

张黎宁 沈丽容 主编

窦立君 蒋安纳 韦素云 章春芳 编

高等教育出版社



教育部大学计算机课程改革项目规划教材

C 程序设计案例教程

C Chengxu Sheji Anli Jiaocheng

张黎宁 沈丽容 主编
 窦立君 蒋安纳 韦素云 章春芳 编

提供 (C) 目录页主标题

、解费所集:出编第第 3

2102, 11 月 10 日 10 时 10 分 10 分

11:40-12:00-12:00-12:00

第 1 卷 第 1 期 第 1 期

11 月 10 日 10 时 10 分 10 分

提供 (C) 目录页主标题

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

11 月 10 日 10 时 10 分 10 分

高等教育出版社·北京



内容提要

本书主要包括3部分内容：一是C语言的简介以及如何学习本课程；二是C程序设计基础部分，即三大基本结构程序的设计（顺序结构、选择结构、循环结构）；三是C的构造数据类型（数组、结构体）、指针类型、函数以及文件的使用。

本书的设计以实际案例问题为主线，随着案例问题的分析逐步融入程序设计的思想，引入编写程序所采用的语法知识点，内容丰富，案例有趣而实用。在案例问题解决的驱动下，内容从简到难，循序渐进，引导读者学习使用计算机分析问题和求解问题的过程、思维模式和基本方法，培养读者的逻辑思维能力和计算思维能力。

本书适合初学者使用，既可以作为高等学校各专业C程序设计课程的教科书，也可作为想学习和使用C语言进行开发的读者的自学参考教材。

图书在版编目(CIP)数据

C程序设计案例教程 / 张黎宁, 沈丽容主编. — 北京: 高等教育出版社, 2015.3
ISBN 978-7-04-042113-2

I. ①C… II. ①张… ②沈… III. ①C语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第026300号

策划编辑 唐德凯

责任编辑 唐德凯

封面设计 于文燕

责任印制 田甜

出版发行 高等教育出版社

社址 北京市西城区德外大街4号

邮政编码 100120

印刷 秦皇岛市昌黎文苑印刷有限公司

开本 787mm×1092mm 1/16

印张 25.75

字数 650千字

购书热线 010-58581118

咨询电话 400-810-0598

网址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

网上订购 <http://www.landaco.com>

<http://www.landaco.com.cn>

版次 2015年3月第1版

印次 2015年3月第1次印刷

定价 34.00元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换

版权所有 侵权必究

物料号 42113-00

前言

目前,高等学校计算机基础教育的核心已经由原来的计算机应用能力的培养转变为计算思维能力的培养。“C 程序设计”作为计算机专业的专业基础课和非计算机专业的公共课,应该更有效地把计算思维所涉及的运用计算机科学的概念去求解问题、设计系统和理解人类的行为等方面贯穿在教学过程中。

本书的编写以高等教育对人才培养的新要求为指导思想,以“培养学生计算思维能力,训练学生编写程序解决实际问题”为目标。教材的设计思路是“用程序学习程序设计”,教材在内容的安排上符合学习者的学习认知过程,从简到难,循序渐进。本书以有趣实用的程序案例为主线,在程序的情景中学习程序设计。通过案例程序引入知识点,在案例程序的分析中逐步融入程序设计的思想和语法,在案例程序的分析中融入计算机分析问题和求解问题的过程、思维模式和基本方法,从而提高 C 程序设计课程教学与学习的效率,同时培养学生的计算思维能力。

本书的标准授课学时为 80 学时,其中理论课为 64 学时,上机实验课为 16 学时。建议学习者在课后增加自主上机的时间,从而达到熟练编程的效果。

本书的作者长期工作在教学第一线,有着丰富的授课经验。具体编写分工如下:沈丽容编写了第 1 章和第 2 章;窦立君编写了第 3 章和第 4 章;蒋安纳编写了第 5 章、第 9 章和贯穿全书的学生成绩管理系统案例;韦素云编写了第 6 章;章春芳编写了第 7 章;张黎宁编写了第 8 章并进行了全文的审阅工作。

