

21 世纪中等职业学校计算机系列规划教材

C 语言程序设计

刘体斌 编著

电子科技大学出版社

21 世纪中等职业学校计算机系列规划教材 编写委员会名单

(以下均按姓氏笔画排序)

编委会主任：周明天

编委会副主任：王海春 匡 松 周察金

编委会委员：马康波 王洪巍 艾 鹏 吕峻闽

刘 金 刘体斌 刘洪涛 刘晓著

吴 宇 陈 辛 罗福强 张晋美

唐 敏 涂 宏 梁庆龙 曾鸿英

缪春池 廖茂萍

图书在版编目 (CIP) 数据

C 语言程序设计/刘体斌编著.—成都：
电子科技大学出版社，2004.4
ISBN 7-81094-442-8

.C... 刘... .C 语言—程序设计—专业学校—教材
.TP312

中国版本图书馆 CIP 数据核字 (2004) 第 021919 号

内 容 简 介

本书根据中等职业技术学校计算机教材大纲的要求，充分考虑中等职业技术学校学生的特点，结合程序设计理论知识编写而成。其主要内容有：C 语言的基本概念、数据类型、语法及其语句规则、函数定义及其调用、指针、预处理、文件操作等。另外，作者根据多年教学和写作经验，编写了大量与书中内容紧密联系的例子。书中内容浅显易懂、易学易用。并且每章后附有习题和上机实习，可供读者练习、复习和巩固学过的知识。

本书可作为中等职业技术学校、高等职业技术学校的教材，还可作为计算机爱好者的自学参考书。

21 世纪中等职业学校计算机系列规划教材

C 语言程序设计

刘体斌 编著

出 版：电子科技大学出版社 （成都市建设北路二段四号）
丛书策划：周友谊
责任编辑：文 利
发 行：电子科技大学出版社
印 刷：电子科技大学出版社印刷厂
开 本：787mm × 1092mm 1/16 印张 15 字数 362 千字
版 次：2004 年 4 月第一版
印 次：2004 年 4 月第一次印刷
书 号：ISBN 7-81094-442-8/TP · 261
印 数：1—4000 册
定 价：17.60 元

版权所有 侵权必究

邮购本书请与本社发行科联系。电话：(028)83201495 邮编：610054。

本书如有缺页、破损、装订错误，请寄回印刷厂调换。

前 言

本书是根据教育部最新颁布的中等职业学校计算机类课程教学基本要求编写的教材，并通过本套书编委会的严格审定。

C 语言是一种通用的程序设计语言，已经被广大的计算机工作者普遍接受和使用。可以说，没有一个编程者是不懂 C 语言的。现在许多流行的语言，诸如 C++、Java、C# 等语言的基础都是 C 语言。另外，C 语言具有直接操纵计算机硬件能力的这一特点也增添了 C 语言的实用性。因此，C 语言有必要被广大计算机爱好者认识和掌握。

本书面向的对象主要是中等职业技术学校计算机专业的学生，因此在安排教学内容上根据学习者的层次，力求浅显易懂，对某些理论上难以理解的内容没有做深入探讨，避免学习者对所学内容产生畏难情绪。

本教材以美国国家 C 语言为基本内容，结合 Turbo C 编译系统，全面、系统地介绍 C 语言及其程序设计方法。全书共分九章。第一章为绪论，介绍 C 语言的发展、C 程序的基本结构及其 C 程序开发平台——Turbo C 2.0 的使用方法。第二章和第三章介绍 C 语言的基本数据类型、运算符及其各种表达式的运算。第四章介绍 C 语言提供的各种语句，逐步掌握程序设计的方法。第五章介绍 C 语言的复杂数据类型，如数组、结构体、共用体、枚举和自定义数据类型等。第六章介绍 C 语言中比较难以理解的指针数据类型，指针这种数据类型是 C 语言的精华所在。第七章介绍程序模块化设计，讲解 C 语言函数的定义及其调用方法。第八章介绍 C 程序中变量的不同存储方式及其预处理命令的一些用法。第九章介绍 C 语言关于文件处理的一些标准函数的使用方法，并在学习中帮助读者利用这些函数实现一些 DOS 命令的功能。最后在附录中给出 C 语言的各种运算符表、程序编译过程中的常见错误信息以及程序设计中常引用的一些库函数的用法。

