



技能型紧缺人才培养系列规划教材

C语言程序设计案例教程

就业导向 任务引领 案例驱动

遵从教学规律 按节细化知识 保证知识体系

沈大林 赵 玺 主 编

王浩轩 许 崇 陶 宁 杨 旭 副主编



 中国铁道出版社



技能型紧缺人才培养系列规划教材

C 语言程序设计案例教程

沈大林 赵 玺 主编

王浩轩 许 崇 陶 宁 杨 旭 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C 语言是功能最强大、被大多数程序员所认可的高级编程语言之一。它也可以完成许多只有低级语言才能完成的、面向机器的底层工作,因此又称“中间语言”。正是由于 C 语言的这些特性,使其成为一种重要的程序设计语言。

本书是《技能型紧缺人才培养系列规划教材》丛书之一。本书具有较大的知识信息量,共讲解了近 70 个案例,提供了 100 多道思考与练习题。本书以计算机实例操作为主线,采用真正的任务驱动方式,展现全新的教学方法。本书贯穿以实例带动知识点的学习,通过学习实例掌握软件的操作方法和操作技巧。

本书适合作为中等职业技术学校计算机专业或高等职业学校非计算机专业的教材,也可作为各种初、中级计算机培训班的教材或初学者的自学用书。

图书在版编目(CIP)数据

C 语言程序设计案例教程 / 沈大林, 赵玺主编.

—北京: 中国铁道出版社, 2012. 1

技能型紧缺人才培养系列规划教材

ISBN 978-7-113-13680-2

I. ①C… II. ①沈… ②赵… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 005620 号

书 名: C 语言程序设计案例教程

作 者: 沈大林 赵 玺 主编

策 划: 刘彦会

读者热线: 400-668-0820

责任编辑: 刘彦会 冯彩茹

封面设计: 刘 颖

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.edusources.net>

印 刷: 北京新魏印刷厂

版 次: 2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷

开 本: 787 mm×1092 mm 1/16 印张: 13.5 字数: 324 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-13680-2

定 价: 25.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 63549504



审稿专家组

审稿专家：（按姓名笔画排列）

丁桂芝（天津职业大学）	毛一心（北京科技大学）
毛汉书（北京林业大学）	王行言（清华大学）
邓泽民（教育部职业技术教育中心研究所）	
冯博琴（西安交通大学）	艾德才（天津大学）
安志远（北华航天工业学院）	曲建民（天津师范大学）
刘瑞挺（南开大学）	吴文虎（清华大学）
宋文官（上海商学院）	李凤霞（北京理工大学）
吴功宜（南开大学）	宋红（太原理工大学）
陈明（中国石油大学）	陈维兴（北京信息科技大学）
张森（浙江大学）	徐士良（清华大学）
钱能（杭州电子科技大学）	黄心渊（北京林业大学）
龚沛曾（同济大学）	蔡翠平（北京大学）
潘晓南（中华女子学院）	



“技能型紧缺人才培养系列规划教材”丛书

丛 书 编 委 会

主 任： 沈大林

副主任： 苏永昌 张晓蕾

编 委：（按姓名笔画排列）

马广月	马开颜	王 玥	丰金茹
王 威	王浩轩	王爱赅	王 锦
王 翠	朱 立	曲彭生	迟 萌
迟锡栋	刘 璐	张凤红	肖柠朴
张 伦	杨 红	杨 旭	杨素生
张 磊	杨继萍	罗红霞	沈 昕
杜 金	沈建峰	陈恺硕	郑 原
郑淑晖	郑 瑜	郑 鹤	赵亚辉
陶 宁	高立军	袁 柳	崔 玥
曾 昊	董 鑫		

本套教材依据教育部办公厅和原信息产业部办公厅联合颁发的《中等职业院校计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》进行规划。

根据我们多年的教学经验和对国外教学的先进方法的分析,针对目前职业技术学校学生的特点,采用案例引领,将知识按节细化,案例与知识相结合的教学方式,充分体现我国教育学家陶行知先生“教学做合一”的教育思想。通过完成案例的实际操作,学习相关知识、基本技能和技巧,让学生在学习中始终保持学习兴趣,充满成就感和探索精神。这样不仅可以让学生迅速上手,还可以培养学生的创作能力。从教学效果来看,这种教学方式可以使学生快速掌握知识和应用技巧,有利于学生适应社会的需要。

每本书按知识体系划分为多个章节,每一个案例是一个教学单元,按照每一个教学单元将知识细化,每一个案例的知识都有相对的体系结构。在每一个教学单元中,将知识与技能的学习融于完成一个案例的教学中,将知识与案例很好地结合成一体,案例与知识不是分割的。在保证一定的知识系统性和完整性的情况下,体现知识的实用性。