本书的编写工作得到了南京林业大学计算机系各位老师的支持,在此表示由衷的感谢。

由于作者水平有限,书中不妥之处在所难免,欢迎各位读者对本书存在的问题提出宝贵意见。作者联系方式:nj.zln@163.com。

作者
2014.12

目 录

第1章 导言	(1)
1.1 为什么要学习程序设计	(1)
1.2 为什么学习 C 语言	(1)
1.3 C 语言的发展历史	(2)
1.4 C 语言的特点	(2)
1.5 C 语言源程序结构	(3)
1.6 如何学习 C 语言	(4)
1.7 Turbo C 2.0 简介	(4)
1.8 Visual C++ 6.0 简介	(10)
习题 1	(15)
第2章 顺序结构	(16)
2.1 字符型数据	(16)
2.1.1 用“*”输出字母“F”的图案	(16)
2.1.2 输出特殊图案	(18)
2.1.3 大小写字母转换	(23)
2.2 整型数据	(26)
2.2.1 输出三位整数的逆序数	(26)
2.2.2 数据加密	(30)
2.2.3 交换两个整数	(34)
2.2.4 生成验证码	(35)
2.3 实型数据	(37)
2.3.1 计算三角形面积	(37)
2.3.2 计算球的表面积和体积	(39)
2.3.3 计算标准体重	(41)
2.4 位运算	(42)
2.4.1 取出一个整数的第 k 位	(42)
2.4.2 将一个整数的第 k 位清 0	(44)
2.4.3 将一个整数的第 k 位置 1	(46)
2.4.4 将一个整数的第 k 位翻转	(47)
习题 2	(49)
第3章 选择结构	(53)
3.1 单分支选择结构	(53)
3.1.1 求两个整数的最大值	(53)
3.1.2 计算三角形面积	(56)

3.2 双分支选择结构	(59)
3.2.1 判断指定年份是否为闰年	(59)
3.2.2 小写字母转大写字母	(60)
3.3 多分支选择结构	(62)
3.3.1 百分制成绩转换为五级制等级	(62)
3.3.2 空气质量指数(AQI)计算器	(65)
3.3.3 求分段函数的值	(67)
3.3.4 是否退休问题	(69)
3.4 switch 语句	(71)
3.4.1 输入整型星期数,输出英语星期名称	(71)
3.4.2 百分制成绩转换为五级制等级	(73)
3.4.3 判断指定的某年某月有多少天	(76)
3.4.4 简单四则运算计算器	(77)
3.4.5 简易菜单制作	(79)
习题3	(83)
第4章 循环结构	(87)
4.1 while 语句	(87)
4.1.1 累加求和问题	(87)
4.1.2 累乘求积问题	(90)
4.1.3 正整数拆分问题	(91)
4.1.4 猴子吃桃问题	(93)
4.2 do...while 语句	(94)
4.2.1 猜数字游戏	(94)
4.2.2 控制光标	(97)
4.2.3 弦截法求方程的根	(99)
4.2.4 计算 π 值问题	(101)
4.3 for 语句	(102)
4.3.1 水仙花数	(102)
4.3.2 素数	(106)
4.3.3 最大公约数和最小公倍数	(108)
4.3.4 高斯日记	(109)
4.4 循环嵌套	(112)
4.4.1 九九乘法表	(112)
4.4.2 杨辉三角形	(113)
4.4.3 年历问题	(114)
4.4.4 五子棋	(118)
习题4	(124)
第5章 数组	(128)
5.1 一维数组	(128)

5.1.1	算错的得分	(128)
5.1.2	兔子产子	(130)
5.1.3	跳水比赛	(133)
5.1.4	逆序存放数字	(134)
5.1.5	十进制/二进制转化器	(136)
5.1.6	排队接水问题	(137)
5.1.7	杰克船长选大副	(144)
5.2	二维数组	(146)
5.2.1	日期转换	(146)
5.2.2	跳水比赛	(148)
5.2.3	矩阵转置	(151)
5.2.4	矩阵对角线之和	(154)
5.2.5	矩阵相乘	(156)
5.2.6	寻找矩阵鞍点	(158)
5.2.7	求解九宫格	(160)
5.3	一维字符数组	(162)
5.3.1	明文加密	(163)
5.3.2	超长的城市名	(167)
5.3.3	确认密码	(169)
5.3.4	分离学号和姓名	(171)
5.3.5	变更文件后缀名	(172)
5.3.6	回文判断	(173)
5.3.7	十六进制/十进制转换器	(175)
5.3.8	删除指定字符	(177)
5.3.9	统计段落的字母个数	(179)
5.3.10	猜首都游戏	(181)
5.4	学生成绩管理系统——数组	(183)
习题 5		(186)
第 6 章	函数	(191)
6.1	自定义函数	(191)
6.1.1	数字魔法秀	(191)
6.1.2	皇冠上的明珠	(196)
6.1.3	分数计算器	(200)
6.2	参数的传递	(204)
6.2.1	大家来找茬	(204)
6.2.2	寻觅好声音	(210)
6.3	嵌套函数与递归函数	(219)
6.3.1	七月份的尾巴	(220)
6.3.2	分解质因数	(224)

