

21 世纪高职高专计算机系列教材

C 语言 程序设计

主 编 王四万

副主编 马 永 段宏斌

主 审 冯博琴

西安交通大学出版社

21 世纪高职高专计算机系列教材

C 语言程序设计

主 编 王四万

副主编 马 永 段宏斌

编 者 李银兴 王伟宇

王 敏 刘立伟

主 审 冯博琴

西安交通大学出版社

内容提要

本书系统地介绍了标准 C 语言的数据类型、表达式、语句及结构特点。系统地讲述了 C 程序设计方法。全书共分 11 章,主要内容包括:数据类型、运算符与表达式;顺序程序设计、选择程序设计、循环程序设计;数组、函数、指针、结构体与共用体、文件。各章配备的习题有相当部分选自于历年全国计算机等级考试的考题,可供参加等级考试的读者参考。“常见错误和程序调试”为初学者上机实践提供了指导;附录为进一步扩展有关知识提供了资料。

本书力求体现“科学严谨”且“少而精”的原则。讲述细致、准确,将复杂繁琐的概念分散或渗透于编程应用之中,重点突出 C 语最基本的知识,理论联系实际,深入浅出,通俗易懂。每章最后有小结,便于学生将各种概念系统化,帮助读者记忆各种规则、明确必须掌握的内容。

本书作为高职高专和高等院校理工科各专业“C 程序设计”课程的教材,也适合计算机类专业本科生使用,还可以作为计算机应用培训教材或自学参考书。

图书在版编目(CIP)数据

C 语言程序设计 / 王四万主编. — 西安:西安交通大学出版社,2003.4
(21 世纪高职高专计算机系列教材)
ISBN 7-5605-1650-5

I. C… II. 王… III. C 语言-程序设计-高等学校:技术学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 003448 号

书 名: C 语言程序设计

主 编: 王四万

策划编辑: 贺峰涛 屈晓燕

文字编辑: 邹 林

出版发行: 西安交通大学出版社

地 址: 西安市兴庆南路 25 号(邮编:710049)

网 址: <http://unit.xjtu.edu.cn/unit/jtupress>

电 话: (029)2668357, 2667874(发行部)

(029)2668315, 2669096(总编办)

电子信箱: eibooks@163.com

印 刷: 陕西向阳印务有限公司

版 次: 2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

开 本: 787mm×1 092mm 1/16

印 张: 17

字 数: 398 千字

书 号: ISBN 7-5605-1650-5/TP·315

定 价: 22.00 元

版权所有,翻版必究!

前 言

C语言自问世以来,因其功能丰富、表达灵活方便、目标代码效率高、实用性强,特别是可移植性好,既有高级语言的优点,又有低级语言的许多(类似汇编语言的)特性。使C语言成为当今最为流行的程序设计语言之一。

目前,C语言已不仅仅是计算机类专业必不可少的主要课程,而且也成为广大非计算机专业人员参加全国计算机等级考试、全国计算机应用技术证书考试(NIT)的考试科目之一。显而易见,学习好C语言程序设计已成为计算机基础素质培养的重要组成部分。

本书是作者总结多年来在C语言课程教学经验的基础上,在听取了各方面的意见,进行了充分的研讨与论证后编写的。它不仅可以作为大专院校计算机专业和其他专业程序设计的教学用书,还可以作为工程技术人员的自学参考书。

本书主要有以下特点:

(1) 本书按初学者掌握程序设计的入门语言为目的安排学习内容,做到由浅入深,循序渐进,重点和难点分散。只要求读者具有计算机文化基础和一定的数学知识,即可学习并掌握本书内容。

(2) 本书在内容安排上注重了教材的简洁性和条理性。力求做到将繁琐内容通俗化、简单化。如将表达“算法”的各种图形概念分散于相关章节;将各种数据类型、表达式等语法规定“表格化”,尽可能减少读者在概念内容上的记忆负担。

(3) 将“算法设计”基本概念渗透到程序设计举例之中。这样做使读者在具体问题中掌握算法设计的要领与方法,一方面可解决抽象概念的难教难学问题,另一方面可使学生更深刻领会“程序=算法+数据结构”的思想。更为重要的是,这样做可以使读者在学习过程中充分体会到,掌握C语言的关键在于多进行程序设计实践。

