



C/C++

程序设计实验指导

张 力 主 编

杜 威 宋雅娟 副主编



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机科学与技术规划教材

C/C++程序设计实验指导

张 力 主编

杜 威 宋雅娟 副主编

中国水利水电出版社

内 容 提 要

本书是《C/C++程序设计教程》的配套实验指导书。

本书设置 11 个实验, 每个实验又包含多个实验项目, 内容丰富, 其中既包含实验范例, 又包含自编程序, 目的是采用循序渐进的方式, 提高读者的分析问题和解决问题的能力。

本教材自成体系, 也可与《C/C++程序设计教程》及其他高级语言程序设计教材配套使用。

图书在版编目 (CIP) 数据

C/C++程序设计实验指导 / 张力主编. —北京: 中国水利水电出版社, 2004
(21 世纪高等院校计算机科学与技术规划教材)

ISBN 7-5084-2134-5

I. C… II. 张… 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 (2004) 第 076490 号

书 名	C/C++程序设计实验指导
作 者	张 力 主编 杜 威 宋雅娟 副主编
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 9.25 印张 209 千字
版 次	2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷
印 数	0001—5000 册
定 价	14.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

目前,各高校理工类专业的计算机公共基础课大多开设了 C 语言程序设计课程,有些院校也开设了 C++ 程序设计课程。这些课程的上机实验与理论教学时间的比例近乎 1:1。程序设计课程是一门实践性很强的课程,上机实验是学习 C/C++ 程序设计的一个重要环节。学生通过上机实验能够尽快地掌握 C/C++ 程序设计语言的语法规则,提高程序设计能力,达到事半功倍的效果。

现在关于 C/C++ 程序设计的实验指导书还不是很多,多数只提供一些实验内容,对于实验的指导内容很少,并且多数是单一的 C 或 C++,将二者结合到一起的很少。为了更好地指导学生上机实验,作者根据多年的教学经验组织编写了《C/C++ 程序设计实验指导》。

本书有以下两大特点:

- (1) C 和 C++ 结合在一起,内容广泛。
- (2) 每个实验都给出一些典型实例,介绍程序的设计思想、代码及程序的分析 and 运行结果,使学生对问题的解决方法有所了解;同时还提供学生自编程序,训练学生的编程能力。

本书所使用的实例都经过反复挑选,既具有实用性,又不乏趣味性,使学生在提高学习兴趣的同时,掌握相关的知识;本书遵循由浅入深、循序渐进的原则,可作为大专院校 C/C++ 语言程序设计课程的实验教材,也可作为广大计算机爱好者的自学读物。

本书由张力主编,并负责全书的统编制定,杜威、宋雅娟任副主编。参加编写的还有戴银飞等。其中实验二至实验四、实验七、实验八由张力编写,实验一、实验九由杜威编写,实验五、实验六由戴银飞编写,实验十、实验十一由宋雅娟编写。

由于时间紧迫,加之作者水平有限,书中难免有不足之处,恳请读者提出宝贵意见。

编 者

2004 年 8 月

目 录

前言

实验一 Turbo C 和 Visual C++的基本操作	1
一、相关知识	1
二、实验目的	2
三、实验内容	2
四、思考题	21
实验二 简单的编程练习——基本数据类型及输入输出函数的使用	22
一、相关知识	22
二、实验目的	27
三、实验内容	27
四、思考题	31
实验三 三种基本结构的程序设计	33
一、相关知识	33
二、实验目的	35
三、实验内容	35
四、思考题	42
实验四 数组的应用	43
一、相关知识	43
二、实验目的	44
三、实验内容	44
四、思考题	50
实验五 函数的应用	52
一、相关知识	52
二、实验目的	53
三、实验内容	53
四、思考题	60
实验六 构造数据类型	61
一、相关知识	61
二、实验目的	62
三、实验内容	62
四、思考题	65
实验七 指针及指针应用	66

一、相关知识	66
二、实验目的	69
三、实验内容	69
四、思考题	84
实验八 文件的读写和定位	86
一、相关知识	86
二、实验目的	89
三、实验内容	89
四、思考题	93
实验九 类和对象	94
一、相关知识	94
二、实验目的	98
三、实验内容	98
四、思考题	106
实验十 继承的应用	108
一、相关知识	108
二、实验目的	111
三、实验内容	111
四、思考题	124
实验十一 多态性的应用	125
一、相关知识	125
二、实验目的	127
三、实验内容	127
四、思考题	142

