



普通高等教育“十二五”规划教材

PUTONG GAODENG JIAOYU "12·5" GUIHUA JIAOCAI

C 语言程序设计

主编◎肖蕾



冶金工业出版社
Metallurgical Industry Press

TP 312C
201437



普通高等教育“十二五”规划教材

C 语言程序设计

主 编 肖 蕾

副主编 郭乐江 胡亚慧 方其庆

编 委 江 巍 程 敏 涂文婕

北 京

冶金工业出版社

2013

内 容 简 介

本书主要内容有:C 语言程序设计基础、算法及其程序设计的方法、基本数据类型与运算、控制语句、数组、函数、指针、编译预处理、结构体与共同体、文件的操作等内容。此外,本书还为读者介绍了 Visual C++6.0 编译系统的使用及 C 语言程序语法错误的调式。

本书可作为高等学校计算机类、信息类专业 C 语言程序设计课程的教材,又可以作为工程技术人员的自学参考书。

图书在版编目(CIP)数据

C 语言程序设计/肖蕾主编. —北京:冶金工业出版社,2013.7

普通高等教育“十二五”规划教材

ISBN 978-7-5024-6351-9

I. ①C… II. ①肖… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 139193 号

出 版 人 谭学余

地 址 北京北河沿大街嵩祝院北巷 39 号,邮编 100009

电 话 (010)64027926 电子信箱 yjcbs@cnmip.com.cn

ISBN 978-7-5024-6351-9

冶金工业出版社出版发行;各地新华书店经销;北京明兴印务有限公司印刷

2013 年 6 月第 1 版,2013 年 6 月第 1 次印刷

787mm×1092mm 1/16; 17 印张; 478 千字; 271 页

33.00 元

冶金工业出版社投稿电话:(010)64027932 投稿邮箱:tougao@cnmip.com.cn

冶金工业出版社发行部 电话:(010)64044283 传真:(010)64027893

冶金书店 地址:北京东四西大街 46 号(100010) 电话:(010)65289081(兼传真)

(本书如有印装质量问题,本社发行部负责退换)

前 言

在计算机教育方面，C 语言是为数不多的与国外保持内容同步的课程之一，可见它的地位之重要不言而喻。由于 C 语言是许多后续专业课程的基础，因此，在本科生教学体系中，C 语言教学仅是起点，而非终点。C 语言既具有低级语言可直接访问内存地址、能进行位操作、程序运行效率高的优点，又具有高级语言运算符和数据类型丰富、结构化控制语句功能强、可移植性好的优点，因此成为程序设计语言的常青树。通过学习，学生可以掌握 C 语言的程序结构、语法规则和编程方法，达到独立编写常规 C 语言应用程序的能力，同时为设计大型应用程序和系统程序打下坚实的基础。本课程是数据结构、面向对象程序设计、操作系统和软件工程等课程的基础，并可为这些课程提供实践工具。

本书以当前广为使用的 Visual C++6.0 编译系统为实现的版本，全面系统地介绍了 C 语言及其程序设计方法。全书共 12 章，第 1 章主要介绍 C 语言程序设计基础，第 2 章介绍了介绍了算法的描述和结构化程序设计的基本方法，第 3 章介绍了 C 语言的基本数据类型、运算符与表达式，第 4~6 章介绍了 C 语言的 3 种基本结构（顺序结构、选择结构和循环结构），第 8 章介绍了函数的定义和使用；第 7、9、10 章介绍了数据的构造类型（包括数组、字符串、结构体和共同体等）和指针类型；第 11 章介绍 C 语言的编译预处理，第 12 章介绍了文件的概念。本书为读者展示了 C 语言灵活、精致的编程方法和在工程、科研中的应用，力求做到 C 语言知识和应用开发能力的融会贯通。

本书总结了作者多年教授 C 语言程序设计的教学与实践的经验。在本书编写的过程中，胡亚慧、程敏，涂文婕，江巍花费了大量的时间和精力，编写教材，同时，得到了陈芳信、郭乐江、方其庆的支持，在此对他们表示感谢。

