



教育科学“十五”国家规划课题研究成果

# C 语言程序设计基础 实验教程

廖雷 罗代忠 主编



高等教育出版社

### 图书在版编目 (CIP) 数据

C 语言程序设计基础实验教程 / 廖雷, 罗代忠主编.  
—北京: 高等教育出版社, 2005. 7  
ISBN 7-04-016705-0

I. C... II. ①廖... ②罗... III. C 语言-程序设计  
—高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 056765 号

策划编辑 付 欣 责任编辑 韩 飞 封面设计 刘晓翔  
版式设计 王 莹 责任校对 尤 静 责任印制 韩 刚

出版发行 高等教育出版社  
社 址 北京市西城区德外大街 4 号  
邮政编码 100011  
总 机 010-58581000

购书热线 010-58581118  
免费咨询 800-810-0598  
网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>

经 销 北京蓝色畅想图书发行有限公司  
印 刷 高等教育出版社印刷厂

开 本 787×960 1/16  
印 张 12  
字 数 210 000

版 次 2005 年 7 月第 1 版  
印 次 2005 年 7 月第 1 次印刷  
定 价 15.50 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 16705-00

## 内容提要

本书是教育科学“十五”国家规划课题的研究成果，是《C 语言程序设计基础》一书的实验配套教程，它为 C 语言初学者上机实验提供指导。全书提供了 13 个方面的验证性、设计性实验，对验证性实验给出了详细的参考实验步骤和相应的程序代码，并对实验进行了适度的拓展分析，穿插讲解了一些常用算法和具有实用价值的程序；教程中还提供了 2 个课程设计，详细讲解实例，循序渐进地启发学生完成课程设计，培养学生实际分析问题、编写程序和动手能力。

本书注重加强学生综合应用能力和创新能力的培养，教程设计的主线是验证性实验、程序分析类实验、设计创作实验。本书以验证性实验为基础，以分析与设计型实验为支撑将课程内的知识相互渗透、融会贯通，从而提高学生综合应用能力和创新能力。编者力求体现编排合理、循序渐进、深入浅出、讲解通俗，使读者在实验中掌握 C 语言程序设计的基础知识和应用。

本书可以作为高等学校工科各专业的实验教程，也可作为计算机培训和计算机等级考试辅导的教学用书，还可供广大程序开发人员和自学 C 语言程序设计上机实验的读者参考。

# 前 言

如何培养满足市场需求的应用型人才,是我们应用型本科院校探索和研究  
的课题,在教育科学“十五”国家规划课题立项研究的基础上,在高等教育出  
版社的支持下,作者编写了这本《C 语言程序设计基础实验教程》,作为《C 语  
言程序设计基础》教材的配套实验教程。

本教程的读者对象主要为高等院校工科应用型各专业学生。针对应用型人才  
培养的目标要求,注重读者学习语言是一个循序渐进的过程,因此本教程的  
编写特色主要体现在:

① 以验证性实验为基础实验,学生入门的门槛要求低。对初学程序设计的  
读者来说,自己独立完成一个完整的程序设计有一定的难度,本教程提供给读  
者初始部分是验证性实验。

② 设计分析性实验和创作性实验注重读者程序分析、设计能力的培养,注  
重读者分析问题、解决问题的能力培养,能高效地提高读者编程应用能力。

③ 特设课程设计实验。语言课程注重边学边练,由于课堂教学的深度和广  
度的限制,练习的深度有一定限制,因此结合实际应用,特设计了两个课程设  
计,以进一步锻炼读者动手解决实际问题的能力。

另外,在编写过程中,为了使读者易于理解、分析程序代码,特对程序代  
码加上了行号。

全书共分 4 章:第一章是基本 C 语言程序设计;第二章 C 语言界面控制管  
理实验;第三章是课程设计;第四章是其他 C 语言集成开发环境简介。本书由  
廖雷、罗代忠主编,陈锦涛参编。廖雷和罗代忠共同拟定全书框架,明确写作  
思路,统稿工作主要由罗代忠完成。全书各章的具体分工如下:第一章 1.1~1.8  
由罗代忠编写,第一章 1.9、第二章 2.1、第四章由廖雷编写,第一章 1.10~1.11、  
第二章 2.2~2.3 和第三章 3.1 由陈锦涛编写,第三章 3.2 由陈锦涛、罗代忠和廖  
雷共同编写。本书由南京大学副教授刘友华博士主审,他提出了很多宝贵意见,  
编者深表感谢。