每个教学单元基本由“案例效果”、“操作步骤”、“相关知识”和“思考与练习”四部分组成。在“案例效果”栏目中介绍案例完成的效果;在“操作步骤”栏目中介绍完成案例的操作方法和操作技巧;在“相关知识”栏目中介绍与本案例单元有关的知识,起到总结和提高自己的作用;在“思考与练习”栏目中提供了一些与本案例有关的思考与练习题。对于程序设计类的教程,考虑到程序设计技巧较多,不易于用一个案例带动多项知识点的学习,因此采用先介绍相关知识,再结合知识介绍一个或多个案例。

丛书作者努力遵从教学规律、面向实际应用、理论联系实际、便于自学等原则,注重训练和培养学生分析问题和解决问题的能力,注重提高学生的学习兴趣和培养学生的创造能力,注重将重要的制作技巧融于案例介绍中。每本书内容由浅入深、循序渐进,使读者在阅读学习时能够快速入门,从而达到较高的水平。读者可以边进行案例制作,边学习相关知识和技巧。采用这种方法,特别有利于教师进行教学和学生自学。

为便于教师教学,丛书均提供了实时演示的多媒体电子教案,将大部分案例的操作步骤实时录制下来,让教师摆脱重复操作的烦琐,轻松教学。

参与本套教材编写的作者不仅有在教学一线的教师,还有在企业负责项目开发的技术人员。他们将教学与工作要求更紧密地结合起来,通过完整的案例教学,提高学生的应用操作能力,为我国职业技术教育探索更添一臂之力。

沈大林

C 语言功能强大,它虽然是高级语言,但也可以完成许多只有低级语言才能完成的、面向机器的底层工作。正是由于 C 语言的这些特性,决定了它成为一种重要的程序语言。例如,使用最多的操作系统 Windows 2000/XP、Linux 等都是用 C 语言编写的。

本书共分 8 章,第 1 章简单介绍了 C 语言的特点和编程工具等,通过一个简单实例说明如何使用 Turbo C++3.0 集成开发环境进行 C 语言程序设计的方法和过程,以及 C 语言的基本数据类型、常量与变量、运算符与表达式;第 2 章结合 17 个实例介绍了程序结构、算法、输入/输出函数等内容;第 3 章结合 12 个实例介绍了 C 语言程序的选择结构程序设计的方法;第 4 章结合 19 个实例介绍了循环结构程序设计的方法;第 5 章结合 20 个实例介绍了函数的定义与调用、函数参数传递与递归、变量作用域和变量存储类型、标准函数应用等;第 6 章结合 10 个实例介绍了数组与字符串;第 7 章结合 11 个实例介绍了指针的定义与应用、数组指针和字符指针等;第 8 章结合 7 个实例介绍了结构体、共用体和编译预处理技术在程序设计中的应用。

本书具有较大的信息量,讲解了近 100 个实例,提供了 100 多道思考与练习题。全书以计算机实例操作为主线,采用任务驱动方式展现全新的教学方法。本书贯穿以实例带动知识点的学习,通过学习实例掌握软件的操作方法和操作技巧。在按实例进行讲解时,充分注意知识的相对完整性和系统性。读者可以跟着本书的操作步骤去操作,从而完成应用实例的制作,在实例制作中轻松地掌握 C 语言程序设计。

本书由浅及深、由易到难、循序渐进、图文并茂,理论与实际制作相结合,可使读者在阅读学习时知其然还知其所以然,不但能够快速入门,而且可以达到较高的水平。教师可以得心应手地使用它进行教学,学生也可以自学。

本书是在任务驱动教学法的基础上总结编写出来的。建议教师在使用本教材进行教学时,一边带学生做各章的实例,一边讲解各实例中的知识和概念,将它们有机地结合在一起,可以达到事半功倍的效果。

本书由沈大林、赵玺主编,王浩轩、许崇、陶宁、杨旭等编著,参与编写工作的还有沈昕、张伦、王爱赫、万忠、郑淑晖、曾昊、崔玥、肖柠朴、曹永冬、沈建峰、郑鹤、于建海、郭政、郭海、郝侠、丰金兰、陈恺硕、孔凡奇、王锦、袁柳、李宇辰、王加伟、徐晓雅、卢贺、王建平、靳轲、张 铮、朱海跃、苏飞、王小兵等。

