

C YUYAN SHIYAN ZHIDAO YU XITI

C语言实验指导与习题

主编 李胜宾 陈汉亚

编者 李胜宾 陈汉亚 陈坚祯

谭立 黄春熹 魏书堤

李正华

CYY

湖南科学技术出版社

高等教育21世纪课程教材

C YUYAN SHIYAN ZHIDAO YU XITI

C语言实验指导与习题

主编 李胜宾 陈汉亚

编者 李胜宾 陈汉亚 陈坚祯

谭立 黄春熹 魏书堤

李正华

江苏工业学院图书馆
藏书章

湖南科学技术出版社

内 容 简 介

本书是《C 语言程序设计》一书的配套教材。全书共分为 3 个部分,第一部分“实验指导”,内容包括 Turbo C 2.0, Visual C++ 6.0 开发环境简介和 16 个实验,分别对应《C 语言程序设计》各章(第 12 章除外)的内容;第二部分为“习题”,按《C 语言程序设计》章节顺序进行编排;第三部分为“试题”。

高等教育 21 世纪课程教材

C 语言实验指导与习题

主 编:李胜宾 陈汉亚

责任编辑:曹 阳

出版发行:湖南科学技术出版社

社 址:长沙市湘雅路 280 号

<http://www.hnstp.com>

印 刷:长沙市飞鹏印刷有限公司

(印装质量问题请直接与本厂联系)

厂 址:长沙市德雅路 226 号

邮 编:410003

出版日期:2003 年 2 月第 1 版第 1 次

开 本:787mm×1092mm 1/16

印 张:12.5

印 数:310000

ISBN 7-5357-3631-9/TP·177

前 言

学习程序设计必须多阅读程序,多动手编程,多上机实践,尤其是要多上机实践。程序设计语言是一门实践性很强的课程,这本书按照教学的要求,为读者编写了一套实验内容、课后作业以及复习考试之用的练习试题。

为了使读者尽快熟悉实验环境,第一部分前两章安排的是 Turbo C 和 Visual C++ 6.0 的简介。考虑到各种需求,除了安排传统的 Turbo C 2.0 系统外,根据当前发展的趋势,着重介绍了 Visual C++ 6.0 的主要特点和使用方法,以便为读者过渡到 Windows 环境下的可视化编程创造良好条件。程序调试是程序设计中必不可少的过程,是一种非常重要的基本功训练。读者应该充分利用软件所提供的功能来寻找、发现程序中发生的错误,这对于编写大型程序以及今后的应用、开发都是十分必要的。为此,结合具体的实例,介绍了程序调试的基本方法和步骤,希望读者能将这一方法用到各实验的程序调试中。第 3 章共安排了 16 个实验,每个实验都有实验的目的、要求、实验内容等,有的实验还指出了实验步骤。考虑到初学者的特点,有的先安排示例性的程序,然后要求读者编写程序。通过这样的练习,能够达到增强动手能力和程序设计能力的培养目的。

第二部分是各种形式的练习题。这部分内容可以结合教材,以加深其基本概念、基本方法的理解。此外,题目的类型也是以规范化的方式出现的,可方便读者参加各种考试。这部分题目均未列出答案,读者可根据自己的情况,选择其中的一部分独立完成。

