

高等学校非计算机专业计算机教育系列教材

C 语言程序设计

楼 静 主编

重庆大学出版社

内 容 提 要

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。

本套教材由用作课堂教学的《C语言程序设计》和用作课后练习与上机指导的《习题集与上机实习指导》两书组成。《C语言程序设计》全书共12章,详细介绍了C语言的基础知识及程序设计方法。主要内容包括:C程序的结构;数据类型、运算符和表达式;数据的输入和输出;控制结构;数组;函数;编译预处理;指针;结构和共用体;文件;位运算;C语言在数据结构中的应用。《习题集与上机实习指导》分为与教学内容对应的12个单元,每个单元包括内容小结、习题、实习内容等,可直接作为学生的课外作业本和上机实习作业本,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,提高学习效果和学习效率。

本书可作为大专院校各专业学生的C语言程序设计教材,也可作为有关人员的自学用书或计算机等级考试(二级C)的参考书。

本教材配有适合多媒体教学的电子课件。

图书在版编目(CIP)

C语言程序设计/楼静主编. —重庆:重庆大学出版社,2001.2

ISBN 7-5624-2289-3

.C... 楼... .C语言—程序设计 .TP312

中国版本图书馆CIP数据核字(2001)第02921号

C语言程序设计

楼 静 主编

责任编辑:曾令维 版式设计:曾令维

责任校对:任卓惠 责任印制:张永洋

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街174号重庆大学(A区)内

邮编:400044

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn(市场营销部)

全国新华书店经销

重庆大学印刷厂印刷

*

开本:787×1092 1/16 印张:14.5 字数:362千

2001年2月第1版 2003年1月第2次印刷

印数:8 501—12 500

ISBN 7-5624-2289-3/TP·273 定价:20.00元

本书如有印刷、装订等质量问题,本社负责调换

版权所有 翻印必究

《高等学校非计算机专业
计算机教育系列教材》编委会

主任:周 荣

副主任:宋焕斌 张怀宁 张启富

委员:(以姓氏笔画为序)

王建良 王建华 车文华

张民坤 庞淑英 郑尚英

秦卫平 耿植林 楼 静

序 言

随着计算机技术和信息技术的飞速发展及与其他学科领域的交叉融合,以计算机技术为基础的信息技术的发展水平、应用水平和教育水平已经成为衡量社会进步程度的重要标志。计算机的广泛普及与应用,使我们传统的工作、学习、生活和思维方式发生了巨大变化。计算机基础已成为面向 21 世纪人才培养方案中必不可少的、最重要的基础之一。计算机基础教育既是文化教育、人才素质教育,又是强有力的技术基础教育。大力发展计算机基础教育已成为信息化社会的需要和各学科发展的需要。

近几年来,面对挑战与机遇并存的发展形势,各高等学校都把计算机基础教学纳入重要的基础课程进行教学,积极探索计算机基础教学的方法、制订基本目标并确立了多层次教学的总体框架与实施方案。在优化计算机基础教学的课程体系结构、制订教学内容、改革教学模式和方法、强化实践教学、改善教学手段、加强基础设施建设、教材建设和师资队伍建设等方面,各高等学校的广大计算机教育工作者做了大量工作,取得了可喜的成果,使非计算机专业学生的计算机应用开发能力及计算机文化素质有了很大的提高。

为适应计算机技术的迅速发展与计算机应用模式的变化,推动教育改革与教学改革的深入发展,完善多层次教学方案,培养学生在本专业与相关领域中的计算机应用和开发能力,培养学生主动利用计算机处理、解决各种问题的意识,培养学生自己学习计算机新技术的意识与能力,使多层次教育的重点向二、多层次推进,使非计算机专业学生的计算机教育跃上新的台阶,昆明理工大学在认真总结前几年教学经验的基础上,组织专门从事非计算机专业学生计算机基础教学并具有丰富的教学经验和教学研究成果的教师编写了本系列教材。目前已编写完成《C 语言程序设计》、《数据库应用基础》、《计算机网络应用——网页设计及网站建设》、《微机组装及维护技术》等,我们将继续跟踪计算机应用发展的新动向,不断充实内容,更新教材,将重点放在计算机教育第二、三层次的教学内容上,不断推出新书,使本系列教材内容涵盖面更广,内容更新、更深,更具有特色。

本系列教材内容丰富、新颖、完整,凝结了我们多年教学实践的经验与智慧,是一套系统化很强的非计算机专业学生计算机基础教学的教材。它不仅适合普通高等学校非计算机专业学生计算机基础教学使用,也可作为成人教育和各种专门培训班及各行各业人员学习计算机基础知识与技能的教学与自学教材。

由于本系列教材的改革幅度较大,编者水平有限,加上时间紧迫,其中肯定会存在不足或错误,诚恳希望同行教师、专家和广大读者不吝赐教。

借此机会,我们谨向所有关心和支持本系列教材的领导、同事及朋友们表示最诚挚的感谢!

