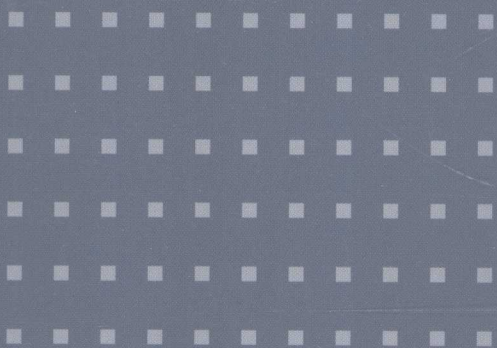


谢延红 李丽 主编

郭长友 王付山 戎丽霞 副主编

C语言实用程序设计 基础教程



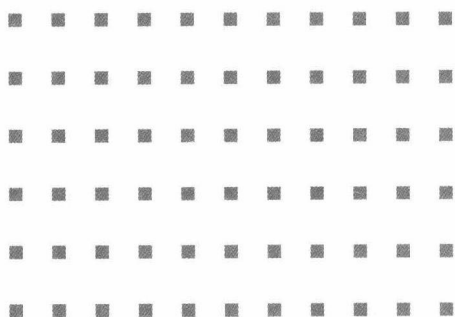
清华大学出版社

谢延红 李丽 主编

郭长友 王付山 戎丽霞 副主编

C语言实用程序设计

基础教程



清华大学出版社
北京

内 容 简 介

本书是山东省精品课程《信息技术基础实践》的研究成果。主要内容包括概述、数据类型和表达式、顺序结构与常用函数、选择结构、循环结构、数组、函数与变量、指针、结构体与共同体、编译预处理、位运算及文件等。全书体系完整,重点突出,并通过一个综合案例将全书内容有机地贯穿起来,真正做到了理论和实践的密切结合。全书内容循序渐进,讲解深入浅出、通俗易懂,图文并茂,案例丰富新颖。

本书适合作为大学本科和专科院校的C语言程序设计课程教学用书,也可作为程序设计人员的参考用书以及全国计算机等级考试(二级C语言考试科目)的培训教材。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C语言实用程序设计基础教程/谢延红,李丽主编.--北京:清华大学出版社,2015

ISBN 978-7-302-39034-3

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

中国版本图书馆CIP数据核字(2015)第017150号

责任编辑:冯 昕 赵从棉

封面设计:常雪影

责任校对:刘玉霞

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印刷者:北京四季青印刷厂

装订者:三河市吉祥印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.25 字 数:467千字

版 次:2015年5月第1版 印 次:2015年5月第1次印刷

印 数:1~2800

定 价:39.00元

产品编号:062579-04

主 编	谢延红	李 丽		
副主编	郭长友	王付山	戎丽霞	
编 委	鲁 燕	李天志	赵丽敏	王文博
	宋秀芹	张建臣	杨光军	曹金凤

前 言

C语言是国内外各高校理工科专业的一门重要基础课程,是一些计算机相关专业的研究生入学考试科目之一,地位至关重要。

本书是山东省精品课程《信息技术基础实践》的研究成果,是在长期从事程序设计类课程教学、教研经验丰富的一线教师教学手稿的基础上修改、整理而成。在本系列教材的规划、编写、整理过程中,不仅对现有较经典的教材进行了充分调研,并多次邀请专家和程序设计类课程的一线教师参会论证,力求博采众长、定位准确、突出特色。

本书具有如下特点:

(1) 全书体系完整,重点突出,深入浅出,循序渐进,图文并茂,讲解透彻,剖析深入,案例丰富新颖。

(2) 关注例题之间的阶梯性和连贯性,所有例题均有程序设计思路分析,这不仅有效降低了学习难度,而且突出了算法思想设计,注重学生编程思维和编程能力的培养。

(3) 注重理论,突出实践。每章后面均有典型例题及程序分析,以实践的形式强化理论,突出易错点,为学生提供解题思路,并通过程序调试方法的详细讲解为实践提供有力保障。

