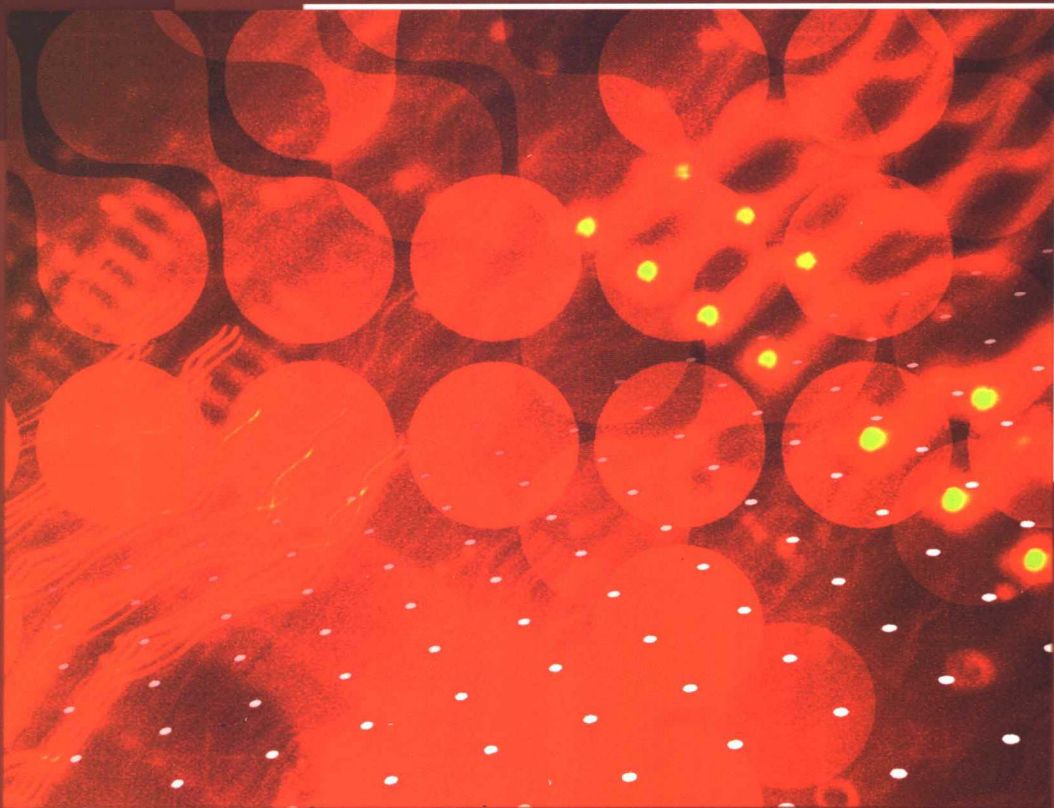


# C 语言程序设计 实验教程



汪同庆 关焕梅 汤洁 主编



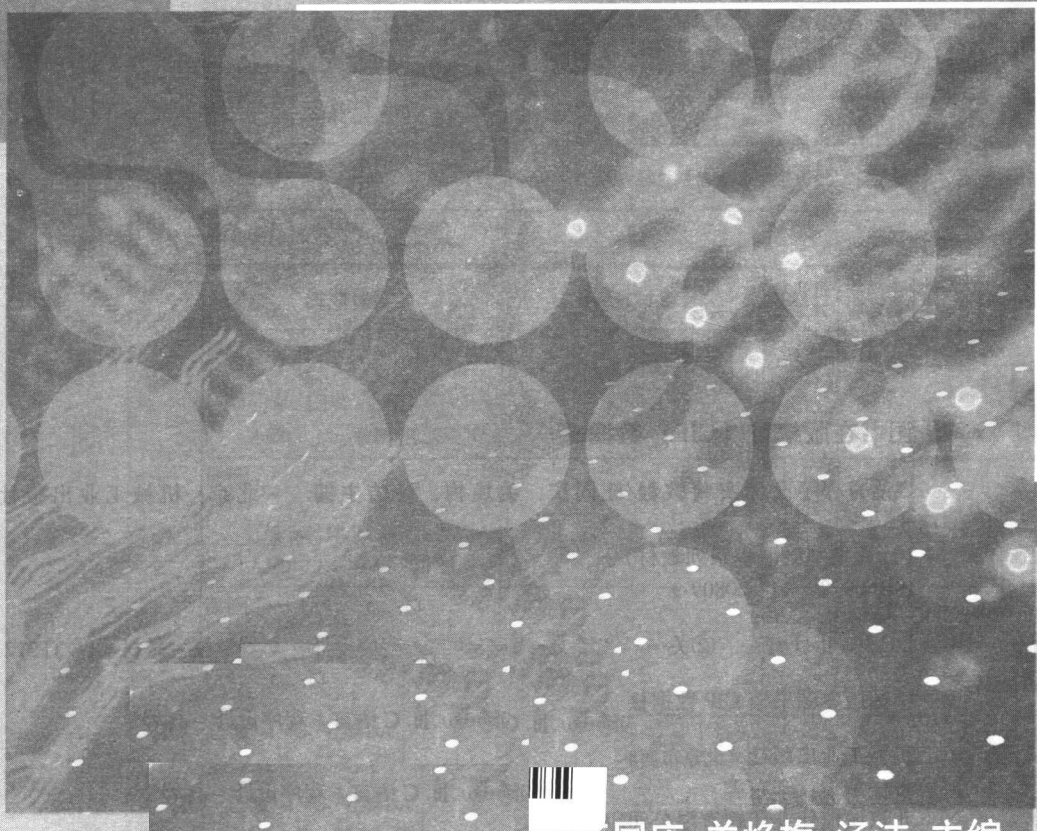
机械工业出版社  
China Machine Press

TP312  
2295C

2007

计算机基础课程系列教材

# C 语言程序设计 实验教程



汪同庆 关焕梅 汤洁 主编  
张华 滕冲 黄磊 杨先娣 黄文斌 参编



机械工业出版社  
China Machine Press

本书是机械工业出版社出版的《C 语言程序设计教程》的配套教材,在理论、操作和编程实践上对教材进行了补充,具有很强的实用性。书中详细介绍了 Visual C++ 6.0 和 Turbo C 2.0 两个集成开发环境的使用,共设置了 35 个实验,并有大量的练习题及其参考答案。

