

21

世纪中等职业教育系列教材

C语言 程序设计

主编：郭 辉

C^c_cC^c_cC^c_c



中国传媒大学出版社



10010101010111010010101010101011001101010101000101
100101010101010101100110101011010001010110101010111011



21

世纪中等职业教育系列教材

C语言 程序设计

主 编：郭 辉
副主编：郑宝昆

江苏工业学院图书馆
藏书章



中国传媒大学出版社



内 容 简 介

本书系统地介绍了C语言程序设计的基本理论和实用技术,适合于中等职业学校计算机及应用专业的学生作为学习和掌握C语言的一本教材。

本书针对职业教育的特点,注重在C语言学习过程中的基础性、实用性和操作性,目标是培养学生的实践能力和自学能力。本书的主要内容包括:C语言概述、基本数据类型、简单程序设计、控制语句、数组、函数、编译预处理、指针、结构与联合、文件。各章都配有学习要点和课后习题。

本书可作为中等职业学校计算机及应用专业或相关专业的教学用书,也可作为计算机等级考试的辅导用书,或者供C语言学习爱好者的参考用书。

图书在版编目(CIP)数据

C语言程序设计/郭辉主编. —北京:中国传媒大学出版社,2006.1

(21世纪中等职业教育系列教材)

ISBN 7-81085-666-9

I. C... II. 郭... III. C语言—程序设计—专业学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2005)第157714号

C语言程序设计

主 编 郭 辉
策 划 王 进 蔡开松
责任编辑 欣 文
责任印制 曹 辉
出 版 人 蔡 翔

出版发行 中国传媒大学出版社(原北京广播学院出版社)
北京市朝阳区定福庄东街1号 邮编 100024
电话:010-65450532 65450528 传真:010-65779405
<http://www.cucp.com.cn>

经 销 新华书店总店北京发行所

印 刷 北京市后沙峪印刷厂
开 本 787×1092mm 1/16
印 张 15.5
版 次 2006年1月第1版 2006年1月第1次印刷

书 号 ISBN 7-81085-666-9/K·666 定 价:20.00元

前言

QIAN YAN

本书是以 C 语言的入门级读者作为主要对象,根据 C 语言程序设计课程的特点,用编排独特、通俗易懂的方法编写的一本教材。全书按照 C 语言中由浅入深、从易到难的学习过程编写,使学生逐步对 C 语言从认识了解到掌握的学习步骤来学习 C 语言。其中主要包括这样几部分:第 1 部分(第 1 章和第 2 章)介绍了程序语言的发展历史、C 语言的基本概念和特点、简单的 C 语言程序、常量与变量的概念、C 语言数据类型、运算符和表达式等;第 2 部分(第 3 章至第 6 章)介绍了简单的程序设计和结构化程序设计的 3 种基本结构(顺序、选择和循环),数组的概念以及函数的定义和使用;第 3 部分(第 7 章至第 10 章)介绍了编译预处理、指针的概念、字符串的处理方法、结构体的概念、链表的基本操作、文件的读写操作等。

在教材编写过程中,始终贯彻课堂讲授与学生实践相结合的教学理念,以学生较熟悉、易理解的方式来阐述,采用启发式的教学方式,配合相应例题和习题引导学生对问题的理解和掌握。

本书适合中等职业学校的计算机及相关专业学生使用,编写过程中同时参照了教育部颁布的全国计算机等级考试大纲,结合实际,深入浅出,着重于对程序设计的实践和应用。希望通过本书的学习,使计算机语言的初学者能够掌握好 C 语言的语法规则,理解计算机程序设计的基本概念和程序语言的设计思想,为今后进一步学习打下基础。

由于作者水平有限,编写时间短促,书中难免有不妥之处,敬请广大读者批评指正。

编 者

2006 年 1 月

目录

MU LU

第一章 C 语言概述	(1)
1.1 程序设计语言的发展	(1)
1.2 C 语言概述	(4)
1.3 认识 C 程序	(5)
1.4 C 语言的字符集和词汇	(8)
1.5 C 程序的上机操作	(10)
思考与练习	(15)