由于编者水平有限，书中存有不妥之处，殷切希望广大读者批评指正！

作者

2013 年 1 月



目 录

1 C 语言概述.....	(1)
1.1 引言	(1)
1.2 C 语言基础.....	(3)
1.3 C 语言编制程序的几个步骤.....	(7)
1.4 C 语言程序的运行环境.....	(9)
1.5 典型问题	(17)
1.6 小结	(18)
2 算法概述	(20)
2.1 算法简介	(20)
2.2 算法描述形式	(22)
2.3 算法分析	(26)
2.4 小结	(30)
3 数据类型、运算符与表达式	(32)
3.1 C 语言的数据类型	(32)
3.2 常量	(33)
3.3 变量	(36)
3.4 数据类型的转换	(43)
3.6 运算符	(44)
3.7 算术运算符和算术表达式	(46)
3.8 赋值运算符和赋值表达式	(50)
3.9 关系运算符和关系表达式	(52)
3.10 逻辑运算符和逻辑表达式	(53)
3.11 位运算符	(55)

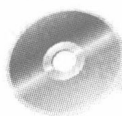


3.12	条件运算符	(57)
3.13	逗号运算符	(59)
3.14	sizeof 运算符	(61)
3.15	强制类型转换运算符	(62)
3.16	指针运算符	(63)
3.17	典型问题	(63)
3.18	本章小结	(66)
4	顺序结构程序设计	(68)
4.1	程序结构概述	(68)
4.2	C 语句概述	(70)
4.3	数据的输入与输出	(73)
4.4	顺序结构程序设计举例	(82)
4.5	小结	(84)
5	选择结构程序设计	(86)
5.1	典型的选择问题的程序设计思路	(86)
5.2	if 语句	(86)
5.3	if 语句的嵌套	(88)
5.4	switch 语句	(97)
5.5	选择结构程序举例	(100)
5.6	小结	(101)
6	循环结构程序设计	(103)
6.1	典型的循环问题程序设计思路	(103)
6.2	用 goto 和 if 语句一起构成循环	(104)
6.3	while 循环结构	(105)
6.4	do...while 循环结构	(106)
6.5	for 循环结构	(109)
6.6	循环的嵌套	(112)
6.7	break 和 continue 语句	(115)
6.8	循环结构程序设计举例	(118)

6.9	小结	(131)
7	数组	(133)
7.1	数组的定义	(133)
7.2	一维数组	(133)
7.3	二维数组	(139)
7.4	字符数组和字符串	(143)
7.5	小结	(147)
8	函数	(148)
8.1	函数的概念	(148)
8.2	函数的调用	(152)
8.3	函数的嵌套调用	(153)
8.4	函数的递归调用	(154)
8.5	数组作为函数参数	(157)
8.6	变量的作用域和生命期	(161)
8.7	函数的储存类型	(163)
8.8	内部函数和外部函数	(166)
8.9	小结	(168)
9	指针	(169)
9.1	地址和指针的概念	(169)
9.2	指针与变量	(170)
9.3	指针与数组	(179)
9.4	指针与字符数组	(188)
9.5	指向数组的指针和指针数组	(190)
9.6	指针作为函数的参数	(195)
9.7	指向指针的指针	(200)
9.8	main 函数的参数	(202)
9.9	有关指针的数据类型和指针运算的小结	(203)
10	结构体与共用体	(205)
10.1	结构体	(205)



10.2	特殊结构体	(209)
10.3	共用体	(221)
10.4	共用体和结构体的内存差异	(223)
10.5	动态存储分配	(224)
10.6	链表	(226)
10.7	枚举类型	(230)
10.8	小结	(232)
11	编译预处理	(234)
11.1	编译预处理的概念	(234)
11.2	宏定义	(234)
11.3	条件编译	(239)
11.4	文件包含	(240)
11.5	常用的编译预处理命令	(242)
11.6	小结	(246)
12	文件	(248)
12.1	文件的概念	(248)
12.2	文件的操作	(249)
12.3	文件的程序应用	(254)
12.4	常用的文件操作函数	(257)
12.5	小结	(259)
	参考文献	(264)



