



高等院校计算机应用技术规划教材

研究式学习— C语言程序设计习题与实验

楼 静 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等院校计算机应用技术规划教材

研究式学习—— C 语言程序设计习题与实验

主 编：楼 静

参 编：耿植林 张燕涛 付湘琼 黎 志

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C 语言是一种结构化程序设计语言, 由于它既可用于编写应用软件, 又可用于编写系统软件, 因此是一种应用广泛的高级程序设计语言。

《研究式学习——C 语言程序设计习题与实验》一书分为 4 个部分, 第 1 部分介绍 Turbo C++ 3.0 开发环境、Visual C++ 6.0 开发环境及程序调试; 第 2 部分是习题; 第 3 部分是实验, 实验类型包括验证型、引导型、扩充型和设计型 4 类; 第 4 部分是综合设计, 包括设计要求、学生设计实例、设计范例等。该书是教材《研究式学习——C 语言程序设计》的配套书, 其编写目的是帮助学生更好地巩固所学知识, 明确上机实习任务, 增强程序设计能力, 培养创新精神, 提高综合应用水平。

本书可作为高等院校各专业学生的 C 语言程序设计课的参考书。

图书在版编目 (C I P) 数据

研究式学习: C 语言程序设计习题与实验/楼静主编;
耿植林等编. —北京: 中国铁道出版社, 2006. 1
高等院校计算机应用技术规划教材
ISBN 7-113-06885-5

I. 研... II. ①楼... ②耿... 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 003224 号

书 名: 研究式学习——C 语言程序设计习题与实验
作 者: 楼 静 等
出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)
策划编辑: 严晓舟 邱秋罗
责任编辑: 严 力 翟玉锋 熊严飞
封面设计: 薛 为
封面制作: 白 雪
责任校对: 李 旸
印 刷: 北京新魏印刷厂
开 本: 787×1092 1/16 印张: 10.5 字数: 251 千
版 本: 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷
印 数: 1~5 000 册
书 号: ISBN 7-113-06885-5/TP·1718
定 价: 15.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

C 语言是一种应用广泛的程序设计语言，许多高校都把 C 语言作为学生学习程序设计的语言。本书有以下几个特点：

- 可直接作为学生的课外作业本和上机实习作业本，它有较强的针对性，能使学生在上机实习和课后练习中目标明确、重点突出。
- 实验内容的组织以及习题的选择，有利于学生加深理解和掌握所学的知识，有利于培养学生编写程序和调试程序的能力，有利于提高学习效率和学习效果。
- 将教材各章内容所对应的实验分为验证型、引导型、扩充型和设计型，可满足不同层次学生的学习需求，激发学生的学习兴趣，全面提高学生的实践能力和应用水平。
- 有关程序调试及测试的内容介绍可增强学生的程序调试能力。
- 通过参考学生设计实例并进行综合设计实践，可对学生进行实验思路、实验设计、实验技术以及发现问题、分析问题、解决问题能力的全面训练，开发学生的创造潜能，激发学生自主学习的意识，营造以学生为主体的学习氛围，从而达到提高教学质量和教学效果的目的。
- 第四部分中的综合设计案例，能帮助学生从理解程序基本要求开始，分析系统需求，划分功能模块，选择适当的数据结构和算法，编写功能模块函数，最后完成代码编写和程序的调试以及测试，从而使学生进一步理解结构化程序设计思想，掌握程序设计的基本方法，提高综合应用 C 语言进行程序设计的能力。

本书分为四个部分。第一部分由付湘琼、楼静、耿植林编写，第二部分由张燕涛、楼静编写，第三部分由楼静编写，第四部分由黎志、楼静、耿植林编写。由楼静负责全书的统稿工作，由李向阳负责审稿。

本书得到昆明理工大学精品课程的资助，得到教务处、教材科及计算中心领导和同志们的大力支持和帮助，在此，对他们表示感谢！同时对本书所有参考书籍、资料的作者们表示感谢！