第三部分是试题。这部分内容可作为提高之用,都有答案供读者参考。通过这些试题,可检查读者掌握知识的程度和使用该语言进行程序设计的能力。

本书由李胜宾、陈汉亚主编,参加编写的还有陈坚桢、谭立、黄春熹、魏书堤、李正华。本书在编写过程中参阅了多种教材和资料,在此对这些作者表示感谢。

因时间仓促、水平有限,错误在所难免。对书中的错误或不妥之处欢迎各位批评指正。

编 者
2003 年 1 月

目 录

第一部分 实验指导

第 1 章 Turbo C 运行环境简介	(1)
1.1 Turbo C 系统的安装和使用	(1)
1.2 程序的调试和运行	(6)
第 2 章 Visual C++ 6.0 开发环境	(9)
2.1 Visual C++ 及其开发环境概述	(9)
2.2 Visual C++ 6.0 的菜单栏	(9)
2.3 Visual C++ 6.0 的工具栏	(16)
2.4 项目与项目工作区	(16)
2.5 资源与资源编辑器	(18)
2.6 联机帮助	(18)
2.7 程序的调试	(22)
第 3 章 实验内容	(27)
实验一 熟悉上机环境	(27)
实验二 输入输出语句程序	(29)
实验三 选择分支程序设计	(31)
实验四 循环程序设计(1)	(35)
实验五 循环程序设计(2)	(38)
实验六 函数(1)	(43)
实验七 函数(2)	(46)
实验八 编译预处理	(48)
实验九 数组(1)	(49)
实验十 数组(2)	(53)
实验十一 结构体和共用体(1)	(57)
实验十二 结构体和共用体(2)	(58)
实验十三 指针(1)	(62)
实验十四 指针(2)	(64)
实验十五 位运算	(66)
实验十六 文件	(66)

第二部分 习题

第 1 章 C 语言概论第 2 章 基本数据类型运算符表达式习题	(68)
第 3 章 数据的输入输出第 4 章 语句习题	(75)
第 5 章 函数习题	(83)
第 6 章 预处理习题	(95)
第 7 章 数组习题	(98)
第 8 章 结构体和共用体习题	(120)
第 9 章 指针习题	(123)
第 10 章 位运算习题	(142)
第 11 章 文件习题	(143)

第三部分 试题

第 1~136 题	(147)
-----------------	-------

第 1 章 Turbo C 运行环境简介

Turbo C 是一个集程序编辑、编译、链接、调试于一体的 C 语言程序开发软件,具有速度快、效率高、功能强等优点,使用非常方便。C 语言程序员可在 Turbo C 环境下进行全屏幕编辑,利用窗口功能进行编译、链接、调试、运行、环境设置等工作。

1.1 Turbo C 系统的安装和使用

1.1.1 Turbo C 2.0 系统安装和设置

关于 Turbo C 2.0 的安装,很多书都有详细介绍,这里不再赘述。只强调两点,一是要注意它的安装文件夹,二是要注意它的环境设置。

因为现在都是在 Windows 环境下安装 Turbo C 系统,所以往往使用任务栏属性在开始菜单程序中添加 TC.exe 程序,或者在桌面上设置 TC.exe 的快捷方式,以便快速进入 TC 系统。

1.1.2 Turbo C 2.0 的使用

1. 进入 Turbo C。

有如下几种方法:从开始菜单启动;从桌面上设置 TC.exe 的快捷方式启动;直接找到 Turbo C 目录下的 tc.exe 程序双击并启动,从而进入主菜单窗口,如图 1.1 所示。

2. 选择工作目录。

工作目录是指用户文件所在的目录。用户进入 TC 目录后,可以在这个目录下再建立一个用户专用的子目录。用户可以在此子目录下进行文件的编辑工作,编译生成的目标文件也存放在此子目录中。这个工作目录可由用户自己建立,再由 Change dir 命令来告诉 Turbo C 你所选定的工作目录。方法见下述。

在进入 Turbo C 后,屏幕上显示出一个 Turbo C 主菜单窗口,这时可以用 F10 键和光标移动键从主菜单中选择所需的功能。步骤如下:先按下 F10 键,可以看到屏幕上部的某一位置会出现亮块,如果亮块不在 File 处,可用光标移动键“←”“→”来移动此亮块;如果将亮块移动到菜单中最左上端的“File”处,表示准备选择“File”类操作,即操作文件。也可不用 F10

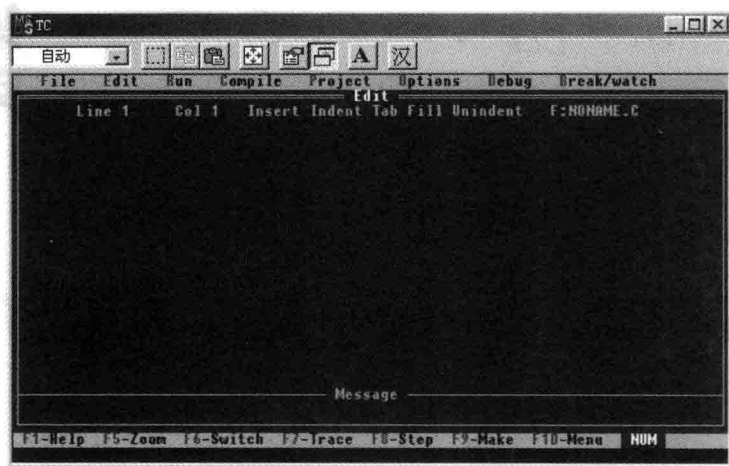


图 1.1 Turbo C 主菜单窗口

键,而直接按 Alt 键和“F”键(同时),使亮块位于“File”处。亮块移动到“File”处后按下回车键,在“File”下方会出现一个子窗口,如图 1.2 所示。

