



高职高专计算机规划教材·案例教程系列

# C语言程序设计 案例教程（第二版）

沈大林 赵 玺 主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE



高职高专计算机规划教材·案例教程系列

# C 语言程序设计案例教程

## (第二版)

沈大林 赵 玺 主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

## 内 容 简 介

C 语言由于其强大的功能,在计算机的各个领域得到广泛应用。C 语言虽然是高级语言,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,是一种重要的程序设计语言。

全书共分 12 章,涵盖了 C 语言程序设计多方面的知识。本书讲解了 110 多个实例,结合知识点介绍了大量小例子,且每章提供多道思考与练习题。本书内容丰富、结构清晰、由浅及深、循序渐进、图文并茂,理论与实际制作相结合,学生不但能够快速入门,而且可以达到较高的编程水平。

本书适合作为高职高专院校计算机专业的教材,也可以作为高等院校非计算机专业的教材、初、中级培训班的教材,以及初学者的自学用书。

### 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. —2  
版. —北京: 中国铁道出版社, 2012. 5  
高职高专计算机规划教材. 案例教程系列  
ISBN 978-7-113-14526-2

I. ①C… II. ①沈… ②赵… III. ①C 语言—  
程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 067597 号

书 名: C 语言程序设计案例教程 (第二版)

作 者: 沈大林 赵 玺 主编

策 划: 秦绪好

读者热线: 400-668-0820

责任编辑: 祁 云 彭立辉

封面设计: 付 巍

封面制作: 刘 颖

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.51eds.com>

印 刷: 北京市昌平开拓印刷厂

版 次: 2007 年 6 月第 1 版      2012 年 5 月第 2 版      2012 年 5 月第 3 次印刷

开 本: 787mm×1092mm 1/16      印张: 17.5      字数: 415 千

印 数: 7 001~10 000 册

书 号: ISBN 978-7-113-14526-2

定 价: 33.00 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

## 高职高专计算机规划教材·案例教程系列

### 专家委员会

主任：邓泽民（教育部职业技术教育中心研究所）

委员：（按姓氏笔画排列）

丁桂芝（天津职业大学）

王首义（黑龙江司法警官职业学院）

邓凯（常州纺织服装职业技术学院）

艾伦（首都师范大学）

刘燕（无锡科技职业学院）

刘兴东（深圳职业技术学院）

李雪（安徽职业技术学院）

连为民（河南商业高等专科学校）

张举华（北京理工大学）

张恩祥（北京联合大学）

张晓云（西安航空技术高等专科学校）

杨云（济南铁道职业技术学院）

贺平（番禺职业技术学院）

洪小达（北京信息职业技术学院）

徐红（山东商业职业技术学院）

蒋川群（上海第二工业大学）

### 委员会

高职高专计算机规划教材·案例教程系列

编委会

编委会

主 编：沈大林 赵 奎

副主编：张晓蕾 魏雪英

编 委：（按姓氏笔画排列）

于建海 万 忠 马广月 马开颜  
王 玥 王 锦 王爱赫 王浩轩  
丰金兰 卢正明 邢芳芳 毕凌云  
曲彭生 朱 立 任心燕 刘 璐  
刘丛然 杜 金 杨 旭 杨素生  
杨继萍 肖柠朴 吴 飞 沈 昕  
沈建峰 张 伦 张士元 张凤红  
陈 炜 罗红霞 郑淑晖 孟昭勇  
郭 政 崔 玥

1982年大学毕业后,我开始从事职业教育工作。那是一个百废待兴的年代,是职业教育改革刚刚开始时期。开始进行职业教育时,我们使用的是大学本科纯理论性教材。后来,联合国教科文组织派遣具有多年职业教育研究和实践经验的专家来北京传授电子技术教学经验。专家抛开了我们事先准备好的教学大纲,发给每位听课教师一个实验器,边做实验边讲课,理论完全融于实验的过程中。这种教学方法使我耳目一新并为之震动。后来,我看了一本美国麻省理工学院的教材,前言中有一句话的大意是:“你是制作集成电路或设计电路的工程师吗?你不是!你是应用集成电路的工程师!那么你没有必要了解集成电路内部的工作原理,而只需要知道如何应用这些集成电路解决实际问题。”再后来,我学习了素有“万世师表”之称的陶行知先生“教学做合一”的教育思想,也了解这些思想源于他的老师——美国的教育家约翰·杜威的“从做中学”的教育思想。以后,我知道了美国哈佛大学也采用案例教学,中国台湾省的学者在讲演时也都采用案例教学……这些中外教育家的思想成为我不断探索职业教育教学方法和改革职业教育教材的思想基础,点点滴滴融入我编写的教材之中。现在我国职业教育又进入了一个高峰期,职业教育的又一个春天即将到来。