在本书的编写过程中得到全国高等学校教学研究中心、高等教育出版社和  
编者所在学校的热情支持和指导,在此表示衷心的感谢。

本书中存在的不足和不当之处,敬请读者批评指正。

编者

2005 年 3 月

## 郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

**反盗版举报电话：**(010) 58581897/58581896/58581879

**传 真：**(010) 82086060

**E - mail：**dd@hep.com.cn

**通信地址：**北京市西城区德外大街4号

高等教育出版社打击盗版办公室

**邮 编：**100011

**购书请拨打：**(010)58581118

# 目 录

|                          |    |
|--------------------------|----|
| 第一章 基本 C 语言程序设计          | 1  |
| 1.1 C 语言程序的编辑与调试         | 2  |
| 1.1.1 C 语言源程序的基本结构       | 3  |
| 1.1.2 Turbo C 2.0 集成环境简介 | 4  |
| 1.1.3 C 语言程序调试           | 5  |
| 1.1.4 TC 环境调用帮助系统        | 7  |
| 1.1.5 建立并运行一个工程文件        | 7  |
| 1.2 程序流程图                | 9  |
| 1.2.1 程序流程图的基本符号         | 9  |
| 1.2.2 程序流程图实例            | 10 |
| 1.2.3 C 语言程序实例           | 10 |
| 1.2.4 熟悉 TC 环境下运行 C 程序实验 | 11 |
| 1.3 数据类型、运算符、表达式及输入/输出实验 | 12 |
| 1.3.1 实验目的及要求            | 12 |
| 1.3.2 实验内容及步骤            | 12 |
| 1.4 分支结构实验               | 18 |
| 1.4.1 实验目的及要求            | 18 |
| 1.4.2 实验内容及步骤            | 18 |
| 1.5 循环结构实验               | 27 |
| 1.5.1 实验目的及要求            | 27 |
| 1.5.2 实验内容及步骤            | 27 |
| 1.6 函数实验                 | 36 |
| 1.6.1 实验目的及要求            | 36 |
| 1.6.2 实验内容及步骤            | 37 |
| 1.7 数组实验                 | 44 |
| 1.7.1 实验目的及要求            | 44 |
| 1.7.2 实验内容及步骤            | 44 |
| 1.8 指针实验                 | 52 |
| 1.8.1 实验目的及要求            | 52 |
| 1.8.2 实验内容及步骤            | 52 |

|                                    |            |
|------------------------------------|------------|
| 1.9 库函数及改错分析 .....                 | 61         |
| 1.9.1 实验目的及要求 .....                | 61         |
| 1.9.2 实验内容及步骤 .....                | 61         |
| 1.10 结构与杂类实验 .....                 | 68         |
| 1.10.1 实验目的及要求 .....               | 68         |
| 1.10.2 实验内容及步骤 .....               | 68         |
| 1.11 文件操作实验 .....                  | 82         |
| 1.11.1 实验目的及要求 .....               | 82         |
| 1.11.2 实验内容及步骤 .....               | 82         |
| <b>第二章 C 语言界面控制管理实验 .....</b>      | <b>93</b>  |
| 2.1 文本屏幕管理实验 .....                 | 93         |
| 2.1.1 实验目的及要求 .....                | 93         |
| 2.1.2 实验内容及步骤 .....                | 93         |
| 2.2 图形屏幕管理实验 .....                 | 102        |
| 2.2.1 实验目的及要求 .....                | 102        |
| 2.2.2 实验内容及步骤 .....                | 103        |
| 2.3 界面设计 .....                     | 124        |
| 2.3.1 实验目的及要求 .....                | 124        |
| 2.3.2 实验内容 .....                   | 124        |
| <b>第三章 课程设计 .....</b>              | <b>137</b> |
| 3.1 课程设计的目的 .....                  | 137        |
| 3.2 课程设计内容 .....                   | 137        |
| 3.2.1 设计制作一个“学生成绩管理系统” .....       | 137        |
| 3.2.2 汉诺塔演示实验 .....                | 154        |
| <b>第四章 其他 C 语言集成开发环境简介 .....</b>   | <b>170</b> |
| 4.1 Turbo C++集成开发环境 .....          | 170        |
| 4.1.1 Turbo C++概述 .....            | 170        |
| 4.1.2 Turbo C++ 3.0 主屏幕简介 .....    | 170        |
| 4.1.3 Turbo C++ 3.0 常用编辑命令简介 ..... | 171        |
| 4.1.4 配置工作环境 .....                 | 172        |
| 4.1.5 调入、编辑和保存源程序 .....            | 173        |
| 4.1.6 编译和链接 .....                  | 173        |
| 4.1.7 运行可执行程序 .....                | 174        |
| 4.1.8 调试 .....                     | 174        |
| 4.1.9 其他 .....                     | 174        |