实验一 Turbo C 和 Visual C++的基本操作

一、相关知识

C 语言是一种简明而功能强大的程序设计语言，它提供了用途广泛的编程工具，并可对硬件进行灵活的控制。C 语言程序较之大多数其他语言更具有良好的可移植性。

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combind Programming Language) 发展演变而来。1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的语言——B 语言，C 语言就是由 B 语言演变而来的。

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，美国国家标准化协会 (ANSI) 为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。

C 语言成为最受欢迎的语言之一，主要是因为它具有强大的功能。许多著名的系统软件，如 Word 就是由 C 语言编写的。用 C 语言加上一些汇编语言子程序，更能显示 C 语言的优势，像 PC-DOS、UNIX 等操作系统就是用这种方法编写的。C 语言对程序员的要求较高，使用 C 语言编写程序会感到限制少，灵活性大，功能强，可以编写任何类型的程序。

Turbo C 2.0 可运行于 286 以上的微机，由 DOS 2.0 或更高版本的操作系统支持，并至少需要 448K 的 RAM，可在任何彩色、单色 80 列监视器上运行，支持数学协处理器芯片，也可进行浮点仿真，这将加快程序的执行。

在使用 Turbo C 之前，必须先将 Turbo C 系统装入用户的磁盘上。用户可以指定安装目录，系统默认的目录为 C:\tc。安装完成后，在此目录下主要有以下的文件和子目录。

C:\tc，其中包含如 tc.exe, tcc.exe, make.exe 等可执行文件。

C:\tc\include，其中包含如 stdio.h, math.h 等各种头文件。

C:\tc\lib，其中包含如 maths.lib, mathl.lib 等各种库函数文件。

C 语言是面向过程的，它支持长久以来推行的结构化程序设计。随着软件需求的迅速膨胀和软件技术的不断发展，面向过程程序设计在许多方面显得力不从心。为此，提出了面向对象程序设计 (Object-Oriented Programming, 简称为 OOP) 的思想。C++ 语言是在 C 语言的基础上发展起来的支持面向对象的程序设计语言。

Visual C++ 是 Microsoft 公司开发的 C++ 编译器，是一个非常有效的具有面向对象特征的程序开发工具。它为 Windows 系统下的应用程序开发提供了完全可视化的集成开发环境。运行 Visual C++ 一般要求至少 586 以上的处理器，32MB 以上的内存，Windows 9x 及其以上版本的操作系统，为 Visual C++ 准备足够的硬盘空间，其大小要根据所选择安装的版本和形式确定，学习版是基本的系统，占用的空间最小；企业版最完善，占用的存储空间

间最大。安装形式分为最小安装、典型安装和选择安装，最小系统安装占用的硬盘空间不小于 125MB，最大系统安装占用硬盘空间不小于 600MB。

与其他 Windows 应用程序的安装一样，Visual C++ 6.0 系统也提供了自动安装程序 Setup.exe，在安装程序的引导下，初学者可以采用典型安装方式和缺省路径，按照提示一步步完成安装。安装完成后，Visual C++ 6.0 系统将被安装在 C:\Program Files\Microsoft Visual Studio\VC98 文件夹中。

注意：与以前的 Visual C++ 版本不同，Visual C++ 6.0 的联机帮助文件都使用 MSDN 文档的帮助方式，其与 Visual C++ 6.0 系统不在同一张 CD 盘上，与 Visual Studio 6.0 产品的帮助文档集合在同一张 CD 盘上，在安装过程中，系统会提示 MSDN 盘。

二、实验目的

掌握 Turbo C 和 Visual C++ 的基本操作方法，并能够熟练地进行程序的调试。

三、实验内容

1. Turbo C 的基本操作及程序调试

(1) 进入 Turbo C 2.0 系统需要调用 tc.exe 文件，可以在 DOS 平台下，在 tc 子目录下键入 tc 回车，进入 Turbo C 2.0；也可以在 Windows 平台下，打开 tc 文件夹，双击 tc.exe 应用程序，进入 Turbo C 2.0。

1) Turbo C 2.0 的工作窗口。

打开 Turbo C 2.0 后，会显示 Turbo C 2.0 的版本信息框，用户只需按任意键，此信息框就会关闭。启动后的 Turbo C 2.0 的工作窗口如图 1-1 所示。