1 C 语言概述

C 语言作为一种计算机高级语言,既适合编写系统程序又适合编写应用程序,同时具有汇编语言和高级语言的双重特性,具有强大的功能,是当今最受欢迎的计算机语言之一。

本章教学基本要求:了解计算机语言的发展;掌握 C 语言程序的组成及其 C 语言程序的调试步骤;理解 C 语言的基本结构。重、难点内容是 C 语言程序的组成和正确调试程序的过程。

1.1 引言

1.1.1 计算机语言与人类语言

自然语言是人类为了交流思想、表达感情、交换信息,因而逐步发明了语言。人工语言是人类为了某种专门需要,而发明的一些新的交流工具。现在人类为了能够更好地与计算机进行交流与通讯,发明了专门与计算机打交道的交流工具——程序设计语言。



图 1.1 自然语言的构成



图 1.2 程序语言的构成

自然语言构成,如图 1.1 所示,的最基本的单元是字,字和语法构成词或者词组,词或者词组和语法一起构成句子或者段落,最后才能形成一篇文章。程序设计语言,如图 1.2 所示的最基本单元是由常量、变量、函数等构成,也就是自然语言中的词或者词组。常量、变量、函数等构造语句;最后形成的文章也就是程序。

计算机语言和自然语言是有很多相通之处,是一门跨国界的语言,通过计算机语言编程,可以表达自己的思想,相互交流。



1.1.2 计算机语言的发展

计算机程序设计语言的发展经历了从机器语言、汇编语言到高级语言的过程。计算机内存储器中存储的所有信息都是二进制形式的代码,指令也是以二进制代码的形式存储的。对于某种型号的 CPU,它能够执行的所有指令的集合就是它的指令系统,也称为机器语言。机器语言全部由 0 和 1 组成,难学、难记、难读,只有少数专业人员可以使用。

1.1.2.1 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数,二进制是计算机语言的基础。计算机发明之初,人们只能用计算机的语言去命令计算机干这干那,一句话,就是写出一串串由“0”和“1”组成的指令序列交由计算机执行,这种语言,就是机器语言。由于计算机的指令系统往往各不相同,所以,在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序,这造成了重复工作。但由于机器语言使用的是针对特定型号计算机的语言,故而其运算效率是所有语言中最高的。机器语言是第一代计算机语言。

例 1.1 用机器语言计算 $1055 - (383 + 545)$ 的 5 条机器指令如下:

```
B8 7F 01
BB 21 02
03 D8
B8 1F 04
2B C3
```

每一条机器指令由操作码和操作数组成。虽然可以直接被计算机执行,但是记不住、难理解、效率低、不易维护。

1.1.2.2 汇编语言

为了减轻使用机器语言编程的负担,人们进行了一种有益的改进:用一些简洁的英文字母、符号串来替代特定指令的二进制串,比如,用“ADD”代表加法,“MOV”代表数据传递等。这样一来,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了。这种程序设计语言就称为汇编语言,即第二代计算机语言。然而计算机是不认识这些符号的,这就需要一种程序,专门负责将这些符号翻译成由二进制数表示的机器语言,这种翻译程序被称为汇编程序。

汇编语言同样十分依赖机器硬件,移植性不好,但效率仍十分高,针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼而质量高,所以至今仍是一种常用且强有力的软件开发工具。

例 1.2 用汇编语言程序计算 $1055 - (383 + 545)$ 。

```
MOV AX 383      /* 将 383 传送到 AX 寄存器 */
MOV BX 545      /* 将 545 传送到 BX 寄存器 */
ADD BX AX       /* 将 BX 内容加 AX 内容,结果在 BX 中 */
MOV AX 1055     /* 将 1055 传送到 AX 寄存器 */
SUB AX BX       /* 将 AX 内容减 BX 内容,结果在 AX 寄存器中 */
```