《高等学校非计算机专业计算机教育系列教材》编委会

2000 年 12 月

前 言

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。本教材的参考学时为 60 学时,其中上课 30 学时,上机 30 学时,以体现“精讲多练”的教学方法。与教材配套的还有适合多媒体教学的电子课件及用作课后练习的《习题集与上机实习指导》一书,该书可直接作为学生的课外作业本和上机实习作业本。本教材在编写中具有以下特点:

1. 本书是为初学程序设计者编写的,注重构建学生对 C 语言的整体认识,注意培养学生阅读程序和编写程序的能力。

2. C 语言涉及到的概念比较复杂,规则繁多,使用灵活,较难掌握,因此,本书只介绍 C 语言的最基本部分,并在内容编排上循序渐进,深入浅出,易于掌握;在文字叙述上力求语言简洁,通俗易懂,层次分明,重点突出。

3. 书中内容可分为三个教学阶段。第一阶段包括第 1 至第 4 章,主要介绍 C 语言的基础知识、数据的输入输出及控制结构。学完这些内容之后,学生应该熟悉 C 语言的语法规则,了解 C 程序的结构,掌握基本类型数据的使用方法及其输入输出,能熟练使用 C 语言提供的控制语句编制程序,能使用 Turbo C 2.0 集成软件调试程序。第二阶段包括第 5 至第 7 章,主要介绍数组、函数及编译预处理。学完这些内容之后,学生应该具备初步的阅读程序的能力,并能编写、调试和运行一些简单的应用程序,了解一些常用的算法。第三阶段包括第 8 至第 12 章,主要介绍指针、结构和共用体、文件、位运算及 C 语言在数据结构中的应用。这一阶段的学习内容是 C 语言的难点,又是 C 语言的特色,学习完这些内容后,学生应该具备较强的阅读程序的能力,并能运用指针、高级数据结构和编程技术来编制较复杂的应用程序。

4. 为了学以致用,在选择例题和习题时尽量多选与 C 语言各应用领域有关的题,书中仅选用了一些最常用、最基本的算法。本书的所有例题均在 Turbo C 2.0 编译环境下调试通过。

5. 《习题集与上机实习指导》一书有较强的针对性,能使学生在上机实习和课后练习中目标明确,思路清晰;有利于学生加深理解和掌握所学的概念,巩固所学知识;有利于培养和提高学生编写程序和调试程序的能力;有利于提高学习效率,增强学习效果。

本书共分 12 章。第 1 章、第 2 章由张启富编写,第 3 章、第 11 章由余鹏程编写,第 4 章由楼静编写,第 5 章由汪俊清编写,第 6 章、第 7 章由李晶编写,第 8 章、第 9 章由耿植林编写,第 10 章、第 12 章由普运伟编写。全书由楼静主编,并负责部分章节的改写及全书的统稿工作。

本书在编写过程中,得到了昆明理工大学副校长周荣教授的关心和支持,同时还得到昆明理工大学教务处、教材科及计算中心领导和同志们的支持、关心和帮助。在此,对他们一并表示衷心的感谢!同时对本书所用参考书籍、资料的作者们表示诚挚感谢!

由于作者水平有限,缺乏经验,书中肯定会有不少错误和缺点,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

2000 年 12 月

目 录

第 1 章 C 程序的结构	1
1.1 C 语言简介	1
1.2 C 语言程序结构和书写格式	2
1.3 C 语言程序的编辑和运行	4
1.4 Turbo C 简介	5
第 2 章 数据类型、运算符和表达式	7
2.1 数据类型及存储形式	7
2.1.1 整型数据	7
2.1.2 字符型数据	8
2.1.3 实型数据	9
2.2 常量与变量	9
2.2.1 标识符	9
2.2.2 常量	10
2.2.3 变量	10
2.3 运算符及表达式	12
2.3.1 算术表达式	12
2.3.2 关系表达式	13
2.3.3 逻辑表达式	14
2.3.4 赋值表达式	15
2.3.5 逗号表达式	15
2.3.6 条件表达式	16
2.3.7 混合运算	16
第 3 章 数据的输入和输出	18
3.1 数据的输入	18
3.1.1 字符输入函数 getchar	18
3.1.2 格式输入函数 scanf	18
3.2 数据的输出	21
3.2.1 转义字符	21
3.2.2 字符输出函数 putchar	21
3.2.3 格式输出函数 printf	22
3.3 应用实例	24
第 4 章 控制结构	28
4.1 结构化程序设计方法	28