---

|                            |     |
|----------------------------|-----|
| 4.2 Visual C++集成开发环境 ..... | 175 |
| 4.2.1 Visual C++概述 .....   | 175 |
| 4.2.2 Visual C++主屏幕 .....  | 175 |
| 4.2.3 源程序的建立、编辑和保存 .....   | 176 |
| 4.2.4 编译和链接 .....          | 177 |
| 4.2.5 运行可执行程序 .....        | 178 |
| 4.2.6 调试 .....             | 178 |
| 4.2.7 其他 .....             | 178 |
| 参考文献 .....                 | 179 |



# 第一章 基本 C 语言程序设计

学习 C 语言程序设计, 必须在编程上多下工夫, 多进行上机实践, 在实践中积累编程经验, 上机实验的目的要从验证程序的基础向设计型实验和综合型的课程设计实验转化, 力求达到:

1. 熟悉并掌握 Turbo C 2.0 环境下运行 C 语言程序的基本步骤和方法;
2. 熟悉并掌握 Turbo C 2.0 环境下调试 C 语言程序的方法与技巧, 能够有效地根据程序编译过程中、连接程序过程中出现的编译错误或连接错误进行分析和定位错误性质, 并能及时改正错误;
3. 保证程序正确并能得到正确的运行结果;
4. 能够运用基本知识独立进行设计型和综合型实验创作, 掌握小型软件设计的基本思想和方法, 能保证自己创作设计的实验程序正常运行。

无论多么小心仔细, 在开始编写 C 语言源程序时总会有一个或多个错误, 因此调试程序是创建正确程序的必要手段。编译无误的程序代码并不表示一定运行正常, 能一次运行正常的程序也并不表示每次都能运行正确。调试程序的目的是纠出程序中这些错误, 保证程序的正确性。程序中常见的错误类型主要有以下两种:

一类是编译类错误, 主要体现在语法错误和词法错误上。在编译程序的过程中会发现编写过程中不按照 C 语言语法规则编写程序代码而造成的语法错误和程序中使用未知的语言元素造成的词法错误, 这些错误在程序编译过程中会显示语法或词法错误类型、相应的错误代码以及错误代码的行号, 通过这些提示信息, 能定位程序中的错误并对源程序中的这些编译类型错误进行修改。

另一类是逻辑错误或运行的错误。运行错误是程序代码在执行时才会产生的错误, 逻辑错误是程序代码所运行产生的结果不是预期的结果。程序中潜在的逻辑错误在程序编译过程中是不能检查出来的, 只有通过程序运行结果才能体现出来, 在对程序运行结果与预期结果进行分析时, 如果程序运行结果与预期结果有差异, 就必须分析和检查源程序中存在的逻辑错误并进行相应的修改。

进行上机实验前应进行的相关准备工作:

- ① 分析任务, 确立解决问题的算法, 按照算法编写好相应的 C 语言源程序;
- ② 准备好需要输入进行测试的数据;

- ③ 预测程序中可能得到的运行结果;
- ④ 编写实验前相应的预习实验报告。

## 1.1 C 语言程序的编辑与调试

与其他编译执行方式的高级语言处理程序一样, C 语言程序的开发过程也要经过编辑、编译、链接、运行以及调试等环节。对于编写 C 语言程序的开发人员来说, 掌握这些环节有助于开发人员编写及调试 C 语言程序。使用 C 语言程序上机一般需要经过以下 4 个步骤:

### 1. 通过编辑, 建立或修改 C 语言源程序

用户通过编辑, 将 C 语言源程序输入计算机并以磁盘文件的形式保存在磁盘上, 这些 C 语言源程序文件的扩展名默认为 .C。C 语言源程序可在 Turbo C 2.0 的环境下进行编辑, 并可对源程序进行插入、删除、修改、复制、移动、保存、打开文件等编辑操作。

### 2. 对源程序进行编译, 生成目标程序

用户在 Turbo C 2.0 环境下编辑的 C 语言源程序, 在计算机上不能直接执行, 必须首先翻译成用特定机器语言描述的目标程序, 这种翻译过程称之为编译。编译过程首先检查 C 源程序中的语法错误或词法错误, 在保证没有错误的情况下, 将 C 语言源程序生成计算机能识别的扩展名为 .OBJ 文件的目标程序; 如果在编译过程中发现 C 语言源程序有语法错误或词法错误, 必须进行修改正确后重新编译才能生成目标程序, 所生成的目标程序文件的基本名与 C 语言源程序文件的基本名相同。

### 3. 链接、装配目标程序, 形成可执行程序

将 C 语言源程序进行编译生成的目标程序并不能直接在计算机上执行, 还必须对目标程序进行链接、装配, 形成可执行程序, 才能在计算机上运行。一般情况下, C 语言源程序由一个独立的源程序模块组成, 可以直接经过链接产生可执行程序; 但对于一个相对较复杂的任务来说, 可能将任务分解成多个独立的程序模块完成, 为此必须首先将这些独立的程序模块分别进行编辑、编译, 生成相应的目标程序, 然后将这些目标程序进行链接、装配, 生成预定任务的可执行程序。

### 4. 调试、运行可执行程序

链接、装配目标程序形成的可执行程序就可以运行, 运行可执行程序就如使用 DOS 的外部命令一样, 在 DOS 提示符下键入相应的可执行程序文件名即可执行该程序; 也可在 Turbo C 2.0 环境下直接执行经编译、链接后生成的可执行程序。

如果程序运行结果与用户预期的结果相一致，证明程序是正确的；如果程序的运行结果与用户预期的结果有差异，源程序中就会有错误存在，需要对 C 语言源程序进行修改、再调试，消除源程序中潜在的错误，直到程序的运行结果与用户的预期结果一致，调试工作就可以结束。调试是保证程序正确的有效手段，是初学者编写 C 语言程序必备的基本技能。

C 语言程序的上机过程一般需经过编辑、编译、链接（装配）、运行（调试）4 个阶段，这 4 个阶段构成一个循环，保证用户编写的程序正确。4 个阶段的关系如图 1.1 所示。

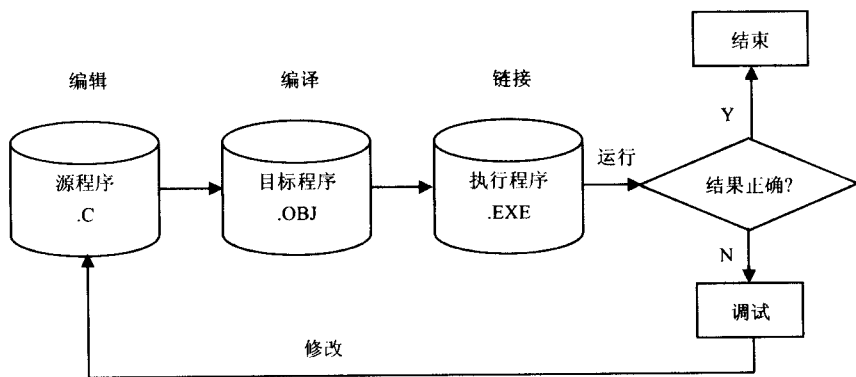


图 1.1 C 语言源程序编辑、编译、链接、运行流程

### 1.1.1 C 语言源程序的基本结构

一个完整的 C 语言源程序应该包含头文件、用户自定义文件、自定义函数、主函数 `main()` 等部分。

【例 1-1】显示输出一个字符串。

相应的 C 语言源程序 `Dispstr.c` 程序的清单如下：

```
#include<stdio.h>                                /*包含文件或用户自定义头文件*/
main()                                             /*主函数*/
{
    /*函数体*/
    printf("This is my first program of C!\n"); /*显示字符串*/
}
```