现在,职业教育类的大多数计算机教材应该是案例教程,这一点似乎已经没有太多的争议,但什么是真正的符合职业教育需求的案例教程呢?是不是有例子的教材就是案例教程呢?许多职业教育教材也有一些案例,但是这些案例与知识是分割的,仅是知识的一种解释。还有一些百例类丛书,虽然例子很多,但所涉及的知识 and 技能并不多,只是一些例子的无序堆积。

本丛书采用案例带动知识点的方法进行讲解,学生通过学习实例掌握软件的操作方法、操作技巧或程序设计方法。本丛书以每一节为一个单元,对知识点进行了细致的取舍和编排,按节细化知识点,并结合知识点介绍了相关的实例。本丛书的每节基本是由“案例描述”、“设计过程”、“相关知识”和“思考与练习”4部分组成。“案例描述”部分介绍了学习本案例的目的,包括案例效果、相关知识和技巧简介;“设计过程”部分介绍了实例的制作过程和技巧;“相关知识”部分介绍了与本案例有关的知识;“思考与练习”部分给出了与案例有关的拓展练习。读者可以边进行案例制作,边学习相关知识和技巧,轻松掌握软件的使用方法、使用技巧或程序设计方法。

本丛书的优点是符合教与学的规律,便于教学,不用教师去分解知识点和寻找案例,更像一个经过改革的课堂教学的详细教案。这种形式的教学有利于激发学生的学习兴趣,培养学生学习的主动性,并激发学生的创造性,能使学生在学习过程中充满成就感和富有探索精神,使学生更快地适应实际工作的需要。

本丛书还存在许多有待改进之处,可以使它更符合“能力本位”的基本原则,可以使知识的讲述更精要明了,使案例更精彩和更具有实用性,使案例带动的知识点和技巧更多,使案例与知识点的结合更完美,使习题更具趣味性……这些都是我们继续努力的方向,也诚恳地欢迎每一位读者,尤其是教师和学生参与进来,期待你们提出更多的意见和建议,提供更好的案例,成为本丛书的作者,成为我们中的一员。

# 第二版前言

FOREWORD

C 语言由于其强大的功能，在计算机的各个领域得到广泛的应用，从 UNIX、DOS 到 Windows、Linux 都使用了 C 语言来进行设计。C 语言虽然是高级语言，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，因此也被称为“中级语言”。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。

全书共分 12 章，涵盖了 C 语言程序设计多方面的知识。第 0 章“绪言”，介绍了 C 语言基本概念、Turbo C 3.0 和 Visual C++ 6.0 开发环境的使用，及本书课程安排；第 1 章介绍了 C 语言程序的基本元素运算符和表达式等内容；第 2 章结合 15 个案例介绍了数据的输出与输入函数的使用方法；第 3 章结合 12 个案例介绍了程序的基本结构和算法、条件分支语句和 switch 开关分支语句的使用方法；第 4 章结合 20 个案例介绍了循环结构程序的设计方法；第 5 章结合 8 个案例介绍了函数的定义与调用，以及函数参数的传递方法；第 6 章结合 11 个案例介绍了标准函数应用、函数的嵌套与递归调用，以及变量的作用域和存储类型；第 7 章结合 13 个案例介绍了数值数组、字符数组的定义与调用方法；第 8 章结合 11 个案例介绍了数组指针、字符指针和函数指针的使用方法；第 9 章结合 7 个案例介绍了共用体和枚举的应用；第 10 章结合 7 个案例介绍了宏定义、文件包含、条件编译和位运算；第 11 章结合 9 个案例介绍了数据文件基本概念、文件的检测与输入/输出函数、文件的定位操作等内容。

本书具有较大的信息量，讲解了 110 多个实例，结合知识点介绍了大量小例子，并提供了 100 多道思考与练习题。每个实例均由实例效果、技术分析和程序解析组成。本书以实例带动知识点的学习，通过学习实例掌握程序设计的方法和技巧，由浅至深，层层引导，能够让学生快速掌握 C 语言，提高编程能力。

本书内容丰富、结构清晰、由浅及深、循序渐进、图文并茂，理论与实际制作相结合，学生不但能够快速入门，而且可以达到较高的编程水平。

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，参加本书编写的有：许崇、陶宁、张秋、杨旭、王浩轩。对本书的出版工作提供了帮助的有沈昕、张伦、王爱赅、万忠、郑淑暉、曾昊、肖柠朴、沈建峰、郑鹤、郭海、陈恺硕、郝侠、丰金兰、袁柳、徐晓雅、王加伟、孔凡奇、卢贺、李宇辰、靳轲、苏飞、王小兵等，在此一并表示感谢。