4.1.1	软件工程	28
4.1.2	结构化程序设计方法	29
4.1.3	程序的三种基本结构	29
4.1.4	算法及其描述	31
4.2	C 语句	33
4.3	选择结构	34
4.3.1	if 语句	34
4.3.2	switch 语句	40
4.4	循环结构	42
4.4.1	for 语句	43
4.4.2	while 语句	46
4.4.3	do-while 语句	47
4.4.4	循环嵌套	49
4.4.5	几种循环的比较	52
4.5	其他控制转向语句	52
4.5.1	break 语句	52
4.5.2	continue 语句	53
4.5.3	goto 语句	54
4.6	应用实例	55
第 5 章	数组	62
5.1	一维数组	62
5.1.1	一维数组的定义	62
5.1.2	一维数组的引用	63
5.1.3	一维数组的初始化	63
5.2	二维数组	66
5.3.1	二维数组的定义	66
5.3.2	二维数组的引用	66
5.3.3	二维数组的初始化	67
5.3	字符数组	69
5.3.1	字符数组的定义	70
5.3.2	字符数组的引用	70
5.3.3	字符数组的初始化	72
5.4	常用的字符串函数	73
5.5	应用举例	75
第 6 章	函数	81
6.1	函数的概念	81
6.2	函数的定义、调用及返回值	82
6.2.1	函数的定义	82
6.2.2	函数的调用	83

6.2.3 函数的返回值	85
6.3 函数参数的传递方式.....	85
6.3.1 值传递	85
6.3.2 地址传递	88
6.4 函数的嵌套调用.....	90
6.5 函数的递归调用.....	91
6.6 变量的作用域及其存储类型.....	93
6.6.1 局部变量及其作用域	93
6.6.2 全局变量及其存储类型	96
6.7 应用实例.....	98
第7章 编译预处理.....	102
7.1 宏定义	102
7.1.1 不带参数的宏定义	102
7.1.2 带参数的宏定义	103
7.1.3 取消宏定义	105
7.2 文件包含	105
7.3 条件编译	106
7.4 程序的组合	108
7.4.1 内部函数和外部函数	108
7.4.2 C程序的组合	109
7.4.3 标准库函数	116
第8章 指针.....	117
8.1 指针的基本概念	117
8.2 指针变量的定义和引用	118
8.2.1 指针变量的定义	118
8.2.2 指针变量的引用	119
8.3 指针与数组	122
8.3.1 指向一维数组元素的指针变量	122
8.3.2 指针运算	124
8.4 指针与字符串	127
8.4.1 指向字符数组的指针	128
8.4.2 指向字符串常量的指针	130
8.4.3 指针数组	132
8.5 指针与函数	133
8.5.1 函数的参数是指针	133
8.5.2 数组名作函数的参数	136
8.5.3 函数返回指针值	137
8.6 指向指针的指针	139
8.6.1 指向指针的指针	139

8.6.2 指向二维数组的指针	139
8.7 应用实例	142
第 9 章 结构和共用体	149
9.1 结构的定义与引用	149
9.1.1 结构类型的定义	149
9.1.2 结构变量的定义	150
9.1.3 结构变量的引用	152
9.1.4 结构变量的初始化	153
9.2 结构和数组	154
9.2.1 结构的成员是数组	154
9.2.2 结构数组	155
9.3 结构的嵌套	156
9.4 结构与指针	157
9.4.1 指向结构的指针	157
9.4.2 结构的成员是指针类型	159
9.5 结构和函数	160
9.5.1 函数的参数是结构类型	160
9.5.2 函数的返回值是结构类型	162
9.6 共用体	163
9.6.1 共用体类型及变量的定义	164
9.6.2 共用体变量的引用	164
9.7 枚举	167
9.7.1 枚举类型的定义	167
9.7.2 枚举型变量的定义和引用	167
9.8 应用实例	168
第 10 章 文件	172
10.1 文件的基本概念	172
10.2 文件操作	173
10.3 文件操作函数	174
10.3.1 文件的打开和关闭	174
10.3.2 单字符的输入和输出	175
10.3.3 字符串的输入和输出	176
10.3.4 数据块的输入和输出	176
10.3.5 格式化输入和输出	177
10.3.6 文件定位函数	178
10.4 命令行参数	179
10.5 应用实例	181
第 11 章 位运算	185
11.1 位运算	185

11.2	应用实例.....	190
第 12 章	C 语言在数据结构中的应用	191
12.1	动态数据结构简介.....	191
12.2	队列.....	192
12.3	堆栈.....	193
12.4	链表.....	194
12.5	二叉树.....	199
12.6	应用实例.....	200
附录	210
附录	ASCII 码表.....	210
附录	C 语言中的关键字	211
附录	运算符的优先级和结合性表.....	212
附录	C 标准库函数	213
参考文献	220