由于作者水平有限,加上编著、出版时间仓促,书中难免有疏漏和不足之处,恳请广大读者批评指正。

编 者

2011 年 11 月

第 1 章 初步了解 C 语言程序设计	1
1.1 C 语言概述	1
1.1.1 了解 C 语言	1
1.1.2 C 语言的编程工具	2
思考与练习 1-1	4
1.2 设计第一个 C 语言程序	5
1.2.1 编辑和保存 C 程序	5
1.2.2 编译与连接、运行和调试 C 程序	9
思考与练习 1-2	14
1.3 基本数据类型、常量与变量	15
1.3.1 标识符与关键字	15
1.3.2 基本数据类型	16
1.3.3 常量与变量	17
思考与练习 1-3	20
1.4 运算符和表达式	20
1.4.1 算术运算符和算术表达式	21
1.4.2 关系运算符和关系表达式	22
1.4.3 逻辑运算符和逻辑表达式	23
1.4.4 其他运算符和表达式	24
1.4.5 运算中数据类型的转换	26
思考与练习 1-4	27
第 2 章 程序结构和算法及数据的输出与输入	30
2.1 程序的基本结构和算法	30
2.1.1 C 程序的基本结构和语句	30
2.1.2 算法	32
思考与练习 2-1	35
2.2 格式化输出函数	36
2.2.1 格式化输出函数	36
2.2.2 格式化输出实例	39
思考与练习 2-2	40
2.3 格式化输入函数	41
2.3.1 格式化输入函数	41
2.3.2 格式化输入实例	44
思考与练习 2-3	47
2.4 字符输入/输出函数	49
2.4.1 字符输入/输出函数的格式与功能	49
2.4.2 字符输入/输出函数的应用实例	50
思考与练习 2-4	51
第 3 章 选择结构程序设计	52
3.1 if 条件分支语句	52
3.1.1 if 单分支语句	52
3.1.2 if-else 双分支语句	54
3.1.3 if-else if-else 多分支语句	56
思考与练习 3-1	59
3.2 switch 开关分支语句和选择结构的嵌套	62
3.2.1 switch 开关分支语句	62
3.2.2 选择结构的嵌套	67
思考与练习 3-2	71
第 4 章 循环结构程序设计	75
4.1 循环结构	75
4.1.1 while 语句	75
4.1.2 do-while 语句	77
4.1.3 for 语句	80
思考与练习 4-1	86
4.2 循环嵌套及中断和转向语句	88
4.2.1 循环嵌套	89
4.2.2 break 和 continue 语句	92
4.2.3 转向语句 goto	95

思考与练习 4-2	99	第 7 章 指针	157
第 5 章 使用函数	103	7.1 指针的定义与应用	157
5.1 函数的定义与调用	103	7.1.1 指针概述	157
5.1.1 什么是函数	103	7.1.2 指针的定义	158
5.1.2 函数的定义	105	7.1.3 指针变量的引用	159
5.1.3 函数的调用	109	思考与练习 7-1	163
思考与练习 5-1	112	7.2 指针与数组	163
5.2 函数参数传递与递归	113	7.2.1 指针的运算	163
5.2.1 函数参数传递	113	7.2.2 数组指针和字符指针	166
5.2.2 函数的递归调用	119	7.2.3 指针数组	169
思考与练习 5-2	123	7.2.4 应用实例	169
5.3 变量作用域和变量存储类型	124	思考与练习 7-2	172
5.3.1 变量作用域	124	第 8 章 结构体、共同体和编译	
5.3.2 变量存储类型（选学）	126	预处理	174
思考与练习 5-3	128	8.1 结构体	174
5.4 标准函数应用	129	8.1.1 结构体的定义	174
思考与练习 5-4	134	8.1.2 结构体变量的引用	178
第 6 章 数组与字符串	135	8.1.3 自定义数据类型和结构	
6.1 一维数组	135	体数组	179
6.1.1 什么是数组	135	8.1.4 应用实例	181
6.1.2 一维数组	135	思考与练习 8-1	184
6.1.3 一维数组的引用和元素		8.2 共用体	185
地址	137	8.2.1 共用体	185
6.1.4 应用实例	138	8.2.2 应用实例	187
思考与练习 6-1	141	思考与练习 8-2	188
6.2 多维数组	142	8.3 编译预处理	189
6.2.1 二维数组的定义	142	8.3.1 宏定义	189
6.2.2 多维数组的定义和引用	143	8.3.2 文件包含	191
6.2.3 应用实例	144	8.3.3 条件编译	192
思考与练习 6-2	147	8.3.4 应用实例	193
6.3 字符数组	148	思考与练习 8-3	195
6.3.1 字符数组	148	附录 A ASCII 码对照表	197
6.3.2 字符串处理函数	150	附录 B 运算符的优先级	
6.3.3 应用实例	152	与结合性	199
思考与练习 6-3	155	附录 C 标准库函数	201

第 1 章 初步了解 C 语言程序设计

1.1 C 语言概述

1.1.1 了解 C 语言

1. C 语言的发展

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,早期的 C 语言主要用于 UNIX 系统。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广泛使用,成为当代最优秀的程序设计语言之一。许多著名的系统软件,例如 Windows 2000/XP、Linux 等都大量使用了 C 语言进行程序设计, DBASE IV 等软件则完全是由 C 语言编写的。C 语言的功能非常强大,它虽然是高级语言,但也可以完成许多只有低级语言(例如汇编语言)才能完成的、面向机器的底层工作,因此也被叫做“中级语言”。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准学会(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言国际标准版本。以后又有 C++ 和 Visual C++ 版本,它们都是在 C 语言基础之上发展的 C 语言的超集,C++ 语言是面向对象的程序设计语言, Visual C++ 是事件驱动的程序设计语言。学会 C 语言有利于 C++ 和 Visual C++ 的学习。

