

21世纪高等学校教材

C语言程序设计 上机实验教程

主编 李 明

上海交通大学出版社

C 语言程序设计上机实验教程

主编 李 明



上海交通大学出版社

内 容 简 介

本书是《C语言程序设计教程》的配套教材,全书共分为6个部分,内容包括:常用的C语言编程环境、实验题目、实验参考答案、习题、习题参考答案、程序设计参考答案。

全书以提高学习者的程序设计能力为基础,以各章的知识点为主线,为每章设计了大量的典型的实验和习题,并为每章的实验和习题配备了参考解答,书中的实验和习题内容丰富,具有启发性、综合性和实用性,所有的代码均在 Visual C 6.0 环境下调试通过,可直接引用。

本书可作为C语言程序设计课程的上机实验教材,也可作为学习C语言的自学教材。

本套教材另配有电子教案(PPT格式)、全部程序代码,同时开发了基于 Windows 操作系统的《C语言程序设计教学网站》,实现了教学互动、网上作业、网上考试及自动化理论知识和编程操作技能的全部自动化判分,可供教材使用单位选用。联络邮箱: baiwen_sjtu@126.com

图书在版编目(CIP)数据

C语言程序设计上机实验教程 / 李明主编. — 上海: 上海交通大学出版社, 2008
21世纪高等学校教材
ISBN 978-7-313-05017-5

I. C… II. 李… 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(2007)第170025号

C语言程序设计上机实验教程

李 明 主编

上海交通大学出版社出版发行

(上海市番禺路877号 邮政编码200030)

电话: 64071208 出版人: 韩建民

上海交大印务有限公司印刷 全国新华书店经销

开本: 787mm x 1092mm 1/16 印张: 12.75 字数: 313千字

2008年1月第1版 2008年1月第1次印刷

ISBN978-7-313-05017-5/TP 681 定价: 20.00元

版权所有 侵权必究

前 言

C 语言程序设计课程是一门实践性很强的课程，没有实验教学，就没有程序设计。通过实验教学，才能理解程序，进而将程序设计真正地应用于实际。为配合《C 语言程序设计教程》的教学，同时为自学者提供一本参考用书，我们组织教学经验丰富的一线教师，编写了此书。

全书共分为 6 章，第 1 章为常用的 C 语言编程环境，介绍了 2 个常用的 C 语言编程环境，即 TC 环境和 VC 环境，并阐述了常用的调试程序的方法。第 2 章为实验题目，为每章精选了大量典型的实验题目。第 3 章为每个实验题目配备了参考答案。第 4 章为主教材的各个章节配备了习题。第 5 章为习题配备了参考答案。第 6 章为主教材的各个章节后的程序设计题配备了参考答案。所有的代码均在 Visual C 6.0 环境下调试通过，可直接引用。

本书由李明任主编，宣善立任副主编。参加编写的人员有：李明、宣善立、胡贤德、江晓舟、赵士阳等。

在编写本书的过程中，参考了大量书籍，得到了许多同志的支持，在此向广大同仁和所有参考书籍的作者表示衷心的感谢。

由于作者才学疏浅，书中错误和不当之处恳请专家、读者批评指正。

编 者

2007 年 10 月

目 录

第 1 章 C 语言上机过程及编程环境.....	1
1.1 C 语言程序开发过程.....	1
1.2 Turbo C 2.0 开发环境	3
1.3 Visual C++ 6.0 开发环境	10
1.4 程序的调试方法	15
1.5 C 语言编程常见错误.....	22
第 2 章 实验.....	27
实验 1 概述	27
实验 2 简单的 C 程序设计.....	28
实验 3 分支和循环的 C 程序设计.....	32
实验 4 数组	35
实验 5 指针	38
实验 6 函数	41
实验 7 结构和联合	45
实验 8 位运算	48
实验 9 文件	51
实验 10 综合程序设计实验	54
第 3 章 实验参考答案.....	66
实验 1	66
实验 2	66
实验 3	68
实验 4	73
实验 5	78
实验 6	80
实验 7	87
实验 8	97
实验 9	99

第 4 章 习题	105
习题 1	105
习题 2	106
习题 3	110
习题 4	116
习题 5	122
习题 6	129
习题 7	134
习题 8	142
习题 9	146
第 5 章 习题参考答案	154
习题 1	154
习题 2	155
习题 3	157
习题 4	158
习题 5	160
习题 6	161
习题 7	162
习题 8	164
习题 9	165
第 6 章 教材各章程序设计题参考答案	167
第 1 章	167
第 2 章	168
第 3 章	170
第 4 章	174
第 5 章	178
第 6 章	182
第 7 章	187
第 8 章	193
第 9 章	194
参考文献	198