第 1 章 C 程序的结构

C 语言是一种通用的程序设计语言,可以在 UNIX、DOS 等环境下运行,常见的版本有 Microsoft C、Turbo C、borland C 和 lattice C 等。本书用 Turbo C(TC)编译系统作为实现的版本,全面系统地介绍了 C 语言及其程序设计方法。

本章介绍 C 语言的发展与特点、C 程序的结构与书写方式、C 程序的编辑与运行。

1.1 C 语言简介

(1) C 语言的发展简史

C 语言的产生和发展与 UNIX 操作系统有着十分密切的关系。它是在研制 UNIX 操作系统的过程中诞生的,并伴随着 UNIX 操作系统的发展而流行。

1969 年,美国贝尔实验室的 Ken Thompson 为 DEC PDP-7 计算机设计出了 UNIX 操作系统。接着,他又根据剑桥大学的 Martin Richards 设计的 BCPL(BASIC Programming Language)语言为 UNIX 设计了一种便于编写系统软件的语言 B。B 是一种无类型的语言,直接对机器字操作。1972 年至 1973 年间,同在贝尔实验室的 Denis Ritchie 改造了 B 语言,为其添加了数据类型的概念,并将原来的解释程序改写为可以直接生成机器代码的编译程序,然后将其命名为 C。1973 年,KEN Thompson 小组在 PDP-11 机上用 C 语言重新改写了 UNIX 的内核。与此同时,C 语言的编译程序也被移植到 IBM 360/ 370, Honeywell 11 以及 VAX-11/ 780 等多种计算机上,迅速成为应用最广泛的系统程序设计语言。

1978 年,Brisn Kernighan 和 Denis Ritchie 出版了被称为 C 语言白皮书的《The C Programming Language》,给出了 C 语言的详细定义。1983 年,美国国家标准化协会(ANSI)又制定了新的标准,称为 ANSI C。现在,市面上大多数的 C 编辑器都是按 ANSI C 开发的。

C++ 语言是在 C 语言基础上,于 20 世纪 80 年代后期研制成功的一种面向对象的程序设计语言,即在 C 语言中引入了类、对象、多继承以及运算符重载等概念。C++ 的出现导致了一场程序设计方法的革命,即目前最为流行的“面向对象的程序设计”。C++ 保留了 C 语言的效率高、功能强、模块结构化等重要特性。

学好 C 语言对进一步学习面向对象的程序设计语言会有很大的帮助。

(2) C 语言的特点

C 语言之所以能迅速发展,广泛流行,具有生命力,是因为 C 语言具有以下一些特点:

1) C 语言是一种兼有高级语言和汇编语言优点的语言。它除了有面向用户、语句简单、容易编程的高级语言的优点,又有对地址操作、位操作、寄存器操作、系统功能调用等汇编语言的优点。

2) C 语言是一种结构化程序设计语言。C 语言不但提供了结构化的控制语句,而且程序以函数为模块,便于实现自顶向下、逐步细化的结构化程序设计方法,很适合于编写大型程序。

3) C 语言数据类型丰富。除了整型、实型、字符型等基本数据类型和数组外, C 语言还允许程序员自己定义结构、共用体等数据类型来描述较复杂的数据对象。尤其是 C 语言的指针类型, 功能强大, 应用灵活, 是 C 程序的重要特色之一。

4) C 语言具有丰富的运算符。除了一般高级语言具有的运算符外, C 语言还包括位运算符、自增和自减运算符等, 共有 34 个运算符, 详细介绍见附录。丰富的数据类型与丰富的运算符相结合, 使 C 语言具有表达灵活和高效率的优点。

5) 高效率的目标代码。C 语言不但具有丰富的运算符, 而且具有汇编语言的一些功能, 加上设计 C 语言编译程序时很注重考虑效率因素, 所以 C 语言程序编译后生成的目标代码长度短, 运行速度快, 在各种高级语言中它的效率最高。

6) C 语言有预处理功能。预处理功能可使大型程序的编写和调试更加方便。

7) C 语言程序可移植性好。C 语言程序不依赖计算机硬件系统, 便于在硬件结构不同的机器上实现程序的移植。

8) 具有丰富的标准库函数。利用库函数可简化程序设计的过程和难度, 提高编程效率及程序的质量。

至今, C 语言已为各种不同的计算机系统编写过系统软件, 如操作系统、编译系统、汇编器及编辑器等; 为各种应用领域编写了应用软件, 如数据库管理软件、CAD/CAM 软件、文字处理软件、图形软件、办公自动化软件、科学计算及工程应用软件等。总之, C 语言已经在广泛的领域里得到了应用。

1.2 C 语言程序结构和书写格式

任何一种程序设计语言都有其特定的语法规则和表达方法。用某种程序设计语言进行程序设计时, 必须按照该语言规定的语法和表达方式编写程序, 只有这样, 才能保证所编写的程序可以正确地执行。

程序的结构是指程序的组织形式, 程序的书写格式是指程序的表达方法。下面通过几个简单的程序来说明程序的结构和书写格式。

程序一:

```
main()  
{  
    printf("Welcome to C !\n");  
}
```

程序运行结果为:

Welcome to C !