本书适合作为高职高专院校计算机专业的教材，也可以作为高等院校非计算机专业的教材、初、中级培训班的教材，以及初学者的自学用书。

由于时间仓促，编者水平有限，书中难免有偏漏和不妥之处，恳请广大读者批评指正。

编者

2012 年 3 月

# 第一版前言

C 语言由于其强大的功能，在计算机的各个领域内得到广泛的应用，从 UNIX、DOS 到 Windows、Linux 都使用了 C 语言来进行设计。C 语言虽然是高级语言，但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，因此也被称为“中间语言”。

本书采用案例带动知识点的方法进行讲解，学生通过学习实例，掌握 C 语言程序设计的基本方法和编程技巧。本书以一节为一个单元，对知识点进行了细致的取舍和编排，按节细化知识点并结合知识点介绍了相关的实例，将知识和案例放在同一节中，知识和案例相结合。本书基本是每节由“案例效果”、“设计过程”和“相关知识”组成。“案例效果”中介绍了学习本案例的目的，包括案例效果、相关知识和技巧简介；“设计过程”中介绍了实例的制作过程和技巧；“相关知识”中介绍了与本案例有关的知识。读者可以边进行案例制作，边学习相关知识和技巧，轻松掌握 C 语言程序设计的基本方法和编程技巧。

全书共分 9 章，涵盖了 C 语言程序设计多方面的知识。第 0 章为绪论，介绍了 C 语言的基本概念以及 C 语言程序开发环境的使用和本书课程安排。第 1 章为 C 语言程序设计基础，主要讲解 C 语言的数据类型、表达式、标准输入/输出语句等内容。第 2 章为算法与程序流程控制，主要讲解程序设计的基本算法，以及顺序、选择、循环三大流程控制结构。第 3 章为数组与字符串，主要讲解数组与字符串的应用、字符串处理等内容。第 4 章为指针，简要介绍了 C 语言中的指针类型，重点讲解指针的概念及使用。第 5 章为函数，讲解了函数的概念、定义及应用，main() 函数的命令行参数，以及标准 C 语言函数的应用。第 6 章为结构体、共用体与枚举，重点介绍了结构体在程序设计中的应用、链表等内容。第 7 章为文件访问，介绍了文件的概念，以及文本文件、随机文件的访问及操作。第 8 章为编译预处理，介绍了宏定义、文件包含、编译预处理等各方面的内容。

最后的附录主要包括一些函数的说明、Turbo C 2.0 程序开发环境介绍等内容。

本书具有较大的知识信息量，从程序设计的基础知识、算法与程序流程控制到最后的文件操作与编译预处理，通过 34 个案例，约 80 个实例的分析讲解，再利用 100 余道习题的练习与巩固，可以使学生快速掌握 C 语言。本书内容丰富、结构清晰、图文并茂，易于教学与个人自学。