在程序设计课程体系中, C 语言是众多后继课程的基本编程工具,特别是与 Windows 编程有关的课程。因此,与计算机相关的专业都把 C 语言程序设计列为基础课程之一。学好 C 语言,对将来学习更为高级的程序设计课程具有重要的意义。

2. C 语言的特点

C 语言的主要特点简介如下:

(1) 中级语言: 编程语言有“高级”和“低级”之分,但并不标志解决问题能力的高低,而是表示与计算机的硬件系统联系的程度。C 语言不仅具有高级语言可读性、可移植性、容易学习等特点,而且兼有低级语言(汇编语言)可以直接控制计算机硬件(可以像汇编语言一样对位、字节和地址进行操作,这是计算机最基本的操作)和运行效率高等优点。

(2) 结构化语言: 层次清晰,便于按模块化方式组织程序,易于调试和维护。结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护和调试。C 语言是以函数形式提供给用户的,

这些函数可方便调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(3) 语言功能强大:具有各种各样的数据类型,并引入了指针概念,可使程序运行效率更高。C 语言具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。另外,C 语言的编程自由度较大,程序书写形式也很自由,大大方便了程序设计人员编程。

(4) 适用范围广泛:表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于 C 语言实现了对硬件的编程操作,因此 C 语言集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也适合于应用软件的开发。

(5) 适合于多种操作系统:可适合于 UNIX、Windows、Linux 等操作系统。正是由于 C 语言的这些特性,决定了它成为一种重要的程序设计语言。通常所使用的软件中,相当多的程序都是由 C 语言编写而成。

(6) 可移植性强:在 C 语言中,没有依赖于硬件的输入/输出语句。程序的输入/输出功能是通过调用函数来实现的,这些函数是由独立于 C 语言的系统程序模块库提供的。C 语言本身并不依赖于计算机的硬件系统,从而便于在硬件不同的各种类型计算机之间实现程序的移植。

3. C 语言程序开发流程

使用高级语言编写的程序称为程序的“源代码”,编写 C 语言程序的过程称为代码编辑。C 程序源代码的输入和编辑可以用专门的代码编辑器来完成,也可以用普通的纯文本处理器(例如 Windows 中的记事本)来进行。

由于计算机只能识别二进制代码指令,源代码不能直接被计算机所识别和执行,必须将 C 语言源代码进行编译,转化为二进制的计算机指令,才能够被计算机所执行。从理论上讲,任何高级语言都有两种执行方式,一种是解释执行方式,解释程序每读一条语句,就翻译成相应的机器语言指令,例如 BASIC 语言就采用该种方式(个别 C 语言在调试阶段采用这种方式);另一种是编译执行方式,编译系统编译源程序后,将所有语句翻译成机器指令,产生一个目标文件。连接目标程序后产生一个可执行文件,然后执行这个可执行文件。

将 C 语言源代码转化为二进制指令的过程称为编译(Compile),这需要有专门的编译器来执行。编译后的二进制代码文件叫做目标文件,该文件的扩展名为.obj。目标文件也不能直接在计算机中运行,还需要通过连接(Link)程序将它与 C 语言的库文件进行连接,最后生成可执行的程序文件。可执行程序文件的扩展名为.exe。

从源文件到可执行程序文件经过的流程如图 1-1-1 所示。

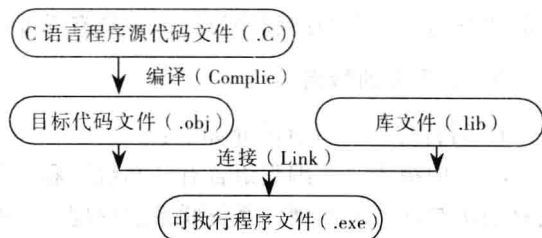


图 1-1-1 C 程序编译流程

1.1.2 C 语言的编程工具

目前,都把负责代码编辑的编辑器、程序调试器、编译目标代码的编译器和连接库文件生成目标文件的连接器综合起来,组成一个 C 语言的编程工具(又称为集成开发环境)。这种编

程工具软件有很多,其中最有名、使用最多的是 Borland 公司的 Turbo C 系列集成开发环境和 Microsoft 公司的 Visual C++ 系列软件。用 C 语言编写程序,必须先学会使用 C 语言的编程工具软件。

1. Turbo C

Turbo C 以其编译速度快、代码执行效率高而著称,是 C 程序员最乐于使用的编程工具之一。Turbo C 版本很多,目前主要使用 Turbo C 2.0 和 Turbo C++3.0 版本,且都有汉化版本。

(1) Turbo C 的安装: Turbo C 的安装极为简单,双击安装程序 INSTALL.EXE 的图标,运行安装程序,即可按照提示信息一步步进行程序的安装,这里不再详细叙述。Turbo C 安装后,在 C 盘根目录下会自动生成一个 TC 文件夹,其内是安装的 Turbo C 的有关文件。