6.4 变量的作用域和存储类型	(228)
6.4.1 ATM 自动取款机	(229)
6.4.2 珠心算比赛	(234)
6.5 标准函数	(242)
6.5.1 编程小秘书	(242)
6.5.2 笑脸,心动的开始	(246)
6.6 学生成绩管理系统——函数	(252)
习题 6	(257)
第 7 章 指针	(262)
7.1 指向变量的指针	(262)
7.1.1 求和问题	(262)
7.1.2 交换问题	(266)
7.1.3 按序存放整数	(268)
7.2 指针与一维数组	(270)
7.2.1 查找元素	(270)
7.2.2 调整队形	(273)
7.2.3 插队问题	(276)
7.2.4 移动积木	(279)
7.3 指针与二维数组	(281)
7.3.1 求矩阵中的最大值	(282)
7.3.2 对调各行元素	(284)
7.3.3 构造有规律的二维数组	(288)
7.4 指针与字符串	(290)
7.4.1 寻找特殊字符	(290)
7.4.2 替换字符	(292)
7.4.3 插入特定字符	(294)
7.4.4 统计子串出现的次数	(298)
7.5 指针数组与二级指针	(301)
7.5.1 根据数字查找星期几	(301)
7.5.2 寻找最大字符串	(303)
7.5.3 main 函数的参数	(306)
7.6 学生成绩管理系统——指针	(307)
习题 7	(314)
第 8 章 结构体、共用体和枚举	(322)
8.1 结构体类型	(322)
8.1.1 计算个人所得税	(322)
8.1.2 统计候选人得票	(327)
8.1.3 杰克船长选大副	(331)
8.2 共用体类型和枚举类型	(336)

8.2.1 不确定类型数据的表示	(336)
8.2.2 取球游戏	(339)
8.3 学生成绩管理系统——结构体数组	(342)
习题 8	(352)
第 9 章 文件	(359)
9.1 文件简介	(359)
9.1.1 重新认识输入输出	(359)
9.1.2 最简单的加法计算	(360)
9.2 文件读写操作	(362)
9.2.1 有损压缩文件	(362)
9.2.2 复读机	(366)
9.2.3 按字符复制文件	(367)
9.2.4 粗心的水果店老板	(368)
9.2.5 按块复制文件	(371)
9.3 文件定位操作	(373)
9.3.1 分割文件	(373)
9.3.2 修改文件内容	(377)
9.3.3 合并文件	(382)
习题 9	(384)
附录	(389)
附录 1 ASCII 码表	(389)
附录 2 运算符的优先级别和结合方向	(391)
附录 3 常用库函数	(392)
附录 4 键盘上常用键的扫描码	(399)
参考文献	(400)

第1章 导言

1.1 为什么要学习程序设计

为什么要学习程序设计？这几乎是每个非计算机专业的学生普遍关心的一个问题。有的学生认为他的专业用不着计算机编程，为什么还要学习程序设计呢？其实，学习计算机编程并不只是为了掌握某种程序设计语言，而是为了学习一种新的思维方式，即计算思维（Computational Thinking）。

2006年3月，美国卡内基·梅隆大学计算机科学系主任周以真（Jeannette M. Wing）教授在美国计算机权威期刊 *Communications of the ACM* 上给出了计算思维的定义：计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。美国中小学进行计算机编程教学的实践证明，学习计算机编程有助于开发学习潜力，提高逻辑推理能力和解决问题的能力。