第 1 章 C 语言上机过程及编程环境

C 语言自诞生以来,其开发环境经过了多年的发展完善,从早期的命令行编译链接方式逐渐演化出集成的开发环境,并随着操作系统的不断提升,其功能也日渐强大和复杂。为顺利完成各个 C 语言程序设计的实验,本部分将介绍两种操作系统(DOS、Windows)下典型的两个 C 语言开发环境最为基本的使用方法。

1.1 C 语言程序开发过程

C 语言是编译型的高级语言,与其他编译型高级语言程序的开发过程类似。一个程序的开发过程在这里是指从输入 C 语言的源程序到生成正确执行程序的过程,其包含了如下几个步骤。

1. 编辑

用户首先根据问题的需要编写相应的 C 语言源程序,然后借助于编辑软件通过键盘将源程序输入计算机内部的过程称为编辑。编辑是通过所谓的编辑软件输入 C 语言的源程序,并将其保存为扩展名为.c 的文件,该文件称为 C 语言的源程序。

在后续的工作中,如果程序出现问题需要修改源程序,此时也必须借助编辑软件对源程序进行修改,如插入、删除、更改、移动等编辑操作。

由此可见,编辑的作用是建立和修改 C 语言的源程序。

2. 编译

编译有以下两个作用。

(1) 检查语法错误。在编辑后,C 语言的源程序已经建立,但其语法是否正确,这就需要调用编译系统进行检查,如果程序中有语法错误,编译给出出错类型、出错位置等信息,供用户参考。出现错误后,必须回到编辑状态,找到并修改错误。再次编译,可能还会出现错误,继续找到并修改。这是一个反复的过程,这个过程不仅需要编译的帮助,也需要用户对程序的理解和经验,重复这个过程直到程序没有语法错误。

(2) 生成目标文件。C 语言源程序是以文本文件的形式存储的,它是高级语言的程序,计算机不能直接识别,因此,还必须借助翻译系统将高级语言的程序翻译为用机器语言描述的目标程序。

当程序没有语法错误后,编译将生成与 C 语言源程序同名的扩展名为.obj 的文件,该文件是二进制机器码描述的文件,也称为目标文件,可被计算机直接识别。

3. 链接

编译后生成的目标文件仍不能被计算机直接执行,因为经过编译的 C 语言源程序可能是由若干个独立的目标模块组成,这些独立的目标模块还必须装配为一个完整的整体,才可以

执行，进而完成预期的目标。

链接通常也称为连接，其主要的作用是将程序中使用的系统函数代码连接到目标文件中，形成可以执行的程序。该程序文件一般与源程序同名，且扩展名为.exe。

4. 执行调试

执行调试的目的是验证程序功能的正确性，一般是通过输入典型测试数据验证程序的结果是否正确。对于较小规模的程序执行调试比较简单，较大规模的程序必须通过 C 语言编程环境提供的工具和方法进行调试，后续内容将对其作进一步的介绍。

当执行结果与程序应该得到的结果不符时，则说明程序的算法有错误，这类错误一般称为逻辑错误。与语法错误不同，编译检查是无法发现逻辑错误的，因此必须通过一定的手段查出逻辑错误，并在源程序中加以改正，再重复以上的各个过程，直到程序正确。

C 语言程序开发的过程，可以表示为图 1-1 所示的流程。

目前的 C 语言开发环境对以上流程的各个操作进行了集成，形成了所谓的 IDE, Integrated Development Environment 即集成的开发环境，用户的编辑、编译、连接、执行调试都可在一个集成界面的环境下实现，从而方便了用户的使用。下面介绍两种常用集成开发环境的基本使用。

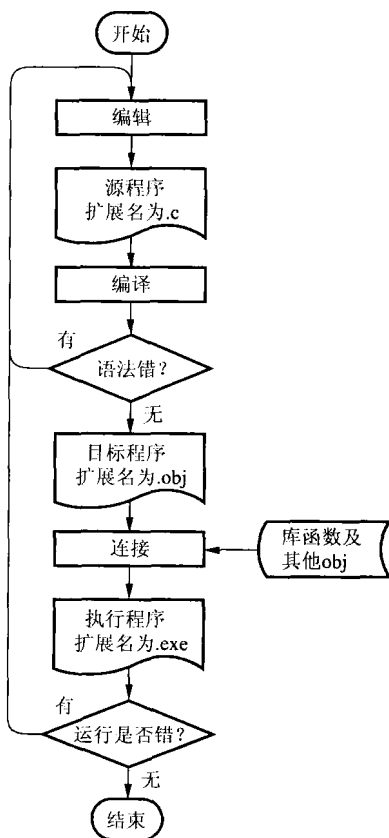


图 1-1 C 语言程序开发流程

1.2 Turbo C 2.0 开发环境

1. 概述

Turbo C 2.0 是 Borland 公司于 20 世纪 80 年代推出的一个 DOS 环境下的 C 语言开发平台, 简称 TC2.0。其特点是具有较为完善的集成开发环境、程序小巧、使用方便。随着操作系统的不断提升, DOS 操作系统已经退出了历史舞台, TC2.0 也退出了实际程序开发环境的行列。但就 C 语言教学而言, 其简单易学掌握的特点使初学者不必在初学伊始就面对复杂的编程环境, 因此 TC2.0 依然作为 C 语言教学的较佳选择。