(2) Turbo C 的启动: 可以选择“开始”→“所有程序”→“附件”→“命令提示符”命令,弹出“命令提示符”窗口,再输入 C:\TC\BIN\TC 命令,然后按 Enter 键,即可启动 Turbo C。也可以双击 C:\TC\BIN 路径下的 TC 程序图标 , 启动 Turbo C。

(3) 汉化 Turbo C: 在原 Turbo C 的基础之上进行汉化,还具有以下特点:

- 纠正了原 Turbo C 中的一些 Bug (错误)。
- 集成了中文 DOS 运行环境,无须另行下载安装即可直接采用中文界面进行操作。
- 可以在 Windows 环境下方便地安装 Turbo C, 本安装程序可以将 Turbo C 安装在任意指定目录,无须修改 Options 菜单里的 Directory 选项,即可正常编译标准 C 程序。
- 提供了源代码编辑器,让习惯于在 Windows 环境操作的用户使用方便。
- 汉化 Turbo C 2.0 和 Turbo C ++ 3.0 的安装程序名称分别为 WY_TC20H_B1009.EXE 或 WY_TCPP30H_B1005.EXE, 安装汉化 Turbo C++ 3.0 后,建立的 C 程序文件将保存在“安装目录\Project”目录下,程序输出的.OBJ 和.EXE 文件将保存在“安装目录\Output”目录下。

(4) Turbo C++ 3.0 版本与 Turbo C 2.0 版本的主要区别如下:

- Turbo C++ 3.0 修正了 Turbo C 2.0 中存在的一些 Bug (错误)。
- Turbo C++ 3.0 支持多窗口操作,窗口间可以快速切换。
- 完全支持鼠标操作(单击选择、拖动和右击等),照顾了习惯于图形界面操作的用户。
- 建立即时帮助系统,只需选定关键字后按 Ctrl+F1 组合键,即可查看详细的帮助。并且,每个函数都具有完整的示例解释说明,只需要复制到新文件即可运行。无论对 C 语言初学者,还是 C++ 高手都是不错的实例教材。
- 可以自定义语句按照语法高亮多色显示,令代码编写、程序查错时更直观方便。
- 提供了一个程序编辑器,该程序编辑器的查找、替换等编辑功能更方便易用,建立和管理项目更方便。
- Turbo C 2.0 程序文件的扩展名是.C, Turbo C++ 3.0 程序文件的扩展名可以以扩展名.C 保存,也可以以扩展名.CPP 保存。
- 建立和管理项目更方便容易。

2. Visual C++ 6.0

除了 Borland 公司的 Turbo C 系列集成开发环境外,Microsoft 公司的 Visual C++ 系列也是常用的 C 语言开发工具。随着近几年 C++ 语言程序的普及,Visual C++ 集成开发环境作为一种

功能强大的程序编译器也被相当多的程序员所使用,使用 Visual C++也能够完成 C 语言的编译。由于 Visual C++集成开发环境运行于 Windows 平台下,对于习惯于图形界面的用户来说是比较易学的。

Visual C++ 6.0 集成开发环境的安装比较简单,双击安装程序 Setup.exe 的图标,运行安装程序后,即可按照提示信息一步步完成程序的安装,这里不再详细叙述。

安装后,选择“开始”→“所有程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令,即可启动 Visual C++ 6.0 集成开发环境。

3. 两种编程工具的比较

(1) Visual C++ 6.0 由于是在 Windows 下运行,具有图形窗口界面,因此比 DOS 下的 Turbo C 更易于操作,但同时也发现,由于 Visual C++ 6.0 集成了太多功能,而且主要用于 C++程序的设计,因此占用资源较多,没有 Turbo C 那样简洁。

(2) Visual C++ 6.0 在对程序进行编译时所耗时间较长,远大于 Turbo C 的编译时间。

(3) 比较编译的结果,使用 Visual C++ 6.0 编译的程序会生成多个中间文件,最终得到的可执行文件较大;使用 Turbo C 编译的程序生成的中间文件较少,最终得到的可执行文件远小于 Visual C++ 6.0 生成的文件。

注意:在 Turbo C 下的数据/变量所占存储空间大小与 Visual C++ 6.0 有所不同,如 int 型变量在 Turbo C 下编译时占 2 个字节,而在 Visual C++ 6.0 下编译时占 4 个字节。因此,在进行内存地址的引用时会有所不同。

本书中的所有例子都可在 Turbo C 2.0、Turbo C++ 3.0 和 Visual C++ 6.0 等 C 软件下正常运行,但为了便于初学者学习,在讲解时主要以 Turbo C++ 3.0 环境为主。

思考与练习 1-1

1. 问答题

- (1) C 语言是功能非常强大的高级语言,为什么也被叫做“中级语言”?
- (2) C 语言主要有哪些特点?
- (3) 常见的 C 语言的编程工具有哪几种?
- (4) 如何安装 Turbo C++ 3.0? 如何启动 Turbo C++ 3.0?
- (5) Turbo C++ 3.0 版本与 Turbo C 2.0 版本的主要区别有哪些?