编程可以使灵感和想法瞬间实现，看到立竿见影的效果，这种即时性的反馈会使学习兴趣变得越来越浓厚。编程的乐趣是其他学科无法比拟的，只要亲身体会过程序运行成功带来的那种喜悦，那种成就感，就会自然而然地迷恋上编程，会觉得自己也有能力去指挥计算机做事了，那种兴奋感将大大超越玩别人预先设定好的游戏软件。12岁的美国孩子托马斯说：“现在的孩子不再只是爱玩游戏，他们还想自己制作好玩的游戏。”

为世界创造了一个又一个奇迹的比尔·盖茨和乔布斯等IT巨人们都具有一个共同的特点，那就是在他们少年时就已经疯狂地迷恋上编程了。虽然他们从小就开始接触计算机，但是他们不但没有沉迷于游戏，而且还用自己的编程能力改变了整个世界。

21世纪是一个信息化的社会，这个社会对人才素质和知识结构都提出了更高的要求。对于高等教育的各个学科，计算机的作用已不仅仅是一种工具，而是各个学科本身的重要组成部分。加强计算机基础教育不仅是为了提高计算机应用水平，而且也是为了促进其他学科教育水平的提高。

1.2 为什么学习 C 语言

为什么选择 C 语言学习编程，而不是 C++、C#、Java 或 VB？这是因为 C 语言比较简洁和单纯，没有很多难以理解的概念，而且 C 语言的应用范围非常广泛，不仅仅用在软件开发上，而且在机械、电子、通信、自动控制等领域都有非常广泛的应用，如单片机以及嵌入式系统开发等。

C 语言具有强大的功能，它既具有高级语言的特点，又具有汇编语言的特点。它既可以作为系统设计语言，编写系统程序，也可以作为应用程序设计语言，编写不依赖于计算机硬件的应用程序。这也是为什么 C 语言自诞生以来历经 40 多年，至今仍然是最受欢迎的编程语言之一。

其实在学习编程时真正关注的并不是使用哪种语言的问题,而是要关注程序背后隐藏的编程思想,领会计算机工作的原理,至于编程语言,它只是实现人们想法的一个工具而已。因为编程语言之间有很多相通之处,一旦领悟了编程的精髓,就不难做到触类旁通了。

1.3 C 语言的发展历史

C 语言是 1972 年由美国的 Dennis M. Ritchie(丹尼斯·里奇,C 语言之父,UNIX 之父,见图 1-1)设计发明的,之所以命名为 C,是因为 C 语言源自 Ken Thompson 发明的 B 语言。1978 年美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时 Brian W. Kernighan 和 Dennis M. Ritchie 合著了著名的“The C Programming Language”(《C 程序设计语言》)一书。现在此书已翻译成多种语言,成为 C 语言方面最权威的教材之一。



图 1-1 C 语言之父 Dennis M. Ritchie

早期的 C 语言主要是用于 UNIX 操作系统。由于 C 语言的强大功能及其各方面的优点逐渐为人们所认识,到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的应用,成为当代最优秀的程序设计语言之一。与此同时,也衍生了 C 语言的很多不同版本。1983 年美国国家标准局(American National Standards Institute, ANSI)成立了一个委员会来制定 C 语言标准。1989 年 C 语言标准被批准,称为 ANSI X3.159-1989 Programming Language C。这个版本的 C 语言标准通常被称为 ANSI C。

1.4 C 语言的特点

1. 语言简洁紧凑、灵活方便

C 语言一共只有 32 个关键字,9 种控制语句,只需用规范的方法就可以构造出功能很强的数据类型、语句和程序结构。它程序简洁,表示方法简单,尽可能地压缩了一切不必要的成分,如用 $i++$ 或 $i+=1$ 表示 $i=i+1$, $--i$ 表示 $i=i-1$,等等。

2. 数据结构丰富

C 语言包含了多种数据结构,如整型、实型、字符型、数组、指针类型、结构体类型和共用体类型等,可以用于实现各种复杂的数据类型运算。

3. 运算符丰富

C 语言的运算符有 34 种,涉及的范围很广,灵活使用各种运算符可以实现难度极大的运算。

4. 语法检查不太严格

一般的高级语言对语法检查非常严格,几乎可以检查出所有的语法错误,但是 C 语言语法检查不太严格,因此程序书写的形式可以比较自由,但是正因为编译器不能检查出所有的错误,所以编程人员要自己认真检查程序,以保证程序正确运行。