子窗口处也有一个亮块,可用“↑”、“↓”键将此亮块移动到 Change dir 处,或者直接按下“C”键来选择 Change dir 操作。输入“↵”键后将会出现一个目录提示框,提示用户输入所选择的目录名,用户输入后再按下“↵”键即可。

如:屏幕显示输入提示,如图 1.3 所示。

用户输入\usr 之后,按下回车键;如果当前不存在此目录,则显示出错信息,可再次输入合法的目录名。

3. 建立工作环境。

即告诉 TC: 包含文件和库文件在什么地方,输出



图 1.2 File 菜单窗口

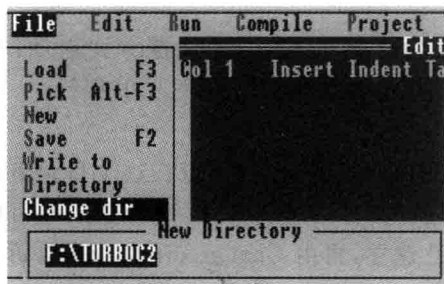


图 1.3 Change dir 提示框

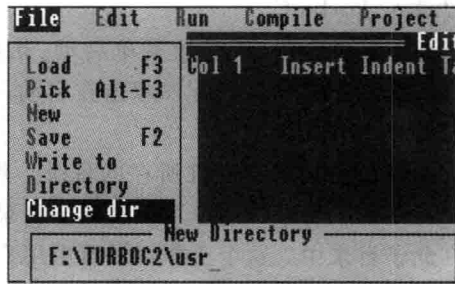


图 1.4 New Directory 框

文件存于何处。可用主菜单上“Option”项来进行。步骤如下:首先在主菜单窗口上,将亮块移到 Option 处,移动亮块的方法前面已作介绍;在亮块移动到“Option”位置以后,按回车键。此时在“Option”位置下面出现一个子窗口,如图 1.5 所示。

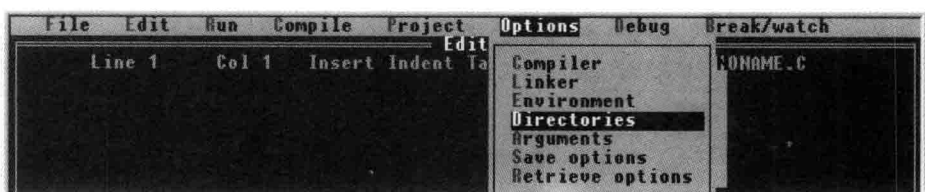


图 1.5 Options 菜单窗口

用光标移动键将亮块移动到“Directories”处,按回车键后又将出现一个子窗口,如图 1.6 所示。

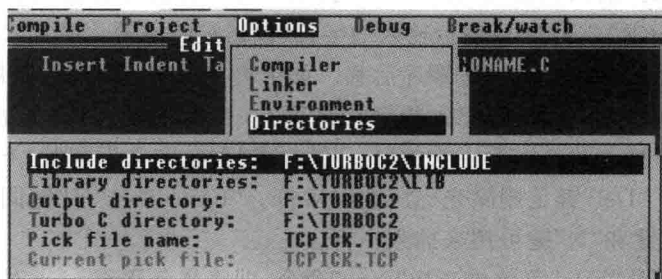


图 1.6 Directories 选项

将亮块移到“Directories”处,按回车键后输入“包含文件”所在的盘符和路径。如输入:

F:\TURBOC2\INCLUDE; F:\TURBOC2\INCLUDE\SYS ✓

表示:包含文件放在 F 盘上的\TURBOC2\INCLUDE 子目录和\TURBOC2\INCLUDE\SYS 子目录中。

包含文件的所在目录输入后,再将亮条移到“Library directories”处,按回车键。输入库函数所在的盘符和目录名。例如:

F:\TURBOC2\LIB ✓

这就告诉 Turbo C: 库函数在 F 盘上的\TURBOC2\LIB 子目录中。

如果用户想将编译链接后的运行文件(即可执行的目标文件)存放在另外的子目录中,可再将亮条移到“Output directory”处,“回车”后输入用户所指定的子目录名。