本书具有基础性、实用性和系统性,同时充分考虑了与其他教材内容的兼容性,可作为各类高等院校非计算机专业计算机公共基础课程的实验教学用书,也可供计算机等级考试和自学参考。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

## 图书在版编目 (CIP) 数据

C 语言程序设计实验教程/汪同庆,关焕梅,汤洁主编. - 北京:机械工业出版社, 2007.3

(计算机基础课程系列教材)

ISBN 978-7-111-20809-9

I. C… II. ①汪… ②关… ③汤… 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 011783 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:李东震

北京诚信伟业印刷有限公司印刷·新华书店北京发行所发行

2007 年 3 月第 1 版第 1 次印刷

184mm×260mm·9.5 印张

定价:16.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换  
本社购书热线:(010) 68326294

# 计算机基础课程系列教材

## 编 委 会

主 任：刘 国 唐碧龙

副主任：刘春燕 许云涛 汪同庆

委 员：熊建强 何 宁 黄文斌

康 卓 吴黎兵 黄 华 高建华

# 序 言

自 20 世纪 80 年代以来,我国计算机基础教育健步发展,已经取得巨大成就。特别是 1997 年教育部高教司颁发了《加强非计算机专业计算机基础教学工作的几点意见》(即[1997]155 号文件)和 2004 年发布了《关于进一步加强高校计算机基础教学的意见》的“白皮书”之后,全国高校计算机基础教育逐步走上了规范化的发展道路,正在向纵深发展。

但是,面向高等学校非计算机专业的计算机基础教学既有它的广泛性,也有它的特殊性。一方面,要让学生掌握必要的基础、最新的知识,以适应市场对人才的使用和需求;另一方面,要将计算机基础教学课程的知识性、技能性和应用性相融合,培养学生综合运用知识的能力,将体验与专业应用接轨。随着目前我国高等学校招生规模的日益扩大,按市场需求培养应用型人才是我国今后高等教育办学的主要方向。

大学非计算机专业的学生除了必须具备扎实的相关专业知识外,还必须掌握计算机应用技术,这是信息化时代对人才素质的基本要求。因此,在进行非计算机专业计算机基础教学过程中,应着力培养学生成为既有扎实的专业知识,又熟练掌握计算机应用技术的复合型人才。

为了适应新的形势,更好地满足高等学校非计算机专业计算机基础教学的需要,我们组织编写了这套“计算机基础课程系列教材”。参加编写的人员都是长期从事计算机基础教学第一线的教师,他们在认真总结多年教学经验的基础上,通过到各类学校调研,反复征求各高校教务部门的意见,取得了共识。

本次推出的系列教材包括:《计算机基础应用教程》、《C 语言程序设计教程》、《数据库技术应用教程》、《计算机网络与应用技术》、《网页制作与应用设计》、《多媒体技术应用基础》、《统计分析系统》、《Visual C++ 程序设计》等,并有配套的实验教程。

本系列教材具有以下特点:

- 选材新颖,构架独特。各书按应用型人才培养模式进行选材,力求在基础性层面上反映当今最新应用成果,摒弃难点中的沉滞部分,新增或扩充重点中的基础内容;在章节的构架上具有新的特色,便于学生自学和老师教学。
- 实用性强,注重应用能力培养。各书尽量不涉及过多的理论问题,强调内容的实用性,注重培养学生分析问题和解决问题的能力,提高学生的创新思维能力。
- 体现案例教学的全新教学思想。凡是涉及应用性知识的章节,各书均以一个或多个案例为引子,然后通过案例导出知识点加以阐述和讲解。这样,学生对所学的知识更容易理解和掌握,同时通过案例分析达到举一反三的效果。
- 具有完备配套的辅助教学资源。(除《统计分析系统》外)各书均配有教学实验教程,以提高学生的实践能力和对知识的体验;各书配有电子教案,教师可登录网站([www. hz-book. com](http://www.hz-book.com))免费下载。

本系列教材主要针对大学非计算机专业学生编写,是一套新颖、实用的应用型教材。它体现了作者们为培养应用型人才辛勤劳动、勇于探索的教学改革精神和成果,也凝聚着他们多年丰富的教学经验和心血。

本系列教材得到武汉大学计算中心、武汉大学东湖分校的领导和老师的大力支持,在此表示衷心感谢。

由于计算机技术发展十分迅速,以及非计算机专业计算机基础教学的广泛性和特殊性,而且限于编者水平,书中难免存在不少缺点和不足,敬请广大读者批评指正。

编委会  
2006 年 7 月  
于武汉大学

# 前 言

C 语言是一种功能强大、编程灵活、特色鲜明、深受广大科技人员和专业编程者喜爱的程序设计语言。但要学好这种语言，不仅要学习基本概念、方法、语法规则，更重要的还要进行实践。只有通过实践体验，积累编程经验，才能真正提高程序设计的能力。

为了适应 C 语言教学的需要，我们认为有必要编写配套的实验教程，以达到以下目的：

- 作为配套教程的辅助教材，可以加强学生对 C 语言概念、方法和内容的巩固；有了实验教程，可增强学生对课程实践重要性的认识深度和投入力度。
- 对编程的思路、方法和技巧起指导、示范作用。
- 可以和其他教材配套使用，作为已有教材的辅导参考书。

本书作为《C 语言程序设计教程》的配套教材，共分 14 章，一共设置了 35 个实验，主要内容包括：C 语言程序开发环境和上机步骤、基本数据类型、运算符和表达式、控制结构、函数、数组、指针、结构体和共用体、编译预处理、文件、高级数据结构。各章附有大量的练习题及其参考答案。

为了学生调试程序和应对等级考试的方便，本书附录中收录了 C 语言常见编译错误和全国计算机等级考试 C 语言二级考试大纲，另附有教学中使用的实验报告模板。

本书具有基础性、实用性、系统性，同时充分考虑了和其他教材内容的兼容性，适用于各类高等院校非计算机专业 C 语言程序设计课程的实验教学，也可供计算机等级考试和自学参考。