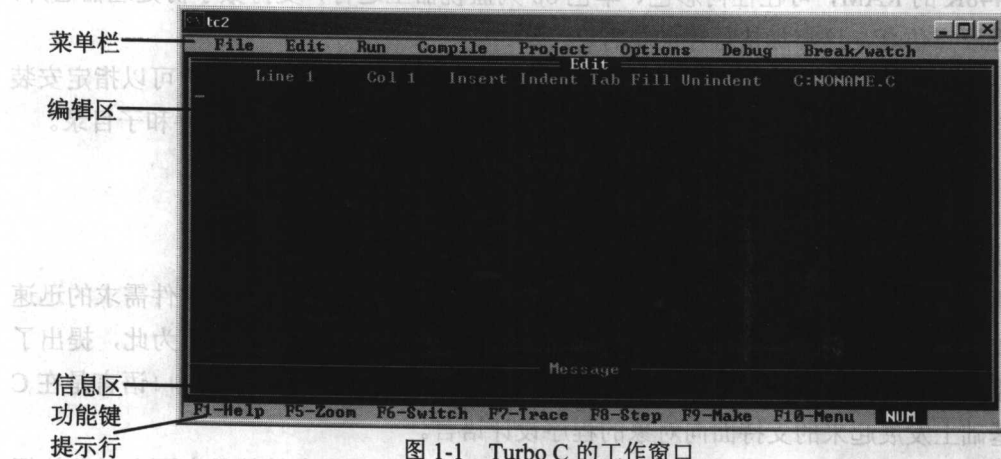


图 1-1 Turbo C 的工作窗口

Turbo C 2.0 窗口包括如下的内容：

① 菜单栏：包括 File（文件）、Edit（编辑）、Run（运行）、Compile（编译）、Project（项目）、Options（选项）、Debug（调试）、Break/watch（断点/监视）主菜单。除 Edit 项外每一个主菜单还有相应的子菜单，只要用 Alt 加上某项中的第一个字母，就可进入该项

的子菜单中。通过菜单可以实现相应的操作。

② 编辑区：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。该窗口的上部有一行说明性标志，如 Line 1 和 Col 1，它们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的文件名（默认的文件名为 NONAME.C）。

③ 信息区：Message 字符标记以下的区域。

④ 功能键提示行：提示一些功能键（快捷键）的作用，包括如下内容：

F1-Help：显示帮助信息。

F5-Zoom：分区控制。将当前活动窗口（编辑窗口或信息窗口）满屏显示。

F6-Switch：转换活动窗口。交替转换编辑窗口和信息窗口为当前活动窗口，当某个窗口为激活状态时，对应的标志字符（Edit 或 Message）将高亮度显示。

F7-Trace：跟踪命令。用于跟踪程序的运行情况。

F8-Step：按步执行。每按一次 F8 键将仅执行一条语句。

F9-Make：进行编译和连接。生成.obj 文件和.exe 文件，但不运行程序。

F10-Menu：返回主菜单。

2) Turbo C 2.0 主要菜单的功能。

① File（文件）菜单，如图 1-2 所示。

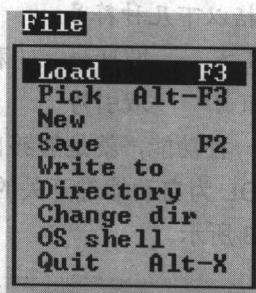


图 1-2 File 菜单

Load（加载）：装入一个文件，可用类似 DOS 的通配符（如*.C）来进行列表选择。也可装入其他扩展名的文件，只要给出文件名（或只给路径）即可。其热键为 F3。

Pick（选择）：将最近装入编辑窗口的 8 个文件列表供用户选择，选择后将该程序装入编辑区，并将光标置在上次修改过的地方。其热键为 Alt+F3。

New（新文件）：说明文件是新的，缺省文件名为 NONAME.C，存盘时可改名。

Save（存盘）：将编辑区中的文件存盘，若文件名是 NONAME.C 时，将询问是否改名。其热键为 F2。

Write to（存盘）：可由用户给出文件名将编辑区中的文件存盘，若该文件已存在，则询问要不要覆盖。

Directory（目录）：显示目录及目录中的文件，并可由用户选择。

Change dir（改变目录）：显示当前目录，用户可以改变显示的目录。

OS shell (暂时退出): 暂时退到 DOS 下, 此时可以运行 DOS 命令, 若想回到 Turbo C 中, 只要在 DOS 状态下键入 EXIT 即可。

Quit (退出): 退出 Turbo C 2.0, 返回到 DOS 操作系统中, 其热键为 Alt+X。

说明: Turbo C 2.0 的所有菜单均可用光标键移动色棒进行选择, 回车则执行。也可用每一项的第一个大写字母直接选择。若要返回主菜单或从它的下一级菜单列表框返回均可用 Esc 键。

② Edit (编辑) 菜单, 进行文本编辑。用 F1 键可获得有关编辑方法的帮助信息。

编辑命令简介:

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	删除光标所在行	Ctrl+T	删除光标处的一个词
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+Q[	查找双界符的后匹配符	Ctrl+Q]	查找双界符的前匹配符

Ctrl+F1 如光标所在处为 Turbo C 2.0 库函数, 则获得有关该函数的帮助信息

说明: Turbo C 2.0 的双界符包括以下几种符号:

花括号 { 和 } 尖括号 < 和 > 圆括号 (和)

方括号 [和] 注释符 /* 和 */ 双引号 " 单引号 '

Turbo C 2.0 在编辑文件时还有一种功能, 就是能够自动缩进, 即光标定位和上一个非空字符对齐。在编辑窗口中, Ctrl+OL 为自动缩进开关的控制键。

③ Run (运行) 菜单, 如图 1-3 所示。

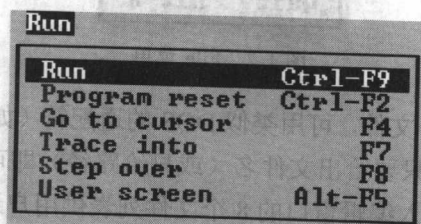


图 1-3 Run 菜单

Run (运行程序): 运行由 Project/Project name 项指定的文件或当前编辑区的文件。如果对上次编译后的源代码未做过修改, 则直接运行到下一个断点(没有断点则运行到结尾)。否则先进行编译、连接后才运行。其热键为 Ctrl+F9。

Program reset (程序重启): 中止当前的调试, 释放分给程序的空间。其热键为 Ctrl+F2。

Go to cursor (运行到光标处): 调试程序时使用, 该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句, 否则提示错误。其热键为 F4。

Trace into (跟踪进入): 在调用子函数时, 若用该项, 将跟踪到该子函数内部去执行。

其热键为 F7。

Step over (单步执行): 执行当前语句, 即使当前是函数调用, 也不会跟踪进入函数的内部, 而是将其作为一条普通语句执行。其热键为 F8。

User screen (用户屏幕): 显示程序运行时在屏幕上显示的结果。其热键为 Alt+F5。

④ **Compile (编译) 菜单**, 如图 1-4 所示。

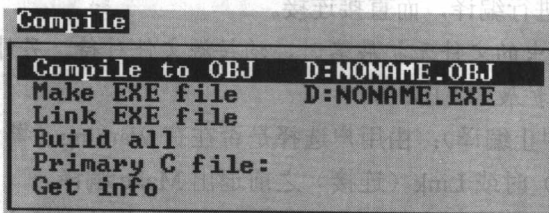


图 1-4 Compile 菜单

Compile to OBJ (编译生成目标码): 将一个 C 源文件编译生成 OBJ 目标文件。其热键为 Alt+F9。

Make EXE file (生成执行文件): 此命令生成一个 EXE 文件。其中 EXE 文件名是下面几项之一:

- 由 Project/Project name 说明的项目文件名。
- 若没有项目文件名, 则为由 Primary C file 说明的源文件名。
- 若以上两项都没有文件名, 则为当前窗口的文件名。

Link EXE file (连接生成执行文件): 把当前 OBJ 文件及库文件连接在一起生成 EXE 文件。

Build all (建立所有文件): 重新编译项目里的所有文件, 并进行装配生成 EXE 文件。该命令不进行过时检查 (上面的几条命令要进行过时检查, 即如果目前项目中源文件的日期和时间与目标文件相同或更早, 则拒绝对源文件进行编译)。

Primary C file: 当没有定义工程文件时, 使用此选项来指定哪一个文件将被编译成 OBJ 文件或生成 EXE 文件。

Get info: 获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息。

⑤ **Project (项目) 菜单**, 如图 1-5 所示。

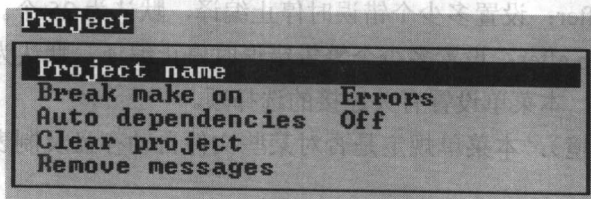


图 1-5 Project 菜单

Project name (项目名): 项目是扩展名为 .PRJ 的文件, 其中包括将要被编译和连接的

文件的文件名。例如有一个程序由 file1.c 和 file2.c 组成, 要将这 2 个文件编译装配成一个 file.exe 文件, 可以先建立一个名为 file.prj 的项目文件, 其内容如下: file1.c、file2.c。此时将 file.prj 放入 Project name 项中, 以后进行编译时, 系统将自动对项目文件中规定的两个源文件分别进行编译, 然后连接成 file.exe 文件。如果其中有些文件已经编译成 OBJ 文件, 而又没有修改过, 可直接写上 OBJ 扩展名, 此时将不再编译而只进行连接。例如: file1.obj、file2.c, 将不对 file1.c 进行编译, 而直接连接。