1.1.2.3 高级语言

人们逐渐意识到应该设计一种这样的语言:这种语言接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用。经过努力,1954 年,第一个完全脱离机器硬件的高级语言——FORTRAN 问世了。50 多年来,共有几百种高级语言出现,有重要意义的有几

十种,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PASCAL、C. PROLOG、Ada、C++、VC、VB、Delphi、JAVA 等。

例 1.3 用 C 语言程序计算 $1055 - (383 + 545)$ 。

```
.....  
a=1055;  
b=383;  
c=545;  
s=a-(b+c);  
.....
```

从以上 3 个例子可以看出,C 语言使程序设计的难度降低,使计算机的发展进入新的阶段。

1.2 C 语言基础

1.2.1 C 语言的发展

C 语言是美国贝尔实验室的 D. M. Ritchie 于 1972 年设计实现的。C 语言直接来源于 B 语言,但它的根源可以追溯到 ALGOL60。ALGOL60 结构严谨,其设计者非常注重语法、分程序结构,因此对于后来许多重要的程序设计语言,如 PASCAL 等都产生过重要的影响。但它是面向过程的语言,与计算机硬件相距甚远,不适合编写系统软件。

1963 年,英国剑桥大学在 ALGOL60 的基础上推出更接近硬件的 CPL 语言,但 CPL 太复杂,难于实现。1967 年,剑桥大学的 Martin Richards 对 CPL 语言作了简化,推出了 BCPL 语言。1970 年,贝尔实验室的 Ken Thompson 以 BCPL 为基础,设计了更简单也更接近硬件的 B 语言(取 BCPL 的第一个字母)。B 语言是一种解释性语言,功能上也不够强,为了很好地适应系统程序设计的要求,Ritchie 把 B 语言发展成称之为 C 的语言(取 BCPL 的第一个字母)。C 语言既保持了 BCPL 语言和 B 语言的优点,又克服了它们的缺点。1973 年,K. Thompson 和 D. M. Ritchie 两人合作把 UNIX 的 90% 以上用 C 改写,即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室的 K. Thompson 和 D. M. Ritchie 开发成功的,是用汇编语言写的,这样 Unix 使分散的计算系统之间的大规模联网以及互联网成为可能。后来,C 语言多次作了改进,但主要还是在贝尔实验室内部使用。

直到 1975 年 UNIX 第 6 版公布后,C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使 C 移植到其他机器时所需做的工作大大简化了,这也推动了 UNIX 操作系统迅速地在各种机器上实现。例如,VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用,C 语言也迅速得到推广。C 语言和 UNIX 可以说是一对孪生兄弟,在发展过程中相辅相成。

1978 年以后,C 语言已先后移植到大、中、小、微型机上,如 IBM System/370、Honeywell 6000 和 Interdata 8/32,已独立于 UNIX 和 PDP 了。现在 Windows 操作系统下大部分代码用标准 C 语言开发,单片机系统中 C 语言逐渐取代汇编语言,C 语言也成为 Unix/Linux 操作系统上的编程语言、Palm 等嵌入式系统下支持的开发语言、黑客和网络安全以及 Tivoli 网管软件的开发语言。现在 C 语言已风靡全世界,成为世界上应用最广泛的几种计算机语言之一。

1.2.2 C 语言的特点

(1) 语言简洁、紧凑、灵活。



C 语言一共只有 32 个关键字,9 种控制语句,见表 1.1,常用符号包括标识符、关键字、常量、字符串、操作符、分割符等组成。程序书写形式自由,标识符的定义非常灵活,支持大小写敏感。

表 1.1 C 语言的 9 种控制语句

if()~else~	for()~	while()~
do~while()	continue	break
switch	goto	return

(2)运算符和数据类型丰富。

C 语言的运算范围很宽,具有非常丰富的运算符,如:自增(++)、自减(--)、位运算符等,共有 34 种运算符。运算符是告诉编译器对运算对象进行某种操作运算的符号。C 语言常用的运算符有三类:算术运算符、关系与逻辑运算符、位运算符。除此之外,还有一些用于完成特种任务的特种运算符,如:强制类型转换运算符(())、条件运算符(?:)、赋值运算符、逗号运算符(,)和自增(++)、自减(--)运算符等。C 语言的数据类型也很多样化,包括整型、实型、字符型、数组类型、指针类型、结构类型、联合类型。