2. 填空题

- (1) 任何高级语言都有两种执行方式,一种是_____方式,另一种是_____方式。
- (2) 使用高级语言编写的程序称为程序的_____,编写 C 语言程序的过程称为_____。C 程序源代码的输入和编辑可以用专门的_____来完成,也可以用普通的_____来进行。
- (3) 将 C 语言源代码转化为二进制指令的过程称为_____,这需要有专门的_____来执行。编译后的二进制代码文件叫做_____,该文件的扩展名为_____。

(4) 目标文件也不能直接在计算机中执行, 还需要通过_____将它与C语言的_____进行连接, 最后生成_____, 该文件的扩展名为_____。

1.2 设计第一个C语言程序

本节介绍采用 Turbo C++ 3.0 设计一个C语言程序 TC1-1.CPP 的方法和步骤, 为本书的学习打下一个良好的基础。TC1-1.CPP 程序运行后, 会在DOS窗口内显示图 1-2-1 所示的文字。该程序的设计、保存、编译与连接, 以及运行程序和调试程序的方法介绍如下, 同时还介绍汉化版 Turbo C++ 3.0 的基本使用方法。



图 1-2-1 C 语言程序的运行结果

1.2.1 编辑和保存C程序

1. 启动和退出汉化版 Turbo C++ 3.0

如果安装汉化版 Turbo C++ 3.0 时选择的安装目录是 C:\TCPP30H, 则会在 C:\TCPP30H 目录下建立一个 Project 文件夹和一个 Output 文件夹。默认情况下, Project 文件夹用来保存C程序文件, Output 文件夹用来保存程序输出的.OBJ 和.EXE 文件。

在安装汉化版 Turbo C++ 3.0 后, 会在“开始”→“所有程序”→“程序设计”→“Turbo C ++3.0 汉化版”命令下创建一组命令, 如图 1-2-2 所示。

选择“Turbo C++ 3.0 汉化版”菜单中的“Turbo C++ 3.0[英文]”命令或“Turbo C++ 3.0”命令, 可弹出 Turbo C++ 3.0 工作环境, 如图 1-2-3 所示。



图 1-2-2 “Turbo C++ 3.0 汉化版”菜单

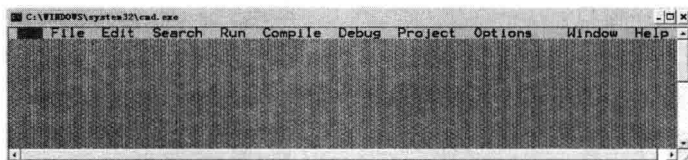


图 1-2-3 英文 Turbo C++ 3.0 工作环境

选择“Turbo C ++3.0 汉化版”菜单中的“Turbo C++ 3.0[汉字]”命令或“Turbo C++ 3.0[汉化]”命令, 可以弹出一个“重要提示”对话框, 单击该对话框内的“是”按钮, 可以调入 CCDOS 汉字系统, 并弹出虚拟的汉化 Turbo C++ 3.0 工作环境, 如图 1-2-4 所示。通常C程序内要显示汉字时, 可以进入汉化的 Turbo C++ 3.0 工作环境。

选择“Turbo C ++3.0 汉化版”菜单中的“Turbo C++3.0 向导”命令, 会弹出“Turbo C ++ 运行向导”对话框, 如图 1-2-5 所示。单击该对话框中“主要功能”栏内不同的选项, 相当于执行相应的菜单命令。

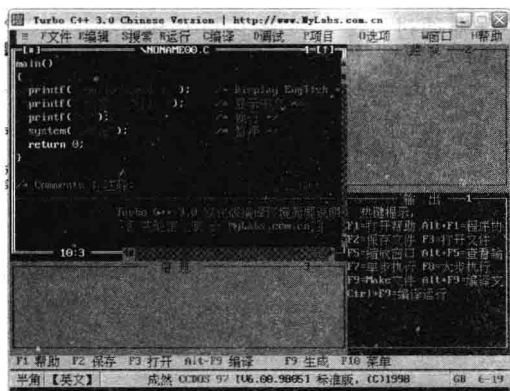


图 1-2-4 汉化的 Turbo C++ 3.0 工作环境

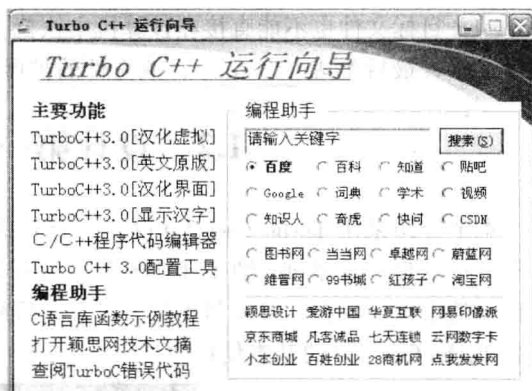


图 1-2-5 “Turbo C++运行向导”对话框

本书如果没有特殊声明，均指在英文或汉化的 Turbo C++ 3.0 工作环境下操作。

在英文 Turbo C++ 3.0 工作环境下，选择 File（文件）→Quit（退出）命令或按 Alt+X 组合键，可以关闭英文 Turbo C++ 3.0 工作环境，退出 Turbo C++ 3.0。