(4) 以一个小型系统为案例,随着课程学习的不断深入逐步完善,最后形成一个功能完整的小型系统。本案例着力突出培养学生利用C语言分析问题、解决问题的能力。

(5) 全方位服务。为方便教师和读者使用,提供了配套的电子课件、例题源程序、习题答案、教学大纲、参考书目等。

全书内容共分为12章,主要内容包括概述、数据类型和表达式、顺序结构和常用函数、选择结构、循环结构、数组、函数和变量、指针、结构体与共同体、编译预处理、位运算及文件等。每章的开篇文字,说明本章的主要内容;每章后面有典型例题分析,帮助读者强化理论内容;全书设有一个综合案例将全书内容有机贯穿起来,使读者随着课程学习的不断深入,能走出局部,以全局的角度综合运用C语言知识点解决实际问题。

本书的编写得益于编写小组的鼎力合作,其中王文博、郭长友负责编写第1、2章,谢延红、宋秀芹负责编写第3~5章,鲁燕负责编写第6章,李天志、张建臣负责编写第7章,赵丽敏负责编写第8章,戎丽霞、杨光军负责编写第9、10章,王付山负责编写第11、12章和附录A、B,曹金凤、戎丽霞负责编写综合案例和附录C。全书由谢延红、李丽统稿并任主编。所有教师均参与了书稿的校稿和程序调试工作。本教材在编写过程中得到了德州学院信息管理学院的大力支持,参考了大量书籍、报刊和互联网等参考文献,一些教师和学生也提出了宝贵的意见和建议,在此一并表示衷心的感谢。

由于编者水平有限,书中疏漏和不足在所难免,诚挚地希望专家和广大读者不吝赐教,提出宝贵意见和建议,我们会认真对待,以期不断改善教材质量。邮件请发至dzxyjsjxc@163.com。

编 者

2015年1月

目 录

第 1 章 概述	1
1.1 计算机程序设计语言的发展	1
1.2 算法	2
1.2.1 算法的主要特征	2
1.2.2 算法的描述方法	2
1.3 C 语言简介	3
1.3.1 C 语言发展历程	3
1.3.2 C 语言的特点	4
1.3.3 C 语言程序示例	4
1.3.4 C 语言程序书写约定	5
1.4 C 语言程序开发步骤	6
1.4.1 C 语言程序开发过程	6
1.4.2 VC++6.0 环境中 C 语言程序运行步骤	6
1.4.3 VC++6.0 环境中其他关键功能	10
1.5 C 语言程序的错误类型及调试方法	11
1.5.1 编译错误及调试方法	11
1.5.2 链接错误及调试方法	12
1.5.3 运行错误及调试方法	14
1.5.4 逻辑错误及调试方法	14
1.6 综合案例	16
习题	18
第 2 章 数据类型和表达式	20
2.1 C 语言字符集与词法规则	20
2.1.1 C 语言字符集	20
2.1.2 C 语言词汇及其组成规则	21
2.2 数据类型	22
2.2.1 基本类型	22
2.2.2 其他数据类型	23

2.3 常量	23
2.3.1 整型常量	24
2.3.2 实型常量	24
2.3.3 字符型常量	25
2.3.4 字符串常量	26
2.3.5 符号常量	26
2.4 变量	26
2.4.1 变量的定义	27
2.4.2 变量赋初值	27
2.4.3 常变量	27
2.5 运算符和表达式	27
2.5.1 运算符的优先级与结合性	28
2.5.2 算术运算符和算术表达式	29
2.5.3 赋值运算符与赋值表达式	31
2.5.4 关系运算符和关系表达式	31
2.5.5 逻辑运算符和逻辑表达式	31
2.5.6 其他运算符与表达式	32
2.5.7 数据的类型转换	33
2.6 典型例题	35
2.7 综合案例	35
习题	36
第3章 顺序结构程序设计及常用函数	38
3.1 C 语句分类	38
3.2 常用数据输出函数	39
3.2.1 单字符输出函数 putchar	40
3.2.2 格式输出函数 printf	40
3.3 常用数据输入函数	44
3.3.1 单字符输入函数 getchar	44
3.3.2 格式输入函数 scanf	44
3.4 其他常用函数	49
3.4.1 常用数学函数	49
3.4.2 常用字符函数	49
3.4.3 其他常用工具函数	49
3.5 典型例题	51
3.6 综合案例	52
习题	53