2. 系统的安装与启动

(1) TC2.0 的安装。TC2.0 环境十分小巧, 整个环境只占 2M 多硬盘空间。用户可以直接将其复制到硬盘下, 例如将其复制到 E 盘(用户可根据需要选择)下, 文件夹名为 TC2。

(2) TC2.0 的启动。在 Windows 操作系统下启动 TC2.0 的步骤是:

① 打开我的电脑, 选择 E 盘, 进入 TC2 文件夹。

② 双击 TC2 文件夹中的 tc.exe(TC2.0 的编程环境程序)文件, 进入 TC2.0 集成开发环境, 如图 1-2 所示。

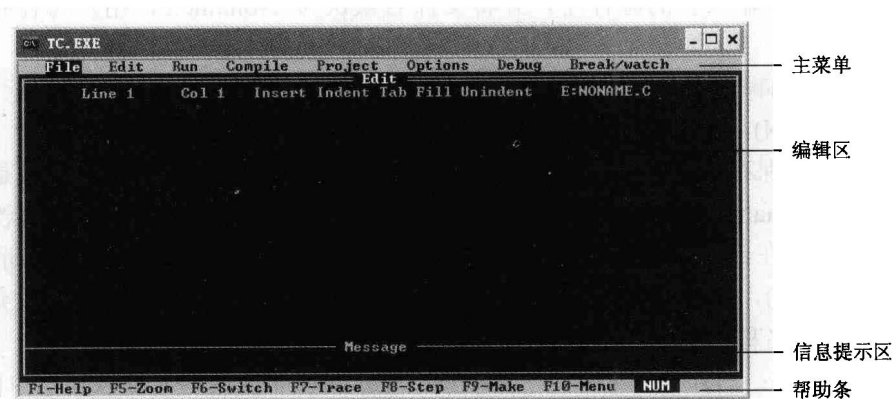


图 1-2 TC2.0 集成开发环境

3. TC2.0 的菜单

TC2.0 集成开发环境, 主要由主菜单、编辑区、信息提示区、帮助条四个部分组成。

(1) 主菜单。主菜单位于屏幕的上方, 其中含有 8 个子菜单项。选择并下拉菜单的最常见方法是按 Alt + 菜单的首字母(在 TC2.0 环境中为红色), 如打开 File 菜单, 按 Alt+F。

① File。此菜单实现与文件相关的各种操作, 其包含以下子菜单项。

a. Load: 加载。选择此子菜单项, 系统打开“加载”对话框, 如图 1-3 所示。在此对话框中, 用户可以选择需要装载的已经存在的源程序位置, 系统将显示此位置下的目录和 C 源程序的清单, 用户可选择需装载的文件名称, 系统将被选中的源程序文件调入内存, 供用户使用。图 1-3 显示的是 E 盘 TC2 子目录下的文件清单。

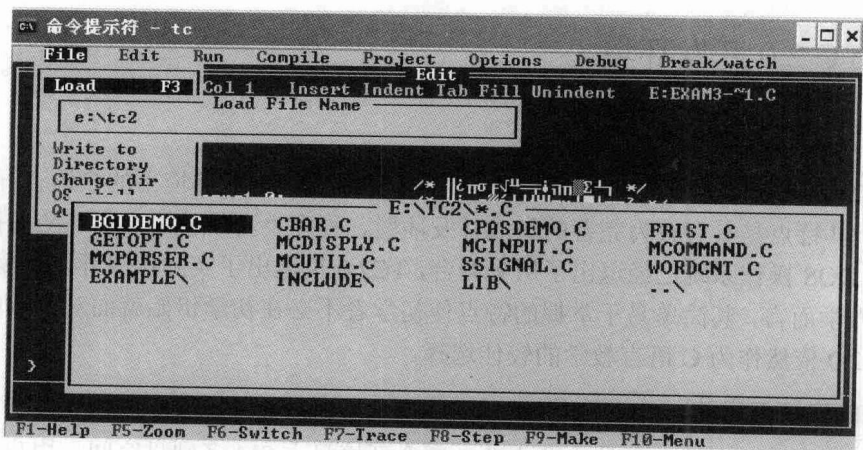


图 1-3 “加载”对话框

b. Pick: 选择。选择此子菜单项，系统将显示 Recent Files 列表框，并在此框中显示最近使用过的几个文件，用户可进行选择，系统将被选中的源程序文件调入内存，如果使用最近使用过的文件，用此方式比较方便快捷。

c. New: 新文件。选择此子菜单项，表示用户需要建立一个新的文件，系统将进入编辑状态，等待用户输入新的源程序，并将文件名默认为 Noname.c，用户可在存盘时为此文件更名。