由于编者水平有限，书中肯定存在不足之处，恳请各位专家、各位老师及广大读者给予批评指正，谢谢！

编 者

2005 年 12 月

第一部分 开发环境及程序调试.....	1
第1章 Turbo C++ 3.0 集成开发环境.....	2
1.1 开发环境简介.....	2
1.1.1 开发环境概述.....	2
1.1.2 环境设置.....	6
1.1.3 编辑、运行一个 C/C++ 程序.....	7
1.2 跟踪调试及设置断点.....	11
1.2.1 跟踪调试.....	11
1.2.2 设置断点和监视表达式.....	13
第2章 Visual C++ 6.0 开发环境.....	17
2.1 Visual C++ 6.0 开发环境.....	17
2.1.1 开发环境概述.....	17
2.1.2 菜单栏各项功能简介.....	18
2.1.3 开发环境的工具栏.....	21
2.1.4 工具栏的操作.....	23
2.1.5 Visual C++ 6.0 中常见的文件扩展名.....	24
2.1.6 新建、编辑、编译、连接、运行一个 C 程序.....	25
2.2 Visual C++ 6.0 中的程序调试过程.....	27
第3章 程序调试及测试.....	32
3.1 查找语法错误.....	32
3.2 逻辑错误.....	33
3.3 常见错误.....	34
3.4 结构测试法.....	38
第二部分 习题及模拟试题.....	41
第1章 C 语言程序入门.....	42
第2章 选择结构.....	46
第3章 循环结构.....	51
第4章 数组.....	55
第5章 函数.....	60
第6章 用户定制数组类型.....	66
第7章 指针.....	68
第8章 文件.....	72
期末考试模拟题.....	74
模拟题参考答案.....	80

第三部分 实验	81
实验一.....	83
实验二.....	85
实验三.....	89
实验四.....	93
实验五.....	97
实验六.....	101
实验七.....	105
实验八.....	109
实验九.....	113
实验十.....	117
实验十一.....	121
实验十二.....	125
实验十三.....	127
第四部分 综合设计	129
综合设计目的及要求.....	130
《C 语言程序设计》课程综合设计成绩考核表.....	132
学生设计实例.....	133
综合设计实例.....	147
参考文献	161

第一部分 开发环境及程序调试

第 1 章及第 2 章介绍 Turbo C++ 3.0 开发环境和 Visual C++ 6.0 开发环境，学生可以根据自己的实际情况，选择需要的开发环境来进行学习和实践。但应注意，不同的 C 版本或 C 开发环境是有一些差别的，在调试相同的程序时可能会出现不同的结果或问题，需要对程序作一些修改。

第 3 章介绍程序调试和测试的基本知识，它能使初学者更快、更好地掌握程序调试方法和测试方法，提高编程能力和效率。

这一部分的内容以自学为主，请课后认真阅读。

1.1 开发环境简介

1.1.1 开发环境概述

1. Turbo C++ 3.0 软件包的主要文件

Turbo C++ 3.0 软件包中主要包含以下几个文件：

• .BOL：一些常用的图形接口和文字处理相关的文件。
• .BIN：某些可执行文件，主要包含编译器和运行库。
• .EXE：某些可执行文件，如运行库。
• .LIB：某些库文件，如运行库。
• .H：某些头文件，如运行库。

• CLASSLIB：某些库文件，如运行库。

• DOC：某些文档文件。

Turbo C (简称 TC) 是美国 Borland 公司的产品, Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件, 如 Turbo BASIC, Turbo Pascal, Turbo Prolog, 这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品, 其中使用了全新的集成开发环境, 即使用了一系列下拉式菜单, 将文本编辑、程序编译、连接以及程序运行一体化, 大大方便了程序的开发。1988 年, Borland 公司又推出 Turbo C 1.5, 增加了图形库和文本窗口函数库等, 而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能, 并可以在 Tiny 模式下直接生成 .COM (数据、代码、堆栈处在同一 64KB 内存中) 文件。还可对数学协处理器 (支持 8087/80287/80387 等) 进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++, 它继承发展 Turbo C 2.0 的集成开发环境, 并包含了面向对象的基本思想和设计方法。

Turbo C++ 3.0 是 Borland 公司为 IBM PC 系列微机及兼容机开发的 C/C++ 语言编译软件包, 集编辑、编译、连接、调试等功能于一体, 是一个功能齐备的集成开发环境。它具有编译速度快、目标代码精练等优点, 在各种 C/C++ 语言编译器中占有十分重要的地位。

Turbo C++ 3.0 的编辑器是一个全屏幕、交互式的文本编辑程序, 提供了非常丰富的编辑命令, 支持鼠标和多文档操作, 使程序员可以方便地输入和修改源程序。Turbo C++ 3.0 的编译器和连接器提供了快速编译能力, 产生的可执行程序代码精练、运行速度快, 还能方便地实现 C/C++ 程序和汇编语言程序等的混合编程。Turbo C++ 3.0 的调试程序提供源程序级上的断点、单步运行功能, 使程序员可以很快地对错误定位, 加快了程序的调试过程。Turbo C++ 3.0 提供了丰富的库函数, 使得许多与 I/O 操作、字符及字符串处理、屏幕与图形操作、系统操作、数学函数等有关的工作均可以通过函数调用来实现, 极大地减少了程序编制的工作量。