本书在编写过程中，电子科技大学出版社朱丹老师对本书的编写提出了各种有益的建议，另外邓辉小姐对本书的出版给予了极大的支持，在此表示诚挚的感谢。

由于成书时间仓促，编写水平有限，书中难免有不妥之处，还望读者海涵。

编 者

2004 年 3 月

序

随着现代科学技术的发展和计算机应用的日益深入和普及，计算机技术与方法已逐渐与其他学科融为一体，成为支撑各学科走向现代化的有机组成部分。对中等职业技术学校各专业学生来说，计算机的作用已不仅仅是一种计算工具，计算机的使用将成为人人都必须掌握的基本技能，计算机的知识将成为人人都必须具有的“第二文化”，不了解计算机知识，不会使用计算机，就像不懂中文、不会写汉字一样，将成为新时期的文盲，将无法在信息社会中有效地生活和工作，因而将会被社会淘汰。

职业技术学校各专业开设计算机课程的目的，首先是为了提高学生的素质，使学生掌握现代计算机的基本知识。同时也是为了让学生具备在信息社会中有效地工作和生活所必需的计算机应用能力，为学生的计算机专业应用打下基础。长期以来，我国从中学到大学的计算机教育，大都以学习计算机高级语言如《BASIC》为主体，把计算机编程作为主要教学内容；另一种倾向认为计算机应用教育主要是教学生学习汉字录入与编辑，片面地以训练学生“打字”速度来代替计算机教育。上述两种倾向都不适应当今社会发展的实际需要。我们认为，职业学校的计算机教育，应立足于培养复合型人才的要求，教育学生把计算机作为一种工具，注重在实际背景下的计算机操作与应用训练，并结合使用有关计算机的大量应用软件，去解决本专业学习和工作中的实际问题，在此基础上，再进一步学习计算机的编程思想和方法。这种以培养学生全面、正确地操作使用计算机并开发专业应用能力的教学模式，更适合于当今职业技术学校学生的计算机教育。

电子科技大学出版社始终将中等职业教育的教材出版作为本

社的出书重点，积累了丰富的经验。为适应近年来计算机技术的迅猛发展，改革传统的教学模式，从整体上提高中等职业技术学校学生的计算机水平，促进教学改革，并结合中职学生的对口升学要求，电子科技大学出版社组织编写了这套中等职业教育计算机系列规划教材。

本套书具有以下突出特点：

1. 针对性强。全套书是根据教育部关于中等职业学校的教学大纲编写而成，同时参照了劳动和社会保障部职业技能鉴定中心颁发的“计算机信息高新技术考试技能培训鉴定规范”。在编写中，针对中等职业教育的特点，突出基础性、先进性、可操作性和实用性，注重对学生创新能力、实践能力和自学能力的培养。

2. 作者实力雄厚。本套书的作者均来自在中等职业教育的教学改革方面走在前列的中等职业学校，他们长期从事中职计算机课程的教学，积累了丰富的教学经验，具有较高的学术水平。强有力的作者队伍是本套书成功的重要保证。

3. 全套书采用任务驱动式、案例式教学法来编写，既体现了常规教材的系统性，又能兼顾学生的职业技能培训，突出职业教育的特色。

相信这套书的出版会对我国中等职业学校的计算机教育起到积极的推动作用。



电子科技大学教授 博士生导师

2004年4月18日

目 录

第一章 绪论	1
1.1 C 语言概述	1
1.1.1 C 语言发展	1
1.1.2 C 语言特点	1
1.2 C 语言程序的基本结构	2
1.3 C 程序的编辑、编译、链接和运行	5
1.3.1 源程序的编辑	5
1.3.2 源程序的编译	5
1.3.3 目标文件的链接	6
1.3.4 运行	6
1.4 Turbo C 2.0	6
1.4.1 安装和主要文件	6
1.4.2 Turbo C 2.0 集成环境的使用	8
1.4.3 Turbo C 2.0 编译错误信息	12
习题一	12
上机实习一	13
第二章 基本数据类型、数据的输入/输出	15
2.1 基本数据类型	15
2.1.1 常量和变量	15
2.1.2 基本数据类型	16
2.2 整型数据的表示方法	17
2.2.1 整型常量	17
2.2.2 整型变量	17
2.2.3 整型常量的类型	19
2.3 实型数据的表示方法	20
2.3.1 实型常量	20
2.3.2 实型变量	20
2.4 字符型数据的表示方法	21
2.4.1 字符常量	21
2.4.2 字符变量	23
2.4.3 字符串	23
2.5 输出函数	24
2.5.1 printf 函数	24