第二章 数据类型、运算符与表达式

2.1 C 语言的数据类型	(17)
2.2 算术运算符和算术表达式	(24)
2.3 赋值运算符和赋值表达式	(26)
2.4 自增、自减运算符	(28)
2.5 逗号运算符和逗号表达式	(29)
思考与练习	(30)

第三章 简单程序设计

3.1 C 程序设计概述	(34)
3.2 输入/输出及其 C 语言的实现	(37)
3.3 数据输出函数	(38)
3.4 数据输入	(42)
3.5 顺序程序设计	(44)
思考与练习	(46)

第四章 控制语句

4.1 关系运算符和关系表达式	(49)
4.2 逻辑运算符和逻辑表达式	(50)
4.3 条件选择语句 if	(52)
4.4 开关语句 switch	(58)
4.5 循环语句	(59)
4.6 break 语句和 continue 语句	(65)
思考与练习	(67)

目录

第五章 数 组

5.1 数组概念的引入	(73)
5.2 数组的说明	(74)
5.3 数组的引用	(74)
5.4 数组的类型	(75)
5.5 数组赋初值	(77)
5.6 字符数组与字符串处理	(80)
思考与练习	(84)

第六章 函 数

6.1 概述	(87)
6.2 函数的定义和调用	(88)
6.3 函数的参数及其传递方式	(93)
6.4 函数的嵌套调用和递归调用	(96)
6.5 变量的作用域及其存储类型	(101)
6.6 内部函数和外部函数	(106)
思考与练习	(108)

第七章 编译预处理

7.1 宏定义	(112)
7.2 文件包含	(117)
7.3 条件编译	(118)
7.4 应用举例	(121)
思考与练习	(122)

第八章 指 针

8.1 指针与地址	(125)
8.2 指针变量的定义	(126)
8.3 指针变量的操作	(127)
8.4 指针与数组	(132)
8.5 字符串的指针和指向字符串的指针变量	(141)
8.6 指针与函数	(144)
8.7 void 型指针	(154)

目录

M U L U

8.8 指针小结	(154)
思考与练习	(156)
第九章 结构与联合	
9.1 结构类型	(161)
9.2 联合类型	(190)
思考与练习	(193)
第十章 文 件	
10.1 文件概述	(196)
10.2 文件类型指针	(197)
10.3 文件的打开与关闭	(198)
10.4 文件的读写	(199)
10.5 文件的定位	(206)
10.6 文件的出错检测与处理	(209)
10.7 非缓冲文件系统	(211)
10.8 文件应用举例	(212)
思考与练习	(213)
自测题	(219)
参考答案	(233)

第一章

C语言概述



学习提示

- 了解程序设计语言的发展过程;
- 了解C语言的特点、C语言与其它高级语言相比有什么异同;
- 掌握C语言程序的基本结构、书写格式;
- 了解C程序中基本的函数调用关系;
- 了解C语言的字符集和词汇;
- 掌握C程序在PC机上的建立、编译和运行过程。

建议学时数: 授课 3 学时 上机 2 学时

重点:

1. C语言的主要特点;
2. C语言在PC机上的运行过程及上机操作过程。



学习内容

1.1 程序设计语言的发展

计算机语言是指一切用于人与计算机通信的语言。编程语言也叫程序设计语言,包括机器语言、汇编语言、以及独立于具体计算机的高级语言。

1.1.1 机器语言

要使计算机按人的意图工作,就必须使计算机懂得人的意图,接受人向它发出的命令和信息。人要和机器交换信息就要解决一个“语言”的问题。计算机并不懂人类的语言(无论是中文或英文)。例如,我们写 $A+B=C$, 机器不能接受,它只能识别0和1两种状态。如光电输入机中纸带有孔的地方,代表1,无孔的地方,代表0。由0和1组成各种排列组合,通过线路转变成电信号,让计算机执行各种不同的操作。

这种直接用0和1组成的机器指令编写的程序,就是机器语言源程序。对计算机来说,这是它唯一能直接“听”得懂的语言。所以,常常称之为面向机器的语言。但是,对使用计算机的人来说,这是十分难懂的语言,它难读、难记、难写,容易出错,不同机型又不通用。显然人和机器之间的通信存在巨大的鸿沟,只有填补上这个鸿沟,使用的人愈是方便容易,机器又能懂得,