1.1 开发环境简介

1.1.1 开发环境概述

1. Turbo C++ 3.0 软件包的主要文件

Turbo C++ 3.0 软件包中主要包含下面 7 个文件夹。

- BGI: 与一些特殊的图形接口卡和文字字体相关的文件。
- BIN: 系统可执行文件夹, 主要包括: 集成开发环境程序文件 (TC.EXE)、命令行编译程序文件 (TCC.EXE) 和其他实用工具程序文件 (系统初始安装时扩展名为 .COM 或 .EXE 的其他文件)。
- CLASSLIB: 扩展的面向对象的类库。
- DOC: 一些帮助文档。

- **EXAMPLES:** 范例文件夹, 后缀名为.C 或.CPP, 是系统带来的实例源程序, 可作为学习 C 语言的范例。通过仔细阅读这些程序, 可以从中学到许多编程技巧。
- **INCLUDE:** 头文件文件夹, 扩展名都是.H, 大致有 50 个。头文件包括库函数的原型、全局变量的说明、宏和结构的定义等, 它们的内容都是文本文件形式, 可以用 Turbo C++ 编辑器查看。在 C 源程序中可以用文件包含预处理语句将头文件包含到源程序中共同编译。例如:

STDIO.H 标准输入输出头文件

STRING.H 字符串处理头文件

CTYPE.H 字符判断头文件

CONIO.H 字符终端显示头文件

DOS.H DOS 调用头文件

- **LIB:** 标准库函数文件夹, 扩展名为.LIB。Turbo C++ 的库函数根据不同的存储模式分装在若干个库文件中, 如 CS.LIB 中存放小模式下的库函数目标代码, CM.LIB 中存放中模式下的库函数目标代码, 有关图形处理的库函数存放在 GRAPHICS.LIB 中等。

2. Turbo C++ 3.0 集成开发环境界面

Turbo C++ 3.0 集成开发环境界面如图 1-1-1 所示。

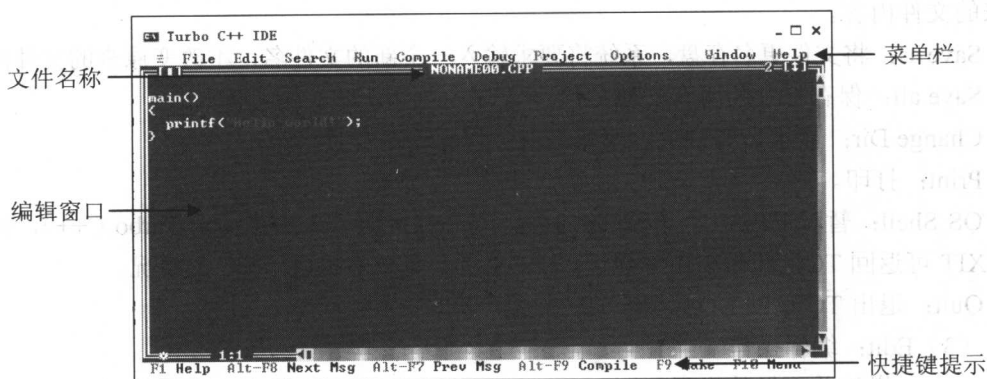


图 1-1-1 Turbo C++ 3.0 集成开发环境界面

3. 菜单操作

Turbo C++ 3.0 支持鼠标操作, 所有操作均可以用鼠标单击菜单命令完成, 操作方法与 Windows 的窗口操作基本类似, 比较直观, 在这里不作详细介绍。

如果用键盘操作, 进入 Turbo C++ 3.0 环境后, 按【F10】键可以随时激活主菜单, 按左、右方向键选择主菜单项, 按回车键确认选中当前菜单项。所有菜单项都有下拉子菜单, 有些菜单项还有 3 级菜单。对于下拉子菜单, 使用上、下方向键选择菜单项, 按回车键确认选中当前菜单项。

也可以通过按住【Alt】键再按住主菜单的首字母 (红色的大写字母) 来选中该菜单项, 如【Alt+F】键用于打开 File (文件) 菜单; 下拉子菜单中的菜单项可直接按首字母 (红色的大写字母) 来选中。

对于常用的菜单项, 可以按热键执行。在下拉菜单项名的右边给出了对应的热键, 如打

开文件的热键是【F3】，则任何时候按【F3】键都相当于执行菜单命令 File→Open。

4. 菜单栏

菜单栏中共有 11 个菜单选项，其主要的功能和子菜单如下。

(1) : 系统菜单

Repaint desktop: 用于重新刷新屏幕。在运行程序的时候可能会造成花屏，例如有时程序的输出结果会破坏集成开发环境的版面显示，执行该命令可以恢复正确的屏幕内容。