2.5.2 putchar 字符输出函数.....	28
2.6 输入函数	29
2.6.1 scanf 函数.....	29
2.6.2 getchar 字符输入函数.....	33
习题二	34
上机实习二	37
第三章 基本运算符及表达式	39
3.1 表达式	39
3.1.1 表达式概念.....	39
3.1.2 表达式的值.....	40
3.1.3 语句和表达式的关系.....	40
3.2 算术运算符.....	40
3.3 赋值运算符.....	42
3.4 关系运算符.....	43
3.4.1 关系运算符.....	44
3.4.2 关系表达式.....	44
3.5 逻辑运算符、位运算符.....	44
3.5.1 逻辑运算符.....	45
3.5.2 按位逻辑运算符.....	46
3.6 其他运算符的运用.....	49
3.6.1 自增运算符和自减运算符.....	49
3.6.2 复合赋值运算符.....	50
3.6.3 逗号运算符.....	51
3.6.4 条件运算符.....	51
习题三	52
上机实习三	55
第四章 控制语句	57
4.1 程序基本控制结构.....	57
4.1.1 顺序结构.....	57
4.1.2 分支结构.....	58
4.1.3 循环结构.....	59
4.2 复合语句	59
4.3 if 条件分支语句	60
4.3.1 一般格式.....	60
4.3.2 嵌套的 if 语句	62
4.4 switch 多分支语句	64

4.5 循环语句	67
4.5.1 while 循环语句.....	67
4.5.2 do ~ while 循环语句.....	68
4.5.3 for 循环语句.....	69
4.5.4 循环语句嵌套.....	71
4.6 break、continue、goto 语句.....	74
4.6.1 break 语句.....	74
4.6.2 continue 语句.....	75
4.6.3 标号与 goto 语句.....	76
习题四	77
上机实习四	82
第五章 数组及其他数据类型	83
5.1 一维数组	83
5.1.1 一维数组的定义.....	83
5.1.2 一维数组的初始化.....	84
5.1.3 一维数组元素的引用.....	85
5.1.4 应用举例.....	85
5.2 二维数组	88
5.2.1 二维数组的定义.....	88
5.2.2 二维数组的初始化.....	89
5.2.3 二维数组元素的引用.....	90
5.2.4 多维数组.....	90
5.2.5 应用举例.....	90
5.3 字符数组	92
5.3.1 字符数组的定义和初始化.....	92
5.3.2 字符数组的输入与输出.....	93
5.3.3 字符串处理函数.....	95
5.3.4 应用举例.....	98
5.4 结构体	98
5.4.1 结构体类型和结构体类型变量的定义.....	99
5.4.2 结构体类型变量的使用.....	101
5.4.3 结构体变量的初始化.....	102
5.5 共用体	103
5.5.1 共用体类型及共用体类型变量的定义.....	103
5.5.2 共用体类型变量的使用.....	104
5.6 枚举类型	104
5.6.1 枚举类型及枚举类型变量的定义.....	104

5.6.2 枚举类型变量的使用.....	105
5.7 typedef 自定义类型.....	106
习题五	107
上机实习五	111
第六章 指针	113
6.1 指针与指针变量.....	113
6.1.1 地址与指针.....	113
6.1.2 变量的指针.....	114
6.1.3 指针变量的定义及赋值.....	115
6.2 指针运算	115
6.2.1 取地址运算“&”与取内容运算“*”.....	115
6.2.2 指针与整数的加减运算.....	116
6.2.3 指针的关系运算.....	117
6.3 指针与数组.....	118
6.3.1 一维数组的指针.....	118
6.3.2 二维数组的指针.....	120
6.3.3 字符串指针.....	123
6.4 指针数组和指向指针的指针.....	125
6.4.1 指针数组.....	125
6.4.2 指向指针的指针.....	126
6.5 指针与结构体.....	129
6.5.1 结构体指针变量.....	129
6.5.2 指向结构体数组的指针.....	130
6.5.3 结构体指针数组.....	131
6.6 指针与链表.....	132
6.6.1 链表	132
6.6.2 内存的动态分配和释放.....	133
6.6.3 链表结点在 C 语言中的表示.....	134
6.6.4 链表的操作.....	135
习题六	141
上机实习六	147
第七章 函数	149
7.1 函数定义和调用.....	149
7.1.1 函数定义.....	149
7.1.2 函数调用.....	151
7.2 函数的递归调用.....	155