如果此时编辑区中正在编辑其他的文件，且此文件尚未保存，系统首先提示 Noname.c not saved. Save?(Y/N)，提示用户先将正在编辑的文件保存，再新建文件。

d. Save: 保存。选择此子菜单项，表示将编辑区中的文件存盘。如果是首次存盘，则系统打开 Rename Noname 对话框，提示用户选择目录和名称；如果是再次存盘，则以当前文件名存在当前目录下。如果此目录下已有同名的文件，系统则提示：Overwrite Noname.c?(Y/N)，选择 Y 表示用新文件覆盖旧文件；选择 N 表示放弃，放弃后用户可重新选择目录和名称。

e. Write to: 另存为。选择此子菜单项，表示将编辑区中的文件以其他的目录或名称存盘，此时系统打开 New Name 对话框，提示用户选择目录和名称，系统则将此文件以用户选择的目录和名称保存。

f. Directory: 目录。选择此子菜单项，系统则打开 Enter File Name 对话框，用户可以在此对话框中选择所需文件路径，系统将显示此路径下的所有文件清单，供用户选择。

g. Change dir: 改变目录。选择此子菜单项，表示需要修改当前目录，系统显示 New Directory 对话框，用户可输入新的目录，系统则将新输入的目录设置为当前目录。

h. Os shell: 暂时退出。选择此子菜单项，可暂时退出 TC2.0 集成开发环境，回到 DOS 状态下，输入 DOS 命令进行操作。或用户可输入 EXIT 命令，回到 TC2.0 集成开发环境下。

i. Quit: 退出。选择此子菜单项，则退出 TC2.0 集成开发环境，回到相应的操作系统中。

② Edit. 选择此菜单，表示需要进行编辑操作。在编辑过程中，用户可以借助以下功能

键或编辑命令。

- a. F1: 打开 TC2.0 集成开发环境的 Help 功能, 以获取帮助信息。
- b. F6: 在编辑窗口和信息窗口之间切换。
- c. F10: 打开主菜单。
- d. Page Up: 向前翻页。
- e. Page Down: 向后翻页。
- f. Home: 将光标移到所在行的首部。
- g. End: 将光标移到所在行的尾部。
- h. Ctrl+Home: 将光标移到文件的首部。
- i. Ctrl+End: 将光标移到文件的尾部。

③ Run。此菜单实现执行程序并查看执行结果的各种操作, 其包含以下子菜单项。

a. Run。选择此菜单, 表示需要执行程序。如果源程序已经编辑、编译、连接通过, 则直接执行程序; 如果程序没有进行编译、连接, 则首先进行编译、连接, 如有错误则提示用户修改, 通过后才执行程序。

b. Program reset。选择此菜单, 表示中止当前程序的执行, 释放分配的程序空间。

c. Go to cursor。选择此菜单, 表示使程序执行到光标所在的行, 用在调试程序阶段。

d. Trace into。选择此菜单, 表示如果程序执行到一条调用其他用户定义的函数时, 程序的跟踪条将跟踪到该函数内部。

e. Step over。选择此菜单, 表示程序将单步执行, 即每一次只执行一条语句, 此方式常用在调试程序中。

f. User screen。选择此菜单, 表示需要察看程序的执行结果。C 的源程序执行后, 仍处在 C 语言的集成开发环境下, 而程序的执行结果显示在用户屏幕下。选择此菜单, 可切换到用户屏幕, 察看程序的执行结果。在用户屏幕下, 按任意键即可返回到集成开发环境。

④ Compile。此菜单完成编译连接等操作, 其包含以下子菜单项。

a. Compile to OBJ。选择此菜单, 表示将一个 C 的源程序文件编译生成.obj 的目标文件。编译分为 2 个阶段, 首先检查程序中的语法错误, 如果无语法错误, 则生成.obj 的目标文件。

b. Make EXE file。选择此菜单, 表示直接生成一个.exe 的文件。如果有语法错误, 则显示错误信息, 不生成.exe 的文件。

c. Link EXE file。选择此菜单, 表示将当前的.obj 的目标文件与各种库文件连接在一起生成.exe 的文件。如果编译通过, 但此项连接不通过, 一般为系统环境的问题, 如路径错误、系统文件丢失等。

d. Build all。选择此菜单, 表示将重新编译项目中的所有文件, 并组装为一个.exe 的文件。一般用在大系统中, 如多模块、多文件的开发中。

e. Primary C file。选择此菜单, 表示指定了主 C 文件, 在以后的编译中, 如果编译有错误, 则将此主 C 文件调入编辑窗口, 以供用户判断和修改。