(3)程序设计结构化、模块化。

C 语言是结构化语言,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,这些函数可方便的调用,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。程序从主函数开始,函数间可相互调用。

(4)生成目标代码质量高。

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。C 语言可以象汇编语言一样对位、字节和地址进行操作,能够直接访问操作内存和进行内存的地址运算。直接支持比特(bit)运算,如移位运算等。它的语法比较灵活,允许程序编写者有较大的自由度。一般只比汇编程序生成的目标代码效率低 10%~20%。

(5)可移植性好。

C 来自于 B 语言。B 语言的精华就是完全参照诺依曼体系结构实现的语言,所以 C 也就继承了 B 的这种特点,在各个平台上,都有相应的 C/C++ 编译器。相对于其他语言来说,C 语言的移植性和通用性要好得多。

1.2.3 C 语言程序的组成

为了说明程序结构的特点,先看以下几个程序。这几个程序由简到难,表现 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解到组成一个 C 语言源程序的基本部分和书写格式。

```
例 1.4 第一个程序 Hello, World!  
/* example1.1 The first C Program */  
#include <stdio. h>  
main()  
{  
    printf("Hello, World!");  
}
```

输出:



Hello, World!

在本程序中,需要注意以下几点:

(1) `/* ... */` 为程序注释部分,只是用于解释程序,对编译和运行不起任何作用。

(2) `#include <stdio.h>` 也可以写成 `#include "stdio.h"`,表示是文件包含命令,表示把 `<>` 或 `"` 内的 `stdio.h` 文件包含到本程序中来, `stdio` 为 `standard input/output` 的缩写,即 C 语言标准输入输出函数的格式均定义在这个文件里。扩展名为 `.h` 的文件称为头文件。

(3) `main` 表示主函数,每一个 C 程序必须有,且只能有一个主函数,它是程序执行的入口,后面的 `()` 为空,表明没有参数。

(4) 程序中的 `{` 为函数开始的符号, `}` 为函数结束的符号,括起来的部分称为函数体。函数体内的 `printf` 函数是 C 语言中定义的标准输出函数,可在程序中直接调用,它的功能是把输出的内容送到显示器上去显示。即把双引号内的字符串按原样输出。

例 1.5 计算两个整数 `a, b` 之和。

```
/*      example1.2      calculate the sum of a and b */
#include <stdio.h>

/* This is the main program */
main()
{
    int a, b, sum;
    a = 10;
    b = 24;
    sum = add(a, b);
    printf("sum = %d\n", sum);
}

/* This function calculates the sum of x and y */
int add(int x, int y)
{
    int z;
    z = x + y; return(z);
}
```

在本程序中,需要注意以下几点:

(1) 语句 `"int a, b, sum;"` 称为变量定义语句, `"int"` 是系统保留的关键字, `"a, b, sum"` 是用户引入的变量,是为通知编译器为这些变量分配内存空间。

(2) 程序由两个函数组成,即主函数 `main` 和用户自定义函数 `add` 组成。函数 `add` 的功能是求两个整数之和并返回给主函数。语句 `"int add(int x, int y)"` 表明函数 `add` 是有两个整型的形参 `x` 和 `y` 并返回一个整型类型值 `z` 的函数。

(3) `%d` 是输入输出的格式化字符串,用来指示输入输出的数据类型和格式转换, `%d` 表示十进制整数类型。

本程序执行过程为:先把 `a, b` 的值读入,然后调用 `add` 函数,并把 `a` 和 `b` 的值传递给 `add` 函数的参数 `x` 和 `y`,在 `add` 函数中计算 `x` 和 `y` 的和赋给变量 `z`,并由 `return` 语句把变量 `z` 的值返回给主函数 `main`,然后赋值给变量 `sum`,最后由 `printf` 函数输出 `sum` 的值。