5. 结构化程序设计

C 语言是一种结构化语言,符合现代编程风格的要求。它用函数作为程序设计的基本单位,以实现程序的模块化。同时,还提供了编写结构化程序所需的控制流语句,如 `if...else`、`for`、`while`、`do...while` 等。

6. 具有低级语言功能

C 语言允许直接访问物理地址,能够对二进制位、字节和地址进行操作,可以直接对硬件进行操作,具有低级语言的许多功能,因此既可以用来开发应用软件,又可以用来编写系统软件和单片机程序。

7. 生成的目标代码质量高

C 语言编写的程序比汇编语言编写的程序可读性好,易于调试和修改,生成的目标代码质量与汇编语言相当,程序执行效率只比汇编语言低 10%~20%。

8. 可移植性好

与汇编语言相比,用 C 语言开发单片机系统软件具有很多优点,如代码效率高、软件调试直观、维护升级方便、代码的重复利用率高、便于跨平台的代码移植等。

1.5 C 语言源程序结构

C 语言源程序的结构如下。

(1) 一个 C 语言源程序可以由一个或多个源文件组成。

(2) 每个源文件可由一个或多个函数组成,函数是构成 C 源程序的基本单位。

(3) 一个源程序不论由多少个文件组成,都有且只有一个 `main` 函数,即主函数。`main` 函数可以位于源程序的任何位置。但是,不管 `main` 函数位于源程序的开头、中间还是末尾,程序都是从 `main` 函数开始执行,并且在 `main` 函数中结束,即 `main` 函数是整个程序的入口和出口。

(4) 每一个语句和声明都必须以分号结尾,分号是构成 C 语句不可缺少的一部分。但是预处理命令和函数首部之后是不加分号的。

(5) 预处理命令通常放在源文件或源程序的最前面。

(6) 关键字和标识符之间必须至少加一个空格以示间隔。

从书写清晰,便于阅读、理解和维护的角度出发,在书写程序时应遵循以下规则。

(1) 一个声明或一条语句占一行。尽管 C 语言允许一行写多个声明或多条语句,但是为了增强程序的可读性,同时也为了便于调试程序,应尽量做到一行只写一个声明或一条语句。

(2) 用 `{}` 括起来的部分,通常表示程序的某一层结构。`{}` 一般与该结构语句的第一个字母对齐,并单独占一行。

(3) 低一层次的语句或声明比高一层次的语句或声明缩进若干格(通常用 `Tab` 键缩进),以使程序层次分明,条理清晰。

(4) 为便于阅读和理解程序,可以在程序中适当地添加一些注释,注释格式为“`/* ..... */`”,注释不属于 C 语句,编译程序时不对注释做语法检查。

在编程时应力求遵循上述规则,以养成良好的编程习惯。

1.6 如何学习 C 语言

学习一门新的程序设计语言的最佳途径就是阅读例程、模仿例程编写程序,通过各种实例程序来学习 C 语言的语法规则、编程的思路和技巧。对于每一章中提到的所有例程,都要仔细研读,直到每一行都理解了,然后合上书本,在计算机上把这些程序输进去并运行。再找几个类似的编程题目,尝试自己把程序写出来,如果写不出来,再回过头来继续研究例程,想想自己为什么写不出来,然后再去写这段程序,反反复复,直到手到擒来为止。学习 C 语言贵在坚持上机实践,不能光看不练,在看书的同时就要输入代码,了解程序运行的各种情况有利于更快更牢固地掌握知识点。只有通过反复的观察、分析、比较、实践、总结,才能熟练掌握 C 语言的各种数据类型、运算符、语句结构的使用方法,才能逐渐地积累编程经验,明白编程的奥秘,才能把书本上的知识变成自己的。

具体而言,学习编程的基本方法分为以下几个步骤。

- (1) 阅读教材上的案例,并按照教材上的案例编写程序,上机运行,查看运行结果。
- (2) 写几个小程序解决一些简单问题,以熟悉基本的算法和语法知识。
- (3) 从自己身边熟悉的事件入手,如成绩统计,编制一段程序,完成一个功能,然后逐步完善。