本书可以作为高等院校非计算机专业教材，也可作为高职高专院校计算机专业教材，还适合作为初学者的自学用书。

由于编者水平有限，加上时间仓促，书中难免有疏漏和不妥之处，恳请广大读者批评指正。

编者

编者

2007年3月

2007年3月



# 目 录

第 0 章 绪言 ..... 1

0.1 C 语言概述 ..... 1

0.1.1 计算机语言的发展 ..... 1

0.1.2 C 语言的由来和特点 ..... 1

0.1.3 C 语言程序开发流程 ..... 3

0.2 C 语言集成开发环境 ..... 3

0.2.1 使用 Turbo C 3.0 开发 C 程序 ..... 3

0.2.2 使用 Visual C++ 6.0 开发 C 程序 ..... 8

0.2.3 调试程序中常见的错误 ..... 12

0.3 程序设计的良好习惯 ..... 14

0.3.1 程序的注释和布局 ..... 15

0.3.2 C 语言程序代码的格式 ..... 15

0.4 教学方法和课程安排 ..... 17

思考与练习 ..... 18

第 1 章 C 语言程序设计基础 ..... 20

1.1 C 语言程序的基本元素 ..... 20

1.1.1 标识符和数据类型 ..... 20

1.1.2 常量与变量 ..... 21

1.2 程序中的运算 ..... 24

1.2.1 算术运算 ..... 24

1.2.2 关系运算 ..... 25

1.2.3 逻辑运算 ..... 26

1.2.4 其他运算 ..... 27

1.2.5 运算中数据类型的转换 ..... 29

思考与练习 ..... 30

第 2 章 数据的输出与输入 ..... 33

2.1 格式化输出函数 ..... 33

2.1.1 格式化输出函数的格式、功能与修饰符 ..... 33

2.1.2 格式化输出实例 ..... 35

2.2 字符输入/输出函数 ..... 38

2.2.1 字符输入/输出函数的格式与功能 ..... 38

2.2.2 字符输入/输出函数实例 ..... 39

2.3 格式化输入函数 ..... 40

2.3.1 格式化输入函数的格式、功能与使用说明 ..... 40

2.3.2 格式化输入实例 ..... 42

思考与练习 ..... 47

第 3 章 算法和程序的选择结构 ..... 52

3.1 程序的基本结构和算法 ..... 52

3.1.1 程序的基本结构和语句 ..... 52

3.1.2 算法 ..... 54

3.2 条件分支语句 ..... 57

3.2.1 if 单分支语句 ..... 57

3.2.2 if...else 双分支语句 ..... 59

3.2.3 if...else if...else 多分支语句 ..... 63

3.3 switch 开关分支语句和选择结构的嵌套 ..... 66

3.3.1 switch 开关分支语句 ..... 66

3.3.2 选择结构的嵌套 ..... 72

思考与练习 ..... 76

第 4 章 循环结构程序设计 ..... 82

4.1 循环结构 ..... 82

4.1.1 for 语句 ..... 82

4.1.2 while 语句 ..... 88

4.1.3 do...while 语句 ..... 90

4.2 循环嵌套及中断和转向语句 ..... 94

4.2.1 循环嵌套 ..... 94

4.2.2 break 和 continue 语句 ..... 98

4.2.3 goto 转向语句 ..... 102

思考与练习 ..... 105

第 5 章 函数定义和参数传递 ..... 112

5.1 函数的定义与调用 ..... 112

5.1.1 函数概述 ..... 112

5.1.2 定义函数 ..... 114

5.1.3 函数声明和调用 ..... 115

5.1.4	应用实例 .....	117	8.1.4	指针的运算 .....	181
5.2	函数参数传递 .....	120	8.1.5	应用实例 .....	183
5.2.1	函数参数和函数返回 .....	121	8.2	数组指针、字符指针和函数 指针 .....	186
5.2.2	函数间参数的传递方式 .....	123	8.2.1	数组指针 .....	187
5.2.3	应用实例 .....	124	8.2.2	字符指针和函数指针 .....	189
思考与练习 .....	127	8.2.3	运算符**和指针数组 .....	191	
第 6 章	函数应用和变量作用域 .....	129	8.2.4	应用实例 .....	194
6.1	标准函数应用 .....	129	思考与练习 .....	198	
6.1.1	标准函数简介 .....	129	第 9 章	复杂数据类型 .....	201
6.1.2	验证程序和应用实例 .....	129	9.1	结构体 .....	201
6.2	函数的嵌套与递归调用 .....	135	9.1.1	结构体的定义 .....	201
6.2.1	函数嵌套调用 .....	135	9.1.2	结构体变量的引用 .....	205
6.2.2	函数的递归调用 .....	136	9.1.3	结构体数组和自定义 数据类型 .....	207
6.2.3	应用实例 .....	138	9.1.4	链表 .....	209
6.3	变量的作用域和存储类型 .....	141	9.1.5	应用实例 .....	211
6.3.1	变量的作用域 .....	141	9.2	共用体和枚举 .....	217
6.3.2	变量存储类型 .....	142	9.2.1	共用体 .....	217
6.3.3	验证程序和应用实例 .....	144	9.2.2	枚举 .....	219
思考与练习 .....	147	9.2.3	应用实例 .....	221	
第 7 章	数组与字符串 .....	148	思考与练习 .....	228	
7.1	数值型一维数组 .....	148	第 10 章	编译预处理 .....	230
7.1.1	一维数组定义和初始化 .....	148	10.1	宏定义 .....	230
7.1.2	数值型一维数组的使用 和元素地址 .....	149	10.1.1	两种宏定义 .....	230
7.1.3	应用实例 .....	151	10.1.2	宏定义的作用范围 .....	231
7.2	数值型多维数组 .....	155	10.1.3	宏定义与函数的区别 .....	232
7.2.1	二维数组的定义 .....	155	10.1.4	应用实例 .....	232
7.2.2	多维数组的定义和使用 .....	157	10.2	文件包含和条件编译 .....	234
7.2.3	应用实例 .....	157	10.2.1	文件包含 .....	234
7.3	字符数组 .....	163	10.2.2	条件编译语句 .....	235
7.3.1	字符数组的定义和使用 .....	163	10.2.3	应用实例 .....	236
7.3.2	字符串和字符处理函数 .....	166	10.3	位运算 .....	239
7.3.3	应用实例 .....	169	10.3.1	位运算符 .....	239
思考与练习 .....	172	10.3.2	位段 .....	243	
第 8 章	指针 .....	177	10.3.3	应用实例 .....	245
8.1	指针的定义与应用 .....	177	思考与练习 .....	246	
8.1.1	指针概述 .....	177	第 11 章	文件 .....	248
8.1.2	指针定义 .....	178	11.1	数据文件基本概念 .....	248
8.1.3	引用指针变量 .....	179			