这是一个完整的 C 语言源程序，程序中以 “/\*” 开头到 “\*/” 结束之间的内容是注释部分，程序在编译时不产生目标代码。

“`#include<stdio.h>`” 的 “`stdio.h`” 中包含了标准输入/输出库函数，有该语

句后，用户就可以在后面的函数中使用“stdio.h”中包含的标准输入/输出库函数。

C 语言源程序由函数组成，每一个函数完成相对独立的功能，源程序中可以包含多个用户定义函数，但一个完整的 C 语言源程序必须有并且只能有一个主函数，即 main() 函数，用户函数可以放在主函数之前或之后，都不会影响程序的执行结果。不管源程序中有多少个用户函数，也不管用户函数放在 main() 函数之前或之后，程序总是从 main() 函数开始执行，即 main() 函数是 C 语言程序的执行入口点，并且也是在 main() 中结束程序。

main() 函数后面有一对“{}”，“{}”中的内容是函数主体内容，称之为函数体，“{}”必须成对出现。函数实现的功能由函数体来实现，一般情况下，函数体由“说明部分”和“执行部分”两部分组成。“说明部分”用于定义数据变量，而“执行部分”是函数的功能实现，由多个语句组成。例 1-1 中的函数体只有“执行部分”而没有“说明部分”。

在 main() 函数的“执行部分”，各个语句后面必须以“;”结束，也就是说，“;”是一个语句的结束符。例 1-1 中的函数体“执行部分”只有一个函数调用语句“printf()”，printf() 函数是标准的输出库函数，该程序的功能是在显示器上输出括号中指定的字符串“This is my first program of C!”。

### 1.1.2 Turbo C 2.0 集成环境简介

#### 1. 启动 Turbo C 2.0

在 Turbo C 2.0 安装目录下运行 TC.EXE 可执行文件将启动 Turbo C 2.0 集成环境，启动后的界面如图 1.2 所示，Turbo C 2.0 集成开发环境集编辑程序、调试程序及其他实用程序于一体，无需独立的编辑、调试、编译、链接程序，在该环境下能建立并运行 C 程序。

#### 2. 创建一个新的 C 语言源程序

选择 File 菜单下的 New 命令，然后按回车键，将弹出一个编辑窗口 Edit 和信息窗口 Message，同时编辑窗口和信息窗口是空白，光标定位在编辑窗口第 1 行第 1 列等待用户输入，源程序文件的默认名称为 NONAME.C，如图 1.2 所示。

#### 3. 存盘

编辑完后，选择 File 菜单下的 Save 命令进行存盘，在弹出的窗口中输入相应的文件名。注意：文件名的后缀为.C（或直接按 F2 功能键），如果当前修改的文件有文件名并希望使用原文件名存盘，用户不需输入文件名而直接保存；如果用户希望换名保存文件，则需输入相应的文件名。