GREP: (Global Regular Expression Print) 全局正则表达式打印，是一种强大的文本搜索工具，类似于 UNIX 的命令 `grep`，它能使用正则表达式搜索文本，并把匹配的行打印出来。

Turbo Assembler: 调用外部的汇编语言的编辑编译环境 Turbo Assembler。

Turbo Debugger: 调用外部的调试工具 Turbo Debugger。

Turbo Profiler: 调用外部的优化工具 Turbo Profiler。

(2) File: 文件操作

New: 新建一个文件，默认的文件名为“NONAME00.CPP”。

Open: 打开文件，如果没有这个文件，则是新建文件。需要输入载入的文件名，该文件名可以带路径或通配符。

Save: 保存文件，如果文件未命名，系统将等待输入一个新的文件名。再次保存将覆盖原来的文件内容。

Save as: 将文件更名存盘，系统将要求输入一个新的文件名，不改变原来的文件内容。

Save all: 保存打开的所有文档窗口。

Change Dir: 显示当前目录，改变当前工作驱动器及目录。

Print: 打印。

OS Shell: 暂时退出 TC，转到 DOS 提示符下（此时并未真正退出 Turbo C++），执行命令 EXIT 可返回 TC。此命令用于想运行 DOS 命令而又不想退出 TC 的情况。

Quit: 退出 TC 返回 DOS。

(3) Edit: 编辑操作

Undo: 撤销前面做的步骤。

Redo: 重做。

Cut: 剪切。

Copy: 复制。

Paste: 粘贴。

Clear: 清除。

Show clipboard: 显示剪切板。

(4) Search: 查找源代码中的具体内容

Find: 查找具体内容，选择命令后弹出对话框。

Replace: 查找并替换某一特定内容。

Search again: 使用先前的参数重新查找。

Go to line number: 跳到源代码文档中具体的某一行。

Previous error: 定位到前一处错误。

Next error: 定位到下一处的错误。

Locate function: 用于查找某函数的定义,若该函数的定义不在当前编辑的程序中,可以将该源程序调入编辑器中。

(5) Run: 运行程序

Run: 运行程序,若自上次编译后源程序有变动,则自动启动 **Make** 命令来重新编译、连接程序。

Program Reset: 终止当前调试,释放分配给程序的内存空间,关闭已打开的文件。

Go to Cursor: 程序从开始执行到当前光标所在行终止,若光标所在行不含可执行语句,则显示一个 **Esc** 框作警告。

Trace Into: 运行当前函数里的下一条语句。若此语句不含调试器可访问的函数的调用,则停在下一条可执行的语句上。

Step Over: 单步执行。不跟踪进入下一级函数,执行当前函数的下一条语句。

Arguments: 允许程序员给出运行程序时的命令行参数,以便在 **TC** 下运行程序时可以从 **DOS** 状态下得到参数并传送到程序中。

(6) Compile: 编译、连接源程序

Compile: 将当前编辑器中的源程序文件编译成 **.OBJ** 文件。编译中会弹出一个窗口显示编译结果,编译完后按任意键即可关闭该消息窗口。若出现编译错误则转到消息窗口的第一条错误上。

Make: 生成可执行文件。

Link: 把当前的 **.OBJ** 文件及库文件等连接在一起,生成 **.EXE** 文件。不进行过时检查。

Build All: 不论是否过时都重建项目里的所有文件。

Information: 获得信息。将打开消息窗口显示工作目录、当前源文件名、当前源文件的字节数、内存可用空间等信息。

Remove Messages: 清除消息窗口中的错误信息。

(7) Debug: 控制集成调试器中除断点和监视表达式以外的选项

Evaluate/Modify: 用于计算变量和表达式的值并显示出来。常用于在跟踪程序执行过程中查看变量或表达式的值。

Call Stack: 在弹出的窗口中显示函数调用栈,说明当前正在执行的函数所嵌套调用的层次关系。

Watches: 包括 **Add Watche...**、**Delete Watche**、**Edit Watche...**、**Remove All Watches**。