(4) 要反复上机练习。C 程序设计是一门实践性很强的课程,只有多上机练习才能提高编程能力,而不能只满足于课堂听讲,还要注意课后作业是否会做,上机编程是否能得到正确结果,程序出现问题后能否调试出正确结果等。只有亲自动手编程才能真正提升自己的计算思维能力,增强自己用计算机解决问题的实践能力,为今后工作奠定坚实的基础。

(5) 循序渐进。高级语言的语句看起来很多很复杂,对编程技巧要求也很高,但是其基本的语句和常用算法并不多,所以在学习过程中要注意对一些常用的基本算法进行归纳和总结,在熟练掌握基本语句和经典算法的基础上再进行举一反三、逐步深入。

(6) 粗细结合。高级语言语法规则繁多,需要记忆的内容也很多,比如整数的范围、实数的范围,它们在机器中存储时所占的字节数,输出时的有效位数,默认宽度等,但是这些细节并不影响高级语言的初步学习,有些规则可通过上机逐步掌握,没必要死记硬背,可以粗一些,但对于基本的函数和语句的书写格式、作用、执行过程等一定要记准,并会灵活应用,这里就一定要细一些,因为高级语言语法是非常严格的,一点小错误,如漏掉一个分号等,都会造成程序错误而无法执行。

(7) 多种手段交流。通过 C 语言编程论坛、教学网站或 QQ 群与教师、同学、网友等进行学习交流,不断吸取别人的长处,丰富编程经验,提高编程水平。

总之,学好 C 语言贵在“实践出真知”,掌握这一原则,就一定会学好 C 语言。

1.7 Turbo C 2.0 简介

Turbo C 是美国 Borland 公司开发的产品。该公司在 1987 年曾首次推出 Turbo C 1.0 产品,其中使用了集成开发环境,大大方便了程序的开发。1988 年, Borland 公司又推出 Turbo C 1.5 版本,增加了图形库和文本窗口函数库等。1989 年推出的 Turbo C 2.0 产品增加了查错功能。

被广泛使用的 Turbo C 2.0 不仅是一个高效快捷的编译软件,而且还有一个使用方便的集成开发环境,利用 Turbo C 2.0 就可以编辑、编译和连接程序,并可以直接运行 C 语言程序,而这些功能只要通过一个简单的子屏幕操作即可实现。

下面将详细介绍 Turbo C 2.0 的使用方法。

(1) Turbo C 2.0 无需安装,将 Turbo C 2.0 解压到某文件夹下,用鼠标双击 Turbo C 2.0 所在目录下的 TC.EXE 文件就能打开 Turbo C 2.0,如图 1-2 所示。



图 1-2 Turbo C 2.0 目录

(2) 按 Alt + O 快捷键,选择 Options | Directories 命令,将 Turbo C directory、Include directories、Library directories 分别指定为 Turbo C 2.0 及其头文件、库文件所在目录。Output directory 用于指定默认的源程序存储路径,可以设定为用户自己创建的文件夹,如图 1-3 所示。

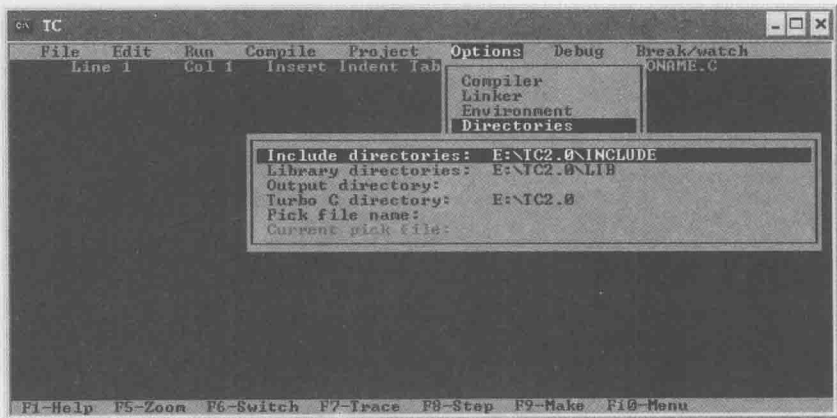


图 1-3 TC 2.0 运行环境配置

选择 Options | Save options 命令,覆盖 TCCONFIG. TC 文件,即保存了环境设置。此后每次打开 Turbo C 2.0 时就无须再进行环境设置。