本书由汪同庆、关焕梅、汤洁主编和统稿。第 1、2 章由张华编写，第 3、4 章由汪同庆编写，第 5 章由黄磊编写，第 6、7 章由汤洁编写，第 8、10 章由滕冲编写，第 9、12 章由杨先娣编写，第 11、13 章和附录由关焕梅编写，第 14 章由黄文斌编写。在本书编写过程中，得到了有关领导和机械工业出版社华章分社的大力支持，在此表示衷心感谢。

受编者水平所限，书中难免存在疏漏之处，恳请广大读者提出宝贵意见。

编 者

2006 年 11 月

于武汉大学珞珈山

# 目 录

编委会  
序言  
前言

|                                                     |    |
|-----------------------------------------------------|----|
| 第1章 C语言概述 .....                                     | 1  |
| 1.1 C语言实验指导 .....                                   | 1  |
| 1.1.1 上机实验的目的 .....                                 | 1  |
| 1.1.2 上机实验前的准备工作 .....                              | 1  |
| 1.1.3 上机实验的步骤 .....                                 | 1  |
| 1.2 C语言集成开发环境 .....                                 | 2  |
| 1.2.1 认识 Turbo C 2.0 集成开发环境 .....                   | 2  |
| 1.2.2 认识 Visual C++ 6.0 集成开发环境 .....                | 7  |
| 1.3 基本练习与参考答案 .....                                 | 15 |
| 1.3.1 基本练习 .....                                    | 15 |
| 1.3.2 参考答案 .....                                    | 16 |
| 第2章 C语言快速入门 .....                                   | 17 |
| 2.1 实验 .....                                        | 17 |
| 2.1.1 实验1: 创建第一个C语言程序 .....                         | 17 |
| 2.1.2 实验2: 创建一个包含算术运算<br>的程序 .....                  | 19 |
| 2.2 基本练习与参考答案 .....                                 | 22 |
| 2.2.1 基本练习 .....                                    | 22 |
| 2.2.2 参考答案 .....                                    | 24 |
| 第3章 基本数据类型 .....                                    | 26 |
| 3.1 实验 .....                                        | 26 |
| 3.1.1 实验1: 基本数据类型数据的<br>使用 .....                    | 26 |
| 3.1.2 实验2: 格式输入输出函数 scanf()<br>和 printf() 的使用 ..... | 27 |
| 3.2 基本练习与参考答案 .....                                 | 29 |
| 3.2.1 基本练习 .....                                    | 29 |
| 3.2.2 参考答案 .....                                    | 30 |
| 第4章 运算符和表达式 .....                                   | 31 |
| 4.1 实验 .....                                        | 31 |
| 4.1.1 实验1: 自增、自减运算符的<br>基本使用 .....                  | 31 |
| 4.1.2 实验2: 运算符的优先级和<br>结合性 .....                    | 32 |
| 4.2 基本练习与参考答案 .....                                 | 33 |
| 4.2.1 基本练习 .....                                    | 33 |
| 4.2.2 参考答案 .....                                    | 34 |
| 第5章 结构化程序设计 .....                                   | 35 |
| 5.1 实验 .....                                        | 35 |

|                                     |    |
|-------------------------------------|----|
| 5.1.1 实验1: if 语句的使用 .....           | 35 |
| 5.1.2 实验2: switch 语句的使用 .....       | 36 |
| 5.1.3 实验3: 循环结构的使用 .....            | 37 |
| 5.1.4 实验4: 嵌套控制结构的使用 .....          | 39 |
| 5.2 基本练习与参考答案 .....                 | 40 |
| 5.2.1 基本练习 .....                    | 40 |
| 5.2.2 参考答案 .....                    | 47 |
| 第6章 函数 .....                        | 50 |
| 6.1 实验 .....                        | 50 |
| 6.1.1 实验1: 函数的定义及其使用 .....          | 50 |
| 6.1.2 实验2: 函数的嵌套调用 .....            | 51 |
| 6.1.3 实验3: 函数的递归调用 .....            | 52 |
| 6.1.4 实验4: 包含多个源文件的程序 .....         | 53 |
| 6.2 基本练习与参考答案 .....                 | 54 |
| 6.2.1 基本练习 .....                    | 54 |
| 6.2.2 参考答案 .....                    | 57 |
| 第7章 作用域和存储类别 .....                  | 59 |
| 7.1 实验 .....                        | 59 |
| 7.1.1 实验1: 变量作用域的应用 .....           | 59 |
| 7.1.2 实验2: 不同存储类别的变量的<br>使用 .....   | 60 |
| 7.1.3 实验3: 内部函数和外部函数 .....          | 60 |
| 7.2 基本练习与参考答案 .....                 | 61 |
| 7.2.1 基本练习 .....                    | 61 |
| 7.2.2 参考答案 .....                    | 65 |
| 第8章 数组 .....                        | 66 |
| 8.1 实验 .....                        | 66 |
| 8.1.1 实验1: 一维数组 .....               | 66 |
| 8.1.2 实验2: 二维数组及数组名作为<br>函数参数 ..... | 67 |
| 8.2 基本练习与参考答案 .....                 | 68 |
| 8.2.1 基本练习 .....                    | 68 |
| 8.2.2 参考答案 .....                    | 71 |
| 第9章 指针 .....                        | 73 |
| 9.1 实验 .....                        | 73 |
| 9.1.1 实验1: 指针变量 .....               | 73 |
| 9.1.2 实验2: 指针与数组 .....              | 74 |
| 9.1.3 实验3: 返回指针的函数和函数<br>指针 .....   | 76 |
| 9.2 基本练习与参考答案 .....                 | 79 |
| 9.2.1 基本练习 .....                    | 79 |
| 9.2.2 参考答案 .....                    | 85 |
| 第10章 字符串 .....                      | 87 |
| 10.1 实验 .....                       | 87 |