f. Get into。选择此菜单, 用户将获得一些相关的信息, 如: 当前路径、源文件名、源文件所占字节的大小、编译中的错误数目、可用空间等。

⑤ Project。此菜单项可以对参加编译、连接的项目文件名等信息进行输入或选择。其包

含以下子菜单项。

a. **Project name**。选择此菜单，表示需要输入或选择参加编译、连接的项目文件。

b. **Break make on**。选择此菜单，用户可以确定退出 **Make** 编译的时机。如：**Warnings**(警告)、**Errors**(错误)、**Fatal Errors**(致命错误)、**link**(连接)之前退出 **Make** 编译。

c. **Auto dependencies**。选择此菜单，用户可以确定编译时是否检查源文件与对应目标文件的时间与日期。当此项开关置为 **ON** 时，表示编译时将检查源文件与对应目标文件的时间与日期；此项开关置为 **OFF** 时，表示编译时将不检查。

d. **Clear project**。选择此菜单，表示将清除 **Project/ Project name** 中的项目文件名。

e. **Remove messages**。选择此菜单，表示从信息窗口中清除所有的错误信息。

⑥ **Options**。此菜单实现系统各种选项的设置，其包含以下子菜单项。

a. **Compiler**。此菜单可以对硬件配置、存储模型、调试技术、代码优化、诊断消息控制、宏定义等进行设置和修改，其下仍有多个子菜单项。

b. **Linker**。此菜单可用于设置与连接有关的各种选择项，其下仍有多个子菜单项。

c. **Environment**。此菜单可用于设置文件是否自动存盘、制表键的大小、屏幕的大小等，其下仍有多个子菜单项。

d. **Directories**。此菜单可用于设置编译、连接时所需文件的路径，其下仍然有多个子菜单项。

e. **Arguments**。选择此菜单，将出现一个对话框，在此对话框中允许用户使用命令行参数。

f. **Save options**。选择此菜单，将把用户对集成环境所做的修改和设置保存到配置文件中，默认的配置文件名名为 **TCCONFIG.C**。

g. **Retrieve options**。选择此菜单，将装入一个配置文件到 **TC** 中，**TC** 将使用这个配置文件中的各种设置。

⑦ **Debug**。此菜单主要用于进行错误的查找和排除，其包含以下子菜单项。

a. **Evaluate**。选择此菜单，可以输入需要显示的表达式，系统则显示此表达式的值，用于跟踪程序的执行。

b. **Call stack**。选择此菜单，可以检查系统堆栈的情况。

c. **Find function**。选择此菜单，可以显示选择的函数情况。

d. **Refresh display**。如果用户偶然重写了编辑窗口的内容，可以用此菜单恢复编辑窗口的内容。

e. **Display swapping**。选择此菜单，可以设置程序调试时屏幕显示的转换方式。

f. **Source debugging**。选择此菜单，可以设置程序连接时可使用的调试程序。

⑧ **Break/watch**。此菜单主要用于监视程序的执行情况，其包含以下子菜单项。

a. **Add watch**。向监视窗口插入一个监视表达式，则程序执行时在监视窗口中显示此监视表达式的值，用于跟踪程序的执行。

b. **Delete watch**。从监视窗口中删除当前的监视表达式。

c. **Edit watch**。在监视窗口中编辑当前的监视表达式。

d. **Remove all watches**。从监视窗口中删除所有的监视表达式。

e. **Toggle breakpoint**。对光标所在的行进行设置或清除断点。

f. Clear all breakpoints. 清除所有的断点。

g. View next breakpoint. 将光标移动到下一个断点处。

(2) 编辑区。编辑区位于屏幕的中间位置, 是用户输入并修改源程序的工作区域。编辑区的第一行是编辑提示, 其中显示了当前光标所在的行、列的坐标, 以及编辑的 C 程序的路径及文件名等。

(3) 信息提示区。信息提示区位于屏幕的下部, 其标题是 Message, 其中主要显示用户在编译或连接过程中出现的错误提示。

(4) 帮助条。帮助条位于屏幕的最下面, 给出了常用操作的快捷键, 如 F10-Menu 表示按 F10 键选择菜单, F1-Help 表示按功能键 F1 查看系统提供的帮助。

提示: 用户可按 F6 功能键在编辑区和信息提示区之间切换。

4. 常用的环境设置

TC2.0 的 Options 菜单包含了系统各种参数的设置, 多数设置较为复杂, 涉及许多计算机的概念, 在此主要介绍 Directories, 即目录选项的设置。目录的设置是确定系统头文件的路径、系统标准库函数的路径、以及用户输出文件的路径。设置的方式是打开 Options 菜单, 选择 Directories, 设置各种路径, 如图 1-4 所示。