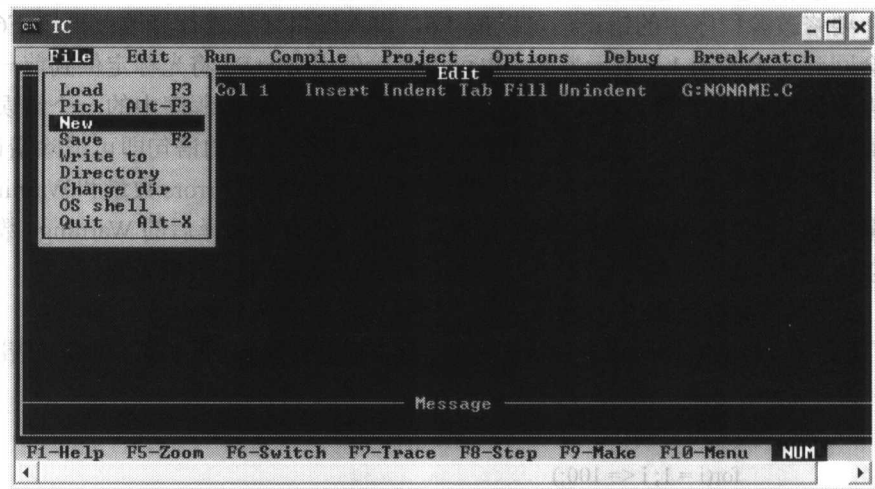


图 1.2 建立新文件

### 1.1.3 C 语言程序调试

#### 1. 纠正语法错误

通过编译程序，程序中存在的错误将在信息窗口 Message 中显示出错误和警告信息。如图 1.3 所示。Message 窗口中显示错误信息和错误代码，通过移动 Message 窗口中的亮条可以定位编辑窗口 Edit 中相应错误的位置，用户就可以进行相应的错误修改。

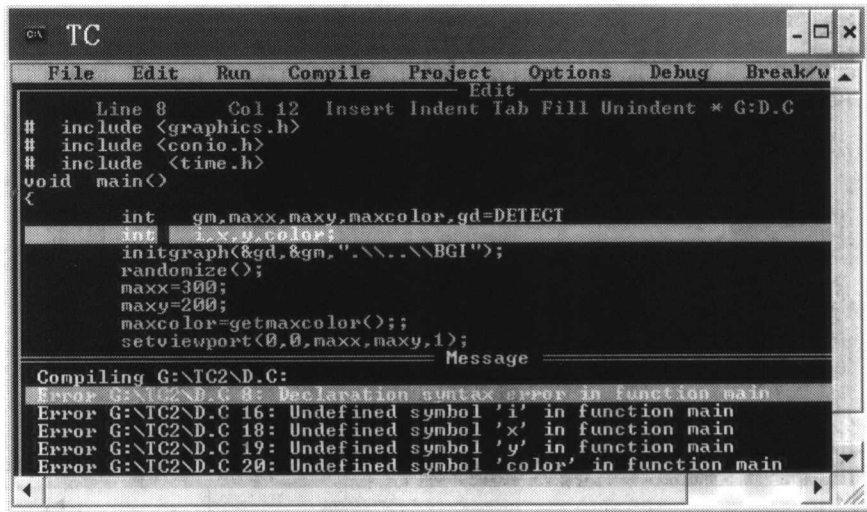


图 1.3 编译错误提示

要修改源程序中的错误,可以按 F6,进入编辑环境进行修改;再按 F6,又会回到原来的图 1.3 所示界面。也可以直接使用 Alt+F7 将光标定位到前一个错误或使用 Alt+F8 将光标定位到后一个错误位置。有时源程序中的一个错误将导致程序中多个错误信息产生,只要修改程序中导致错误的语句即可消除错误。

当程序纠正错误后再进行编译,直到 Message 窗口中 Errors 数量和 Warnings 数量为 0,程序则编译通过。但程序要正常运行,有时编译后的 Warnings 数量可以不为 0,为保证程序的正确,最好使编译后 Warnings 数量为 0。

## 2. 跟踪运行错误

【例 1-2】在 Turbo C 2.0 环境下调试、跟踪查看变量 i 与累加和 s 值变化情况。

```
main()
{
    int s = 0, i;
    for(i = 1; i <= 100;)
    {
        s = s + i;
        i = i + 1;
    }
    printf("1+2+3+...+100=%d", s);
}
```

如果想在调试过程中查看循环过程中 i 与累加和 s 的变化情况,可以选择 Break/watch → Add watch → 在 Add watch 窗口中输入 i,再选择 Break/watch → Add watch → 在 Add watch 窗口中输入 s, s 和 i 显示在 Watch 窗口中,如图 1.4