计算机才能发挥更大的作用。为此,人们研究了一种汇编语言。

1.1.2 汇编语言

把用二进制数表示的指令,用一些符号来表示,如用表示操作的英文缩写来代替指令代码,用 16 进制数表示数字。如 LD 代表取一个操作数指令,ADD 代表一个加法指令。这种用符号代替后的指令,就叫汇编语言,又称符号语言,像 LD、ADD 等这类符号称为指令符号或助记符。用汇编语言编写的程序,称为汇编语言源程序,常简称为汇编语言程序。

这种语言,相对机器语言就容易读、容易记、容易写了。但是机器却不能识别。因此,计算机是无法直接执行的。一个不懂汉语的外国人到中国来要和中国人直接交谈,那是无法进行对话的,所以,只好求助于翻译。在计算机中,也同样采取这种方法。人们编写程序用汇编语言,然后请一位翻译,把汇编语言程序翻译成机器能懂得的机器语言程序,这个翻译过程,叫做“汇编”。汇编后产生的机器代码称为目标程序。翻译可以由人手完成,但做起来既繁琐单调,又容易出错。实际上,我们是让计算机来做。因而就研制出了担任翻译的程序,取名叫汇编程序。汇编过程如图 1.1 所示。

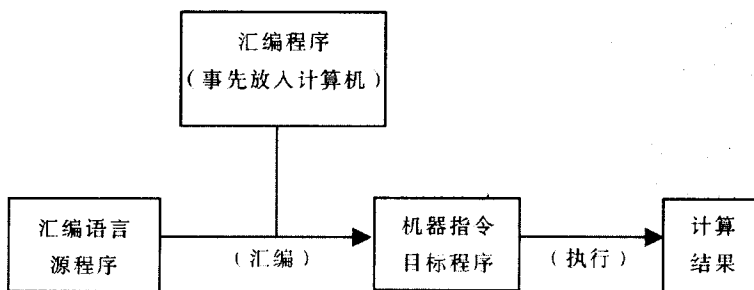


图 1.1 汇编过程

汇编语言使程序设计工作前进了一大步,但是仍然存在很多缺点:第一,不便于我们求解问题过程的描述。如一个数学公式,汇编语言的表达形式与人们的习惯表达形式差距很大;第二,它仍是面向机器语言。不同机型,汇编语言也不一样,因此用它编制的程序,没有通用性。为了克服这些不足之处,人们进一步研制出了高级语言。

1.1.3 高级语言

它是用更接近人的自然语言和数学表达式的一种语言,它由表达不同意义的“关键字”和“表达式”按照一定的语法规则组成,完全不依赖机器的指令系统。这样的高级语言为人们提供了很大的方便,编制出来的程序易读易记,也便于修改、调试,大大提高了编制程序的效率,也大大提高了程序的通用性,便于推广交流,从而极大地推动了计算机的普及应用。

例如,使用高级语言 BASIC,要想得到 $3 \times 8 \times \sin(\pi/2)$ 的结果,只需写出 `print 3 * 8 * sin(3.14159/2)` 即可,计算机将计算并输出结果。

我们看到高级语言离人们的理解愈加接近了,但离计算机的理解就愈远了。计算机是不能直接理解那些英语单词、数学表达式的。所以,为了填补人机之间的鸿沟,还是得求助于翻译。这种“翻译”,通常有两种做法,即编译方式和解释方式。编译方式是:事先编好一个称为编译程序的机器指令程序,并放在计算机中,把用高级语言编写的源程序输入计算机,编译程

序便把源程序整个翻译成用机器指令表示的目标程序。然后执行该目标程序,得到计算结果。如图 1.2(a)所示。

解释方式是:事先编好一个称为解释程序的机器指令程序,并放在计算机中,把用高级语言编写的源程序输入计算机。它并不像编译方式那样把源程序整个翻译成用机器指令表示的目标程序,而是逐句地翻译,译出一句立即执行,即边解释边执行。见图 1.2(b)。这种方式比编译方式费机器时间,但可以少占计算机的内存。