这是一个非常简单但却完整的 C 语言程序, 它仅由一个主函数 main() 构成。程序中包含一个语句, 其功能是调用标准库函数 printf() 在屏幕上输出一个字符串。输出字符串中转义符“\n”的作用是换行。

程序二:

```
#include "stdio.h"           /* 用文件 stdio.h 中的全部内容来替换此句 */  
main()                       /* 主函数 */
```

```

{
    int a, b, c;           /* 定义三个整型变量 a, b, c */
    scanf("%d, %d", &a, &b); /* 从键盘上输入变量 a, b 的值, & 是取地址运算符
*/
    c = max(a, b);       /* 调用函数 max(), 并将返回值赋给变量 c */
    printf("max = %d", c); /* 输出变量 c 的值, %d 是整数的输入输出格式说明
*/
}
int max(x, y)           /* 定义整型函数 max(), 其中 x, y 为形式参数 */
int x, y;              /* 对形参 x, y 进行数据类型说明 */
{
    int z;              /* 定义变量 z */
    if(x > y) z = x;    /* 用选择控制结构找出 a, b 中的大值并赋给变量 z */
    else z = y;
    return(z);         /* 返回 z 值并使程序的执行返回主函数 */
}

```

程序运行结果为:

输入:12,50[回车]

输出:max = 50

该程序包含两个函数:主函数 `main()` 和被调函数 `max()`。`max()` 的功能是找出变量 `a, b` 中的大值并放在变量 `z` 中, 然后通过返回语句把 `z` 的值赋给主函数中的变量 `c`。`main()` 的功能是输入、输出数据。

从上述两个程序可以看出:

(1) C 语言是函数式的语言, 构成 C 程序的基本单位是函数, 函数是具有相对独立功能的程序模块。C 语言的函数有三种: 第一种是主函数, 名为 `main()`。每个程序中必须包含一个且只能包含一个主函数, 程序总是从主函数中的第一个语句开始运行, 即其他函数都是在执行主函数时被调用执行的。第二种是用户自己定义的函数, 如 `max()`。C 程序中可包含一个或多个这样的函数, 也可以没有。第三种是 C 语言提供的标准库函数, 如 `printf()`、`scanf()` 等, 它们是开发者为用户预先编写的具有特定功能的一系列函数。

(2) 程序的书写格式如下:

- 1) 函数由语句组成。
- 2) 关键字及编译预处理命令用小写字母书写, 如: `int`, `return` 等。
- 3) C 语言程序没有行的概念, 一行中可以有多条语句, 一条语句也可以写在多行上。以分号“;”作为语句的结束符, 注意“;”不是分隔符。
- 4) 用花括号对 { 和 } 表示程序的层次范围, 一般总是让每一对 { 和 } 按列对齐。一个完整的程序模块要用一对 { } 包括, { 标志一个程序功能块的开始, } 标志一个程序功能块的结束。
- 5) 不能在变量名、函数名、关键字中插入空格和空行。
- 6) 可以用 `/* ... */` 对 C 语言中的任何部分作注释, 以增加程序的可读性。
- 7) 把程序中的各语句组按其功能和嵌套关系缩进编排, 可使程序模块和复合关系变得十

分明显。

8) 在程序中加上适当的空格或空行可使程序更加清晰。

(3) 函数的一般形式为：

```
类型说明符  函数名(形式参数表) /* 定义函数的数据类型,说明形参的个数 */  
形式参数说明; /* 说明形参的数据类型 */  
{ /* 表示函数体的开始,其后不加分号 */  
数据说明部分; /* 对函数体内各数据进行类型说明 */  
执行语句部分; /* 由一组完成函数功能的语句组成 */  
} /* 表示函数体的结束,其后不加分号 */
```

1.3 C 语言程序的编辑和运行

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,C 程序的运行一般要经过四个步骤,即源程序的编辑、源程序的编译、目标程序的连接、可执行程序运行。整个运行过程如图 1-1 所示。