(3) 按 Alt + F 快捷键可以打开 File 菜单,选择其中的 New 子菜单项可以编辑新的源程序,如图 1-4、图 1-5 所示。

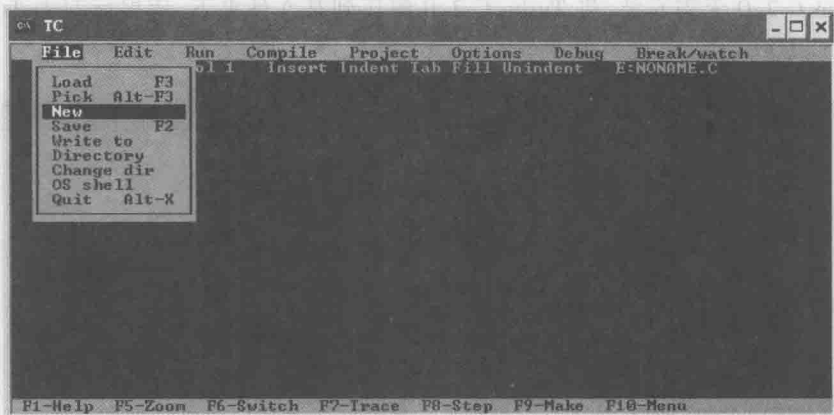


图 1-4 在 Turbo C 2.0 中编辑新的源程序

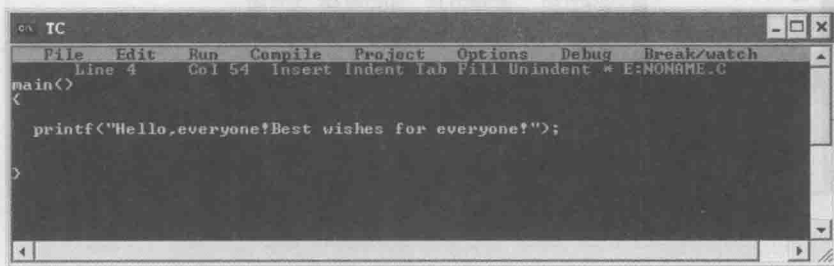


图 1-5 编辑好的源程序

(4) 选择 Compile | Compile to OBJ 命令对源程序进行编译,编译成功后再选择 Compile | Link EXE file 命令,生成 EXE 文件,如图 1-6、图 1-7、图 1-8 和图 1-9 所示。