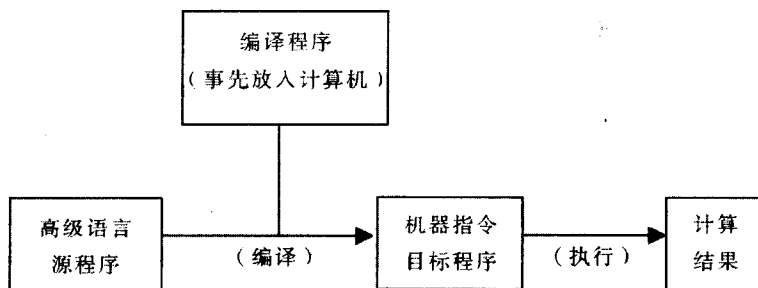


图 1.2(a) 编译过程

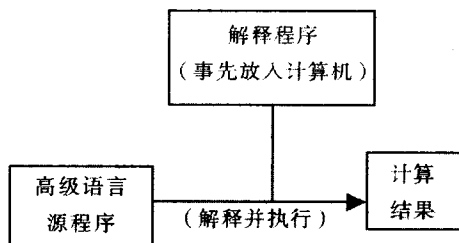


图 1.2(b) 解释过程

FORTRAN、ALGOL、COBO 等高级语言采用编译执行方式,而大多数 BASIC 高级语言采用解释执行方式。

由于编译(解释)程序代替了人工把高级语言源程序翻译为机器指令程序,大大节约了使用者的工作量。自从有了高级语言后,一般的科技人员和大、中学生以及职工,都能很快地学会使用计算机,可以完全不顾什么机器指令,也可以不必深入懂得计算机的内部结构和工作原理,就能方便地使用计算机进行各种科学计算或事务处理等。因此有人说,高级语言的出现是计算机发展中“最惊人的成就”。

目前,世界上已有一百多种高级语言,最流行的有几十种。例如:

1. FORTRAN(Formula Translator 的缩写):它是世界上最早出现的高级语言,从 1954 年问世以来,经过几次大的发展,功能有很大的增强。现在流行的是 FORTRAN 77 版本。它特别适于科学、工程计算。
2. COBOL(Common Business Language 的缩写):适于非数值计算的商业、管理领域。
3. PASCAL 语言是最早出现的结构化语言,适合于计算机教育。
4. PL/1 语言是一种大型语言,功能强,数值计算和数据处理均适用。
5. Ada 语言是一种工程化的大型语言,适合于大型软件工程。

6. C 语言是近年来广泛推广的结构化语言,适于编写系统软件。

7. BASIC 语言是一种简单会话式语言,在世界上应用最广泛。

1.2 C 语言概述

1.2.1 C 语言的发展过程

在 C 语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件,其可读性和可移植性都很差;但一般的高级语言又难以实现对计算机硬件的直接操作(这正是汇编语言的优势),于是,人们盼望有一种兼有汇编语言和高级语言特性的新语言。

C 语言是一种过程化的程序设计语言。C 语言的前身是 ALGOL 语言。1960 年,ALGOL60 出版,它是一种面向问题的高级语言,它离计算机硬件远,不宜用来编写系统程序。1963 年,剑桥在 ALGOL 基础上推出 CPL(复合程序设计语言),它接近硬件,但 CPL 规模大,难以实现,学习和掌握困难,没有流行。1967 年,剑桥的马丁·理查德(Martin Richards)对 CPL 进行了简化,推出 BCPL(基本复合程序设计语言)。这是一种计算机软件人员在开发系统软件时,作为记述语言使用的程序语言。1970 年美国贝尔实验室的肯·汤普逊(Ken Thompson)和丹尼斯·M·利奇(Dennis Ritchie)完成了 UNIX 的初版,与此同时,他们还改写了由 Martin Richards 开发的 BCPL 语言,形成了一种称为 B 的语言。此后,B 语言又进一步被进行了改进和完善,形成了称之为 C 的语言。