第4章 选择结构程序设计	55
4.1 if 条件语句	55
4.2 条件表达式	61
4.3 switch 语句	63
4.4 典型例题	66
4.5 综合案例	70
习题	72
第5章 循环结构程序设计	75
5.1 while 语句	75
5.2 do-while 语句	78
5.3 for 语句	79
5.4 循环语句的嵌套	82
5.5 break 语句和 continue 语句	84
5.5.1 break 语句	84
5.5.2 continue 语句	85
5.6 典型例题	86
5.7 综合案例	88
习题	90
第6章 数组	94
6.1 一维数组	95
6.1.1 一维数组的定义	95
6.1.2 一维数组的使用	97
6.1.3 一维数组的初始化	98
6.2 二维数组	100
6.2.1 二维数组的定义	101
6.2.2 二维数组的使用	104
6.2.3 二维数组的初始化	105
6.3 字符串	107
6.3.1 字符串常量	107
6.3.2 字符串与字符数组	108
6.3.3 字符串的输入输出	111
6.3.4 字符串处理函数	113
6.4 典型例题	117
6.5 综合案例	120
习题	121

第7章 函数与变量	126
7.1 函数定义	126
7.2 函数的调用	128
7.2.1 函数的调用形式	128
7.2.2 函数的调用过程	128
7.2.3 函数的嵌套调用	130
7.3 函数原型声明	131
7.4 函数的参数传递	132
7.4.1 传值方式	132
7.4.2 传址方式	133
7.5 递归函数	138
7.6 变量的作用域	140
7.6.1 局部变量	141
7.6.2 全局变量	142
7.7 变量的存储类型	142
7.7.1 自动变量	143
7.7.2 静态变量	144
7.7.3 寄存器变量	144
7.7.4 外部变量	145
7.7.5 变量汇总	146
7.8 典型例题	146
7.9 综合案例	149
习题	151
第8章 指针	157
8.1 指针与指针变量	157
8.1.1 指针变量的基本概念	157
8.1.2 指针变量的定义与初始化	159
8.1.3 指针变量的使用	160
8.1.4 二级指针	162
8.2 指针与数组	163
8.2.1 一维数组和指针	164
8.2.2 二维数组和指针	167
8.2.3 指向字符串的指针	172
8.3 指针与函数	174
8.3.1 指针变量作为函数参数	174
8.3.2 指向函数的指针	178
8.3.3 返回值为指针的函数	180

8.3.4	main 函数的参数	181
8.4	典型例题	182
8.5	综合案例	188
	习题	189
第 9 章	结构体与共用体	195
9.1	结构体类型	195
9.1.1	定义结构体类型	195
9.1.2	结构体变量的定义	196
9.1.3	结构体变量的使用	198
9.1.4	结构体变量的初始化	199
9.1.5	结构体变量的赋值	200
9.2	结构体数组	201
9.2.1	结构体数组的定义	201
9.2.2	结构体数组的初始化	201
9.2.3	结构体数组的使用	202
9.3	结构体类型指针	203
9.3.1	指向结构体变量的指针	203
9.3.2	指向结构体数组的指针	205
9.4	结构体与函数	206
9.4.1	结构体变量作函数参数	206
9.4.2	指向结构体变量(或数组)的指针作函数参数	207
9.4.3	函数的返回值为结构体类型	208
9.5	链表	209
9.5.1	链表概述	209
9.5.2	动态存储分配函数	210
9.5.3	链表的基本操作	210
9.6	共用体	216
9.6.1	共用体类型的定义	216
9.6.2	共用体类型变量的定义	217
9.6.3	共用体变量的使用	217
9.7	枚举类型	218
9.8	typedef 类型定义	220
9.9	典型例题	222
9.10	综合案例	224
	习题	225
第 10 章	编译预处理	233
10.1	宏定义	233