11.1.1 文件分类.....	248	11.2.2 文件的输入/输出函数 .....	253
11.1.2 文件与指针 .....	249	11.2.3 应用实例 .....	254
11.1.3 文件系统.....	249	11.3 文件的定位操作 .....	259
11.1.4 文件的打开与关闭 .....	250	11.3.1 文件指针位置的 设置和获取 .....	259
11.1.5 应用实例 .....	251	11.3.2 应用实例 .....	260
11.2 文件的检测与输入/输出函数...	252	思考与练习 .....	264
11.2.1 文件的检测函数 .....	252		

## 第0章 绪言

**【本章提要】**本章简要介绍了C语言的历史和特点、C语言集成开发环境 Turbo C 3.0 与 Microsoft Visual C++ 6.0 开发环境,以及C语言程序的格式和结构。另外,还介绍了使用 Turbo C 3.0 和 Visual C++ 6.0 开发C程序的方法,为全书的学习打下基础。

### 0.1 C语言概述

C语言是 Combined Language(组合语言)的简称,是一种计算机程序设计语言。随着计算机的迅速发展和广泛应用,C语言已经成为目前最流行的计算机语言之一。

#### 0.1.1 计算机语言的发展

按照计算机语言的使用和发展,又将其分为以下3个时代:

(1)第一代计算机语言使用机器语言(Machine Language),也就是计算机中央处理器(CPU)本身所使用的语言。机器语言可以直接被CPU使用,所以表达准确、运行速度非常快。但是,由十六进制数字组成的机器语言对大多数人来说很难编写、阅读和理解。

(2)第二代计算机语言使用汇编语言(Assembly Language),它的出现使编写计算机程序变得容易。有表达意义的命令名称代替了一组组简单的数字,语句的出现也使程序内容变得清楚,易于理解。但是,计算机只能读懂机器语言,所以用汇编语言编写的程序要先用汇编程序(Assembler)翻译成机器语言,CPU才能运行。如今,汇编语言被计算机高级专业人员广泛使用,但是汇编语言需要用许多条语句去完成一个极其简单的任务,编写过程烦琐且耗时长,所以它很难普及到一般使用者。

(3)第三代计算机语言使用高级语言(High-Level Language)。它使每一条语句的功能大大加强,同时有易写、易读和易于理解的特性。通过对高级语言的学习,编写计算机语言对普通人来说终于不再是一座不可征服的高山,但是对于计算机的CPU来说,高级语言太笼统太简单了,它需要一个功能强大的翻译器来帮助理解。

C语言也常被看做是一种中级语言,它既具有高级语言的特点,又具有汇编语言的特点。

#### 0.1.2 C语言的由来和特点

##### 1. C语言的由来

C语言是一种面向过程的计算机程序设计语言,它是目前众多计算机语言中最优秀的结构程序设计语言之一。

1970年,美国贝尔实验室的程序员 Ken Thompson 在 BCPL 的基础上开发出了 B 语言,并用 B 语言编写了第一个 UNIX 操作系统。1972年,贝尔实验室的另一位程序员 D.M. Ritchie 在 B 语言的基础上进行了重新改写并命名为 C 语言。之后,二人又联手用 C 语言重写了 UNIX 操作系统,大大提高了其可移植性与兼容性,并创造了使用非汇编语言编写操作系统的先河。1978年,贝尔实验室正式发表了 C 语言,并广泛地应用到大、中、小及微型计算机上。如今,C 语言以其简练、灵活、功能强大、高效等优点闻名于世,特别是在系统软件开发领域,许多著名的系统软件(比如 DBASE IV)都是用 C 语言编写的。用 C 语言加上一些汇编语言子程序更能凸显 C 语言的优势,例如 PC-DOS、WORDSTAR 等就是用这种方法编写的。此外,C 语言在编写二维、三维图形和动画等方面,也深受人们的欢迎。

## 2. C 语言的特点

C 语言与其他许多语言相比,具有如下特点:

(1) C 语言是一种“中级语言”,它把高级语言的基本结构和语句与低级语言的实用性结合起来。编程语言的“高级”、“中级”、“低级”并不是标志语言解决问题能力的高低,而是与计算机硬件联系的程度。C 语言不仅有高级语言的可读性好、可移植性好、容易学习等特点,还兼有汇编语言运行效率高、可直接控制计算机硬件的特点。C 语言可以和汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

(2) C 语言是结构式语言。结构式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、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可方便地调用,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) C 语言功能齐全。C 语言具有各种各样的数据类型,并引入了指针概念使程序效率更高。C 语言还具有强大的图形功能,支持多种显示器和驱动器。此外,C 语言的计算功能、逻辑判断功能也比较强大。

(4) C 语言自由度大。C 语言的编程自由度大,语法限制少,例如对数组下标出界、函数参数虚实转换不做检查。变量类型的使用也比较灵活,例如整型、字符型和逻辑型数据在特定条件下可以通用等,这大大方便了编程人员。