从以上例子可以分析出 C 语言程序的基本特点、基本组成结构为:

(1) 一个 C 语言程序可以有一个或多个文件组成。每个文件可由一个或多个函数组成。

(2) 函数有两类:库函数和用户自定义函数。库函数是系统提供的,用户可直接调用,如输入函



数和输出函数等。对于库函数中没有的功能,需要用户自己实现,即用户自定义函数。用户自定义函数的数目由程序的复杂度决定。

(3)函数是构成 C 语言程序的基本单位,它完成相对独立的功能。它由函数头和函数体两部分组成。函数的一般形式为:

函数名(参数表)

{数据说明部分;

执行语句部分;}

函数名小括号中的“参数表”可以根据需要存在,可能有参数,也可能没有参数。若有多个参数,它们之间应该用逗号隔开。若没有参数,则参数说明部分可以省略。但函数名后的一对圆括号不能省略。函数名和参数说明部分统称为函数说明部分。一对大括号中间包含的全部内容叫做函数体。

(4)程序由一个或多个函数组成;必须有且只能有一个主函数 `main()`;程序执行从 `main` 开始,在 `main` 中结束,其他函数通过嵌套调用得以执行。

(5)程序体部分全部放在“{}”内,在程序部分主要包括两种语句,分别为声明语句和执行语句。声明语句主要对使用的变量和函数做出声明,执行语句体现了函数的功能。语句是构成程序的基本单位。

(6)一个好的程序一定要有适当的注释,即 `/* ... */`。* 和 / 之间不能有空格;不能嵌套;不产生编译代码;允许出现在程序的任何位置。

由此可见一个完整的 C 语言程序一般包含以下 4 个部分:

(1)编译预处理命令。

(2)全局变量及函数说明。

(3)`main()` 函数。

(4)用户自定义函数。

1.2.4 C 语言字符集

1.2.4.1 C 语言的基本字符集

字符是组成语言的最基本的元素。C 语言字符集由字母,数字,空白符,标点和特殊字符组成。在字符常量,字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

A. 字母

小写字母 `a~z` 共 26 个。

大写字母 `A~Z` 共 26 个。

B. 数字

`0~9` 共 10 个。

C. 空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略不计。因此在程序中使用空白符与否,对程序的编译不发生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

D. 标点和特殊字符

空格 ! % * & --- (下划线) + = -

< > / \ ' " ; . , () {} 【】



1.2.4.2 C 语言的标识符

标识符是一个名称。不同的程序设计语言使用不同形式的标识符。C 语言规定:标识符只能由字母(A~Z 和 a~z)、数字(0~9)和下画线(_)组成,并且首字符必须是字母或下画线。

由用户根据需要定义的标识符称为用户标识符。用户标识符一般被用来给变量、函数、数组、语句标号或文件等命名。例如:m,n,x,y,sum,max,color,song,yes_no 等均是合法的用户标识符;而-3x(以减号开头),123MYM(以数字开头),float(系统保留字)等均是非法的用户标识符。

ANSI C 标准没有规定标识符的长度(字符个数),但不同的 C 编译系统规定不同。MS C 规定标识符长度不能超过 8 个字符,超过 8 个字符时取前 8 个字符作为标识符。Turbo C 规定标识符长度不能超过 32 个字符,超过 32 个字符其后面的字符无效。

用户标识符是用户用于标识某个量的符号,因此,命名应尽量具有相应的意义,以便阅读理解。

注意:

(1)“见名知义”,如 sun(求和),max(最大),PI(π),BT(β)等。

(2)大小写有别。习惯上变量名、函数名等用小写,如 a,max()等;符号常量用大写,如 A,PI,BT 等。其中小写的 a 和大写的 A 是两个不同的标识符。

(3)用户标识符决不能与系统规定的保留字同名,如 int=3,float=3.0 是非法的;最好也不要与系统提供的标准库函数名同名。

(4)尽量不要用下画线开头,且长度不超过 8 位,原因是 Turbo C 2.0 系统内部使用了部分用下画线开头的标识符,如_fd,_cs,_ds,_ss 等十几个扩展关键字;虽然 ANSI C 不限制标识符的长度,但它受到各种版本 C 语言编译系统的限制,同时也受到具体机器的限制。