10.1.1 不带参数的宏定义	233
10.1.2 带参数的宏定义	235
10.2 文件包含	236
10.3 条件编译	236
10.4 典型例题	238
10.5 综合案例	239
习题	239
第 11 章 位运算	242
11.1 位运算符	242
11.2 位运算的应用	243
11.3 位段及其应用	249
11.4 典型例题	251
习题	252
第 12 章 文件	255
12.1 文件概述	255
12.2 文件指针	257
12.3 文件的打开与关闭	258
12.4 文件的定位与检测	261
12.5 文件的读写操作	263
12.5.1 按字符方式文件读写函数 fgetc 和 fputc	263
12.5.2 按字符串方式文件读写函数 fgets 和 fputs	265
12.5.3 按格式化方式文件读写函数 fscanf 和 fprintf	266
12.5.4 按数据块方式文件读写函数 fread 和 fwrite	268
12.5.5 文件的随机读写	270
12.6 典型例题	271
12.7 综合案例	273
习题	273
附录 A C 语言常用库函数	278
附录 B 常用字符与 ASCII 代码对照表	284
附录 C 综合案例参考源代码	285
参考文献	295

概 述

自从 20 世纪 40 年代电子计算机诞生以来,无论是计算机的硬件方面还是软件方面,都得到了迅猛发展,计算机及其应用也渗透到社会的各个领域。程序设计语言是人与计算机之间交流的重要工具,在众多程序设计语言中,C 程序设计语言(简称 C 语言)有其独特之处,是高级程序设计语言的典型代表,也是国内外广泛使用的一种编程语言。

本章首先简要介绍了计算机程序设计语言的发展、算法的特征和表示方式,然后介绍了 C 语言的发展历程,并以实例方式介绍 C 语言的构成特点,最后介绍 C 语言程序的开发步骤和错误分类及调试方法。

1.1 计算机程序设计语言的发展

语言可分为自然语言与人工语言两大类。自然语言是人类在自身发展过程中形成的语言。人工语言是指为某种目的而设计的语言,计算机语言就是人工语言的一种。

自从世界上第一台电子计算机 ENIAC 于 1946 年问世以来,随着计算机硬件的不断更新换代,计算机程序设计语言也有了很大的发展。在过去的几十年间,大量程序设计语言被发明、取代、修改或组合在一起。计算机语言的发展经历了机器语言、汇编语言和高级语言三个阶段。

1. 机器语言

机器语言是第一代计算机语言,是用二进制代码表示的、计算机能直接识别和执行的一种机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和执行效率高等特点。用机器语言编写程序,编程人员要熟记所用计算机的全部指令代码和代码的含义,但这是一件十分烦琐的工作。编写程序花费的时间往往是实际运行时间的几十倍或几百倍,并且,程序全是数字 0 和 1 组成的二进制指令代码,直观性差,容易出错。

2. 汇编语言

20 世纪 50 年代中期,为了减轻人们使用机器语言的负担,开始采用一些助记符号来表示机器语言中的二进制指令,比如,用“ADD”代表加法,

“MOV”代表数据传递等。这样,人们很容易读懂并理解程序功能,纠错及维护都变得方便了,这种程序设计语言称为汇编语言。然而,计算机不能直接执行用汇编语言编写的程序,需要一种专门的程序负责将这些符号翻译成二进制表示的机器语言,这种翻译程序称为汇编程序。汇编语言依赖于机器硬件,移植性不好,但效率较高。

3. 高级语言

由于汇编语言依赖于硬件体系,且助记符量大难记,于是人们又发明了更加易用的高级语言。这种语言的语法和结构近似于人类自然语言中的英语,且由于远离对硬件的直接操作,使初学者更加易学易用。1954年,第一个完全脱离机器硬件的高级语言——FORTRAN 问世,40多年来,共有几百种高级语言出现,其中具有重要意义的有几十种,C语言就是典型的代表之一。