C 语言是在 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。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标准 C 语言。1983 年,美国国家标准委员会(ANSI)对 C 语言进行了标准化,于 1983 年颁布了第一个 C 语言标准草案(83 ANSI C),后来于 1987 年又颁布了另一个 C 语言标准草案(87 ANSI C)。

早期的 C 语言主要是用于 UNIX 系统。C 语言形成后,1973 年丹尼斯·M·利奇(Dennis Ritchie)把 UNIX 系统中的 90% 又用 C 语言进行了改写。随着 UNIX 的移植、推广,C 语言也得到移植和推广。C 语言同时具备低级语言和高级语言的特征,所以有人说它是中级语言。由于 C 语言本身强大的功能,自其面世以来,倍受广大程序员的青睐。到了 80 年代,C 开始进入其它操作系统,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。

可以说,C 语言发展的大致历程为:

ALGOL60 → CPL → BCPL → B → C

1.2.2 C 语言的特点

1. 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字,9 种控制语句,程序书写自由,主要用小写字母表示。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,而这是计算机最基本的工作单元。

2. 运算符丰富

C 的运算符包含的范围很广泛,共有种 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 的运算类型极其丰富表达式类型多样化,灵活使用各种运算符可以实现在其它高级语言中难以实现的运算。

3. 数据结构丰富

C 的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据类型的运算。并引入了指针概念,使程序效率更高。另外 C 语言具有强大的图形功能,支持多种显示器和驱动器。且计算功能、逻辑判断功能强大。

4. C 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。

5. C 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

6. C 语言允许直接访问物理地址,可以直接对硬件进行操作

因此既具有高级语言的功能,又具有低级语言的许多功能,能够象汇编语言一样对位、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,可以用来写系统软件。

7. C 语言程序生成代码质量高,程序执行效率高

一般只比汇编程序生成的目标代码效率低 10~20%。

8. C 语言适用范围大,可移植性好

C 语言有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,也适用于多种机型。

1.2.3 C 语言的版本

目前最流行的 C 语言有以下几种:

Microsoft C 或称 MS C

Borland Turbo C 或称 Turbo C

AT&T C

这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便、完美。在本书以 ANSI C 为基础叙述。

1.3 认识 C 程序

1.3.1 Hello World 程序

我们首先通过一个简单的程序例子,来大致了解 C 程序的特点。

C 语言的 Hello World 程序。该程序的功能是在屏幕上显示“Hello World!”的信息,这种程序通常作为开始学习编程语言时的第一个程序:

程序代码如下:

【例 1.1】

```
#include <stdio.h>
void main( ) /* void 是函数的返回值类型,为空 */
{
    printf("Hello World!");
}
```

如果在 C 语言的编译器下运行该程序,将会在计算机屏幕上显示:

Hello World!

这样的信息。

通过观察,我们发现 C 程序由:

```
main( )
{
    .....
}
```

这样的框架构成,该框架称为主函数或 main 函数。其中,void 是空“类型”的标识符,是 main 函数的返回值类型,此处是为说明主函数没有返回值的意思,具体它的意义和用法在后续章节中阐述。对 main 函数来说,void 通常可以省略。main 为函数名,圆括号里一般有参数(main 函数一般没有参数),花括号内为函数体。函数体由 C 语句(程序指令)或 C 函数组成,关于 C 语句,我们今后会逐步学习的。main 函数是 C 语言本身函数库已定义好的标准函数,C 编译器能对它进行正确的编译,不会存在不认识的情况。可能有人会问,是否所有的 C 程序都必须有 main 函数呢?答案是肯定的,一个 C 程序必须有一个 main 函数,否则,程序将无法运行,且只能有一个主函数(main 函数),程序从 main 函数位置开始执行。此外,printf()也是标准函数,printf 函数的功能是把要输出的内容送到显示器去显示。printf 函数是一个由系统定义的标准函数,可在程序中直接调用。

1.3.2 C 程序中的函数调用

下面,我们看一个有关函数调用的 C 语言程序的例子:

【例 1.2】

```
main( ) /* 主函数 */
{
    void proc( ); /* 函数声明 */
    int m = 5; /* 指定 m 为整数类型的变量,并赋予初始值为 5 */
    proc( ); /* 调用函数 proc,无返回值 */
    m = func( ); /* 调用函数 func,结果返回给变量 m */
    printf("It is an example of c program. \n");
}
```

```

void proc() / * 定义函数 proc , void 指定该函数返回值为空 * /
{
printf("Hello. \n");
}

int func() / * 定义函数 func , int 指定该函数返回一个整数 * /
{
return(1) ; / * 返回整数 1 * /
}

```

在这个程序中,涉及到了函数的概念。除了主函数外,其中还定义了名为 proc 和 func 两个函数。程序中 / * * / 表示对程序的说明(称为注释),不参与程序的运行,只是方便其他人看懂该程序而用的。注释文字可以是任意字符,如汉字、数字、英文字符等。

整个程序的执行过程为:

- 程序从 main() 函数处开始执行。
- 定义变量 m, 代表一个整数类型,且初始值为 5。
- 执行 proc() 函数;在屏幕上显示 Hello. 其中 \n 为转义字符,代表换行的意思。
- 执行 func() 函数;并将该函数运行的结果赋给变量 m。此时, m 的值为 1。
- 在屏幕上显示提示信息 "It is an example of c program. "。

该程序执行的结果是在屏幕显示两行信息:

Hello .

This is an example of c program.

从上面的程序中,我们看到,C 程序为函数模块的结构,所有的 C 程序都是由一个或多个函数构成,其中必须只能有一个主函数 main()。程序从主函数开始执行,当执行到调用函数的语句时,程序将控制转移到调用函数中执行,执行结束后,再返回主函数中继续运行,直至程序执行结束。

任何函数(包括主函数 main())都是由函数说明和函数体两部分组成,其一般结构如下:

函数类型 函数名(形式参数) 函数说明部分

```

{
说明语句部分;           函数体部分
执行语句部分;
}

```

其中,函数说明部分包括:函数类型、函数名和圆括号中的形式参数。函数说明各部分如图 1.3 所示。

函数类型	函数名	函数参数表
↓	↓	↓
int	max (int x ,	int y)

图 1.3 函数说明部分示例

1.3.3 C 语言程序的结构特点

一个 C 源程序最基本的结构特点为:

1. 一个 C 语言源程序可以由一个或多个源文件组成。
2. 每个源文件可由一个或多个函数组成。
3. 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 函数,即主函数。
4. 源程序中可以有预处理命令(include 命令为预处理命令的一种),预处理命令通常应放在源文件或源程序的最前面。
5. 每一个说明,每一个语句都必须以分号结尾。但预处理命令,函数头和花括号“}”之后不能加分号。
6. 标识符,关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

1.4 C 语言的字符集和词汇

1.4.1 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、空格、标点和特殊字符组成。

1. 字母 小写字母 a~z 共 26 个,大写字母 A~Z 共 26 个。
2. 数字 0~9,共 10 个。
3. 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。在其它地方出现时,只起间隔作用,编译程序对它们忽略。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。

4. 标点和特殊字符 特殊字符有 39 个,包括运算符和操作符。如:!,#,%,&,'(',')*+, -,./:<=>?[\\]^_`{|}~。

1.4.2 C 语言词汇

在 C 语言中使用的词汇分为六类:标识符、关键字、运算符、分隔符、常量、注释符等。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。标识符用于命名变量、类型、函数和其他各种用户定义的对象。除由系统定义的库函数的函数名外,其余都由用户自定义。C 规定,标识符只能是字母(A~Z,a~z)、数字(0~9)、下划线(_)组成的字符串,并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

_6m, x, x2, myboy1, xyz5;

以下标识符是非法的:

1w 以数字开头;

a*B 出现非法字符*;

-3x 以减号开头;

bowy-1 出现非法字符- (减号);

在使用标识符时还必须注意以下几点:

(1)标准C不限制标识符的长度,但它受各种版本的C语言编译系统限制,同时也受到具体机器的限制。例如,Turbo C区分前32个字符。

(2)在标识符中,大小写是有区别的。例如,BOOK和book被认为是两个不同的标识符。

(3)标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名应尽量有相应的意义,以便阅读理解,做到“顾名思义”。