|                                       |     |                                    |     |
|---------------------------------------|-----|------------------------------------|-----|
| 10.1.1 实验1: 字符数组 .....                | 87  | 12.2.2 参考答案 .....                  | 116 |
| 10.1.2 实验2: 字符串指针变量和<br>字符串处理函数 ..... | 88  | 第13章 文件 .....                      | 117 |
| 10.2 基本练习与参考答案 .....                  | 89  | 13.1 实验 .....                      | 117 |
| 10.2.1 基本练习 .....                     | 89  | 13.1.1 实验1: 文件的顺序存取 .....          | 117 |
| 10.2.2 参考答案 .....                     | 93  | 13.1.2 实验2: 文件的随机存取 .....          | 118 |
| 第11章 结构体、共用体和枚举 .....                 | 96  | 13.2 基本练习与参考答案 .....               | 119 |
| 11.1 实验 .....                         | 96  | 13.2.1 基本练习 .....                  | 119 |
| 11.1.1 实验1: 结构体变量 .....               | 96  | 13.2.2 参考答案 .....                  | 120 |
| 11.1.2 实验2: 结构体数组 .....               | 97  | 第14章 高级数据结构 .....                  | 123 |
| 11.1.3 实验3: 结构体与函数 .....              | 98  | 14.1 实验 .....                      | 123 |
| 11.1.4 实验4: 共用体 .....                 | 100 | 14.1.1 实验1: 自引用的结构体变量的<br>使用 ..... | 123 |
| 11.1.5 实验5: 枚举 .....                  | 102 | 14.1.2 实验2: 实现线性链表的基本<br>操作 .....  | 124 |
| 11.2 基本练习与参考答案 .....                  | 103 | 14.2 基本练习与参考答案 .....               | 127 |
| 11.2.1 基本练习 .....                     | 103 | 14.2.1 基本练习 .....                  | 127 |
| 11.2.2 参考答案 .....                     | 106 | 14.2.2 参考答案 .....                  | 127 |
| 第12章 编译预处理 .....                      | 108 | 附录1 常见编译错误 .....                   | 129 |
| 12.1 实验 .....                         | 108 | 附录2 实验报告模板 .....                   | 140 |
| 12.1.1 实验1: 宏定义和文件包含 .....            | 108 | 附录3 全国计算机等级考试二级C语言<br>考试大纲 .....   | 141 |
| 12.1.2 实验2: 条件编译 .....                | 109 | 参考文献 .....                         | 144 |
| 12.2 基本练习与参考答案 .....                  | 112 |                                    |     |
| 12.2.1 基本练习 .....                     | 112 |                                    |     |

# 第1章 C语言概述

## 1.1 C语言实验指导

### 1.1.1 上机实验的目的

“程序设计”是一门实践性很强的课程，在学习过程中必须十分重视实验环节，有条件的情况下读者可以参观或参与实际项目工程的软件研发过程。作为初学者绝不能满足于知道了一些语法和看得懂书上的程序，而应该熟练掌握程序开发的全过程，即能够独立编写出程序的源代码，独立上机运行、测试和调试程序。因此，上机实验的目的不仅是为了验证新学到的内容或者简单地验证自己所编写的程序能否正确运行。

C语言上机实验的主要目的是：

1) 加深对新知识的理解。C语言中的语法规则较多，如果只是看书或在课堂上听老师讲解，很快就会感到枯燥无味，且容易忘掉，但它们正是写出正确程序的前提条件。通过不断地上机实验，就能自然地掌握C语言。

2) 了解和熟悉C语言编程环境的基本操作。编程环境包括所使用的计算机硬件系统、安装的操作系统以及C语言编译系统和程序开发工具。编程人员应该了解，为了运行一个C语言程序需要哪些必要的外部条件（例如硬件配置和软件配置），可以利用哪些系统功能来帮助自己开发程序。对于这些，各种编程环境都不尽相同，读者只需要熟练掌握一两种编程环境的基本操作，再遇到其他系统时便会触类旁通，很快就能学会。

3) 学习上机测试、调试程序的基本方法和技巧。也就是要善于发现程序中的错误，并且能够很快地排除这些错误，使程序能正确运行。测试和调试程序的能力是每个程序设计人员都应当掌握的一项基本功。经验丰富的编程人员在编译连接出现错误提示时，一般能够很快地判断出错误原因和位置，并改正它。初学者应该通过自己的大量编程实践来不断积累经验。

因此，在做实验时不要在程序通过后就认为任务完成了，而应当多设计几组输入数据对程序进行测试，在程序中增加输出语句跟踪程序的执行过程，考虑给程序增加更多的功能，对算法进行优化等。这样不仅增加了学习的主动性，也使学习成为一件快乐的事情。

### 1.1.2 上机实验前的准备工作

在上机实验前事先做好准备工作可以提高上机实验的效率。准备工作包括：

1) 了解所用的编程环境的一般操作方法和实验室的规定。

2) 复习与本次实验相关的教学内容。

3) 准备好上机时用到的程序。可以在纸上写出程序的源代码，应该书写整齐，并经检查无明显的语法和逻辑错误。切记不要在问题还没完全弄清楚，对程序的目标模糊不清时就直接上机编写代码。

4) 对运行中可能会出现的问题事先作出估计，对程序中有疑问的地方作出标记，以便在上机时给予注意。

5) 准备好测试程序所用的数据。

### 1.1.3 上机实验的步骤

上机实验一般应包括以下几个步骤：

1) 创建自己的工作文件夹，用来存放本次实验的全部程序文件。

2) 启动集成开发环境，例如 Turbo C 2.0 或 Visual C++ 6.0。

3) 输入程序的代码，并检查之。

4) 编译和连接。根据错误提示找到错误位置和原因,并改正之,再编译和连接,直至程序可以正确运行。

5) 输入不同的测试数据,运行程序并分析结果。修改程序,再测试,直至程序运行结果正确。

6) 完成实验报告。

需要强调的是,在上机过程中出现错误,一般应自己独立处理,不要轻易询问指导老师。特别是面对编译和连接的错误信息时,应该自己分析判断,这是学习调试程序的良好习惯。

## 1.2 C 语言集成开发环境

现在就让我们先来了解即将使用的 C 语言开发环境的基本操作。因为 Turbo C 2.0 是全国计算机等级考试所采用的上机环境,所以我们将对该开发环境的基本操作进行简要介绍。而 Visual C++ 6.0 是我们推荐使用的开发环境,因此将对其作详细介绍,并且会在以后的实验中逐渐深入。

### 1.2.1 认识 Turbo C 2.0 集成开发环境

Turbo C (简称 TC) 是 Borland 公司开发的基于 MS-DOS 操作系统的 C 语言编译系统,它是一个集程序编辑、编译、连接和调试为一体的 C 语言程序开发环境。TC 可以运行在 Windows XP 的“命令提示符”窗口中。TC 2.0 是被广泛采用的版本。