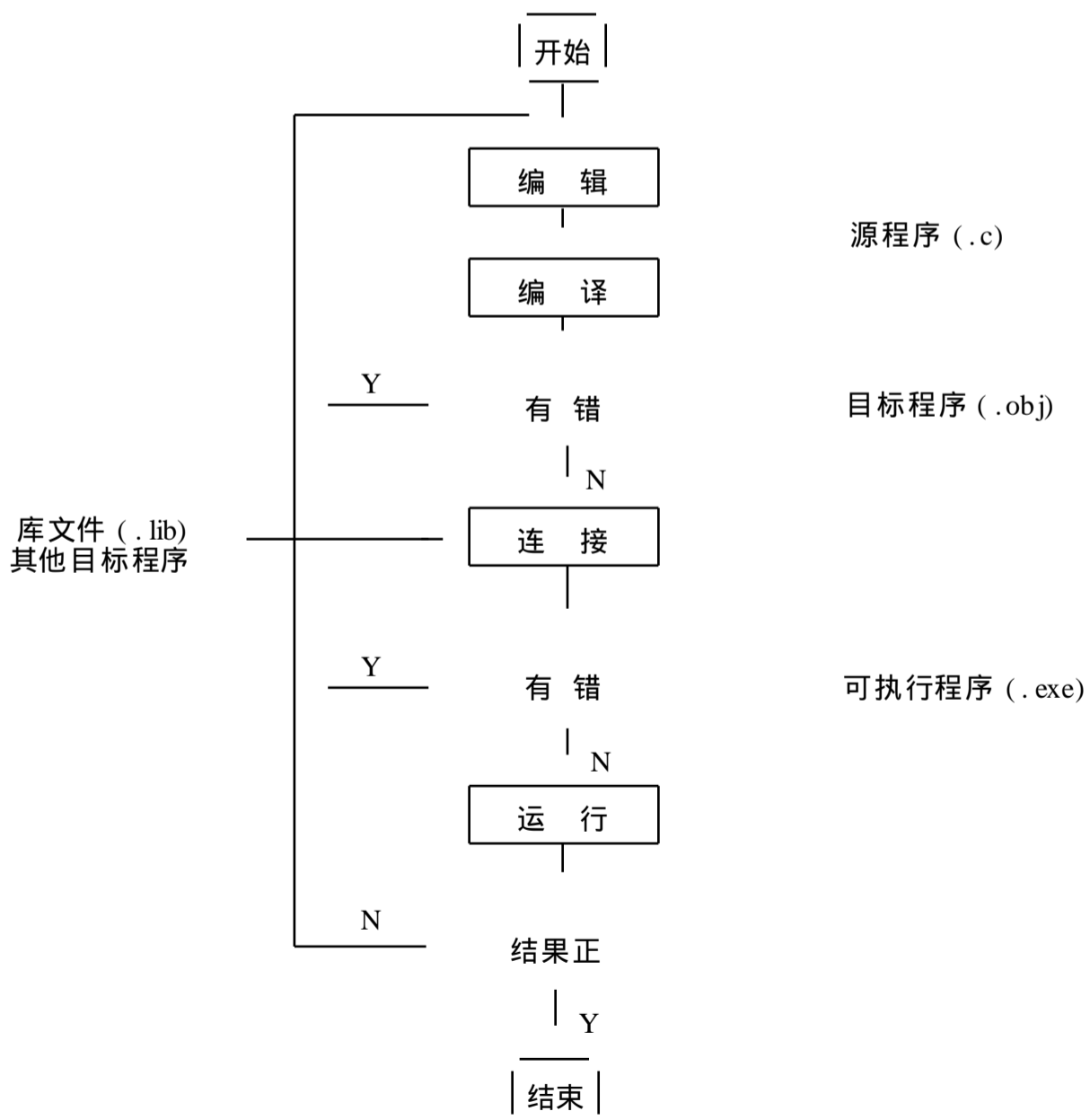


图 1-1

(1) 编辑

编辑是指用编辑程序(即编辑器)把编写好的源程序输入计算机,并以文本文件的形式存放在磁盘上。通过编辑建立的 C 源程序文件的扩展名为“ .C”,如 file1.c、test.c 等。编辑软件有很多,本套教材只介绍 Turbo C 编辑器的用法。

(2) 编译

编译是指将编辑好的扩展名为“ .C”的源程序文件翻译成二进制目标代码的过程。通过编译建立的目标程序文件的文件名与源程序文件名相同,扩展名为“ .obj”。编译时,编译程序先对源程序文件中的每一个语句进行语法检查,发现错误时,就在屏幕上显示出错的位置和错误类型的信息。若编译时发现错误,可使用编辑器对源程序文件进行查错修改,然后再编译,直到修改完所有的错误为止。