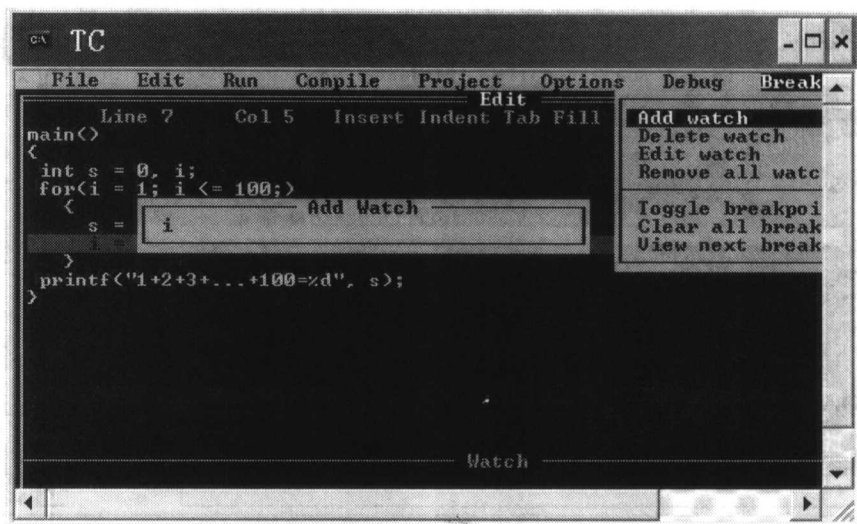


图 1.4 增加变量值监视窗口

所示；当用 F8 单步跟踪调试时，s 和 i 值的变化将在 Watch 窗口中显示出来，如图 1.5 显示的是当 i=3 时，s=6 的情况，用户可以通过跟踪 i 和 s 的值的变化来确认自己编写的程序运行是否正确，这是 C 语言中调试程序最常使用的方法。

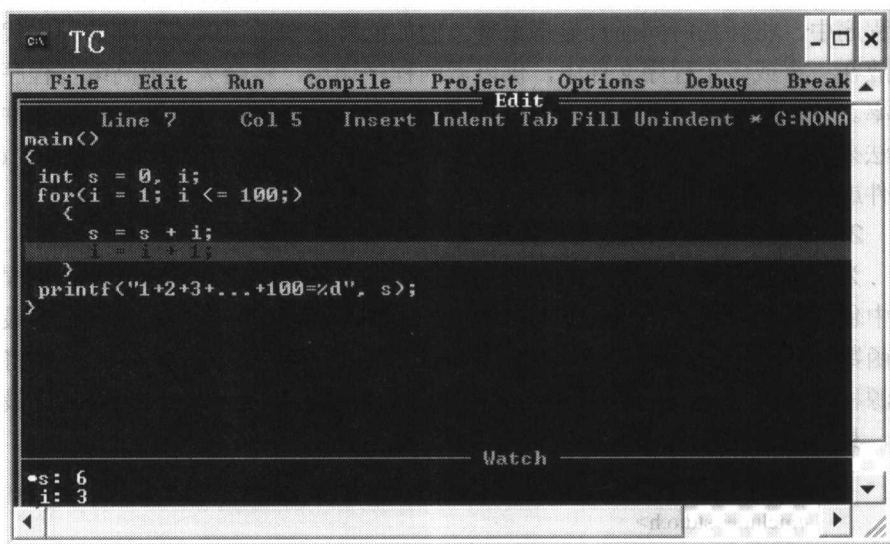


图 1.5 监视窗口中变量值变化

#### 1.1.4 TC 环境调用帮助系统

按 F1 即可调用帮助系统，出现的帮助窗口将解释用户当前所处位置的含义，在所有帮助屏幕关键字处按 Enter 键，便可得到该选项更详细的帮助信息。使用 Home 键或 End 键，可分别选择屏幕上第一个或最后一个关键字。按 Alt+F1，可以回到前次帮助屏幕（总共有 20 个帮助屏幕）。在进入帮助系统后，再按 F1，可以得到帮助索引。

当使用 TC 编辑程序时，可能需要了解各种函数信息，将光标置于需了解的函数名上，按 Ctrl+F1，将会得到有关该函数信息的帮助屏幕。要退出帮助系统返回原来的菜单选择，按 Esc 键。