Toggle breakpoint: 设置或除去光标所在断点,断点被设置后就用高亮显示。

Breakpoints: 断点管理对话框,能对断点进行定位、删除、编辑和修改。

(8) Project: 对工程文件进行管理

Open Project: 选择一个工程文件名,其扩展名为 **.PRJ**,工程文件名也是后面要生成的可执行文件名和 **MAP** 文件名。

Close project: 关闭当前的工程。

Add item: 添加一项到工程中,例如可以添加一个 **.CPP** 类的源程序到工程中。

Delete item: 从工程中删除一项。

Local options: 修改工程中具体一个源代码文件的一些参数选项,例如:修改编译后的输出名,源代码为 STARS.CPP,那么目标文件默认为 STARS.OBJ,在这里可以修改为其他名称。

Include files: 显示工程中所有的包含文件所在的路径。

(9) **Options:** 设置集成开发工作环境

Compiler: 编译器选项设置。包括硬件配置、存储模型、调试技术、代码优化、诊断信息控制及宏定义。

Linker: 设置连接器的选项。包括映射文件、段初始化、默认库、图形库等。

Environment: 环境设置。

Directories: 设置系统文件的查找路径和生成文件的存放路径。常常需要根据本机上 Turbo C 系统的安装情况进行相应的设置,否则可能出现编译、连接错误。

Save: 将上述的选项设置参数保存起来,供下次启动 TC 时使用。可以分别选择保存环境、桌面和项目等。

(10) **Window:** 显示各种窗口

Size/Move: 查看窗口大小或移动窗口。

Zoom: 使编辑窗口放大到充满整个开发界面。

Tile: 平铺多个窗口。

Cascade: 层叠多个窗口。

Next: 下一个窗口。

Close: 关闭当前文档窗口。

Close All: 关闭所有文档窗口。

Message: 显示消息窗口。

Output: 显示输出窗口。

Watch: 显示监视窗口。

User screen: 显示当前的 DOS 屏幕,按任意键可返回 Turbo C 屏幕。

Register: 显示寄存器的状态。

List all: 列出所有窗口。

(11) **Help:** 帮助

Contents: 帮助的内容。

Index: 对帮助的索引。

Topic search: 按帮助的字母顺序进行查找。

Previous topic: 先前的帮助主题。

Help on help: 对于帮助的帮助,介绍帮助的使用方法。

About: 关于,介绍软件的版本开发商等。

1.1.2 环境设置

Turbo C++在编译和连接时常常要用到一些头文件和库文件,还会生成一些新文件。系统必须知道这些文件分别放在什么地方。在编译程序时,如果有类似于“Unable to open include

file 'STDIO.H'”的错误信息,说明头文件的路径指定错误;而“Unable to open include file 'COS.OBJ'”之类的错误则是库文件的路径指定错误。此时,需要修改标准查找路径。

设置 TC++ 工作目录的操作步骤如下。

(1) 选择菜单命令 File→shell,临时返回到 DOS 状态。

(2) 用 DIR *.H /S 命令查找头文件所在目录,记下其路径,如 D:\TC30\INCLUDE。

(3) 用 DIR *.LIB /S 命令查找库文件所在目录,记下其路径,如 D:\TC30\LIB。

(4) 执行命令 EXIT,返回 TC++ 集成工作环境。

(5) 选择并执行菜单命令 Options|Directories,在弹出的对话框的“Include Directories”文本框中输入头文件所在路径,如 D:\TC3.0++\INCLUDE,则头文件所在目录已设置好。在“Library Directories”文本框中输入库文件目录,如 D:\TC3.0++\LIB;在“Output Directories”文本框中输入编译连接过程中产生的目标文件和可执行文件的存放目录,未指定则存放在当前目录下;在“Source Directories”文本框中设置源代码的工作目录,未指定则存放在当前目录下。环境设置对话框如图 1-1-2 所示。

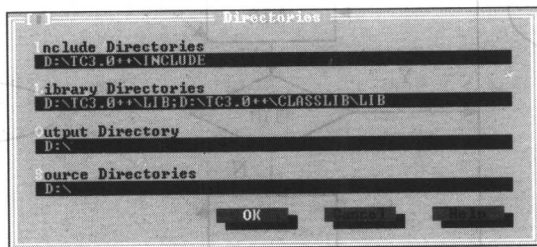


图 1-1-2 “Directories”环境设置对话框

(6) 设置完成后用鼠标单击【OK】按钮,再重新进入 Options 菜单,选择执行 Save 命令,保存刚才完成的设置,如图 1-1-3 所示。

说明:以上(1)~(4)步也可在 Windows 下通过搜索文件和文件夹来完成。

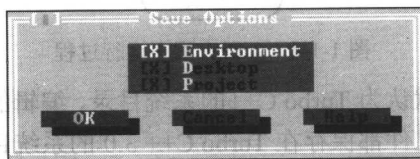


图 1-1-3 “Save Options”对话框

1.1.3 编辑、运行一个 C/C++ 程序

计算机只能识别用机器语言编写的程序,用高级语言编写的程序必须经过解释或编译后计算机才能执行。C 语言和其他编译型的高级语言一样,运行过程一般要经过 4 个步骤,即源程序的编辑、源程序的编译、目标程序的连接和可执行程序的运行。整个运行过程如图 1-1-4 所示。