2. 关键字

关键字是由C语言规定的具有特定意义的字符串,通常也称为保留字。关键字是由系统定义的固定的字符串,用户定义的标识符不应与关键字相同。C语言的关键字共有32个,必须用小写字母。根据关键字的作用,可分为数据类型关键字、控制语句关键字、存储类型关键字和其它关键字四类。

(1)数据类型关键字(12个):char, double, enum, float, int, long, short, signed, struct, union, unsigned, void;

(2)控制语句关键字(12个):break, case, continue, default, do, else, for, goto, if, return, switch, while;

(3)存储类型关键字(4个):auto, extern, register, static;

(4)其它关键字(4个):const, sizeof, typedef, volatile。

3. 运算符

C语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式,表示各种运算功能。运算符由一个或多个字符组成。

4. 分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中,分隔各个变量;空格多用于语句各单词之间,作间隔符。在关键字和标识符之间必须要有一个以上的空格符作间隔,否则将会出现语法错误,例如把int a;写成inta;C编译器会把inta当成一个标识符处理,其结果必然出错。

5. 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第二章中将专门给予介绍。

6. 注释符

C语言的注释符是以“/*”开头并以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时,不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来,使翻译跳过不作处理,待调试结束后再去掉注释符。

1.5 C 程序的上机操作

1.5.1 C 程序的上机过程

对 C 语言的上机操作就是要学习在计算机上如何实际地编辑 C 源程序、编译源程序、连接目标文件和运行程序这样的过程。通常上机运行一个 C 程序,必须经过以下三个步骤:

1. 编辑 C 源程序文件

所谓 C 源程序,就是存放 C 源程序的文件。C 源程序文件可以随意命名,但其扩展名必须是 .C(如 myProgram.C)。

编辑 C 源程序文件有两种含义:建立或修改 C 源程序文件。如果磁盘中没有相应的文件,则编辑的目的就是建立新文件,即输入你的程序到文件中;如果相应的文件已经存在,则编辑的目的就是要修改文件。无论是建立或是修改,最终目的都是要得到一个正确的 C 源程序文件。将程序正确地保存在文件后,编辑工作即告结束。

2. 编译和连接

(1) 编译的概念及其目的

任何高级语言程序(源程序)计算机都不能直接识别,要执行源程序,首先必须翻译成机器语言即目标代码,计算机才能直接识别。将高级语言源程序翻译成目标代码程序的过程称为“编译”,编译工作由专门的编译程序完成,编译后得到的目标代码文件的扩展名为 .OBJ(如: A.OBJ)。

(2) 连接的概念及其目的

编译后得到的目标代码虽然计算机能直接认识,但还不能直接执行,因此目标模块可能只是整个程序中的一个模块,并不是整个程序的完整模块;另外在目标模块中往往使用了一些未在本模块中定义的外部引用(如外部函数等),因此,编译后还必须把各目标模块组合起来,同时把有关的各种代码装配在一起产生一个完整的可执行文件后,才能直接执行。组合和装配的过程就称为“连接”,由专门的连接程序完成,连接后得到的文件称为可执行文件,其扩展名为 .EXE(如 myProgram.EXE)。

3. 执行程序

经过编译和连接,最后得到了扩展名为 .EXE 的可执行文件,就可以直接拿来运行。当可执行文件运行时,系统将 CPU 的控制权交给运行程序,同时按照程序设计的步骤一步步去执行,直到程序执行完毕为止。

对以上三个步骤,每一步骤都对应相应的程序命令,不过,不同版本的 C 语言其操作命令会有所不同。这里,我们选用了 Turbo c 2.0 作为我们上机实习的工具。Turbo c 集成环境是一个集编辑、编译、连接、调试、运行和文件管理为一体的工具,对 C 程序上机过程的三个步骤,都可在此集成环境中完成。在这一节中将介绍 Turbo C 环境中常用的功能,以及在该环境中如何完成从编辑源程序到运行程序这样一个过程。

1.5.2 Turbo C 概述

Turbo C 为 C 语言程序的编写、编译和运行提供了一个软件环境,用户可以在这个集成环