(5) C 语言可移植性强。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,而这些函数是由独立于 C 语言的系统程序模块库所提供的。所以,使用 C 语言编写的程序可以在硬件不同的机种之间被移植。

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。目前,使用最多的计算机操作系统 Windows 系列、Linux 系列等就有相当多的部分是用 C 语言编写的。可以这样说,在程序设计中,只要能想得到,几乎就没有 C 语言做不到的。

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,对将来学习其他程序设计相关课程,具有重要的意义。

### 0.1.3 C 语言程序开发流程

使用 C 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 程序源代码的编辑可以由专门的代码编辑器来完成，也可用普通的纯文本编辑器来进行，比如 Windows 中的记事本。由于计算机只能识别二进制代码指令，源代码不能被计算机所识别和执行。为了使其能够被计算机所执行，必须对其进行编译，将其转换为二进制指令。从源文件到可执行程序中经过的流程如图 0-1 所示。

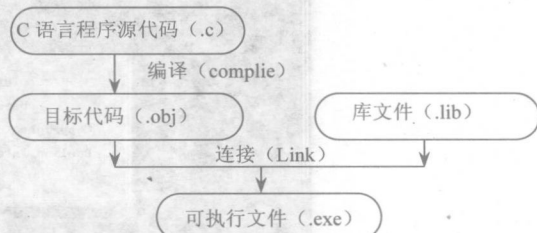


图 0-1 程序编译流程

将 C 语言源代码转化为二进制指令的过程称为编译 (Compile)，C 语言的编译需要有专门的编译器来执行。编译完成的二进制代码文件称为目标文件，其扩展名为 .obj。目标文件也不能直接在计算机中执行，还需要通过连接 (Link) 程序将它与 C 语言的库文件进行连接，最后生成可执行程序文件，其扩展名为 .exe。现在，通常把负责代码编辑的编辑器、编译目标代码的编译器与连接库文件生成目标文件的连接器以及在程序中对程序进行调试的程序综合起来，组成一个软件，称为集成开发环境 (IDE)。

## 0.2 C 语言集成开发环境

C 语言集成开发环境是指一类具有用户界面，旨在帮助用户更快捷、更方便地运用 C 语言各种功能的软件。许多公司都推出了自己的 C 语言开发工具，虽然它们的集成开发环境不尽相同，侧重点也不一样，但是在 C 语言基本应用上是一致的。这里介绍两种比较常用的集成开发环境：Borland 公司的 Turbo C 3.0 系列以及 Microsoft 公司的 Visual C++ 系列（早期为 Microsoft C/C++）集成开发环境。在这两种不同的集成开发环境下，本书涉及的大部分内容是相同的，程序运行结果也是一样的，但是变量长度是不一样的。例如，整型变量占用的内存单元数，在 Turbo C 3.0 系列集成开发环境下是 2，在 Visual C++ 系列集成开发环境下是 4。另外，在 32 位和 64 位两种 Windows 系统时，Visual C++ 的编译器也会改变字节数，所以 Visual C++ 不是跨平台的，更换系统需要重新编译。

### 0.2.1 使用 Turbo C 3.0 开发 C 程序

Turbo C 以其编译的速度快、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。Turbo C 系列中广泛使用的是 Turbo C 3.0，它是 Turbo C 2.0 的升级版本。

Turbo C 3.0 (以下简称 TC 3.0) 是在 DOS 下运行的程序，但也可以运行在 Windows 98/2000/XP 等系统下（在“命令提示符”窗口中运行）。Turbo C 3.0 的安装极为简单，只要插入安装盘，在 DOS 提示符下输入 Install，再按 Enter 键，接下来按提示信息进行操作就能够完成。

#### 1. Turbo C 3.0 集成开发环境简介

在 Turbo C 3.0 安装完成后，在“命令提示符”窗口中，改变路径提示符到安装后的文件

夹下的 bin 目录内, 然后输入 Turbo C 3.0 的可执行文件名 (例如, TC), 按 Enter 键, 即可启动 Turbo C 3.0, Turbo C 3.0 的启动界面如图 0-2 所示。

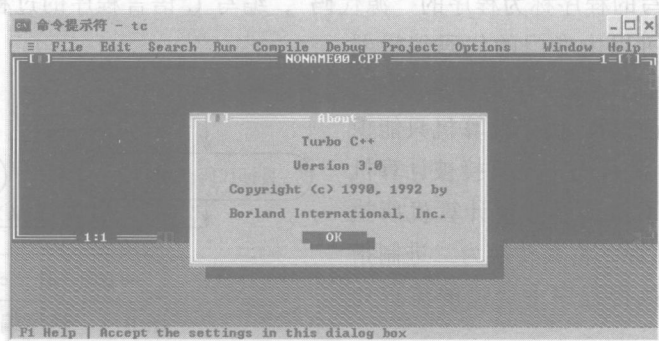


图 0-2 Turbo C 3.0 的启动界面

单击 OK 按钮, 进入 Turbo C 3.0 工作环境, 可以看到在程序编辑窗口中有光标在左上角闪烁, 此时就可以进行程序的编辑。

在 Turbo C 3.0 工作环境的最上方是 Turbo C 3.0 的菜单区, 菜单中有程序编辑、编译、调试以及环境设置的各种命令。下面简单介绍几种最常用的菜单命令及其子菜单命令。