4. 编辑源文件。

在“主菜单”状态下,将亮块移动到左上角的“File”处,然后按回车键,屏幕出现“File”子窗口。在“File”菜单窗口中,将亮条移至“Load”,用“回车”选择此命令,或在出现主菜单窗口时直接按“F3”键。这时出现一个包含 *.C 的提示框,如图 1.7 所示:

用户可输入要编辑的文件名,“回车”即可。

如果记不清所要编辑的源文件名,想看一下当前目录中有哪些源文件,可以在子窗口出现上述“*.C”时直接按回车键,Turbo C 就会显示出当前目录下的所有后缀为“.C”的文件名。利用光标键将亮条移动到需要编辑的文件名处,按“回车”后,该输入文件的内容即显示在屏幕上,供用户编辑、修改。

如果用户用“Load”输入的文件名是已存在的文件,这时屏幕上将显示出文件内容,可供修改;否则屏幕上是一片空白,表示文件无内容(是新文件),用户可以用键盘输入文件内容。在编辑过程中除用到各字符键外,还可以用到“Ins”和“Del”键。“Ins”键是一个切换键,用来

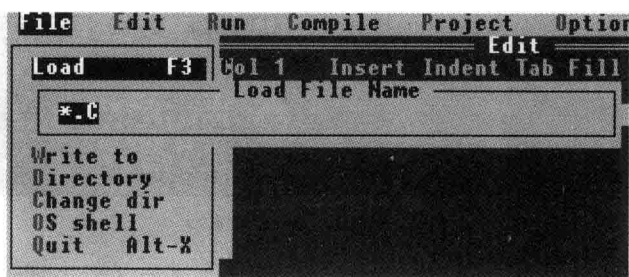


图 1.7 Load 输入提示框

控制工作状态是否为“插入状态”。按下“Ins”键后，可以看到在屏幕编辑窗口的状态行上有一个英文单词“INSERT”，这时从键盘输入的字符(包括控制字符，如“回车”)会插入到屏幕的当前光标处，光标后的字符会自动按顺序后移。如果再按一下“Ins”键，则取消插入状态，状态行上的“INSERT”单词消失，此后键入的字符将覆盖(而不是插入)光标处的字符。Turbo C 设置的初始状态是“插入状态”，第一次按“Ins”键改成“覆盖状态”，再按“Ins”则又改为插入状态，如此反复切换。“Del”键是删除光标所在的字符。“Ctrl”键和“Y”键同时按下可删除光标所在的一行；“Ctrl”键和“N”键可用来插入一行。

编辑完成后，可从“File”菜单中选择“save”命令(即将亮条移至 save 处后，再“回车”)将文件存盘；也可直接按下 F2 键，将文件存盘。

5. 编译、链接。

加工程序时，先编译源代码，生成目标文件(扩展名为. OBJ)；然后将目标文件进行链接，生成可执行文件(扩展名为. EXE)。

对单文件程序的编译、链接的方法是在将文件存盘后，按 F10 键，将亮条移至 Compile 处后“回车”；也可直接按 Alt-C，即可产生一个子窗口，如图 1.8 所示。

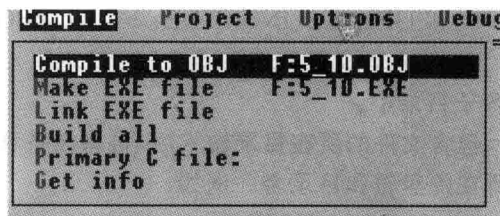


图 1.8 Compile 菜单窗口

表示编译后生成 5_10. obj 文件，链接后生成 5_10. exe 文件，源文件名设定为 5_10. c。将亮条移到“Make EXE File 处(也可直接按 F9 键)，则 TC 将对文件进行编译、链接并生成运行文件。

若程序有错，则在屏幕底部的“Message”窗口显示出错及警告信息，这时可进行修改。改完后，再重复进行编译、链接。

另外，TC 最显著的特点之一是能进行分别编译，可以把一个长的程序分别编辑在多个文件中，分别编译。Turbo C 提供了一个工具，可直接将这些文件编译、链接并生成一个完整的运行程序，而不必由用户显式地分别编译、链接。这个工具就是 Project 菜单项，使用步骤如下：