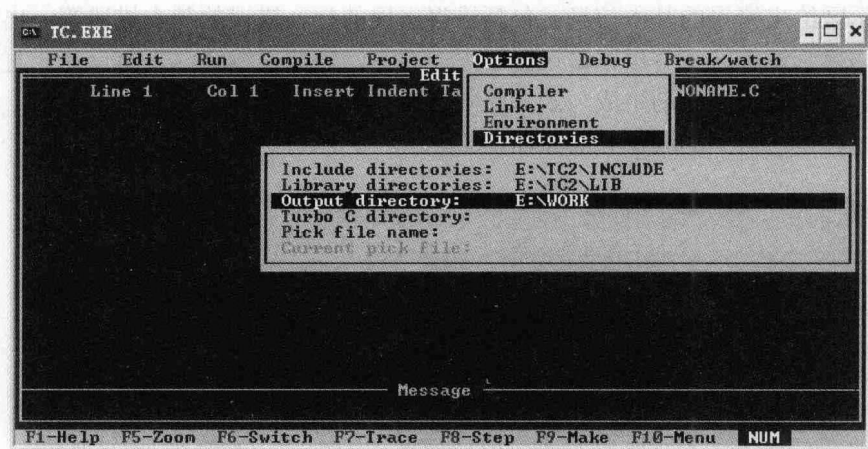


图 1-4 TC2.0 路径设置

选择 Directories 选项回车后, 在其下方出现各种路径的设置, 选中要设置的路径回车后即可输入路径。图 1-4 中 TC 复制在 E 盘的 TC2 文件夹下, 则必须将 Include directories 设置为 E:\TC2\INCLUDE, Library directories 设置为 E:\TC2\LIB。以上两个路径必须根据 TC2.0 环境安装的位置正确设置, 否则在编译时无法找到系统头文件, 连接时无法连接系统库函数。

Output directory 可以设置为用户建立的一个目录, 如本例为 E:\WORK, 则用户操作时的 .obj 及 .exe 文件都保存在 E:\Work 目录中。

设置完后可以按 Esc 键逐级返回到编辑区。

5. 程序开发步骤

下面通过一个简单程序, 说明在 TC2.0 中程序的开发过程。程序实现输入两个整数, 并输出两数之和, 代码如下:

```
#include <stdio.h>
void main(void)
{
    int a,b,c;
    printf("please enter 2 int data: ");
    scanf("%d,%d",&a,&b);
    c = a + b;
    printf("a+b=%d\n",c);
}
```

/* 注意此处丢失右大括号 */

开发步骤分为编辑、编译、连接、执行、查看结果 5 个过程。

(1) 编辑。启动 TC2.0，在程序编辑区中输入以上程序，完成后选择 File(文件)菜单中的 Save(存盘)命令，系统提示输入保存的文件名，用户可以输入保存文件的路径及文件名，本例是 E:\first.c。如此，编辑操作就完成了。

当然编辑时也可以调入已经存在的程序，可以选择 File 菜单的 Load 命令，输入编辑好的文件路径及文件名，即可调入对应程序。

(2) 编译。选择 Compile 菜单的 Compile to OBJ 命令，即可对已建立的 E:\first.c 文件进行编译。编译主要有两个作用：一是检查语法错误，二是生成.obj 文件。当有语法错误时，将在信息提示区给出错误的基本信息，包括错误出现的行、错误的基本描述等。出现语法错误后按 F6 切换到编辑区，并根据错误提示逐一改正，然后再编译，不断反复，直到没有语法错误。编译无语法错时自动生成与.c 文件同名的.obj 文件。

编译上述程序，得到的编译界面如图 1-5 所示。

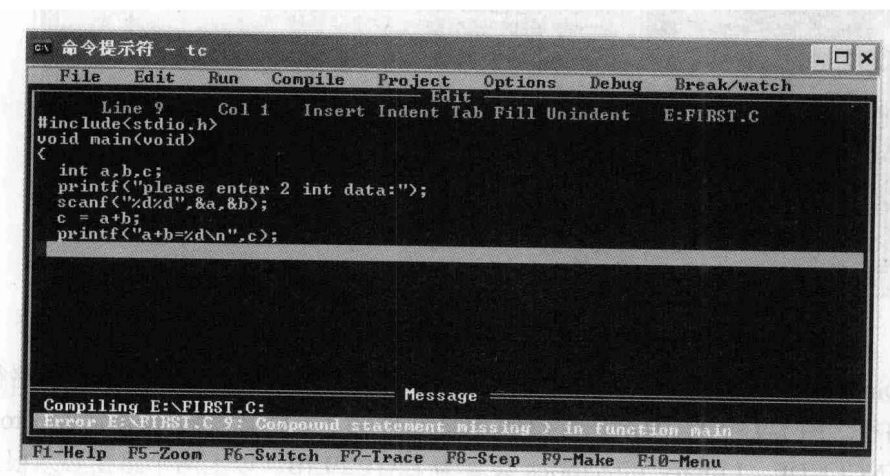


图 1-5 有错误的编译界面

从图 1-5 中可以看出，信息提示区中给出了错误的基本信息，即第 9 行产生了 1 个错误，错误信息是：Compound statement missing } in function main，即丢失了右大括号，同时编辑区中的光标位于相应的位置。按 F6 切换到编辑区，在相应的位置增加右大括号，再次编译，结果如图 1-6 所示。