下面以一个简单的 C/C++ 程序为例来说明整个操作过程。

1. 启动 TC

如果 Turbo C++ 3.0 系统安装在 D:\TC30 子目录下,可以在 DOS 命令窗口中执行下列 DOS 命令来启动 TC。

D: [回车]

CD\TC30\BIN [回车]

TC [回车]

也可以在 Windows 窗口中, 通过双击目录 D:\TC30\BIN 下的 TC.EXE 文件来启动 TC。

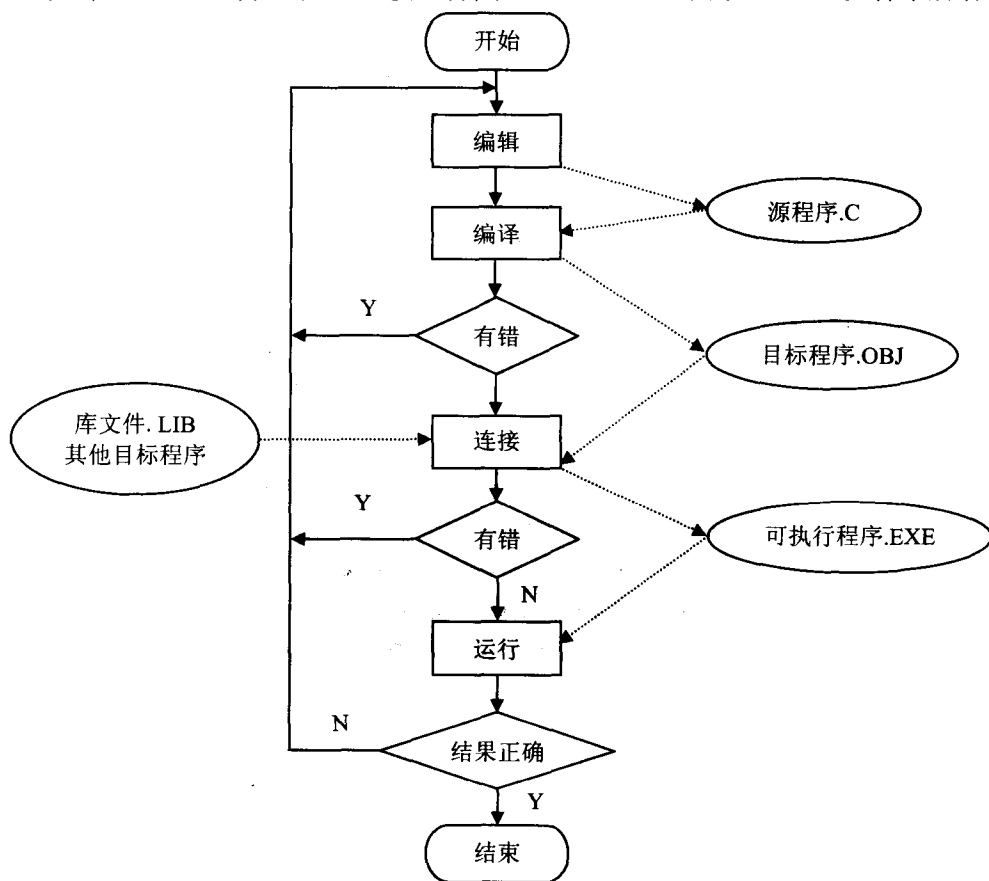


图 1-1-4 C 语言程序运行过程

启动 TC 后的工作目录默认为 Turbo C++ 的系统目录, 编辑的源程序文件、编译生成的目标文件、连接生成的可执行文件都保存在 Turbo C++ 3.0 的系统目录下, 这样不利于系统的安全与维护, 特别是当多个人在同一台微机上编程时还会带来更多的麻烦。因此, 建议在上机时先创建用户自己的工作目录, 并且每次上机时都进入自己的目录中操作。例如, 某学生的学号为 00156001, 在 E 盘根目录下以学号为目录名创建子目录的 DOS 命令为:

E: [回车]

MD \00156001 [回车]

在进入自己的工作目录后, 再设置如下所示的 TC 可执行文件的查找路径, 就可以在自己的工作目录下启动 TC 了。

CD \00156001 [回车]