(1)假设有两个文件组成一个程序，首先要生成两个源文件(如 File1. c 和 File2. c)。

(2)构造 Projec 文件。在编辑状态下，编辑一个后缀为 PRJ 的文件(文件名可由用户选择，如:MYPROG. PRJ)，此文件内容如下：

```
File1
File2
```

后缀.C可有可无,而且 File1,File2 的顺序无所谓。当 File1.c 和 File2.c 不在同一个目录中时,可在 Project 文件 MYPROG.PRJ 中给出各自的路径,例如文件内容可为:

```
\Turbo C\USR\File1
```

```
\Turbo C\File2
```

(3)在主菜单窗口中选择 Project 功能项,即将亮条移至 Project 处回车(也可按 Alt+p 键得到),此时出现如图 1.9 子窗口。



图 1.9 Project 菜单

将亮条移到“Project name”处,按“回车”后,输入 Project 文件名(如 MYPROG.PRJ)“回车”,然后按 F9 键即产生相应的可执行文件。此运行文件名即为用户给出的 Project 文件名(如 MYPROG.EXE),运行时可用下述的 RUN 命令,也可在操作命令状态下直接输入运行文件名(如 MYPROG ↵)。

(4)停止编译的方式

如果同时编译的几个程序中有错误,就须停止编译,由用户进行修改。用户可以指定两种方式来停止编译。

如果用户希望同时编译的几个文件中,出现一个错误时就停止编译,可用如下方法:

将 Project 菜单中的亮条移至“Break make on”处,“回车”后又可出现一子窗口,如图 1.10 所示。

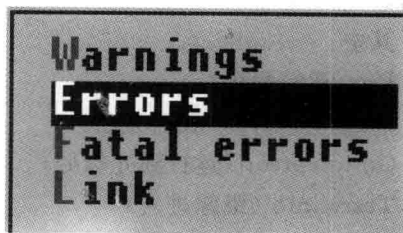


图 1.10 Break Make On 窗口

将亮条移至 Errors 或 Warnings 处,“回车”后,

在 Break Make On 右边即出现此选择项,如图 1.11 所示。

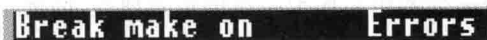


图 1.11 Errors 选项弹出窗口

这时,当编译的几个文件中只要出现一个错误时,就立即停止编译,亮条停在出错处,指示用户进行修改。

如果用户希望把所有文件中存在的错误都找出来以后,再修改错误,可用如下方法:

在“Break make on”子窗口中,将亮条移至“Fatal errors”或“Link”处,此时“Break make on”右边出现的是“Fatal Errors”或“Link”。这样,在编译出现于所有错误后,才停止编译,用户这时再修改源文件。

关于 Project 菜单的详细说明请参见 Turbo C 手册。

6. 运行。

编译链接完成一个文件后,可利用主菜单中的 RUN 命令或直接按 Ctrl-F9 键来运行程序。

其实,当用户认为自己的源程序不会有编译、链接错误时,在源程序编译完成后就可直接用 RUN 命令或直接按 Ctrl-F9 键。这时 Turbo C 将一次完成从编译、链接到运行的全过程,这是运行 Turbo C 程序最简便、常用的方法。

程序运行后仍回到 TC 屏,这时若想看运行结果,可用“RUN”菜单(如图 1.12 所示)中的 User screen 命令(即亮条出现在“RUN”处,“回车”后可出现一个子窗口,将亮条移至“User screen”处“回车”即可);也可以直接按 Alt+F5 键转到用户屏,程序运行的结果显示在用户屏上,看完后可按任一键回到 TC 屏。

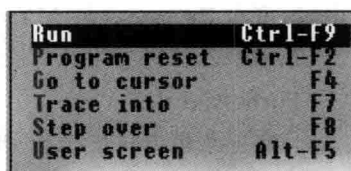


图 1.12 RUN 菜单

1.2 程序的调试和运行

1.2.1 RUN 菜单

把高亮度块移至 RUN 菜单后按下回车键,或用 Alt+R 可进入 RUN 菜单,如图 1.12 所示。

其中

Run(运行程序)

Program reset (程序重启)

Go to cursor (运行到光标处)

Trace into (跟踪进入)

Step over (单步执行)

User screen (用户屏幕)

需要强调的是:Trace into 在调用函数时将跟踪到该函数内部去(热键为 F7),这在调试、运行函数的过程中非常有用。另一个是 step over,选此项(或按下 F8 键)程序将单步运行,即每次只执行一条语句。因为函数调用是作为一条语句处理的,因此不会跟踪到函数内部。