7.3 函数与数组.....	159
7.3.1 数组元素作函数的参数.....	159
7.3.2 数组名作为函数的参数.....	160
7.4 函数与指针.....	161
7.4.1 指针作为函数的参数.....	161
7.4.2 返回值是指针的函数.....	162
7.4.3 指向函数的指针.....	163
7.4.4 函数与链表.....	165
7.5 命令行参数.....	169
习题七.....	171
上机实习七.....	176
第八章 变量存储类别及预处理.....	178
8.1 变量的存储类别.....	178
8.1.1 局部变量和全局变量.....	178
8.1.2 动态存储和静态存储.....	180
8.2 预处理.....	184
8.2.1 宏定义.....	184
8.2.2 文件包含.....	186
8.2.3 条件编译.....	187
习题八.....	189
上机实习八.....	192
第九章 文件操作.....	194
9.1 文件.....	194
9.2 打开、关闭文件函数.....	195
9.2.1 打开文件函数 fopen.....	196
9.2.2 关闭文件函数 fclose.....	196
9.3 标准流式文件.....	197
9.4 文件读写函数.....	198
9.4.1 fputc 和 fgetc 函数.....	198
9.4.2 fputs 和 fgets 函数.....	200
9.4.3 feof 函数.....	201
9.4.4 fread 函数和 fwrite 函数.....	202
9.4.5 格式输出函数 fprintf 和格式输入函数 fscanf.....	205
9.5 文件定位函数.....	207
9.5.1 rewind 函数.....	207
9.5.2 fseek 函数.....	207

9.5.3 ftell 函数	208
习题九	209
上机实习九	211
附录一 Turbo C 2.0 编译常见错误信息	213
附录二 C 语言运算符	217
附录三 Turbo C 2.0 库函数	218

第一章 绪 论

【学习目标】

- (1) 了解 C 语言的发展；
- (2) 掌握 C 语言的特点和 C 程序的结构；
- (3) 掌握 Turbo C 语言环境的功能及使用方法。

1.1 C 语言概述

1.1.1 C 语言发展

C 语言是计算机通用的一种高级程序设计语言,它是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司(AT&T)贝尔实验室在开发 UNIX 系统的过程中,一种兼有高级语言和低级语言特点的 C 语言正式发表。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的“THE C PROGRAMMING LANGUAGE”一书。通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的 C 语言标准。1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本对 C 语言的发展进行了扩充,制订了新的标准,称为 ANSI C。1987 年,ANSI 又公布了新标准——87 ANSI C。本书的内容主要是基于该标准编写的。

早期的 C 语言主要应用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐被人们认识,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使用,成为当代最优秀的程序设计语言之一。在 C 的基础上,1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。但是,C 是 C++ 的基础,C++ 语言和 C 语言在很多方面是兼容的。因此,掌握了 C 语言,再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。

1.1.2 C 语言特点

自计算机发明以来,控制计算机工作的编程语言有许多,但是流行的语言却不多。C 语言作为一种最具长久生命力、最适合编写系统软件的计算机语言,总有优越于和不同于其他语言的特点。其特点主要有以下几个方面:

1. 数据类型丰富

C 语言除了整型、字符型、浮点型等基本数据类型外，还允许用户定义诸如数组、结构体、共用体、枚举、指针等高级数据类型来描述复杂的数据结构，如链表、树等。特别是指针类型，功能强大，灵活多变，大大简化了程序结构，提高了程序的运行效率，是 C 语言的重要特色之一。当然指针也是 C 语言中较难掌握的内容，希望读者在学习到这部分内容时，应引起重视。

2. 运算符丰富

C 语言的运算符非常丰富，其他语言中的一些语法规则，甚至一些语句，在 C 语言中也将它作为运算符来处理，例如赋值运算、括号等。C 语言中有直接求变量存储地址的运算符，也有进行二进制运算的运算符。正是由于 C 语言运算符的多样性，使得 C 语言的表达式具有多样化的特点。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。有时看似一个非常复杂的问题，用 C 语言可能仅是一个表达式。因此，希望初学者学会正确理解 C 程序中一些表达式的含义。

3. 结构化程序设计语言

C 语言提供的分支、循环和模块的基本结构和语句，便于实现由顶向下、逐步求精的结构化程序设计方法。程序可读性好，便于调试和维护。C 语言的主要结构成分是函数。程序以函数为模块有利于把程序分割成若干个相对独立的功能模块，同时也为程序模块之间的相互调用及数据传递提供了许多方便。这一特点有利于大型软件模块化，为多人共同开发大型软件的软件工程技术提供了强有力的支持。