#### 1. 启动 Turbo C 2.0

确认实验机房的机器上已安装好 Turbo C 2.0, 如果安装目录为: D:\TC2, 那么在 Windows XP 上可以采用以下几种方法启动 TC:

- 单击“开始”按钮打开“开始”菜单,选择“程序”→“附件”→“命令提示符”,打开“命令提示符”窗口,然后输入以下命令启动 TC:

D: \TC2 \TC

- 在资源管理器中打开 D:\TC2 文件夹,找到该文件夹中的 tc.exe 文件,双击该文件名,即可启动 TC。
- 在桌面上创建指向 D:\TC2\tc.exe 文件的快捷方式,双击该快捷方式图标即可启动 TC。

#### 2. Turbo C 2.0 的窗口

启动 TC 后,首先看到的是如图 1-1 所示的欢迎界面,窗口中央是 TC 的版本信息框。

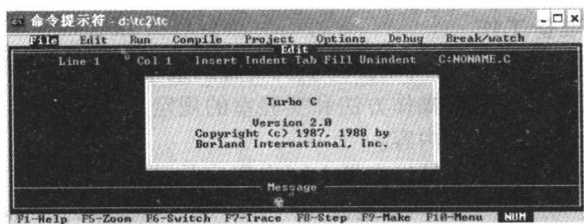


图 1-1 TC 的欢迎界面

因为 TC 不支持鼠标操作,所以必须按下键盘上的任意键,版本信息框才会消失,然后将出现 TC 完整的工作界面,如图 1-2 所示。

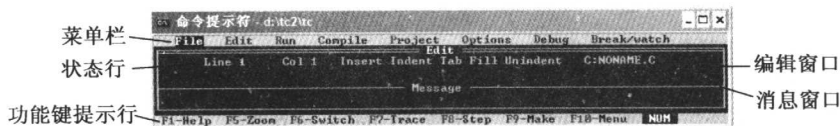


图 1-2 TC 的工作界面

TC 的窗口包括以下部分：

- 菜单栏。包括 8 个菜单：File（文件）、Edit（编辑）、Run（运行）、Compile（编译）、Project（项目）、Options（选项）、Debug（调试）和 Break/watch（断点/监视）。
- 编辑窗口。用来输入和编辑程序的源代码。
- 状态行。Line 和 Col 后面的数字表示编辑窗口中光标的位置，即光标所在的行号和列号。最右端显示的是正在编辑的文件的名称，及其所在的磁盘符号。中间部分显示的是当前的编辑状态和设置。
- 消息窗口。用来显示编译和连接时的提示信息。
- 功能键提示行。显示可用的功能键及其作用。

### 3. 菜单操作

可以用两种方法打开菜单：

- 按 <Alt + 菜单首字母> 组合键。例如要打开“File”菜单就按 <Alt + F> 组合键。
- 按 <F10> 键之后用方向键选择某一个菜单，按 <Enter> 键打开该菜单。

按 <Esc> 键关闭打开的菜单。

在打开的菜单中通过方向键选择菜单项，按 <Enter> 键执行该菜单项对应的功能。

### 4. 设置工作目录

工作目录是指用户存放文件的目录，例如 E:\work。具体操作是：打开 File 菜单，选择“Change dir（改变目录）”，按 <Enter> 键，打开“New Directory（新目录）”对话框，如图 1-3 所示。在该对话框中输入 E:\work，按 <Enter> 键确定。

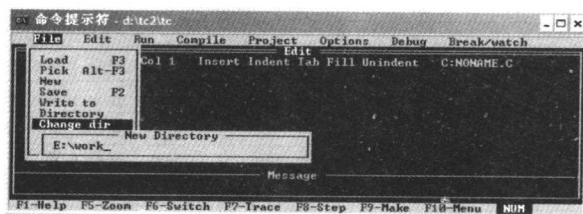


图 1-3 设置工作目录

### 5. 设置系统目录

系统目录是指为了让 TC 正确的工作必须设置的目录。打开 Option 菜单，选择“Directories（目录）”菜单项，按 <Enter> 键打开系统目录设置对话框，如图 1-4 所示。

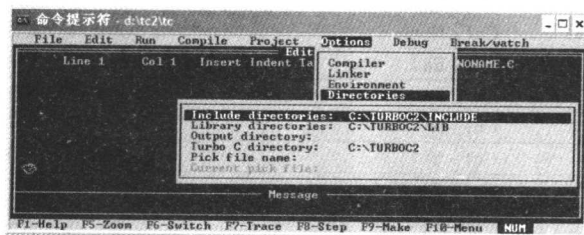


图 1-4 系统目录设置对话框

选中一项后按 <Enter> 键进行修改即可。

系统目录设置对话框中各项功能如下：

- Include directories（包含文件目录）：标准库对应的头文件所在的目录。
- Library directories（库目录）：标准库所在的目录。
- Output directory（输出目录）：存放编译和连接生成的目标文件和可执行文件。

- Turbo C directory (TC 的目录): TC 的安装目录。

## 6. 创建新文件

TC 启动之后会自动新建一个文件,当然我们也可以通过 File 菜单(见图 1-5)的“New (新建)”菜单项新建一个文件。由于 TC 一次只能显示一个文件,所以新建文件时将关闭已打开的文件。

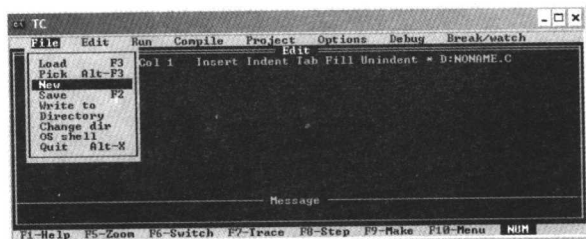


图 1-5 创建新文件

## 7. 编辑文件

通过按 <Alt> + <E> 组合键,选择 Edit 菜单,即可进入 TC 的编辑状态。在编辑状态下可以输入和编辑程序的源代码,这与在 Windows 的记事本中输入源代码类似,但操作起来不是很方便。表 1-1 列出 TC 的常用编辑键。