1.2 算 法

算法(algorithm)是对一个问题求解方法和步骤的一种描述。针对一个需要求解的问题,除了确定适合的数据结构外,关键是确定有效的算法,然后才能用具体的程序设计语言编写实现程序。对于一个问题,可实现的算法并不唯一,在保证算法正确的前提下,一般用算法的时间复杂度(算法执行所用时间)和空间复杂度(算法执行所需存储空间)区分各种算法的优劣。

1.2.1 算法的主要特征

所谓算法具有以下主要特征:

- (1) 有穷性。一个算法应能够在执行有限步后结束,并且每一步能够在有限的时间内完成。
- (2) 确定性。算法中的每一步都有确切的含义,不具有二义性。
- (3) 可行性。算法中的操作能够用已经实现的基本运算执行有限次来实现。
- (4) 零个或多个输入。零个输入就是算法已经确定了初始条件,不需要再输入数据。
- (5) 一个或多个输出。算法设计的目的是要获得问题的结果,因此需要将结果以输出的方式反馈给用户。

1.2.2 算法的描述方法

算法的描述方法有自然语言、伪代码、N-S图、流程图等。在此,仅简要介绍本书采用的算法描述方法——流程图。流程图是一种传统的算法描述方法,主要由图 1.1 中所示的几种图形组成。



图 1.1 流程图基本组成图形

起止框: 在框内标注“开始”表示程序开始,在框内标注“结束”表示程序结束,一个完整的流程图始末必须是起止框。

处理框：是用来表示执行赋值、计算、传送运算结果等的图形符号，算法中处理数据需要用到的算式、公式等根据执行顺序分别写在不同的处理框中。

条件判断框：一般有一个入口和两个出口，在条件成立的出口处需注明“是”或“Y”，在条件不成立的出口处需注明“否”或“N”。如果是多分支判断，则可有多个出口。

流程方向指示线：带箭头的流线表示执行的先后顺序。

【例 1.1】 输入两个数，找出其中较大的数。

此算法的流程图如图 1.2 所示，具体执行过程为：

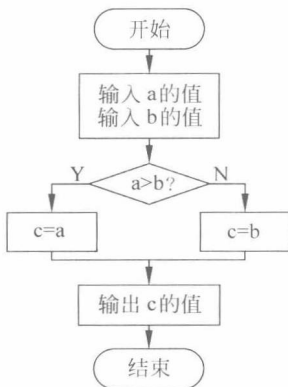


图 1.2 例 1.1 算法流程图

- (1) 算法开始。
- (2) 输入两个数，分别存放到变量 a、b 中。
- (3) 如果 a 大于 b，则将 a 赋给变量 c；否则将 b 赋给变量 c。
- (4) 输出变量 c 中的值，即较大数。
- (5) 算法结束。

可以看出，用流程图表示算法，形象直观，逻辑清晰，交流方便。当算法不太复杂时，采用流程图进行描述不失为一种较好的方法。

1.3 C 语言简介

1.3.1 C 语言发展历程

C 语言是当今世界最流行的程序设计语言之一，它比较接近硬件，有着和汇编语言比较相近的高效率，但又比汇编语言直观易懂。用 C 语言既可以编写系统软件如 UNIX、Linux，也可以编写应用软件 MATLAB，还可以进行嵌入式系统开发。C 语言虽然不适合开发 Windows 应用程序，但也是 Windows 应用程序开发语言（如 C++、C#）的基础。

1960 年出现的 ALGOL 60 是一种面向问题的高级语言，但它不接近于硬件，不适宜系统程序的编写。1963 年英国剑桥大学推出了 CPL (Combined Programming Language) 语言。CPL 语言虽然比 ALGOL 60 接近硬件一些，但规模较大。1967 年剑桥大学的 Martin Richards 对 CPL 语言进行了简化，推出了 BCPL (Basic Combined Programming Language)

语言。1970 年美国贝尔实验室的 Ken Thompson 又将 BCPL 语言进一步简化而设计出 B 语言(取 BCPL 的首字母),并用 B 语言开发了 UNIX 操作系统。1972—1973 年,贝尔实验室的 D. M. Ritchie 又设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 B 语言精简、接近硬件的优点,又克服了其无数据类型等缺点。