(1) 在 File (文件) 菜单中, New (新建) 命令用来建立一个新的文件, 默认文件名为 NONAME00.CPP, 在保存文件时可以更改文件名; Open (打开) 命令用来打开一个现存的文件; Save 命令用来保存编写的文件, 如果文件名是 NONAME00.CPP, 则会询问是否更改文件名; DOS shell 命令用来暂时退出 Turbo C 3.0 回到 DOS 提示符下, 此时可以运行 DOS 命令, 如果想回到 Turbo C 3.0 中, 在 DOS 状态下输入 EXIT 即可; Quit (退出) 命令用来退出 Turbo C 3.0, 返回到 DOS 操作系统中。

(2) 在 Edit (编辑) 菜单中, Undo (取消) 命令用来取消上一步操作; Redo (恢复) 命令用来恢复上一步取消的操作; Cut (剪切) 命令用来剪切选中的代码内容; Copy (复制) 命令用来复制选中的代码内容; Paste (粘贴) 命令用来粘贴选中的代码内容; Clear (清除) 命令用来清除选中的代码内容。

(3) 在 Search (搜索) 菜单中, Find (查找) 命令用来在代码中查找指定内容; Replace (替换) 命令用来在代码中替换指定内容; Go to line number (定位到行) 命令用来定位光标到指定行。

(4) 在 Run (运行) 菜单中, Run 菜单命令用来运行当前程序; Program reset (程序重启) 命令用来中止当前的调试, 释放分给程序的空间; Go to cursor (执行到光标处) 命令在调试程序时使用, 可使程序运行到光标所在行, 光标所在行必须为一条可执行语句, 否则提示错误。

(5) 在 Compile (编译) 菜单中, Compile (编译) 命令用来编译当前程序; Make (生成) 命令用来生成一个 .EXE 文件, 并显示生成的 .EXE 文件名; Link 命令用来把当前 .OBJ 文件及库文件连接在一起生成 .EXE 文件; Build all 命令用来重新编译项目里的所有文件, 并进行装配生成 .EXE 文件。

菜单区下面是程序的编辑区域, 也称为编辑窗口, 在编辑窗口上方中间显示程序的名称, 左下方显示光标所在位置的行号和列号。

编辑窗口的下方是 Message (信息) 窗口, 在编译程序时, 该窗口显示相关的编译信息。界面的最下方是编辑时最常用的快捷键, 例如, F1 显示帮助、F9 编译程序、F10 激活菜单等。

## 2. 程序开发步骤

使用 TC 3.0 开发 C 语言程序的步骤如下:

(1) 输入和编辑程序: 进入 TC 3.0 界面后, 选择 File (文件) 菜单中的 New (新建) 命令, 创建一个新文件, 此时就可以在编辑窗口中输入和编辑程序代码。例如, 在编辑窗口中输入如下代码:

```
/* TC 3.0 My First C Program */
#include "stdio.h"
void main()
{
    printf("Good Morning!\n");
}
```

下面对这些程序代码进行简单说明。

- ◎ 代码的第一行是程序文件头, 对程序做相关说明。“/\* ... \*/” 符号表示这一部分为注释性文字, 不作为程序代码运行, 在程序编译时会被忽略。“/\*” 与 “\*/” 必须成对出现, 两者之间的所有字符 (可以是多行) 均为注释文字。
- ◎ 第二行以 #include 开始, 这是程序的预处理命令 (在以后的章节将对它作详细解释), 引用了 C 语言的标准库 stdio.h, 使下面的 printf() 语句能够得以顺利执行。
- ◎ C 语言源程序的基本单位是函数, 第三行的 main() 是 C 语言程序的主函数, 每个 C 语言程序有且仅有一个主函数, 所有的 C 语言程序都是从这里开始执行。main() 函数前面的 void 表示该函数没有返回值。
- ◎ 第四行的左大括号 “{” 与最后一行的右大括号 “}” 之间是函数的主体, 它们必须成对出现。
- ◎ 第五行函数体中使用了 C 语言的输出函数 printf() 来输出字符串 "Good Morning!"。在 C 语言中, 字符串都必须用半角双引号括起来, \n 是转义字符, 表示换行。除编译预处理语句 (以 # 开头的特殊语句, 如前面的 #include "stdio.h") 外, 所有的 C 语言的语句都必须以分号 “;” 表示语句的结束。