表 1-1 TC 的常用编辑键

| 分 类   | 按键命令         | 功 能           |
|-------|--------------|---------------|
| 移动光标  | ↑            | 光标上移一行        |
|       | ↓            | 光标下移一行        |
|       | ←            | 光标左移一个字符      |
|       | →            | 光标右移一个字符      |
|       | Ctrl + A     | 光标左移一个单词      |
|       | Ctrl + F     | 光标右移一个单词      |
|       | PgUp         | 上翻一页          |
|       | PgDn         | 下翻一页          |
|       | Home         | 光标移至行首        |
|       | End          | 光标移至行尾        |
|       | Ctrl + PgUp  | 光标移至页首        |
|       | Ctrl + PgDn  | 光标移至页尾        |
|       | Ctrl + Home  | 光标移至文件首       |
|       | Ctrl + End   | 光标移至文件尾       |
| 编辑状态  | Ins          | 在插入状态和改写状态间切换 |
| 插入与删除 | Ctrl + N     | 在当前光标位置插入新行   |
|       | Ctrl + Y     | 删除光标所在的行      |
|       | Ctrl + Q + Y | 删除光标位置至行尾     |
|       | Backspace    | 删除光标左边的一个字符   |
|       | Del          | 删除光标处的一个字符    |
|       | Ctrl + T     | 删除光标右边的一个单词   |
| 块操作   | Ctrl + K + B | 将光标处标记为块首     |
|       | Ctrl + K + K | 将光标处标记为块尾     |
|       | Ctrl + K + C | 将标记的块复制到当前光标处 |
|       | Ctrl + K + V | 将标记的块移动到当前光标处 |
|       | Ctrl + K + Y | 删除标记的块        |
|       | Ctrl + K + H | 隐藏或显示块标记      |

## 8. 保存文件

TC 为所有新建的文件都自动命名为 NONAME.C, 因此, 在第一次保存文件时应将其改名。可以选择 File 菜单的“Save (保存)”菜单项或按 <F2> 功能键来保存文件, 第一次保存时会打开重命名对话框, 输入文件保存的路径和文件名, 按 <Enter> 键保存。如图 1-6 所示, 把新建的文件保存到 E:\work 中, 文件名为 first.c。

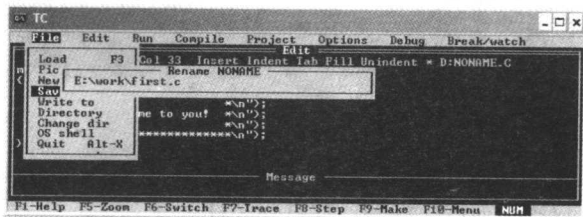


图 1-6 保存文件

如果想把当前文件重命名, 或者保存到另一个位置, 那么可以选择 File 菜单的“Write to (另存为)”菜单项。执行该命令时将打开一个对话框, 如图 1-7 所示, 输入文件保存的路径和文件名即可。

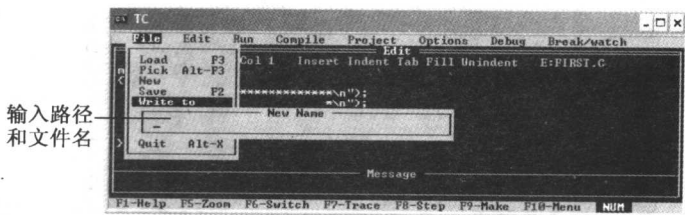


图 1-7 文件另存为

## 9. 打开文件

如果想打开一个已存在的源文件, 可以选择 File 菜单的“Load (装载)”菜单项或直接按 <F3> 功能键。TC 将弹出一个对话框, 如图 1-8 所示, 此时可以输入要打开的文件名。如果不输入路径, 则 TC 在当前目录下查找该文件并装载。如果输入了路径, 则 TC 在指定的位置查找文件并装载。

在如图 1-8 所示对话框时, 可以不输入文件名, 而直接按 <Enter> 键, 则 TC 会打开文件列表窗口, 如图 1-9 所示。在该窗口中找到要打开的文件, 按 <Enter> 键即可。

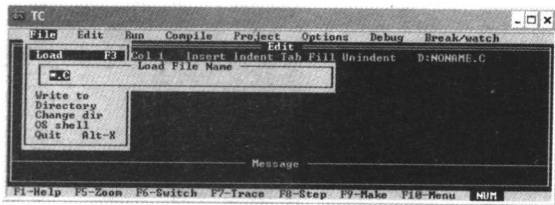


图 1-8 打开文件

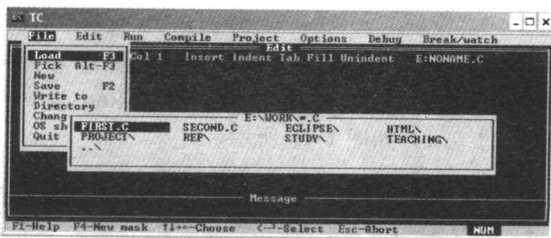


图 1-9 文件列表窗口

## 10. 编译程序

要编译当前打开的源文件可按 <Alt> + <C> 组合键打开 Compile 菜单, 如图 1-10 所示。

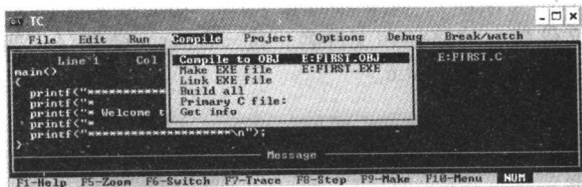


图 1-10 Compile 菜单

Compile 菜单的各菜单项功能如下:

- Compile to OBJ: 仅将当前源文件编译成目标文件。
- Make EXE file: 顺序完成前两项的功能。
- Link EXE file: 将当前的目标文件与所需的库文件连接在一起生成可执行文件。