2. 新建和打开 C 程序

（1）新建 C 语言程序：英文 Turbo C++ 3.0 工作环境如图 1-2-3 所示，选择 File（文件）→New（新建）命令，会弹出 C:\TCPP30H\ Project 文件夹中的 NONAME00.C 程序，该程序内给出了一段 C 语言小程序，同时还给出了 Turbo C++ 3.0 汉化版编译环境的简要说明。读者可以用记事本打开 NONAME00.C 程序，观看 Turbo C++ 3.0 汉化版编译环境的简要说明，其内包括 Turbo C++ 常用快捷键和 CCDOS 的基本操作等。

用鼠标拖动选中 NONAME00.C 程序内的所有内容，再选择 Edit（编辑）→Clear（清除）命令或 Ctrl+Delete 组合键，将选中的内容删除。也可以选择 Edit（编辑）→Cut（剪切）命令或按 Shift+Delete 组合键，将选中的内容剪切到剪贴板内。

如果 C:\TCPP30H\ Project 文件夹内没有 NONAME00.C 程序，则会新建一个空 C 语言程序窗口。如果已经打开了一个 C 语言程序窗口，则选择其中的 File（文件）→New（新建）命令，新建一个空 C 语言程序窗口。

（2）打开 C 语言程序：选择 File（文件）→Open（打开）命令，弹出 Open a File（打开一个文件）界面，如图 1-2-6 所示（还没有输入文件名称）。在 Name（名字）文本框内输入要打开的文件名称（此处输入 TC1-1.C），该文件应该在 C:\TCPP30H\ Project 文件夹内。如果要打开其他文件夹内的 C 语言程序文件，应输入文件的路径。

然后，单击 Open a File（打开一个文件）界面中的 Open（打开）按钮，即可打开 TC1-1.C 程序的窗口。

如果单击 Replace（替换）按钮，可以关闭原 C 语言程序窗口，打开指定的 TC1-1.C 程序窗口，完成替换工作。如果单击 Cancel（取消）按钮，可以关闭 Open a File（打开一个文件）界面，取消打开操作。

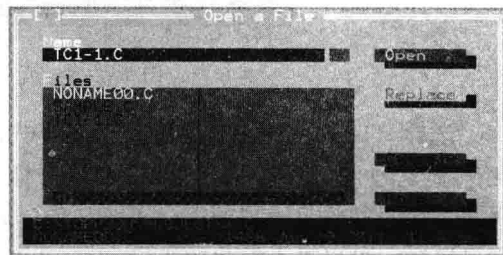


图 1-2-6 Open a File（打开一个文件）界面

3. 输入和编辑C程序

(1) 输入C语言程序：可以在英文 Turbo C++ 3.0 工作环境内新建的C语言程序窗口中直接输入C语言程序。

另外,可以在 Windows 的记事本内输入C语言程序,再以扩展名.C或.CPP保存在 C:\TCPP30H\Project 文件夹内。也可以选择“Turbo C++ 3.0 汉化版”菜单中的“代码编辑器”命令,弹出“代码编辑器”窗口,如图 1-2-7 所示,利用它可以轻松输入和编辑C语言程序,再将该程序保存在 C:\TCPP30H\Project 文件夹内。

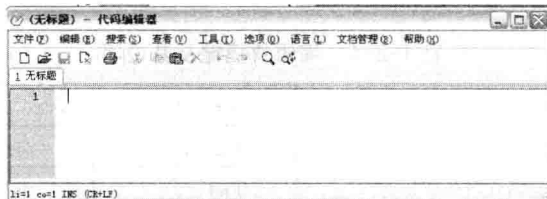


图 1-2-7 “代码编辑器”窗口

然后,在英文 Turbo C++ 3.0 工作环境内打开需要的C语言程序。

这里在英文 Turbo C++ 3.0 工作环境内输入如下的 TC1-1.C 程序。此时的英文 Turbo C++ 3.0 工作环境如图 1-2-8 所示。

```
/** TC1-1 */  
/*****/  
#include "stdio.h"  
main()  
{  
    printf("Welcome to Turbo C++ 3.0 World ! \n");  
}
```

程序中,前2行代码对程序做相关说明,“/*”和“*/”符号之间的内容(可以是多行)是注释内容,用来帮助读懂程序,不参与程序的执行。“/*”与“*/”必须同时存在。