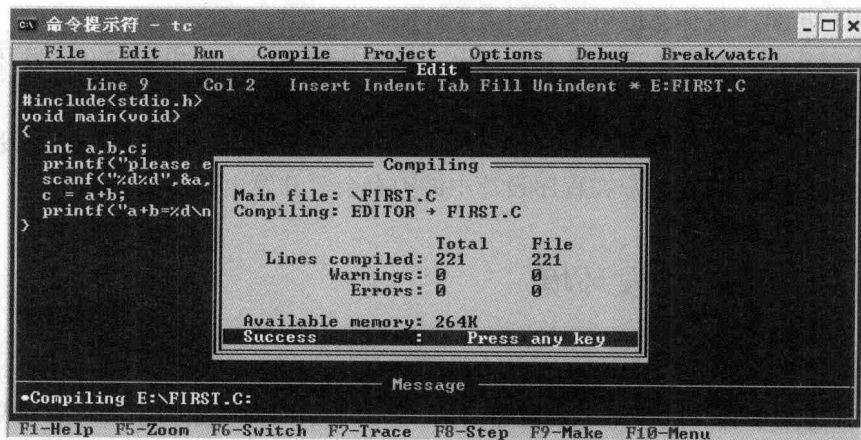


图 1-6 无错误的编译界面

从图 1-6 中可以看出，编译正确，即可进行下面的连接操作。

(3) 连接。连接的作用是将编译后生成的.obj 文件与系统的库函数连接，并生成.exe 文件。连接可选择 Compile 菜单的 Make EXE file、Link EXE file、Build all 命令。当没有连接错误时，则自动生成与.c 文件同名的.exe 文件。连接正确后，出现图 1-7 所示的信息框。

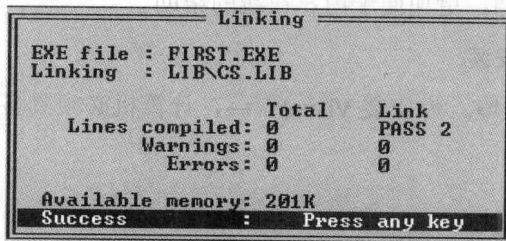


图 1-7 连接正确后的信息框

(4) 执行。生成执行文件后，用户必须执行该程序并验证结果的正确性。执行选择 Run 菜单的 Run 命令，也可直接按系统提供的快捷键 Ctrl+F9。Run 命令是一个复杂的批命令，如果用户完成编辑后没有编译连接，Run 命令则按编译、连接、执行的顺序完成各项操作。如果已经编译连接完成，则直接执行程序。如果程序中有输入，则执行提供输入窗口供用户按输入格式输入相应的数据。程序执行后，直接回到集成环境。

(5) 查看执行结果。程序执行后输出的结果在后台的用户窗口中，用户在集成环境中是看不到的。查看结果必须通过 Run 菜单的 User screen 命令或按快捷键 Alt+F5 切换到用户屏幕，用户屏幕包含了程序执行的输入输出的结果。本例的用户屏幕，如图 1-8 所示。

屏幕中的第一行 4,5 是程序执行时用户输入的数据，第二行 a+b=9 是程序的执行结果。根据程序的功能可以验证输入与输出的关系是否正确。如果结果不正确说明程序功能错误，这类错误是编译检查无法发现的，统称为逻辑错误。逻辑错误的检查要对程序的逻辑功能进

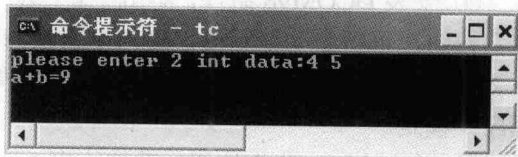


图 1-8 用户屏幕

行分析。在用户屏幕下按任意键可以返回集成环境。

程序开发结束后,可选择 File 菜单的 Quit 或按快捷键 Alt+X 退出 TC2.0 开发环境,返回到操作系统。

以上简单介绍的是 TC2.0 开发程序的过程,随着使用的深入,用户可以逐渐发掘编程环境中菜单的各项功能,使得用户在程序设计中游刃有余,开发出高质量的程序。

1.3 Visual C++ 6.0 开发环境

1. 概述

C++ 语言是在 C 语言的基础上发展起来的,它增加了面向对象的编程思想,成为目前主流的程序设计语言之一。Visual C++是微软公司开发的面向 Windows 平台的 C++语言开发环境,它不仅支持 C++语言,也支持 C 语言,成为常用的 C++语言和 C 语言的开发平台。下面简单介绍在 Visual C++环境下,执行 C 语言程序的过程。