在编译和连接后 TC 都会弹出一个信息框报告编译和连接的结果, 如图 1-11 和图 1-12 所示。在键盘上按下任意键将关闭该信息框。

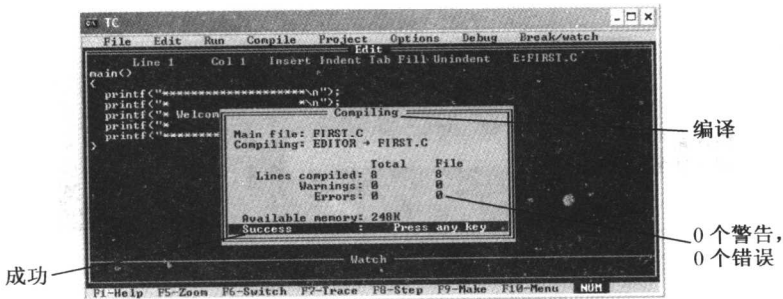


图 1-11 编译结果信息框

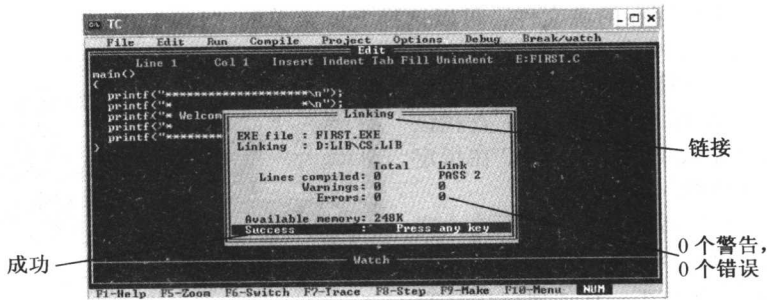


图 1-12 连接结果信息框

## 11. 运行程序

选择 Run 菜单的“Run (运行)”菜单项或直接按 <Ctrl> + <F9> 组合键则自动执行“Make EXE file”, 生成可执行文件, 然后执行该文件。选择 Run 菜单的“User screen (用户屏幕)”菜单项, 可以查看执行的结果, 如图 1-13 所示。

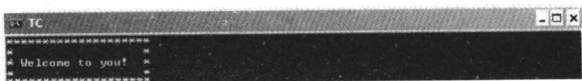


图 1-13 查看运行结果

按任意键，将关闭该窗口并返回到 TC 的主界面。

## 12. 调试程序

编写程序难免会有错误，调试程序就是要发现可能存在的错误，并加以改正。编译器能检查出程序中的语法错误，所以一旦编译时发现错误，TC 将不会生成目标文件，并会在编译信息框中报告错误。如果把 first.c 中的倒数第 2 行末尾的分号删除，然后编译该文件，则得到如图 1-14 所示的信息框。

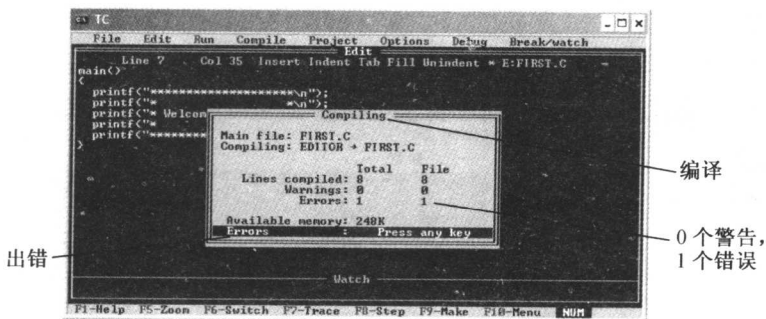


图 1-14 编译出错信息框

编译器给出的错误提示有两类：一类称为“Error（错误）”，另一类称为“Warning（警告）”。编译器发现错误（Error）就不会生成目标文件，所以把这一类错误称为致命错误，必须找到并改正致命错误。而警告则属于轻微错误，如果程序代码中只是出现警告错误，编译器仍然可以生成目标文件，也不会影响连接，但在运行时可能会出错。因此，严格来讲应该修改程序代码直至既无致命错误，也无警告。

按下任意键后，消息窗口中将显示错误信息，如图 1-15 所示。用方向键在多条错误信息间切换时，编辑窗口中将高亮显示与当前错误信息对应的发现错误的代码行。按下 <Alt> + <E> 组合键进入编辑状态，修改错误然后再编译，如果还有错误，则再修改，再编译，……直至无编译错误。

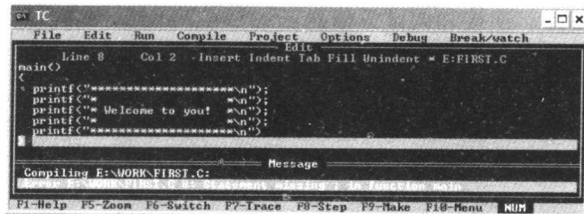


图 1-15 编译错误信息

有了编译器的错误提示信息，找到编译错误就会相对容易。但是，对于逻辑错误和运行时错误编译器就无能为力了，大多数情况下需要通过跟踪程序的执行过程，观察和分析程序执行的中间结果才能找到错误位置。TC 支持以下的调试方法：

- 让程序执行到特定的位置时暂停以便观察阶段性结果。
- 在监视窗口中添加变量，以便观察程序执行过程中这些变量的值的变化。
- 单步执行或者跟踪执行某些可能有问题的语句。

对于这些调试方法的具体操作请读者查阅相关资料，这里不再详述。

### 1.2.2 认识 Visual C++ 6.0 集成开发环境

Visual C++（简称 VC）是微软公司开发的被广泛使用的基于 Windows 平台的 C 和 C++ 语言

编程环境，可以在其中编辑、编译、连接、运行和调试 C 语言程序。现在常用的版本是 VC 6.0。VC 运行在 Windows 操作系统上，对于熟悉图形用户界面的用户而言较易学习。本书基于 Windows XP 环境介绍 VC 6.0 的基本操作。

### 1. 启动 VC++ 6.0

确认实验机房的计算机上已安装 VC 6.0，单击 Windows XP 的“开始”按钮，选择“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”，就可以启动 VC 6.0。也可以在桌面上创建 VC 6.0 的快捷方式，双击该快捷方式的图标，即可启动 VC 6.0。