#### 1.1.5 建立并运行一个工程文件

利用 C 语言程序进行大型程序设计（譬如进行课程设计）时，一般会根据结构化程序设计方法将程序划分为多个源程序，每个源程序包括一个完成独立功能的函数，这样一个函数就是一个程序模块，程序的各个部分除了必要的信息交流外，互不影响，特别适合大程序开发，也是避免错误的有效方法。对此

类文件的编译采用分块编译,即以一个个源文件为单位进行编译产生目标文件,然后用链接程序将生成的目标程序链接成执行文件。

### 1. 工程文件

在大型程序设计中,如果将自己编写的一些通用的或标准的程序代码嵌入到程序中,将耗费时间而且显得累赘。假如能将这些通用的或标准的程序代码存放在一个或多个程序文件中,供以后的程序调用,将会大大提高编写程序的效率。对这类程序的组织管理在 C 语言中最常用的方法是采用建立工程文件的方法来实现,工程文件可以管理多个文件的程序,这些文件可以是.c 文件、.obj 文件或.lib 库文件。

### 2. 建立工程文件的源文件程序

为使描述简单,这里采用在两个数中求最大数为例建立工程文件。工程文件中包括两个源程序,一个是主程序 promain.c 文件,另一个是包含有求最大数函数 max(int, int)的源程序文件 promax.c 文件。在主程序 promain.c 文件将调用源程序文件 promax.c 文件中定义的求最大数的函数 max(int, int)并输出最大数,具体的程序清单如下。

```
/*主程序 promain.c 文件*/
#include<stdio.h>
main()
{
    int  max();          /*声明调用函数*/
    int  a, b, t;
    printf("Please input a and b :");
    scanf("%d%d", &a, &b);
    t = max(a, b);        /*调用 promax.c 程序中定义求最大数的函数 max(int, int)*/
    printf("max = %d", t);
}

/*源程序文件 promax.c 文件,定义求最大数的函数 max(int, int)*/
int max(int x, int y)
{
    if( x > y )
        return  x;
    else
        return  y;
}
```



### 3. 建立并保存工程文件

建立好工程文件的源文件后, 需要建立工程文件 (.prj), 建立方法与建立 C 源程序文件一样, 只是扩展名为 .prj。在 C 语言编辑环境建立工程文件操作步骤:

选择“File”菜单中的“New”后, 在编辑区输入

promain.c

promax.c

再选择“File”菜单中的“Save”, 保存工程文件 MYPROG.PRJ。

**注意:** 工程文件中的源程序文件的排列顺序并不重要, 除非特殊要求的编译顺序。

### 4. 建立工程文件的执行文件

建立工程文件的执行文件, 首先选择“Project”菜单中选择“Project name”项, 命名工程文件的文件名, 然后按 F9, 产生工程文件的可执行文件, 按 Ctrl+F9, 运行工程文件。

## 1.2 程序流程图

描述一个程序的算法可以使用的描述方法有很多种形式, 可以采用简单易懂的算法描述性语言, 也可采用图例形式的程序流程图或 N-S 图或 PAD 图。图例形式描述的算法直观、明了, 让人很容易理解。在三种图例算法描述中, 程序流程图是最早出现的算法描述图例表示, 也是最经典的算法描述方法, 这里仅介绍这种算法描述方法。

### 1.2.1 程序流程图的基本符号

程序流程图又称为程序框图, 是历史悠久、使用最广泛的描述程序算法的方法。它使用的以下基本符号来描述程序算法, 基本符号如图 1.6 所示。

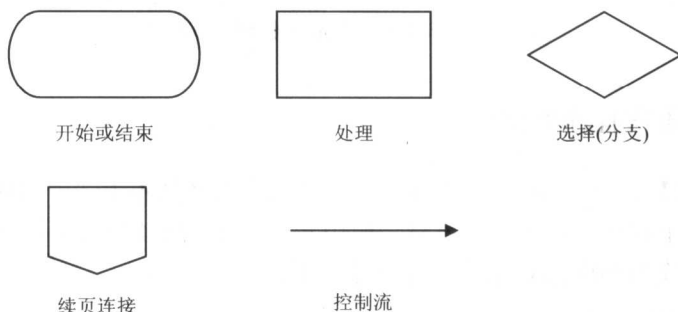


图 1.6 程序流程图基本符号