SET PATH=%PATH%;D:\TC30 [回车]

TC [回车]

以上设置 TC++ 查找路径的操作也可以在自动批处理文件 C:\AUTOEXEC.BAT 中用 PATH 命令一次完成, 而不必每次上机都做。具体操作过程请参阅有关 DOS 命令手册, 或向

上机指导教师请教。

子目录的建立也可以在 Windows 下通过新建子文件夹来完成。

2. 编辑源程序文件

进入 Turbo C++ 3.0 集成开发环境后,系统默认新建了一个文件名为“NONAME00.CPP”的文件,在编辑窗口中会出现闪烁的编辑光标,表示可以输入源程序的内容。如果启动 TC 后就自动打开了一个源程序文件,可用鼠标单击 File 菜单,选择 New 命令来新建一个源程序文件,系统默认的文件名为“NONAME00.CPP”。

输入下列语句,每个语句结束后按回车键,注意字母的大小写,标点符号用英文标点。

```
#include "stdio.h"
#include "conio.h"
main()
{ char name[81];
  clrscr(); /*清屏*/
  printf("What's your name?");
  scanf("%s",name);
  printf("\nHELLO,%s\n",name);
}
```

若输入中出现错误,可用鼠标定位,也可用上、下、左、右方向键移动编辑光标到错误字符位置,按【Delete】键删除错误字符。请注意编辑窗口中光标的形状,若形状为竖线,表示处于插入状态,新输入的字符将插到当前光标前;若为方框,表示处于改写状态,新输入的字符将改写当前光标位置的字符。按【Insert】键可以切换插入/替换状态。

另外,复制、粘贴、删除等编辑命令能大大提高编辑源代码的速度,这些操作与 Windows 编辑操作一样,只是一些快捷键不相同。例如:拖动鼠标选择要复制的语句,此时被选中的语句用反白色表示,选择 Edit 菜单中的 Copy 命令(或用快捷键【Ctrl+Insert】),将选中的语句复制到剪贴板中,再移动文字插入光标到想插入文字的地方;选择 Edit 菜单中的 Paste 命令(或用快捷键【Shift+Insert】),此时剪贴板中的语句就被复制到插入点的后面,插入的内容用反白色表示。

3. 保存源程序文件

选择 File 菜单中的 Save 命令(或直接按【F2】键),出现“重命名”文本输入框,按退格键删除 NONAME.C,再输入新文件名,如 EXP1.CPP,按回车键或单击【OK】按钮。此时,编辑窗口上方中部的文件名已从“NONAME00.CPP”变成了“EXP1.CPP”,如图 1-1-5 所示。

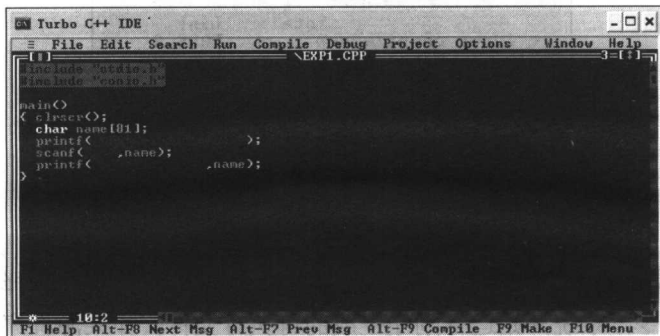


图 1-1-5 文件名为“EXP1.CPP”的窗口

4. 编译源程序

选择 Compile 菜单中的 Compile 命令，系统开始编译源程序，逐行检查语法错误，并显示编译消息窗口报告编译情况，如图 1-1-6 所示。

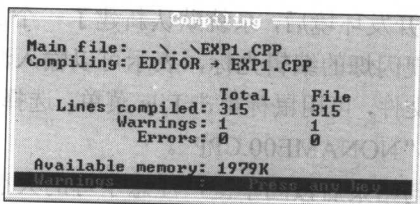


图 1-1-6 编译消息窗口

编译成功则生成目标文件“EXPI.OBJ”，并出现“Success: Press any key”的提示，按任意键关闭编译消息窗口。编译中出现错误则在末行出现“Error: Press any key”的提示，按任意键关闭编译消息窗口后，在消息（Message）窗口显示错误信息，如图 1-1-7 所示。再修改语法错误，重复编译，直到通过编译生成目标文件为止。