1.2.2 Debug 调试菜单

将高亮度块移到 Debug 处按下或者按 Alt+D 快捷键,将弹出 Debug 子菜单,如图 1.13 所示。

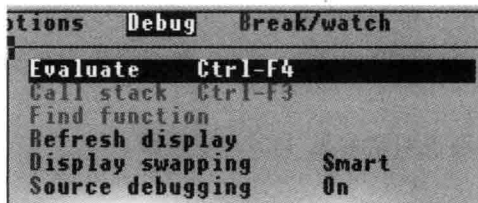


图 1.13 Debug 菜单

Evaluate(求值)选此项可进入对话框(如图 1.14 所示)。

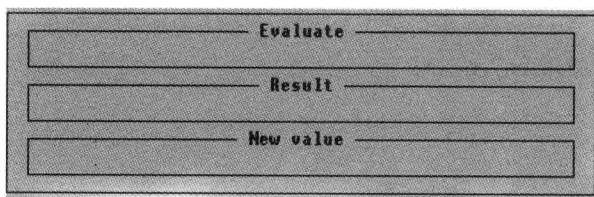


图 1.14 Evaluate 对话框

Expression 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

Call stack (调用堆栈) 用于检查堆栈情况

Find function(查找函数)用于显示规定的函数

Refresh display (刷新显示)编辑窗口被用户窗口重写了,可用此命令恢复编辑窗口的内容

Display swapping Smart(显示方式交换)设置程序调试时,屏幕显示转换方式

Source debugging On(调试源程序)设置程序连接时,能使用何种调试程序

1.2.3 Break / watch (断点及监视表达式)

将高亮度块移到 Break / watch 处按下,或按下 Alt+B 可进入 Break / watch 子菜单,如图 1.15 所示。

Add watch(增加监视)向监视窗口插入一个监视表达式。要查看程序中的变量值的变化情况,可输入该变量,则在监视窗口中会显示该变量的状态。

Delete watch (删除监视)从监视窗口中删除监视表达式

Edit watch (编辑监视)在监视窗口中编辑了一个监视表达式

Remove all watches(消除所有监视)从监视窗口中删除所有监视表达式

Toggle breakpoint 对光标所在行设置或清除断点

Clear all breakpoints 清除所有断点

View next breakpoint 将光标移到下一个断点处

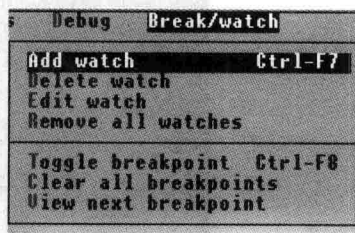


图 1.15 Break/watch 子菜单

1.2.4 函数程序的调试

【例 1.1】 示例程序如下:

```
#include<stdio.h>
main()
{
    int n;
    long s;
    long fact(int);
```

```
printf("Input n=");  
scanf("%d",&n)  
s=fact(n);  
printf("%d! =%ld\n",n,s);  
}  
long fact(int x)  
{  
    long int t;  
    if(x==0) t=1;  
    else t=fact(x-1)*x;  
    return t;  
}
```

使用 Trace into(或 F7 键)跟踪进入函数 fact 内,使用 Add watch 插入监视表达式 x,观察 x 值的变化情况以及函数递归调用的执行步骤。如图 1.16 所示。

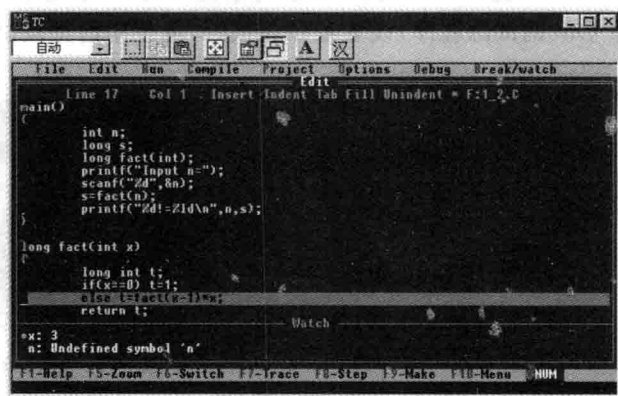


图 1.16 跟踪调试实例

假如调用函数时输入 $n=5$,在 watch 框内可以观察到 x 从 5 变化到 0 的过程;接着又能观察到 x 从 0 变化到 5 的过程;最后将结果带回到主调函数并返回主函数结束。