1.2.4.3 C 语言的关键字

C 语言中的关键字又称为保留字,它是由系统提供的,表示特定的语法成分,不能当做用户标识符使用。所有的保留字均为小写字母,专供系统本身使用。下面分类给出 ANSI C 规定的 32 个保留字。

A. 数据类型关键字,包括 char, double, enum, float, int, long, short, signed, struct, union, unsigned, void 共 12 个。

B. 控制语句关键字,包括 break, case, continue, default, do, else, for, goto, if, return, switch, while 共 12 个。

C. 存储类别关键字,包括 auto, extern, register, static 共 4 个。

D. 其他关键字,包括 const, sizeof, typedef, volatile 共 4 个。

1.3 C 语言编制程序的几个步骤

1.3.1 用自然语言写文章的步骤

(1)确定书的内容。

(2)选择适当的形式。

(3)确定书的格式。

(4)开始写书的内容。

(5)校验、修改文章。



(6)交付出版社审核,若审核通过,进行下一步;否则返回上一步,直至出版社审核通过为止。

(7)正式出版发行。

1.3.2 用计算机语言编制程序的步骤

(1)了解需求。

(2)确定核心算法。

(3)制定总体方案划分模块。

(4)画出程序流程图。

(5)编写程序源代码,检查源代码是否正确,若正确,进行下一步;否则修改语法错误,直至改对为止。

(6)编译源程序。

(7)链接库函数。

(8)执行程序。

在计算机语言编制程序的步骤中,从第5个步骤开始,是由程序员进行的工作,也即C语言的调试工作。下面将详细介绍C语言程序的调试过程。

1.3.3 C语言程序的调试过程

C语言程序的源代码不能被计算机执行。一般来讲,C程序开发步骤为:

如图1.3所示,C语言的设计过程应该分为以下4步:

第一步:编辑。

使用程序设计语言提供的编辑器编辑源程序,即录入程序代码,并保存源程序。C语言源程序的扩展名为.c。许多文本编辑器都可以用来编辑源程序,例如Windows的写字板、记事本以及Word等。C语言源程序的存储格式必须是文本文件格式,在保存时需加以注意。编写C程序时,习惯上使用小写英文字母,常量和和其他用途的符号可用大写字母。C语言对大、小写字母是有区别的。关键字必须小写。

第二步:编译。

程序编译是指将编辑好的源文件翻译成二进制目标代码的过程。目标程序文件的主文件名与源程序的文件名相同,扩展名为.obj。编译过程是使用C语言提供的编译程序完成的。不同操作系统下的各种编译器的使用命令不完全相同,使用时应注意计算机环境。编译时,编译器首先要对源程序中的每一个语句检查语法错误,当发现错误时,就在屏幕上显示错误的位置和错误类型的信息。此时,要再次调用编辑器进行修改,然后再进行编译,直至排除所有语法和语义错误。正确的源程序文件经过编译后在磁盘上将生成目标文件*.obj。