(4) 每章最后一节安排有“小结”,以对该章内容进行归纳与总结。目的在于帮助读者能够提纲挈领,抓住重点掌握该章学习内容。

(5) 注重操作和实用型人才培养的需要。本书的选例取材力求典型且少而精,以减少读者阅读量。编者认为,作为程序设计语言入门学习教材,应当将最基本的知识教给学生。因而本书未涉及太多的非数值计算程序设计内容,将不常用的“位运算”写入附录,供读者选学。

本书由王四万主编,马永、段宏斌担任副主编。全书共分11章。第1章、第7章由马永编写,第2章、第4章由段宏斌编写,第3章、第11章由李银兴编写,第5章、第6章由王伟宇编写,第8章由刘立伟编写,第9章由王敏编写,第10章和附录由王四万编写。

本书编写过程中,编者参考了国内外有关书籍与教材,吸收了各书的编写经验,引用了其

中一些材料,在此谨向各书的作者和出版社表示深切感谢。西安交通大学冯博琴教授、西安石油学院陈建铎教授给予了很多有益的指导。冯博琴教授在百忙中审阅了全书。西安交通大学出版社为本书出版给予了大力支持,编辑贺峰涛、屈晓燕付出了许多辛劳,在此一并致谢。同时对参与本书讨论和提出宝贵意见的同志致以谢意。

由于我们水平有限,编写经验不足,加上时间仓促,书中缺点错误在所难免,欢迎读者批评指正。

编者

2003 年 1 月于西安

21 世纪高职高专计算机系列教材编委会

顾 问：冯博琴

主 编：陈建铎

副主编：谢膺白 王四万 何东健 龚尚福

编 委：(以姓氏笔画为序)

王 津 王四万 王佑元 王丽芳 王晓奇

何东健 张水平 张俊兰 张晓云 李银兴

陈建铎 段宏斌 龚尚福 谢膺白

策划编辑：贺峰涛 屈晓燕

序

随着我国科学技术的发展,全民高等教育已经成为时代的要求。扩大招生规模,发展高等职业教育,已经成为各级政府和广大教育工作者的共识。为了指导和推动全国高等职业教育的健康发展,国家教育部先后制定了“高职高专教育基础课程教学基本要求”和“高职高专教育专业人才培养目标和规格”两个文件。在此基础上,许多出版社先后出版了相关的系列教材,对推动我国的高等职业教育起到了积极的作用。

但是,时代在前进,科学技术在发展,尤其是计算机信息技术发展的速度更是惊人。这就要求高等学校的教学内容应能跟上科学技术的发展,应能满足新技术对新型人才的需求。因此,其教材应当不断地修改和更新。故此,我们组织高校中长期从事高等职业教育的专家、学者编写了“21 世纪高职高专计算机系列教材”。在编写过程中,我们以教育部上述两个文件为依据,参阅同类教材,汲取多年来在高等专科教育、成人教育中培养应用型人才的成功经验,充分体现高职高专实用型人才的特征,“以应用为目的,以必须、够用为度”,尽量做到从实际应用的需求出发,减少枯燥乏味的纯理论和概念,使学生理论联系实际,学中有用,边学边用。通过学习,提高学生的应用和解决实际问题的能力。在编排顺序方面,尽量做到由浅入深,循序渐进,内容多样,结构合理,语言简练,文字流畅,使学生易学,易懂,易掌握。

这套教材目前已列入选题的有 18 种,既有专业基础知识,又有最新技术,可作为高职高专基础课、专业基础课以及最新技术课的教材,也可供自考和学历文凭教育使用。

在人类社会进入新世纪以来,我国高等职业教育迅猛发展的格局已经形成。这就要求教育界的志士仁人奋发努力,以自己的心血和汗水去培养时代所需要的一代有理想、有道德、有知识、有能力的高素质、高水平的应用型专业人才。

