法,检查每一步的正确性。

(1) 选择“Break/watch”主菜单的“Add watch”子菜单,在弹出的“Add watch”对话框中输入变量 a,然后按回车键,如图 1-7 所示。用同样的方法分别添加监视变量 b 和 t。

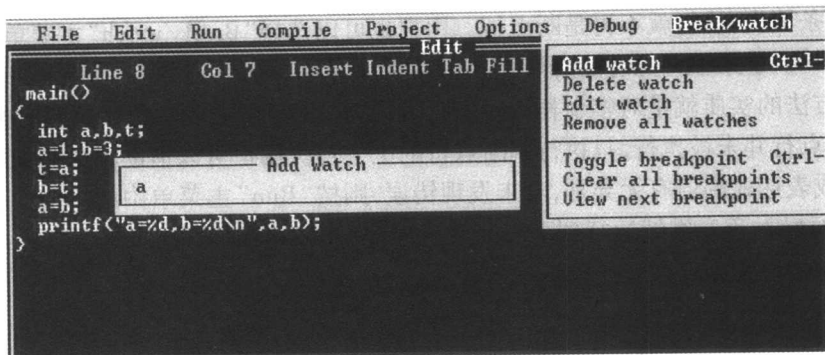


图 1-7

(2) 开始运行,按 F8 键,可以看到编辑窗口中的源程序中的第 1 行处呈现高亮度显示,表示准备进入 main 函数。同时可以看到屏幕下部的 Message 窗口变成了 Watch 窗口,它用来观察所监视的变量值的变化。

(3) 每按 F8 一次,程序运行一行。当程序运行完第 4 行后,在 Watch 窗口中观察到变量 a 和 b 的值分别为 1 和 3;运行完第 5 行后,变量 t 获得变量 a 赋给的值 1;运行完第 6 行和第 7 行后,变量 b 和变量 a 的值都被更新为 1,如图 1-8 所示。

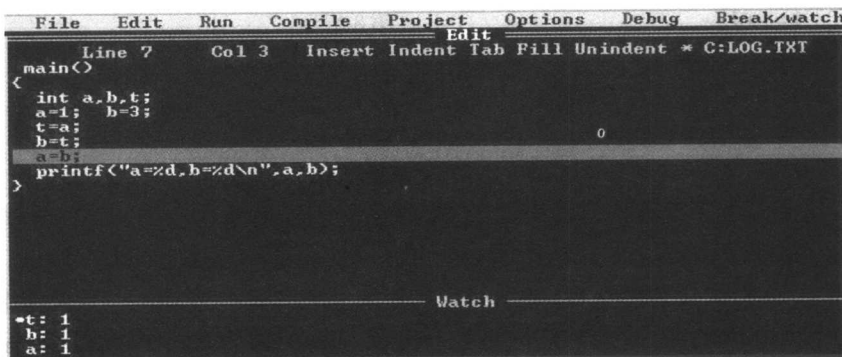


图 1-8

(4) 根据观察变量值的变化情况,可以分析得出逻辑错误出现在第 6 行和第 7 行,这是因为变量 b 原来的值 3 没有得到保留就被更新为 1,所以在第 7 行变量 b 赋给变量 a 的值是更新后的 1,而不是原来的 3。

(5) 根据出错的原因进行修改,将第 7 行和第 6 行代码进行交换,即在变量 b 原来的值 3 赋给变量 a 之后再更新为 1。

(6) 程序再次运行,结果正确。

2. 断点设置。

单步执行的方法虽然能够有效地逐行检查错误,但对于较长的程序,运行步骤多,较费时。我们可以在程序中采用设置若干断点的方法快速找到出错处。

设置断点的方法是:将光标移到某一行,然后选择“Break/watch”主菜单中的“Toggle breakpoint”子菜单或者按 Ctrl + F8 快捷键,此行就以红色条显示,即为断点行。用此方法,可在程序中设置若干断点行。若想取消断点行,则将光标移到断点行,再按一次 Ctrl + F8 快

按键,颜色条消失,该行就不再是断点行。用户也可以选择“Break/watch”主菜单中的“Clear all breakpoints”子菜单,一次消除程序中所有的断点行。

这种方法的实质就是对整个长程序进行分段,逐段检查错误。程序运行到断点处会暂停,注意断点行并未被执行,只执行到断点行的上一行。用户可以借助添加监视等方法分析此时变量或表达式的值有无异常,若未发现错误,则按“Run”主菜单中的“Run”子菜单或按 Ctrl + F9 快捷键,程序继续运行到下一断点。如此一段一段地检查。

下面举例说明断点在程序调试中的使用。

【题目】 求 $s = \frac{1}{10} \sum_{i=1}^{10} i$ 。

程序如下:

main()	第1行
{	第2行
int i, sum = 0; float ave;	第3行
i = 1;	第4行
do	第5行
{	第6行
sum = sum + i;	第7行
i = i + 1;	第8行
} while(i <= 10);	第9行
ave = sum/10;	第10行
printf("ave = %f\n", ave);	第11行
}	第12行

分析:这个源程序没有任何语法错误,能够顺利通过编译和连接。程序经运行结果为 ave = 5,而正确结果应为 5.5。我们用设置断点和添加监视的方法进行程序调试。

(1) 将第10行设置为断点行,并添加监视变量 sum 和 ave,按 Ctrl + F9 快捷键使程序运行,程序运行到第10行暂停,此时观察到监视变量 sum 的值为 55。由此可知,程序第1行到第9行未出错。

(2) 按 F8 功能键,程序第10行被运行,观察到变量 ave 的值变为 5,如图 1-9 所示。由此可知,错误出在第10行,分析原因为运算符“/”左右两个操作数均为整型,表达式 sum/10 的值也只能为整型,故将第10行改为“ave = sum/10.0;”。

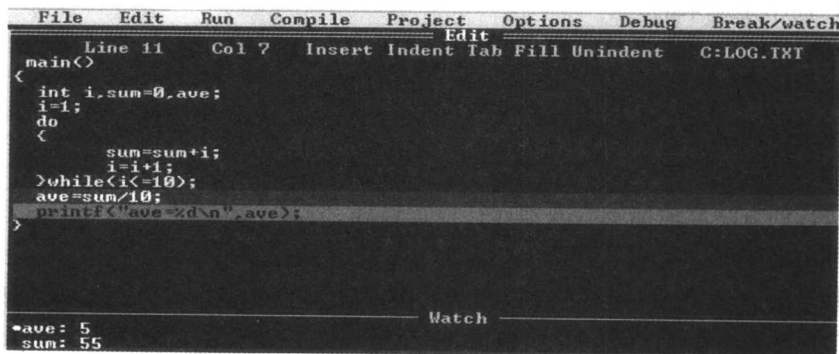


图 1-9