### 2. VC 的窗口

启动后将看到 VC 的主窗口，如图 1-16 所示。

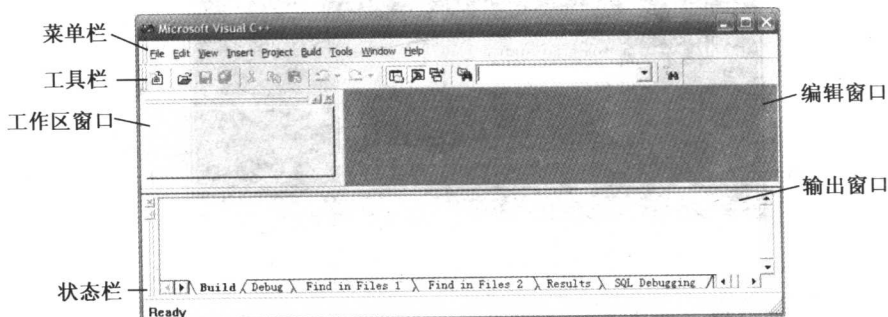


图 1-16 Visual C++ 6.0 的主窗口

主窗口主要包括以下部分：

- 菜单栏。包含 9 个菜单项：File（文件）、Edit（编辑）、View（查看）、Insert（插入）、Project（项目）、Build（构建）、Tools（工具）、Window（窗口）和 Help（帮助）。
- 工具栏。VC++ 提供了几个工具栏。可以在菜单栏或工具栏的空白区单击鼠标右键，在弹出的菜单中选择要显示或关闭工具栏的名称。现在显示出来的是标准工具栏。
- 工作区窗口。用来显示所设定的工作区的信息。现在没有打开的工作区。
- 编辑窗口。用来输入和编辑源程序，每个源文件将显示在一个独立的编辑窗口中。
- 输出窗口。显示编译、连接和调试等信息。
- 状态栏。显示操作提示信息和编辑状态。

### 3. 新建文件

简单的 C 语言程序只包含一个源文件。选择 File 菜单的 New 菜单项，打开 New 对话框，如图 1-17 所示。

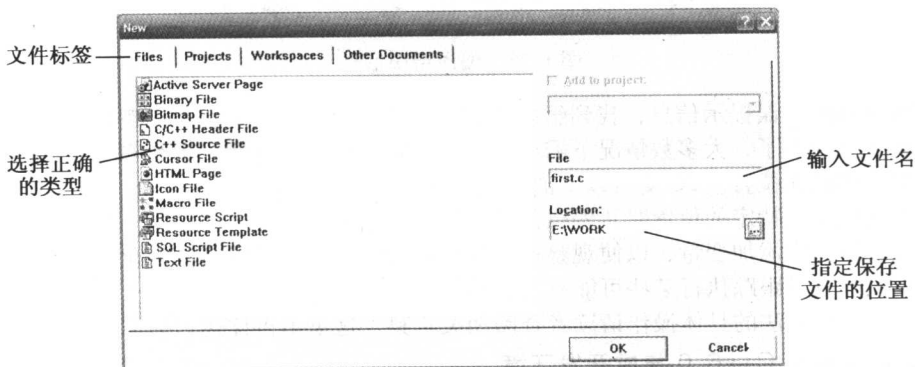


图 1-17 新建文件对话框

单击 Files 标签, 在左边列表框中选择 C++ Source File。然后, 在右边的 File 文本框中输入文件名, 例如 first.c, 其中.c 是 C 语言源文件的扩展名。

接下来, 在“Location (位置)”文本框中输入源文件的存放路径, 要确保该路径存在, 否则会导致创建失败。也可以点击该文本框右边的“...”按钮, 在打开的“Choose Directory (选择目录)”对话框(图 1-18)中直接选择存放位置。

单击 New 对话框中的 OK 按钮, 可在指定的位置创建一个源文件, 并打开一个编辑窗口, 如图 1-19 所示。

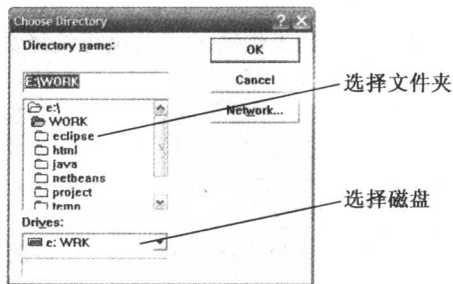


图 1-18 选择目录对话框

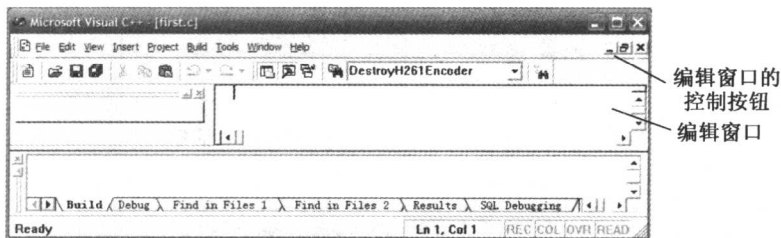


图 1-19 新建文件的编辑窗口

#### 4. 编辑文件

在文件编辑窗口中输入程序代码, 如图 1-20 所示。由于完全是 Windows 界面, 所以在 VC 中编辑文件非常方便。

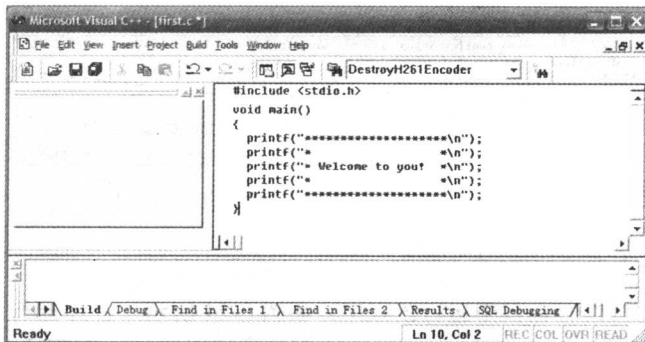


图 1-20 编辑文件

#### 5. 保存文件

在编辑文件后选择 File 菜单的 Save 菜单项保存所作的修改。

#### 6. 关闭文件

直接关闭源文件的编辑窗口, 即可关闭该文件, 也可以选择 File 菜单中的 Close 菜单项来关闭当前文件。

#### 7. 打开文件

如果要打开一个已存在的源文件, 可以选择 File 菜单的 Open 菜单项, 在弹出的 Windows 的“打开”对话框中选择要打开的文件。

#### 8. 编译程序

打开 Build 菜单, 如图 1-21 所示。可以选择 Compile 菜单项, 仅对 first.c 文件进行编译。也