#include "stdio.h"语句是程序的预处理命令(在以后的章节将对它做详细解释),引用了C语言的标准库 stdio.h,使下面的 printf()语句得以顺利执行。

main()是C语言程序的主函数入口,所有的C语言程序都从这里开始执行。下一行的左花括号“{”与最后一行的右花括号“}”之间是函数的主体,它们必须成对出现。

printf("Welcome to Turbo C++ 3.0 World ! \n");语句使用了C语言的输出函数 printf()用来输出双引号内的字符串 Welcome to Turbo C++ 3.0 World ! \n。C语言中的字符串都必须用半角双引号“”括起来。其中的\n是转义字符,表示换行。该行语句的最后是分号,所有的C语言语句都必须以分号结束。

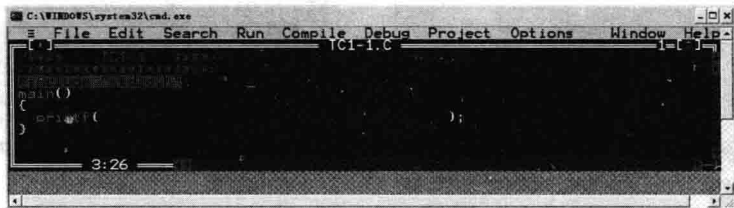


图 1-2-8 在英文 Turbo C++ 3.0 工作环境中输入程序

(2) 编辑 C 语言程序: 通过按方向键可以移动光标。在英文 Turbo C++ 3.0 工作环境的 Edit (编辑) 菜单中提供了一些命令, 利用它们可以简化程序的输入和编辑。

- 选择 Edit→Clear (清除) 命令或 Ctrl+Delete 组合键, 将选中的内容删除。
- 选择 Edit→Cut (剪切) 命令或按 Shift+Delete 组合键, 将选中的内容剪切到剪贴板中。
- 选择 Edit→Copy (复制) 命令或按 Ctrl+Insert 组合键, 将选中的内容复制到剪贴板中。
- 选择 Edit→Paste (粘贴) 命令或按 Shift+ +Insert 键, 将剪贴板中的内容粘贴到当前光标处。

另外, 英文 Turbo C++ 3.0 还提供了许多编辑快捷键, 如表 1-2-1 所示, 利用它们也可以简化程序的输入和编辑。

表 1-2-1 编辑快捷键及其说明

编辑快捷键	说 明	编辑快捷键	说 明
PageUp	向前翻一页	PageDn	向后翻一页
Home	将光标移到所在行的开始处	End	将光标移到所在行的结尾处
Ctrl+N	在光标所在行的上边插入新的一行	Ins	插入/改写方式切换
Delete	删除光标处的字符	Ctrl+QY	从光标处删除到行尾
Ctrl+Y	删除光标所在的一行	Ctrl+T	删除光标所在处右边的词
Ctrl+KW	保存选中的内容到文件	Ctrl+KP	打印选中的内容
Ctrl+QF	查找文字	Ctrl+QA	查找并替换文字
Ctrl+F1	获得有关函数的帮助信息		

4. 保存 C 程序和改变当前目录

在英文 Turbo C++ 3.0 工作环境的 File 菜单中提供了一些命令, 用来保存 C 语言程序、改变默认目录和切换到 DOS 状态。简介如下:

(1) Save (保存): 选择该命令, 可以将当前的 C 语言程序以原来的名称保存, 如果是第 1 次保存文件, 则和执行 Save as (另存为) 命令的作用一样。

(2) Save as (另存为): 选择该命令, 会调出 Save File As (保存文件为) 对话框, 如图 1-2-9 所示。输入文件名称后, 单击 OK 按钮, 即可将当前的 C 语言程序保存在 Project 文件夹中。如果输入新路径和文件名称, 可将程序保存在指定的文件夹内。

(3) Save all (全部保存): 选择该命令, 可以保存当前所有打开的 C 语言程序。

(4) Change dir (改变目录): 选择该命令, 可以弹出 Change Directory (改变目录) 对话框, 如图 1-2-10 所示。利用该对话框可以显示当前目录, 也可以改变显示的目录。

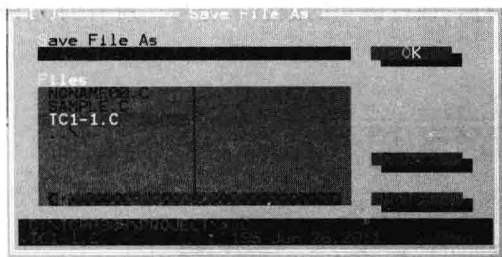


图 1-2-9 Save File As (保存文件为) 对话框

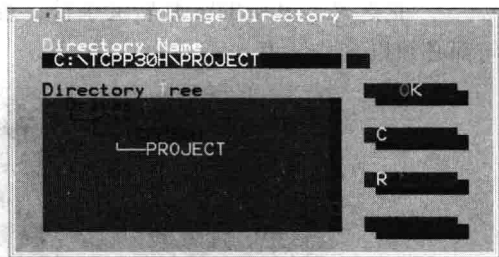


图 1-2-10 Change Directory (改变目录) 对话框