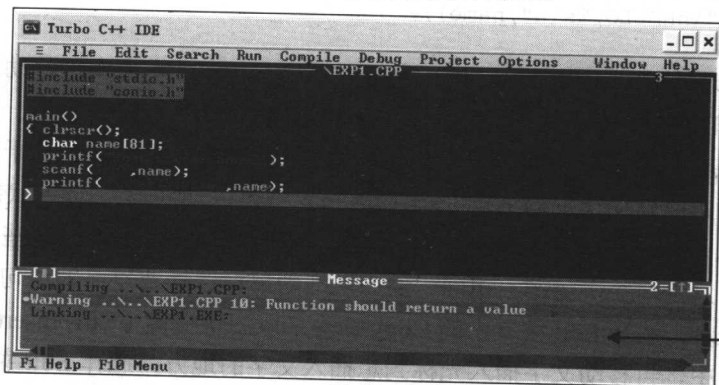


图 1-1-7 消息窗口

5. 生成可执行文件

选择 Compile 菜单中的 Make 命令，或者选择 Link 命令，或者选择 Build All 命令，将连接目标文件生成可执行文件“EXPI.EXE”。连接消息窗口报告连接情况，如图 1-1-8 所示。

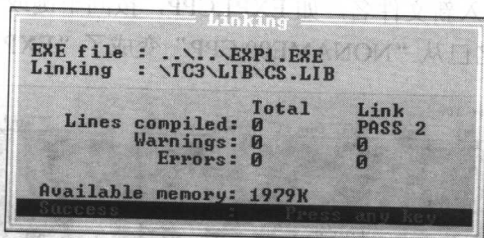


图 1-1-8 连接消息窗口

6. 运行程序

可以在 TC++ 集成开发环境中运行程序，用鼠标单击 Run 菜单，选择 Run（或者直接按【Ctrl+F9】键）命令，系统开始运行刚才生成的可执行文件。如果源程序尚未编译连接，此操作会自动编译源程序、连接目标程序生成可执行文件，并自动运行可执行文件。

程序运行时,在屏幕上显示“**What's your name?**”,等待键盘输入姓名,如图 1-1-9 所示。当输入姓名并按下回车键后,系统立即返回 TC++ 编辑环境。

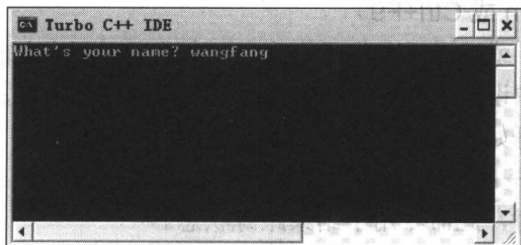


图 1-1-9 程序运行界面

7. 查看运行结果

选择 Window 菜单中的 User screen 命令(或者直接按【Alt+F5】键),可以在用户窗口查看运行结果,如图 1-1-10 所示。查看完毕按任意键立即返回 TC++ 编辑环境。

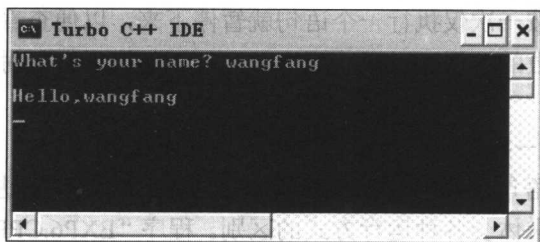


图 1-1-10 运行结果界面

1.2 跟踪调试及设置断点

1.2.1 跟踪调试

由于程序中存在分支、循环、函数嵌套调用、递归等结构,造成了程序运行时的变化规律和其静态结构之间存在一定的差异,因此仅靠阅读程序本身很难掌握程序运行时各种变量内容的动态变化,这就给调试程序和测试程序带来了很大的困难。如果能在程序运行过程中动态地显示程序执行的流向和各变量的内容,则有助于程序员了解程序的动态运行情况,从而更好、更快地调试程序。

Turbo C++ 集成开发环境中的集成调试器可以使程序在任何地方停止运行,以便程序员检查、修改变量的值。要调试程序时,必须将编译环境参数设置正确,即将调试开关打开。操作方法为:选择菜单命令并将 Options→Compiler→Advance Code generation→Option→OBJ debug information 设置为 On(系统默认值即 On),将 Debug→Source Debugging 设置为 On(系统默认值就是 On)。

跟踪调试工具主要来自两个下拉子菜单:Run 和 Debug。搭配使用这些功能将使程序调试变得非常容易。Run 菜单中的选项负责跟踪程序代码,使程序处于被跟踪的状态;Debug 菜单中的选项可以计算、显示表达式的值,为变量临时赋上一个新值,可以设置程序断点,增加监视表达式以便随时显示某变量或表达式的值等。