关于 Turbo C 更详细的信息,请参看 Turbo C 手册。

第2章 Visual C++ 6.0 开发环境

2.1 Visual C++ 及其开发环境概述

Visual C++ 是 Windows 环境下最主要的应用、开发系统之一。

Visual C++ 开发面向对象应用程序的两种方法：

一种是使用 Windows 提供的 Windows API 函数。API 是应用程序编程接口 (Application Programming Interface) 的缩写。API 为应用程序提供系统的各种特殊函数加上数字常量、宏、结构、类型以及其他项的集合, 这些编程元素是用 C 语言写的。有上千个 API 函数(窗口管理、图形设备、系统服务), 大多数 API 定义在 Windows.h 文件中, 也可以从软件开发工具包 (SDK, Software Development Kit) 中得到。尽管 Windows 本身是使用 C 语言编写的, 但它不可避免地使用了许多对象, 如窗口、对话框、控件、画笔和字体等, 每个对象都包含了许多数据和操作。

另一种是直接使用 Microsoft 提供的 MFC (Microsoft Foundation Class) 类库。MFC 不仅表示类, 它还表示 Windows 应用框架。MFC 是使用 C++ 语言编写的。MFC 类库中的 CWinApp 类封装了许多基本的应用程序。

Visual C++ 6.0 不仅是 C++ 语言的调试环境, 而且也是 C 语言极好的调试环境。为使读者尽快掌握它的使用方法, 下面将对这一开发环境进行介绍。

单击“程序”项中 Microsoft Visual Studio 6.0 中的 Microsoft Visual C 6.0, 进入 Visual C++ Developer Studio, Visual Studio 窗口如图 2.1 所示。

Visual Studio 通常称为 Developer Studio (开发工作室); 名词 IDE 表示集成化开发环境 (Integrated Development Environment)。本章将对常用的窗口、菜单、按钮做简要介绍。

2.2 Visual C++ 6.0 的菜单栏

Visual C++ 6.0 和 Word 2000, Excel 2000 等一样, 都是微软的产品, 很多用法都是相似的。在窗口的不同地方单击鼠标右键可以弹出相应的快捷菜单。在工具栏上单击鼠标右键可以弹出工具快捷菜单, 通过它可以增减工具栏上的工具, 一些常用的菜单命令都可以由工具按钮来完成。工具快捷菜单如图 2.2 所示。

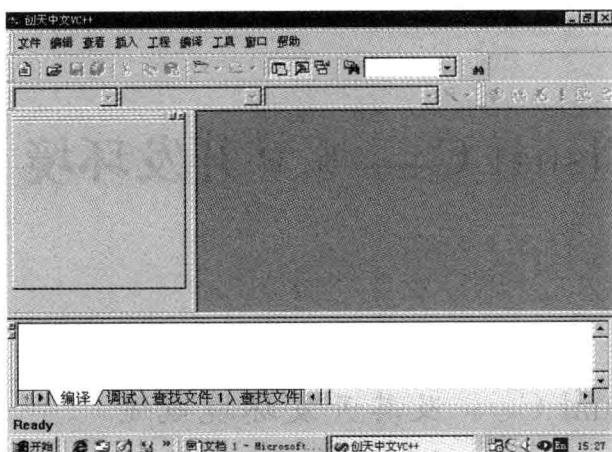


图 2.1 Visual Studio 窗口

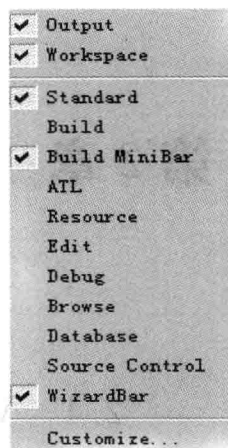


图 2.2 工具快捷菜单

2.2.1 File 菜单

File 菜单是对文件进行操作的,因此是最基本的,必须掌握。像新建、打开、存储、关闭(包括 workspace)文件等。

New 选项(新建文件)有下列 4 个选项,即文件(Files)、工程(Projects)、工作区(Workspaces)和其他文档(Other Documents)。缺省的是工程选项。