(3) 连接

连接是指把一个或多个目标程序文件及系统提供的标准库函数连接在一起,生成可以运行的可执行文件的过程。通过连接建立的可执行程序文件的文件名与源程序文件名相同,扩展名为“ .exe”。如果连接过程出错,则连接程序在屏幕上显示出相应的连接错误信息,但不显示出错的位置。若连接时发现错误,可使用编辑器对源程序文件进行查错修改,然后再编译、连接,直到修改完所有的错误为止。

(4) 运行

运行是指运行可执行程序文件并得到运行结果的过程。通过编译和连接过程排除了源程序文件中的语法和语义错误后,还不能保证源程序文件正确。当得到的运行结果有错时,说明源程序文件中还包含着一些难以查找和判断的逻辑或算法设计错误,此时同样要使用编辑器对源程序文件进行查错修改,然后再编译、连接,直到修改完所有的错误并得到正确的运行结果为止。

1.4 Turbo C 简介

Turbo C 是一种能完成 C 程序的编辑、编译、连接及运行的集成软件。Turbo C 是 1987 年由 Borland 公司首次开发出来的。它包含一套支持各种操作的程序,有一个使用方便的集成开发环境,可以在该环境中依次完成程序的编辑、编译、连接和运行而不必退出该环境,提高了程序生成、编译、调试、检验等各方面的速度。Turbo C 的编辑程序是一个全屏幕、交互式的文本编辑程序,利用它提供的命令可以方便地编辑修改源程序,快速地编译和连接程序,能产生运行较快、代码较小的程序。Turbo C 以它功能齐备的集成开发环境和无与伦比的编译速度在各种 C 的编辑器中占有十分重要的市场,是使用最广泛的一种 C 语言开发系统。

(1) Turbo C 2.0 的安装及运行

Turbo C 2.0 的内容被分装在两张高密软盘上,其中的第一张软盘为安装盘,用户可通过执行安装盘上的安装程序 install.exe,把 Turbo C 2.0 的全部内容安装到指定硬盘的根目录下的 TC 子目录中。TC 目录下除包含 Turbo C 2.0 的文件外,还包含两个子目录 include 和 lib。include 子目录下存放的是头文件,lib 子目录下存放的是标准库函数。

TC 目录下有一些 C 语言的源程序,它们是实例程序,用以说明某些 C 成分的使用方法,例如 calc.c、hello.c、demo.c 等。初学者仔细阅读这些程序,可以学到许多编写 C 语言程序的

编程技巧。

安装好 Turbo C 后,直接执行 TC 目录下的可执行文件 tc .exe 即可进入集成环境。集成环境的操作界面如图 1-2 所示。

图 1-2 中的第一行称为主菜单,它由 8 个英文单词组成。每个英文单词代表一类操作,每一类操作中又包含若干个命令;最后一行是参考行,给出现行常用功能键的意义;主菜单下方一个较大的窗口是用于编辑源程序文件的编辑窗口;在编辑窗口下方的一个较小的窗口是用于显示出错信息的信息窗口。