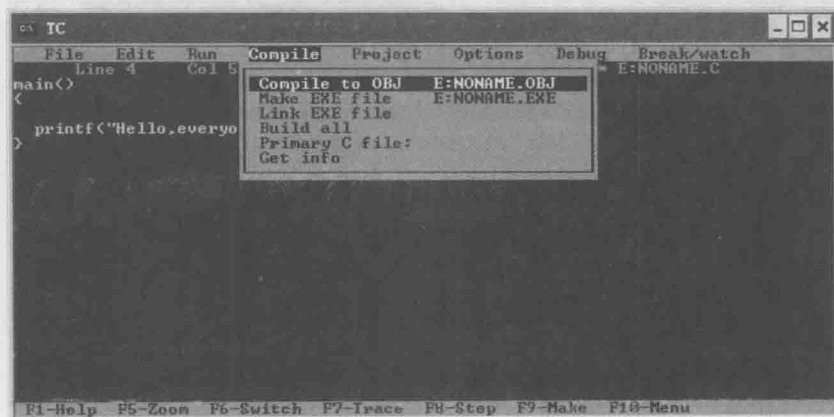


图 1-6 在 Turbo C 2.0 中编译源程序

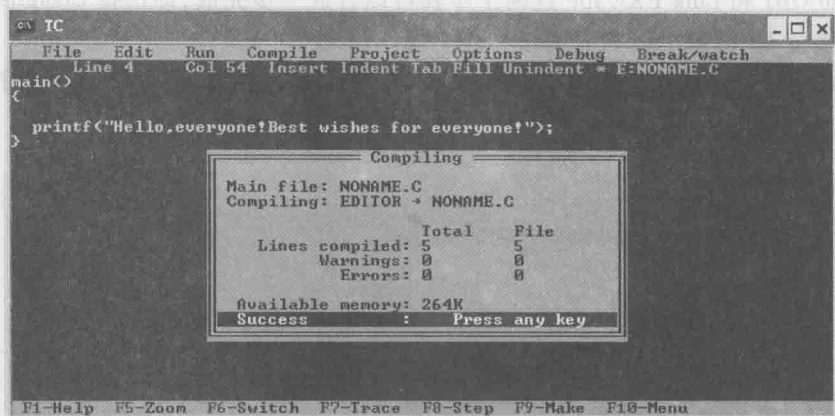


图 1-7 程序编译成功

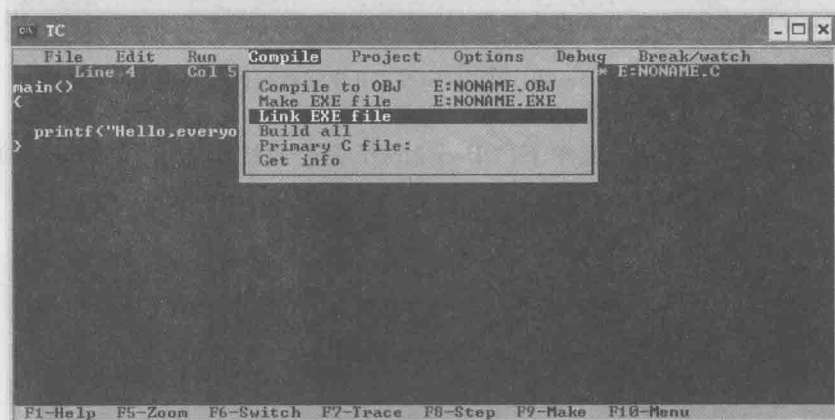


图 1-8 连接库函数并生成 EXE 文件

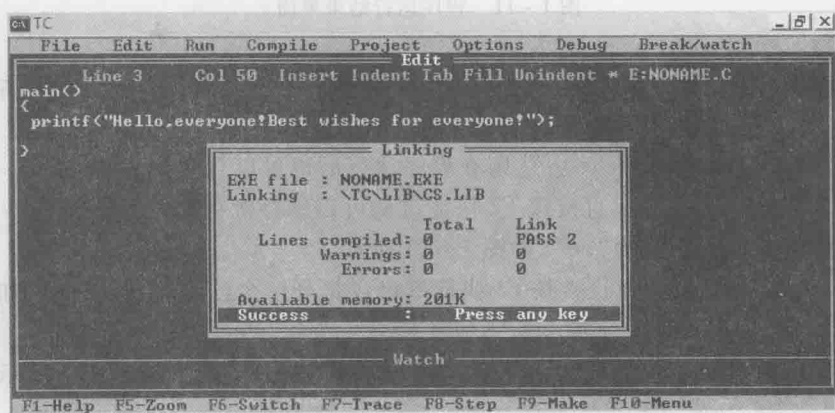


图 1-9 连接成功并生成 EXE 文件

Compile to OBJ 和 Link EXE file 这两个步骤可以合成一步完成,即选择 Compile | Make EXE file 命令或直接按 F9 键都可以完成编译和连接操作。

(5) 按 Alt + R 快捷键选择 Run 菜单或直接按 Ctrl + F9 快捷键都可以运行程序,如图 1-10 所示。



图 1-10 在 Turbo C 2.0 中运行程序

(6) 程序执行完后选择 Run 菜单下的 User screen 命令或按 Alt + F5 快捷键看到程序执行结果,如图 1-11 所示。



图 1-11 程序运行结果界面

如果在程序末尾加一句“system(pause);”,则在程序运行之后可以直接看到结果。“Press any key to continue...”表示按任意键可以回到程序编辑窗口。

如果在程序末尾加一句“getch();”,则在程序执行后也可以直接看到结果。

(7) 选择 File | Save 命令或直接按 F2 快捷键都可以保存源程序文件,如图 1-12 所示。保存文件时文件的扩展名必须为“.c”,如图 1-13 所示。

(8) 选择 File | Load 命令或直接按 F3 快捷键可以打开一个已经存在的文件,如图 1-14 所示。

(9) 选择 File | Write to 命令可以改变文件名或文件存储路径,如图 1-15 所示。