注意: 当项目文件中的文件无扩展名时, 均按源文件对待。另外, 其中的文件也可以是库文件, 但必须写上扩展名 LIB。

Break make on (中止编译): 由用户选择是否有 Warning (警告)、Errors (错误)、Fatal Errors (致命错误) 时或 Link (连接) 之前退出 Make 编译。

Auto dependencies (自动依赖): 开关为 on, 编译时将检查源文件与对应的 OBJ 文件的日期和时间, 否则不进行检查。

Clear project (清除项目文件): 清除 Project / Project name 中的项目文件名。

Remove messages (删除信息): 把错误信息从信息窗口中清除掉。

⑥ **Options (运行环境设置菜单)**, 如图 1-6 所示。该菜单对初学者来说要谨慎使用。

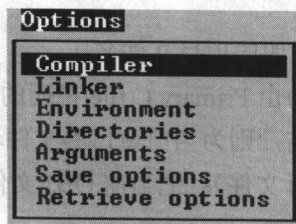


图 1-6 Options 菜单

Compiler (编译器): 本选择项又有许多子菜单, 可以让用户选择硬件配置、存储模型、调试技术、代码优化、对话信息控制和宏定义。例如:

- **Model:** 共有 Tiny, Small, Medium, Compact, Large, Huge 六种不同模式, 供用户选择。
- **Calling convention:** 可选择 C 或 Pascal 方式传递参数。
- **Floating point:** 可选择仿真浮点、数学协处理器浮点或无浮点运算。
- **Identifier length:** 说明标识符有效字符的个数, 默认为 32 个。
- **Errors stop after:** 设置多少个错误时停止编译, 默认为 25 个。
- **Warnings stop after:** 设置多少个警告错误时停止编译, 默认为 100 个。

Linker (连接器): 本菜单设置有关连接的选择项。

Environment (环境): 本菜单规定是否对某些文件自动存盘及制表键和屏幕大小的设置。例如:

- **Edit auto save:** 是否在 Run 或 Shell 之前, 自动存储编辑的源文件。
- **Backup file:** 是否在源文件存盘时产生后备文件 (.BAK 文件)。
- **Tab size:** 设置制表键大小, 默认为 8。

- **Zoomed windows:** 将现行活动窗口放大到整个屏幕, 其热键为 F5。

Directories (路径): 规定编译、连接所需文件的路径, 有下列各项:

- **Include directories:** 包含文件的路径, 多个子目录用 “;” 分开。
- **Library directories:** 库文件路径, 多个子目录用 “;” 分开。
- **Output directories:** 输出文件 (.OBJ, .EXE, .MAP 文件) 的目录。
- **Turbo C directories:** Turbo C 所在的目录。

Arguments (命令行参数): 允许用户使用命令行参数。

Save options (保存设置): 使用户可以将 Compiler, Linker, Environment, Debug 以及 Project options 中进行的设置保存起来, 生成配置文件, 默认的文件名是 TCCONFIG.TC。