Files Projects Workspaces Other Documents

Project 如图 2.3 所示的选项卡中有 3 个选项要特别注意,它们是:

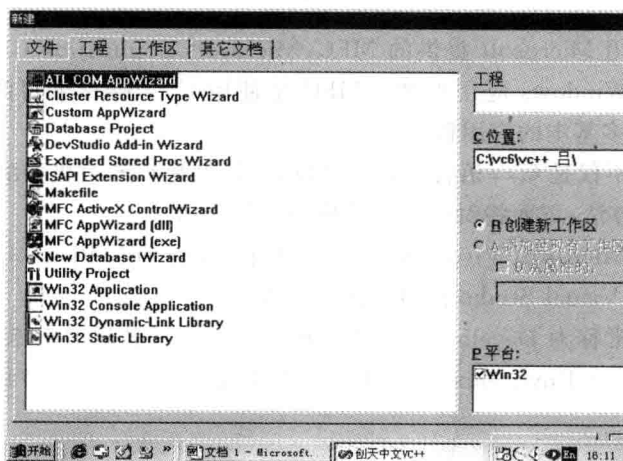


图 2.3 Project 选项

1. Win32 Console Application: 用于创建 Win32 控制台应用程序。调试 C/C 程序使用该选项。

2. Win32 Application: 用于创建 Win32 应用程序。使用 API 函数进行可视化编程时要使用这一选项。

3. MFC AppWizard[exe]: 用于创建 MFC 可执行程序。使用微软基础类库编写程序时要选该选项。

File 选项如图 2.4 所示。其中有两个选项用于编写 C/C++ 源程序。即：

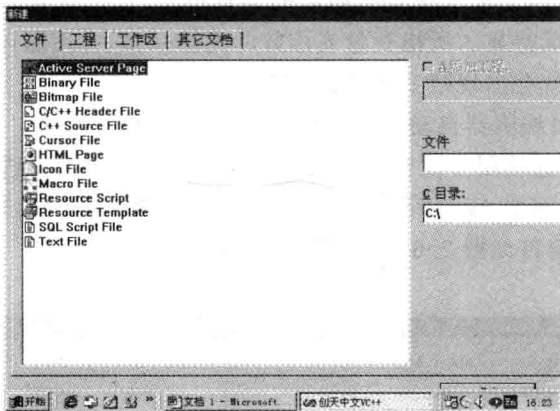


图 2.4 File 选项

C/C++ Header File: 创建 C/C++ 头文件。

C++ Source File: 创建 C++ 源程序文件(创建 C 源程序文件时,扩展名为 .c)。

Workspaces 选项卡和 Other Documents 选项卡分别用于创建新的工作区和创建新的文档。这里就不再详述,有兴趣的读者可参考有关书籍。

2.2.2 Edit 菜单

Edit 的菜单如图 2.5 所示。最下方的 4 个命令是 Microsoft Visual Studio 6.0 增添的功能,当用 MFC 编程时显得非常方便。

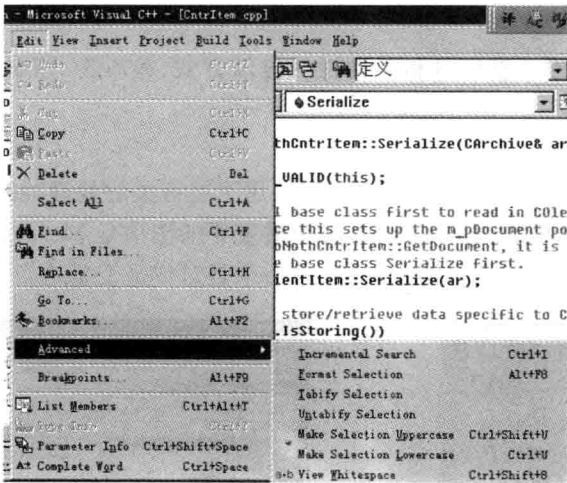


图 2.5 Edit 菜单

List Member: 列出光标左边的类或结构的有效成员变量和成员函数。即在变量名之后键入“.”或“->”,那么系统会自动列表显示所有有效的成员名,这时只要键入成员的前几个字母就可从列表中选中该成员,按 Tab 键即可完成输入;也可用滚动条找到待输入的成员名,然后用鼠标双击成员名来完成。

Type Info: 显示标识符的完整声明。即如果将鼠标指针停在某一变量或函数上,系统将