4. 能生成高效率的目标代码

利用 C 语言编写的程序，生成的目标代码质量很高。无论参考程序运行的时间还是参考程序所占据的存储空间，这两方面的高效都是其他高级语言所不能比拟的。它仅比用汇编语言生成的目标代码低 10% ~ 20%。即使有的系统在设计初期，基于程序员的知识，可能是用其他高级语言实现的，但是最后却常常由于系统运行速度慢，达不到用户的要求而将系统用 C 语言进行改写。

5. 可移植性好

目前 C 语言可在许多机器上实现，用 C 语言编写的程序一般不需要做修改就能移植到不同的机型、不同的操作系统中。

1.2 C 语言程序的基本结构

任何一种程序设计语言都具有特定的语法规则和规定的表达方式。一个程序只有严格按照语法和表达方式编写，才能保证编写的程序在计算机中能够被正确执行。同时在编写程序时，要求程序员遵循良好的程序设计风格，便于其他读者阅读和理解。

用 C 语言编写的程序称为 C 语言源程序。一个 C 语言源程序文件由一个或多个函数构成，函数是 C 程序的基本构成单位。每个 C 程序必须有且只能有一个名为 main 的主函数，程序的执行总是从主函数开始的。

下面我们通过两个例子来说明 C 程序的基本结构。

【例 1-1】

```
/* 在屏幕上显示：How are you! */  
main()  
{  
    printf("How are you!\n");  
}
```

运行结果为：

How are you!

该 C 语言源程序非常简单，仅由一个 main 主函数构成。在主函数中仅仅只有一个语句，该语句是调用 C 语言中提供的 printf 标准函数形成的，printf 函数的功能是把要输出的内容送到显示器去显示。例子输出信息中的“ ”表示“回车换行键”。程序中每个语句末以分号作为语句的结束标志。

源程序中的第一行是 C 语言的注释。C 语言的注释是由“/*”和“*/”括起来的任何文字，可以出现在程序中的任何地方。C 语言中的注释是用来说明程序段的功能、变量的作用以及程序员认为应该向程序阅读者说明的任何内容。在将 C 程序编译成目标代码时，所有的注释都将被忽略。C 语言注释的多少不影响目标代码的效率。注释可以使程序清晰易懂，便于调试、修改、维护，便于程序员之间相互交流与协作。但是需要注意的是：注释标志不允许嵌套。例如：“/*C 语言注释/*非法 1*/请注意*/”和“/*C 语言注释非法 2*/请注意*/”都是错误的注释形式。

【例 1-2】

```
/* 求最大值 */  
main()  
{  
    int i,j,z;  
    scanf("i=%d,j=%d",&i,&j);  
    z=max(i,j);  
    printf("max=%d\n",z);  
}  
/* 求两整型数据的最大值函数 */  
int max(x,y)  
int x,y;  
{  
    int m;  
    m=x>y?x:y;  
    return(m);  
}
```

运行的结果为：

输入：i=10 , j=16

输出：max=16

该 C 语言源程序由两个函数构成，从主函数的第一条语句开始执行。首先定义了三个整型变量，然后输入变量 i 和 j 的值。scanf 函数与 printf 函数相对应，是从终端给相应的变量输入值。第三个语句表示主函数调用用户自定义的 max 函数，将 i、j 参数的值传递给 max 函数的 x、y 参数，执行 max 函数中的语句，通过 return 语句返回 x、y 的最大值并赋给 z 变量，最后将求得的最大值在屏幕上输出显示。执行完主函数的所有语句，整个程序执行完毕。

在例题中的函数体分为两部分：一部分为说明部分，另一部分为执行部分。说明部分是指变量的类型说明。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。本例中如主函数中 i、j、z 变量的说明。函数体中说明部分后的部分称为执行语句部分，用以完成程序的功能。