Retrieve options (恢复设置): 调用以前保存的配置文件, 使用其中的设置来配置 Turbo C 系统。

- ⑦ **Debug (调试) 菜单,** 如图 1-7 所示。该菜单主要用于程序调试、查错、设置等。

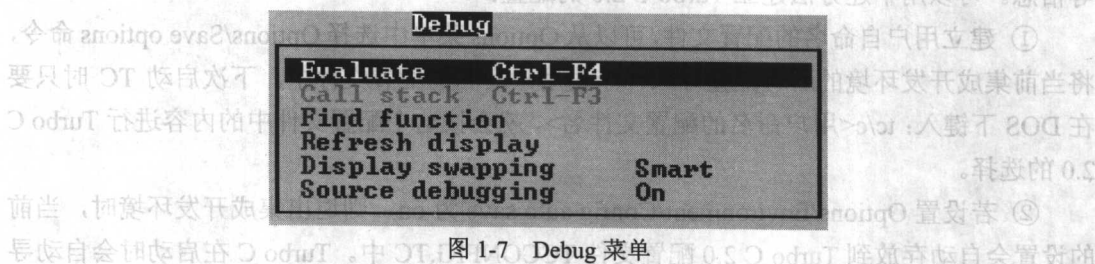


图 1-7 Debug 菜单

Evaluate (计算): 当程序运行时, 此命令可以使用户查看各变量或表达式的值, 并且可以修改这些值。此命令会弹出一个对话框, 如图 1-8 所示。

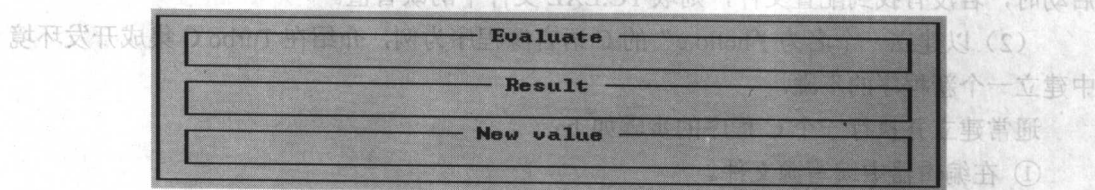


图 1-8 Evaluate 对话框

各项的含义如下:

- **Evaluate:** 要计算结果的表达式。
- **Result:** 显示表达式的计算结果。
- **New value:** 赋给新值。

- ⑧ **Break/watch (断点及监视表达式) 菜单,** 如图 1-9 所示。

Add watch: 向监视窗口插入一个监视表达式。

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。

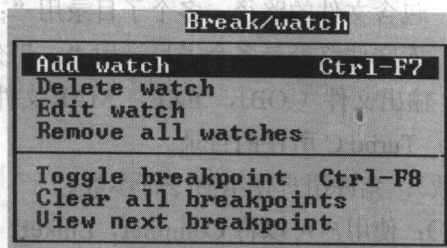


图 1-9 Break/watch 菜单

3) Turbo C 2.0 的配置文件。

所谓配置文件是包含 Turbo C 2.0 有关信息的文件, 其中存有编译、连接的选择和路径等信息。可以用下述方法建立 Turbo C 2.0 的配置:

① 建立用户自命名的配置文件, 可以从 Options 菜单中选择 Options/Save options 命令, 将当前集成开发环境的所有配置存入一个由用户命名的配置文件中。下次启动 TC 时只要在 DOS 下键入: tc/c<用户命名的配置文件名>, 就会按这个配置文件中的内容进行 Turbo C 2.0 的选择。

② 若设置 Options/Environment/Config auto save 为 on, 则退出集成开发环境时, 当前的设置会自动存放到 Turbo C 2.0 配置文件 TCCONFIG.TC 中。Turbo C 在启动时会自动寻找这个配置文件。

③ 用 TCINST 设置 Turbo C 的有关配置, 并将结果存入 TC.EXE 文件中。Turbo C 在启动时, 若没有找到配置文件, 则取 TC.EXE 文件中的缺省值。

(2) 以建立一个名为“hello.c”的 C 语言源程序为例, 介绍在 Turbo C 集成开发环境中建立一个新程序的步骤。

通常建立并执行一个 C 程序的步骤如下:

- ① 在编辑器中编写源文件。
- ② 编译和连接。
- ③ 生成可执行文件。

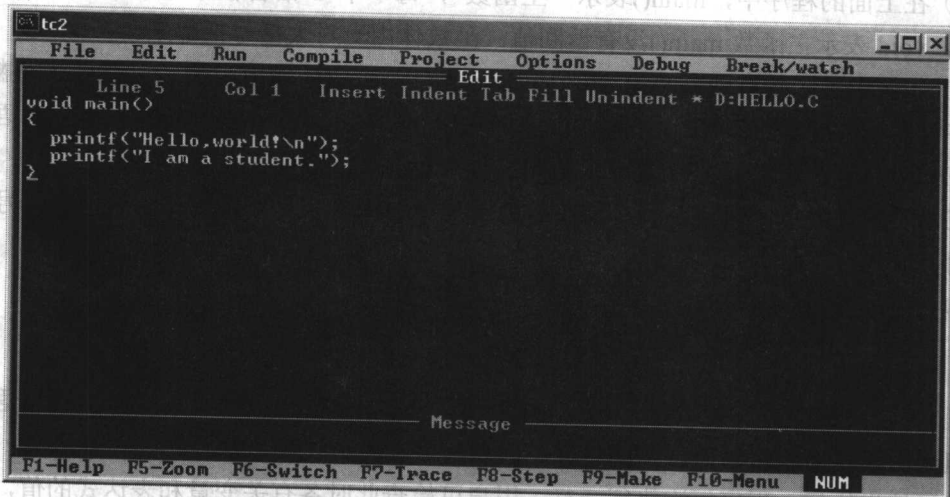
进入 TC 工作窗口后, 按 F3 键, 即可在随之出现的框中输入文件名, 文件名可以带“.c”也可以不带(此时系统会自动加上)。输入文件名后, 按回车, 即可将文件调入。如果文件不存在, 就建立一个新文件(也可用下面例子中的方法输入文件名)。系统随之进入编辑状态, 就可以输入或修改源程序了。源程序输入或修改完毕以后, 按 Ctrl+F9。则进行编译、连接和执行, 这三项工作是连续完成的。