此后,C 语言又改写了多次,直到 1978 年,贝尔实验室才正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的 *The C Programming Language* 一书,本书中的 C 语言被称为标准 C。后来由美国国家标准学会在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C;1987 年再次颁布新标准,称之为 87 ANSI C。1990 年,国际标准化组织 ISO 将 87 ANSI C 作为 ISO C 的标准。目前所使用的 C 编译系统均以 ISO C 作为基础,但不同版本如 Microsoft C、Turbo C 和 Quick C 等稍有不同。本书的内容基本上是以 87 ANSI C 为基础。

1.3.2 C 语言的特点

C 语言之所以经久不衰,是因为其本身具备不同于其他语言的突出特点。

(1) C 语言生成的目标代码质量高。它可以直接对硬件进行操作,可以直接访问地址,能进行位(bit)运算,因此,C 语言源程序生成的目标代码质量很高。实验表明,C 语言代码效率只比汇编语言代码效率低 10%~20%。

(2) C 语言简练、紧凑,使用方便、灵活。C 语言严格区分大小写,一共有 32 个全是小写字母的关键字和 9 种流程控制语句。相对其他计算机语言,较容易学习和记忆,源程序较短,编写程序时工作量较少,容易编写和调试。

(3) C 语言功能全面。C 语言有结构化的流程控制语句,有实现程序模块化的函数;数据类型丰富,能实现各种复杂的数据结构的运算,指针类型的引入使程序的效率更高;运算符众多,从而实现了运算类型、表达式类型的多样化;C 语言系统提供的函数库进一步增强了 C 语言功能。

(4) C 语言程序的可移植性好。C 语言程序本身不依赖于机器硬件系统,基本上不用修改就可以应用于硬件结构不同的计算机和各种操作系统。

(5) C 语言程序设计自由度大,语法限制不太严格。C 语言书写格式自由,语法检查宽松,给编程人员较大的自由空间。这对于熟练的程序员是有益的,但也加大了初学者的学习难度。C 语言对数组的下标是否越界、指针变量是否已赋初值等不做检查,导致程序容易出现运行错误或逻辑错误。因此初学者一定要严格检查、验证程序,不要认为只要程序编译、链接成功就编写成功了。

C 语言还有一些其他优点,读者需要在学习和实践中慢慢体会。虽然 C 语言也有一些缺点,如类型转换较随意、运算优先级太多难以记忆等,但因其上述突出的优点,仍然是非常优秀的程序设计语言之一。

1.3.3 C 语言程序示例

在学习 C 语言的具体语法之前,先通过一个简单的 C 语言程序示例,初步了解 C 语言程序的基本结构。

【例 1.2】 已知两数 a 和 b,按公式 $s=a+a*b$ 计算,并输出结果。

```
预处理命令→#include <stdio.h>           //编译预处理命令
main 函数→void main()                     //主函数
{
    {
        int a,b,s;                         //定义变量 a,b 和 s
        a=3;                             //给变量 a 赋值
        b=5;                             //给变量 b 赋值
        s=a+a*b;                         //按公式计算
        printf("a+a*b= %d\n",s);         //输出变量 s 的值
    }
}
```

由例 1.2 可以看出 C 语言程序的构成特点如下:

(1) C 程序由若干函数组成,其中一个特殊的函数是主函数 main。一个 C 程序必须有且只能有一个 main 函数,它是程序执行的入口。

(2) 一对花括号括起来的是 main 函数的函数体。函数体由若干以分号为结束符的语句组成。C 语言中语句的书写非常自由,如“a=3;b=5;s=a+a*b;”这三条语句既可以写在一行,也可以每条语句单独占一行,既可以左端对齐,也可以不对齐。但为了提高程序的可读性,建议一条语句占一行,相同级别的语句要左对齐。