从上述两个例子我们可以了解到 C 语言的函数有以下三种：

(1) 主函数 main

每个程序中必须有一个主函数且只能有一个。在运行时，程序从主函数的第一条语句开始执行，执行完最后一条语句，整个程序就执行完毕。

(2) 用户自己编写的函数（也称为自定义函数或子程序）

程序中用户自定义函数的数量不限，当然也可以没有。实际上，一个 C 语言应用程序通常都包含若干功能单一、结构简单的函数，能提高程序的可读性，便于调试和维护。

(3) 标准函数（也即库函数）

C 语言提供了若干个不同功能的标准函数，便于程序员编写各种应用程序。C 语言的开发者将许多常用的算法和功能编写成函数，然后将编译后生成的目标文件统一存放在库函数中。这样，程序员在编写程序时如果实现某些功能，而标准函数中具有完成这些功能的函数，那么就不必自己再编写程序，只需直接调用相应的标准函数就可以了。但是需要注意的是：由于这些标准函数的目标程序是存放在相关的文件中，所以需要在引用这些标准函数的程序中指明这些文件的位置。利用 C 语言的#include 预编译命令可以实现。

各种版本的 C 语言提供的标准函数的数量也各不相同，一般都在数百个以上。本书将对一些常用的标准函数做具体介绍，并且要求读者掌握。对于其他的标准函数读者可以查看相关资料，掌握这些标准函数的使用方法，本书不做具体介绍。

现在，我们可以把编写 C 程序的基本要点归纳如下：

(1) 一个 C 语言程序可以由一个或多个源文件组成。每个源文件可由一个或多个函数组成。一个源程序不论由多少个文件组成，都有一个且只能有一个 main 主函数。

(2) C 程序习惯使用小写英文字母，常量和特殊用途的符号可以用大写字母。C 语言区分大小写字母，如果将主函数名 main 写成 MAIN，程序将不能运行。

(3) C 程序不存在程序行概念。一行中可以写多个语句，有时也可以将一个语句写在多行上。分号是一个语句的结束标志，多个语句之间必须用分号间隔。

(4) 用一对花括号“{ }”标志一个程序块的开始与结束，为了提高程序的可读性，一般总是让每一对花括号按列对齐。

(5)在程序中增加适当的空格或空行,可以将各语句组按其功能和嵌套关系进行缩排,使程序的模块和复合关系变得明显,增加程序层次的清晰性,以培养用户良好的程序设计风格。

(6)充分利用注释功能。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性和可维护性。

1.3 C 程序的编辑、编译、链接和运行

计算机是一种传输、存储和处理信息的电子设备,其基本工作包括:数据的输入、数据的存储处理和数据的输出。数据是信息的载体,计算机无论处理什么类型的数据,都是通过执行人们所编制的程序来完成预定目标的。因此,用计算机解决一个实际问题,首先要进行程序设计,其一般过程是:

(1)分析问题;

(2)找出解决问题的算法和确定合适的数据结构;

(3)选用一种高级语言描述算法,编写解决问题的程序。

C语言和其他编译型高级语言一样,其运行一般要经过四个步骤,即:源程序的编辑、源程序的编译、目标程序的链接、可执行程序的运行。这一过程可用如图 1-1 所示的流程图描述。

1.3.1 源程序的编辑

程序员使用一种编辑软件将自己编写的程序输入到计算机中并以文本文件的形式存储,文件的扩展名一般是“.C”,以备程序员在调试和修改时能够识别是C语言程序。现有的编辑软件很多,如常用的Word、WPS、Notepad(记事本)、Wordpad(写字板)等,也可以利用任何一个程序开发平台来编辑。后面我们将着重介绍基于C语言的Turbo C集成开发平台。

1.3.2 源程序的编译

编译是指借助于对应的编译程序将编辑好的源程序翻译成二进制文件的过程,生成的

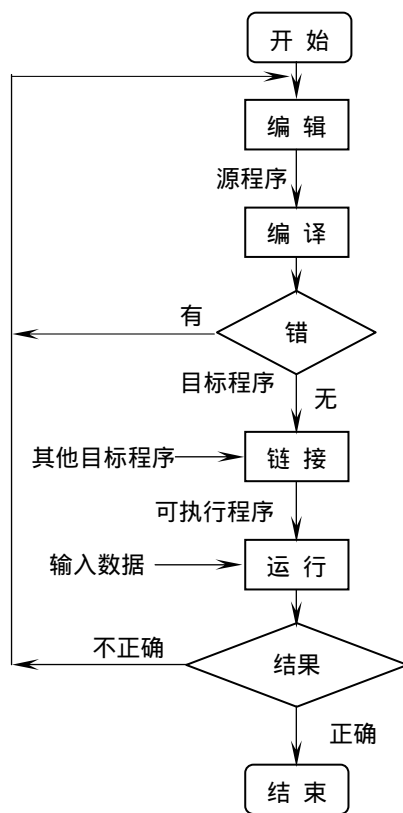


图 1-1