1) 建立并执行程序的步骤。

- ① 进入 Turbo C 集成开发环境, 通过键盘输入如下程序, 如图 1-10 所示。

```
void main()                /* 主函数 */
{
    printf("Hello, world!\n"); /* 输出语句 */
}
```

```
printf("I am a student."); /* 输出语句 */
```



② 程序存盘：在编辑窗口下，可以使用 File 菜单中的 Save 命令存盘，也可直接按 F2 键将文件存盘，这时，系统将弹出一个 Rename NONAME 对话框，这里将建立一个名为 hello.c 的文件。如图 1-11 所示。

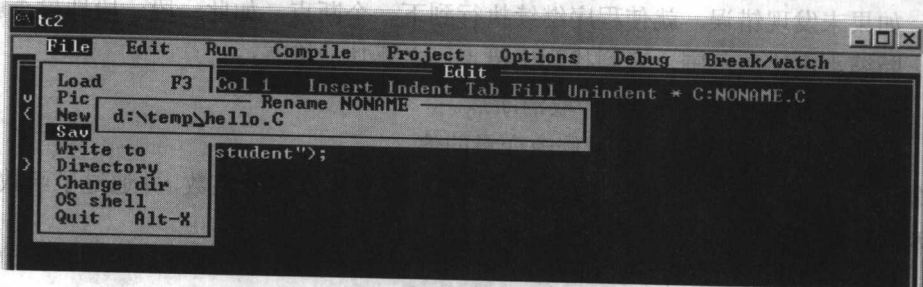


图 1-11 程序命名存盘

③ 编译一个程序：按 Alt+F9 进入编译状态后，会出现一个编译窗口，几秒钟后，若显示信息：Success: press any key，则表示编译成功。如果编译时产生警告 Warning 或出错 Error 信息，则会显示在屏幕下部的信息窗口中，可按提示对源程序进行修改，重新进行编译。

④ 运行程序：源程序经编译无误后，按 Ctrl+F9 运行，这时屏幕会出现一个连接窗口，显示 Turbo C 正在连接程序所需的库函数。连接完毕后，屏幕会突然一闪，后又回到编辑窗口，此时可按 Alt+F5 切换到程序输出窗口，查看输出结果。再按任意键，就又回到编辑窗口。

这时，在磁盘上生成了 3 个文件：hello.c、hello.obj、hello.exe。其中文件 hello.c 是 C 语言源文件，文件 hello.obj 是 Turbo C 编译程序产生的二进制机器指令（目标代码），文件 hello.exe 是 Turbo C 连接程序产生的可执行文件。

2) 程序分析解释。

① 在上面的程序中, `main()`表示“主函数”, 每一个 C 语言程序都必须有一个主函数 `main()`。 `void` 表示主函数 `main()`没有返回值。函数体由一对大括号括起来。

② 本例中主函数有两条输出语句, `printf` 是 C 语言中的输出函数。双引号(双撇号)内的字符串按原样输出, “`\n`”是换行符, 即在输出“Hello, world!”后回车换行, 语句的最后有一个分号, 在 C 语言中是一行语句的结束标志。

③ `/*`和`*/`之间的部分是注释, 这将有助于程序的阅读和理解, 要注意这两个符号应用对使用, 缺一不可。

3) 程序运行结果。

```
Hello, world!
I am a student.
```

(3) 程序的编译和连接没有错误不等于运行时一定正确。编译系统能检查出语法错误, 但无法检查出逻辑错误。下面将介绍两种程序动态调试的方法。

- 按步执行: 程序每次执行一行, 用户可检查此时各有关变量和表达式的值, 以便发现错误。主要的功能键有:

<F7>键: 按步执行程序, 每按一次, 执行一条语句。

<Ctrl>+<F7>组合键: 观察变量和表达式的值。

- 设置断点: 程序执行到断点时将暂停, 用户可检查此时各有关变量和表达式的值。如果未发现错误, 就使程序继续执行到下一个断点, 如此一段一段地检查就能不断缩小出错的范围, 直到找到出错点。主要功能键有:

<Ctrl>+<F8>组合键: 设置/取消断点。

<Ctrl>+<F4>组合键: 查看修改变量的值。

1) 输入如下程序, 该程序在主函数 `main()`中从键盘接收两个整数, 并通过子函数 `max()`来求得两者的最大值, 并在主函数 `main()`中输出。

```
int max(int num1,int num2)
{
    if(num1 > num2)
        return num2;
    else
        return num1;
}
main()
{
    int x, y, maxNum;
    printf("\nx=");
    scanf("%d", &x);
    printf("\ny=");
    scanf("%d", &y);
    maxNum = max(x, y);
    printf("\nThe max is: %d\n", maxNum);
}
```

2) 按步执行调试程序。编辑、编译、连接和运行这个程序,可以发现这个程序没有任何语法错误,能顺利通过编译和连接。在程序运行时输入 5 和 8,程序的运行结果为:

The max is:5

这个结果显然是不对的。这时可采用如下所述的按步执行程序的方法,检查每一步的正确性。

① 重新开始运行,按<F7>键,可以看到编辑窗口中源程序的主函数 main()处高亮度显示,表示准备进入主函数 main()。同时可以看到屏幕下部的 Message 窗口变成 Watch 窗口,用于观察数据,如图 1-12 所示。

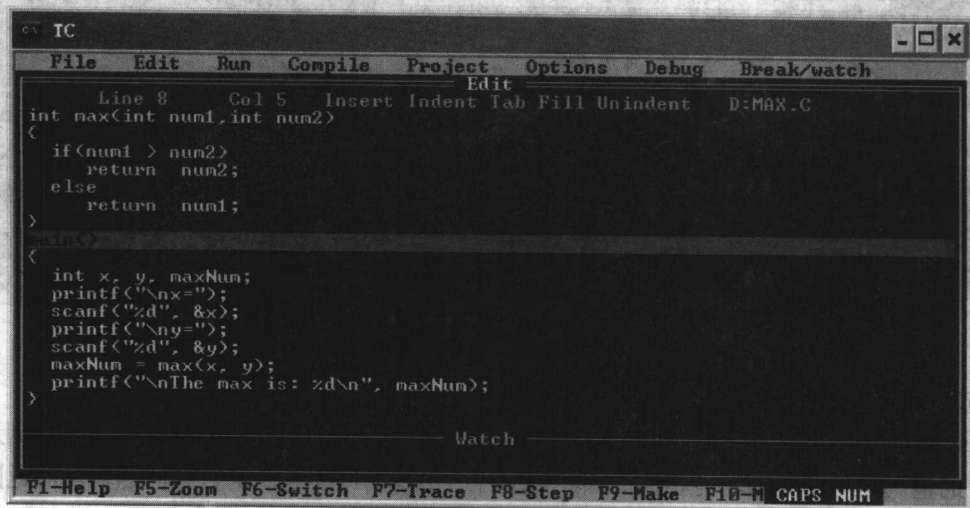


图 1-12 用<F7>键跟踪的初始窗口

② 再次按下<F7>键,高亮条移到主函数中的第二行(第一行是对变量的定义,不是执行语句,故被跳过),表示已进入了主函数 main()和左花括号,但并未执行 printf 函数语句,只是表明下一步要执行此语句。

③ 按<F7>键,此时执行主函数的第二行语句,在用户屏幕上将输出“x=”的信息。可以用<Alt>+<F5>组合键查看用户屏幕,然后按<Esc>键返回编辑窗口。

④ 按<F7>键,此时执行主函数的第三行语句,由于 scanf 函数语句需要输入数据,所以程序将切换到用户屏幕,输入:

8↵

按回车键后,屏幕返回编辑窗口,高亮条移到下一行,表示 scanf 语句已执行完毕。这时,变量 x 已被赋值为 8。查看变量的方法是按<Ctrl>+<F7>组合键,将弹出一个 Add Watch 对话框,输入想查看的变量名 x,如图 1-13 所示。按回车键后,该对话框消失,在屏幕下部的 Watch 窗口显示变量 x 的当前值为 8。

⑤ 按同样方法执行下两步,并在用户屏幕输入 5,即将 5 赋值给变量 y。并在 Watch 窗口添加变量 y,可看到 y 的值为 5。

⑥ 当高亮度条停留在语句 maxNum = max(x,y); 上时要注意,按下<F7>键后,程序将