第三步:链接。

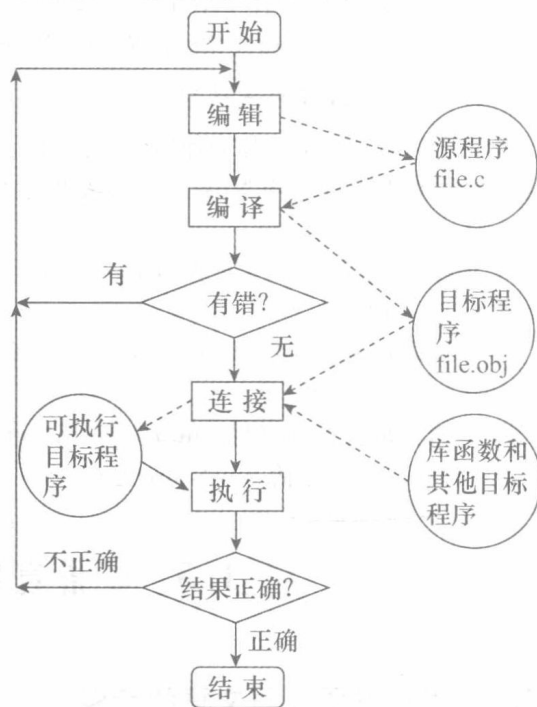


图 1.3 C语言程序的开发步骤



编译后产生的目标文件是可重定位的程序模块,不能直接运行。链接就是把目标文件和其他分别进行编译生成的目标程序模块及系统提供的标准库函数链接在一起,生成可以运行的可执行文件的过程。链接过程使用C语言提供的链接程序完成,生成的可执行文件存在磁盘中。可执行文件的扩展名为.exe。

第四步:运行。

通过上述四个过程得到的扩展名为*.exe的文件可以直接在操作系统下运行,不再依赖于具体的编译系统。运行完程序后,如果结果符合要术,是正确的,则整个程序设计过程结束。否则,必须进一步查找算法步骤中的错误并修改源程序,不断重复编辑—编译—链接—运行的过程,直到得到正确结果为止。

VC6.0集成开发环境提供了相应的编辑工具、编译工具和连接工具。一个源代码有时要经过多次的修改才能编译通过,因此这一步有时是很困难的。程序在编译时,如果不能通过,则会有错误提示信息,程序员要根据错误提示信息调试程序。

下面介绍实用最广泛的C语言集成开发环境——VC6.0。

1.4 C 语言程序的运行环境

1.4.1 VC6.0 集成开发环境介绍与使用

1.4.1.1 Visual C++6.0 的启动与退出

(1)启动 Visual C++6.0 的操作方法如下:

在 Windows 桌面上,单击【开始】|【程序】|【Microsoft Visual C++6.0】菜单,在下拉菜单中用鼠标点击 Microsoft Visual C++6.0 命令,启动 Microsoft Visual C++6.0 的主窗口,如图 1.4 所示。

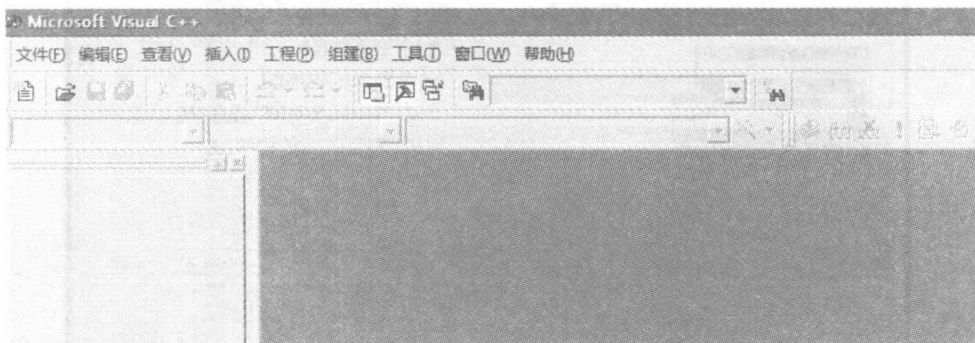


图 1.4 Visual C++6.0 的主窗口

(2)退出 visual c++6.0 的操作方法如下:

单击 Visual C++6.0 的主窗口的【文件】|【退出】菜单命令,或者单击窗口的【退出】按钮,就可以退出 Visual C++6.0。

1.4.1.2 新建工程

在“E:\c—study”目录下,新建一个名为“hello”的工程。新建工程 hello 的操作方法如下:

(1)启动 Visual C++。

(2)在 Visual C++主窗口中,单击【文件】|【新建】命令,弹出“新建”对话框。