(3) 以“//”开头的是 C 程序的注释。注释是为程序语句添加的功能说明信息,目的是增加程序的可读性,程序在进行编译和链接时会把注释忽略掉。“//”为单行注释,只能将其后当前行的信息作为注释处理,如果要连续的多行信息作为注释处理,可以采用在需要注释信息的第一行行首加“/*”,在最后一行行尾加“*/”。

(4) “#include <stdio.h>”是一条编译预处理命令,需要放在程序的最前面,应注意的是该命令最后没有分号。程序中用到的 printf 是包含在 stdio.h 中的函数。C 语言编译系统为用户提供了很多库函数,根据功能分别包含在不同的头文件中。如 math.h 中包含了一些和数学有关的库函数,如求平方根、正弦、余弦等;string.h 中包含了和字符串处理相关的函数。如果用这些库函数,就需要用#include 命令将相应的头文件包含进来。

1.3.4 C 语言程序书写约定

虽然 C 语言程序对书写格式要求很低,但为了提高程序的可读性、可调试性和可维护性,养成良好的程序设计风格,建议读者书写程序时遵守如下约定:

(1) 一条语句或命令或左、右花括号均单独占一行。

(2) 用分层缩进的写法显示嵌套结构层次。同一个层次相应的左花括号和右花括号对齐,层次中的语句缩进一个 Tab 键的位置。在 VC++6.0 中,只要书写满足第一条约定,程序会自动调整对齐,非常方便。

(3) 标识符尽量做到“顾名思义”,可采用和其实际含义有关联的单词或单词组合,如用标识符 length 表示长度。

(4) 适当地加入注释信息,不同的功能块之间用一个空行隔开。

1.4 C 语言程序开发步骤

1.4.1 C 语言程序开发过程

C 语言程序的开发,一般要经过编辑、编译、链接和运行 4 个步骤,如图 1.3 所示。

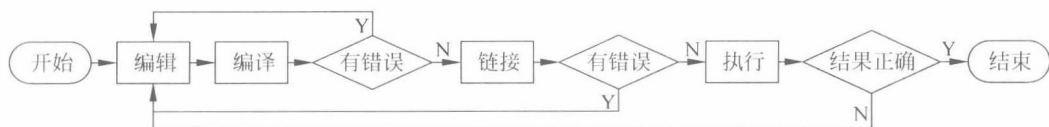


图 1.3 C 语言程序开发过程示意图

1. C 语言程序的编辑

针对一个实际问题,首先要根据题意设计问题求解算法,可以先用流程图等把算法表示出来,然后转换为 C 语言程序,也可以直接把算法用 C 语言程序的形式表示出来。编辑就是将写在纸上或记在头脑中的 C 语言程序输入到计算机中,以文件的形式存放在磁盘上。这种在编辑方式下建立起来的程序称为源程序,C 语言的源程序扩展名为 .c。

2. C 语言程序的编译

源程序是用 C 语言写的,而计算机能直接识别的只有由二进制指令组成的机器语言,因此,需要一个称为编译器的程序把源程序翻译成机器语言,这个过程称为编译。编译器创建的机器语言指令称为目标代码,包含目标代码的文件称为目标文件,一般目标文件的文件名与相应的源程序文件名相同,但扩展名为 .obj。

3. C 语言程序的链接

目标程序还不能直接在机器上运行,需要一个称为链接程序的程序把程序中用到的库函数和多个目标文件链接为一个扩展名为 .exe 的可执行文件。

4. C 语言程序的运行

运行可执行文件,就可以得到程序运行结果。

无论是在编译、链接还是运行阶段,如果发现错误,都必须返回到编辑阶段对源程序进行修改,然后重新编译、链接、运行,直到成功为止。

1.4.2 VC++6.0 环境中 C 语言程序运行步骤

Visual C++6.0(简称 VC++6.0)是微软公司为 C++ 语言设计开发的集成编译环境,同时兼容 C 语言。因此,C 语言和 C++ 语言程序的编辑、编译、链接、运行以及调试全过程均可在此环境中完成。在此,以例 1.2 为例简要介绍 C 语言程序在 VC++6.0 中的运行方法和步骤。