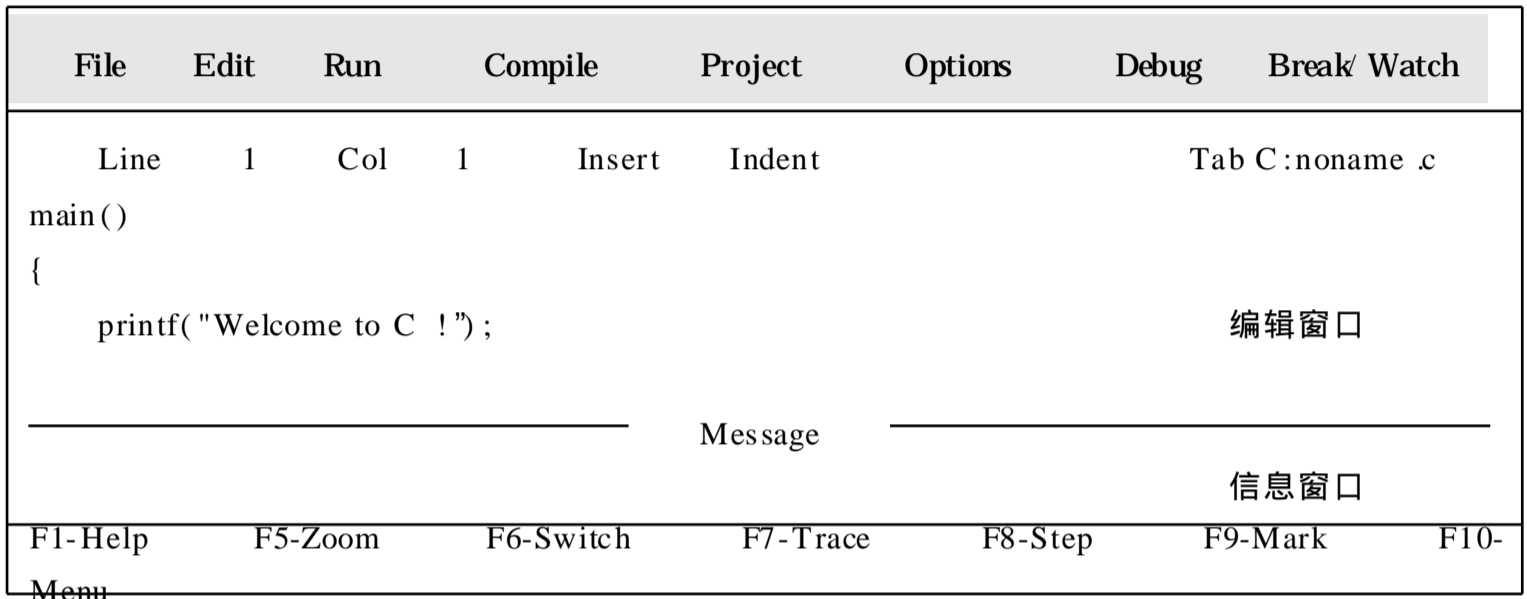


图 1-2

(2) Turbo C 主菜单中各类操作的功能

File: 用于管理文件和目录,装入或保存文件,调用 DOS 命令,退出 Turbo C 环境。

Edit: 进入 Turbo C 的编辑程序,输入和修改源程序。

Run: 编译、连接并运行装入执行 Turbo C 环境的当前程序。

Compile: 编译和连接装入 Turbo C 环境的当前程序。

Project: 管理工程文件。

Options: 设置编译程序和连接程序的各种选择项。

Debug: 设置各种调试选择项。

Break/ Watch: 设置调试用的断点在调试阶段需要监视的变量名。

对于 Turbo C 的详细介绍请阅读与本书配套的《习题集与上机实习指导》一书。

第 2 章 数据类型、运算符和表达式

本章主要介绍 C 语言的基础知识,如:数据类型、标识符、常量、变量、运算符及表达式等,同时还将介绍数据的存储形式、数据类型转换和运算符的优先级与结合性。

2.1 数据类型及存储形式

数据是程序处理的基本对象,数据类型是指数据的内在表现形式。C 语言根据数据的不同性质和用途将其分为不同的数据类型。各种数据类型具有不同的存储长度、取值范围和允许的操作。C 语言提供了丰富的数据类型,如图 2-1 所示。

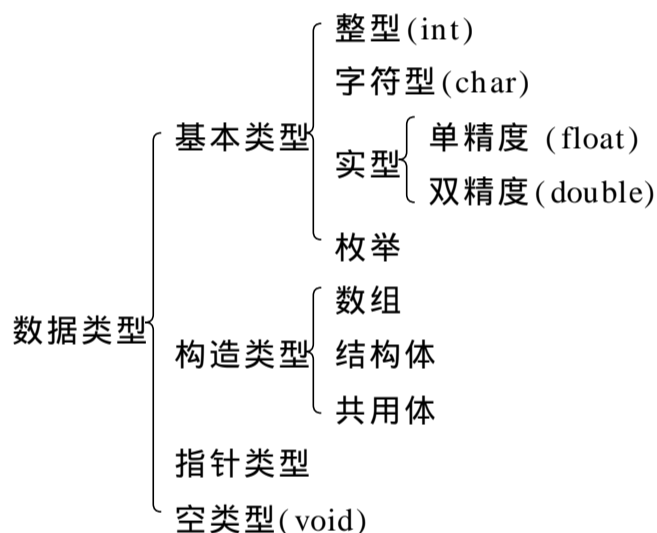


图 2-1

本章只介绍基本类型中的整型、字符型和实型,其他数据类型将在后续章节中介绍。

2.1.1 整型数据

整型数据用来表示数学中的整数,包括正整数、负整数及 0,其定义及取值范围如表 2-1 所示。在 C 语言中,整数可用三种不同的数制表示:十进制数,直接给出数字;八进制数,第 1 位数字前加 0;十六进制数,第 1 位数字前加 0x。例如:

$$65(\text{十进制}) = 0101(\text{八进制}) = 0x41(\text{十六进制})$$

$$-125(\text{十进制}) = -0175(\text{八进制}) = -0x7d(\text{十六进制})$$

位数	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
12:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
-12:	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0

符号位,0 表示正数,1 表示负数

图 2-2

由于大部分计算机都使用二进制补码,因此,有符号整数 12 与 -12 在计算机中的存储形