Visual C++6.0 是 Microsoft 公司提供的基于 Windows 的 C++/C 集成开发环境,它提供了各类应用程序的开发环境,如命令行、Windows 应用程序等。Visual C++ 6.0 内置了 Microsoft Foundation Class(MFC),通过简单的继承可以生成丰富的 Windows 应用程序,同时提供了 MSDN 丰富的在线联机文档,更加便于用户在线的帮助。

2. Visual C++6.0 的安装

(1) Visual C++6.0 的环境。为安装 Visual C++, 计算机必须具备如下软硬件条件:

- ① Pentium 处理器。
- ② 32M 以上内存。
- ③ 200M 以上的可用硬盘空间。
- ④ 高速光驱。
- ⑤ 800×600 以上像素的显示器。
- ⑥ Windows98、Windows2000、Windows NT、Windows XP 操作系统。

目前市场上的计算机基本都具备以上条件。

(2) Visual C++6.0 的安装。Visual C++6.0 系统一般存放在光盘上,安装过程如下:

- ① 插入 Visual C++6.0 光盘,打开“安装向导”对话框。
- ② 单击“下一步”按钮,再单击“接受协议”按钮,此时系统提示输入产品序列号、用户名等信息,按要求输入。
- ③ 输入完成,单击“下一步”按钮,出现信息提示框,单击“确定”按钮,重新启动计算机,进入下一步的安装。
- ④ 安装 DCOM98 组件,单击“下一步”按钮,安装完成后,出现信息提示框,单击“确定”按钮,重新启动计算机,进入下一步的安装。
- ⑤ 在安装模式中一般选择“自定义”模式,也可根据需要选择“产品”或“服务器应用程序”模式。
- ⑥ 单击“下一步”按钮,系统给出“选择公用安装文件夹”的路径,一般按默认值,也可根据需要改变。

⑦ 单击“下一步”按钮，等待片刻，再单击“继续”按钮，进入正式安装过程，此时屏幕显示产品序列号。

⑧ 单击“确定”按钮，出现对话框，提示用户选择需要安装的组件。

⑨ 单击所需组件前面的复选框，再单击“继续”按钮，系统开始复制文件。

⑩ 安装完成，按照系统提示重新启动计算机。

3. Visual C++ 6.0 简介

(1) Visual C++ 6.0 的启动。Visual C++是一个集成的语言开发环境，首先必须在计算机系统中安装，安装完毕后，可按以下多种方式启动。

① 如果桌面上有 Visual C++6.0 的快捷方式图标，双击即可启动。

② 直接双击一个 C 语言或 C++语言的源程序，启动 Visual C++ 6.0。

③ 选择开始菜单的程序，选中程序选项中的 Microsoft Visual Studio 6.0 下的 Microsoft Visual C++ 6.0，也可启动 Visual C++6.0。

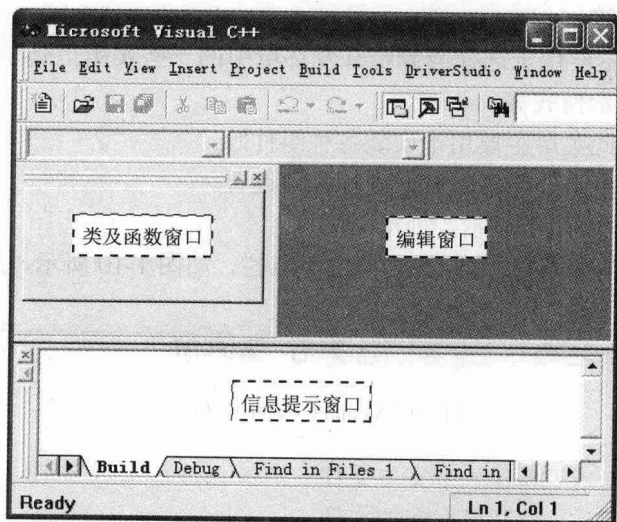


图 1-9 Visual C++主窗口

Visual C++启动后，其主界面如图 1-9 所示，由以下几个部分组成：

① 标题栏：标题栏位于屏幕的最上方，用于显示标题信息。

② 菜单栏：菜单栏位于标题栏的下方，为用户的各种操作提供菜单。

③ 工具栏：工具栏位于菜单栏的下方，为用户的各种操作提供方便快捷的工具按钮。

④ 类窗口：类窗口包含了用户的一些信息，如类、项目文件、资源等。

⑤ 编辑窗口：编辑窗口用于建立和修改源程序代码，以及资源文件、文档文件等。

⑥ 信息提示窗口：信息提示窗口用于显示编译时出现的错误信息，以及调试和查询的结果，以帮助用户查找错误。

⑦ 状态栏：状态栏位于屏幕的最下方，用于显示当前操作的状态、注释、光标所在的行和列的位置等信息。

(2) Visual C++ 6.0 的菜单。利用 Visual C++6.0 开发软件时，主要的工作都是通过菜单