如果安装了 Turbo C 3.0 中文版, 可使用 “代码编辑器” 软件来输入和编辑 C 程序。

(2) 保存程序: 选择 File (文件) → Save (保存) 命令, 或者按 F2 快捷键, 弹出 Save File As (文件另存为) 窗口, 默认的保持路径是 DOS 系统当前路径, 默认文件名为 NONAME00.CPP, 如图 0-3 所示。

输入文件名 0-1.CPP 后单击 OK 按钮, 此时编辑窗口上方的文件名由 NONAME00.CPP 改变为 0-1.CPP, 如图 0-4 所示。

**注意:** 在编译程序前最好先保存程序, 避免因错误引起的程序崩溃而丢失文件。

(3) 编译和连接程序: 选择 Compile (编译) → Make (生成) 命令或者按 F9 快捷键, 集成开发环境将先把程序编译为目标文件, 然后连接为可执行文件, 此时会出现如图 0-4 所示的信息窗口, 提示编译成功。编译后得到的可执行文件为 0-1.EXE。



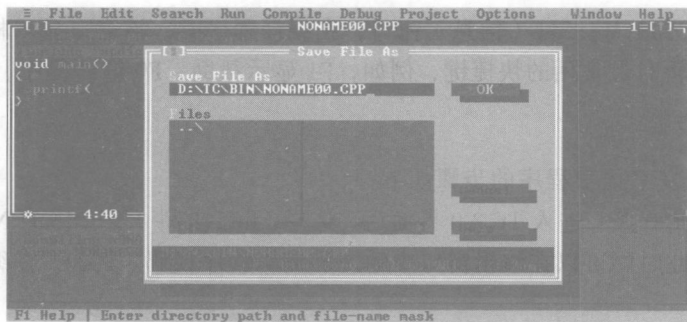


图 0-3 Save File As (文件另存为) 窗口

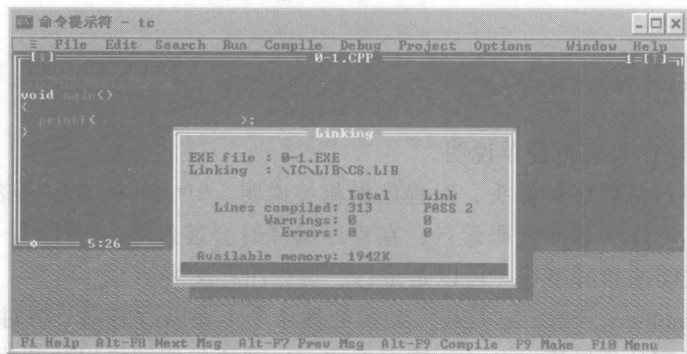


图 0-4 将源文件编译为可执行文件

如果程序有错误,此时将弹出提示编译出错的信息窗口,并在屏幕下方的 Message (信息) 窗口中显示相关的错误信息。

(4) 运行程序:选择 Run (运行)→Run (运行)命令,或按 Ctrl+F9 组合键,虽然界面没有任何变化,但此时程序已执行完成。

然后,选择 Window (窗口)→User screen (用户屏幕)命令,或按 Alt+F5 组合键可以将屏幕切换到用户屏幕。这时,可以看到如图 0-5 所示的程序运行结果。

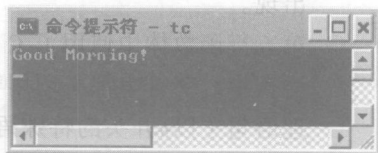


图 0-5 程序 0-1.CPP 的运行结果

在用户屏幕按任意键可以回到编辑窗口。

现在,这个简单的 C 语言程序就设计完成了,接下来看看在代码编辑和编译与连接过程中生成了哪些文件。选择 File (文件)→Quit (退出)命令,或按 Alt+X 组合键,退出 Turbo C 3.0 集成开发环境。在 DOS 提示符下,输入 dir 0-1.\*命令,按 Enter 键,可以看到如图 0-6 所示的文件列表。

可以看到有 3 个名为 0-1 的文件,这 3 个文件都是在程序设计中创建的。其中,0-1.CPP 为 C 语言程序的源文件,也就是前面输入的程序代码;0-1.OBJ 为目标文件,它包含的是编译器所生成的二进制机器指令代码;0-1.EXE 是 Turbo C 所生成的可执行文件,它包括了 0-1.OBJ 中的代码以及在程序连接时从相关运行库所得到的必需的支持过程。

现在,就得到了一个可执行程序 0-1.EXE,这个程序可以直接在 DOS 下执行,在 DOS 提示符后输入“0-1”,再按 Enter 键就可以执行。