陈建铎

2002 年 10 月

目 录

第 1 章 概述

1.1 程序设计语言的发展	(1)
1.1.1 机器语言	(1)
1.1.2 汇编语言	(1)
1.1.3 高级语言	(1)
1.2 C 语言的历史与特点	(2)
1.2.1 C 语言的发展历史	(2)
1.2.2 C 语言的特点	(3)
1.3 C 程序简介	(4)
1.4 C 程序的运行环境及其编译与执行	(5)
习题	(9)

第 2 章 数据类型、运算符与表达式

2.1 常量与变量	(10)
2.1.1 常量与符号常量	(10)
2.1.2 变量	(11)
2.2 整型数据	(13)
2.2.1 整型常量	(13)
2.2.2 整型变量	(13)
2.3 实型数据	(14)
2.3.1 实型常量	(14)
2.3.2 实型变量	(15)
2.4 字符型数据	(16)
2.4.1 字符常量	(16)
2.4.2 字符变量	(17)
2.4.3 字符串常量	(18)
2.5 各类数值型数据间的混合运算	(18)
2.5.1 数据类型转换规则	(19)
2.5.2 数据类型转换方法	(19)
2.6 算术运算符和算术表达式	(20)
2.6.1 基本的算术运算符	(20)
2.6.2 算术表达式	(21)

2.7 赋值运算和赋值表达式·····	(21)
2.7.1 简单赋值运算·····	(21)
2.7.2 复合赋值运算·····	(23)
2.7.3 自增运算符和自减运算符·····	(23)
2.8 逗号运算符和逗号表达式·····	(24)
2.9 小结·····	(25)
习题·····	(26)

第3章 顺序程序设计

3.1 概述·····	(29)
3.1.1 C语句分类·····	(29)
3.1.2 何谓顺序程序设计·····	(30)
3.2 赋值语句、变量赋初值·····	(30)
3.2.1 赋值语句·····	(30)
3.2.2 变量赋初值·····	(31)
3.3 C程序处理数据的输入与输出·····	(31)
3.3.1 数据的输入与输出概念·····	(31)
3.3.2 字符数据的输入与输出·····	(32)
3.4 格式输入与输出函数·····	(33)
3.4.1 格式输出函数——printf函数·····	(33)
3.4.2 格式输入函数——scanf函数·····	(36)
3.5 程序举例·····	(37)
3.6 小结·····	(38)
习题·····	(38)

第4章 选择结构程序设计

4.1 关系运算符和关系表达式·····	(43)
4.1.1 关系运算符·····	(43)
4.1.2 关系表达式·····	(44)
4.2 逻辑运算符和逻辑表达式·····	(44)
4.2.1 逻辑运算符·····	(44)
4.2.2 逻辑表达式·····	(44)
4.3 条件运算符和条件表达式·····	(46)
4.4 选择语句·····	(48)
4.4.1 流程图和N-S图·····	(48)
4.4.2 if语句的语法格式·····	(48)
4.4.3 if语句的嵌套·····	(50)
4.5 switch和break语句·····	(51)
4.5.1 switch语句·····	(51)

4.5.2 break 语句	(53)
4.6 程序举例.....	(53)
4.7 小结.....	(56)
习题	(57)

第5章 循环结构程序设计

5.1 概述	(60)
5.2 goto 语句与循环	(60)
5.3 while 语句	(61)
5.4 do-while 语句.....	(63)
5.5 for 语句	(65)
5.6 几种循环结构的比较.....	(67)
5.7 循环的嵌套.....	(68)
5.8 循环控制语句.....	(69)
5.8.1 用 break 语句终止循环.....	(69)
5.8.2 continue 语句	(70)
5.8.3 应用举例.....	(70)
5.9 算法及其表示方法简介.....	(71)
5.9.1 算法及其特性.....	(71)
5.9.2 算法表示方法.....	(72)
5.9.3 用程序设计语言实现算法.....	(76)
5.10 程序举例	(77)
5.11 小结	(80)
习题	(81)

第6章 数组

6.1 概述.....	(83)
6.2 一维数组.....	(84)
6.2.1 一维数组的定义.....	(84)
6.2.2 一维数组元素的引用.....	(84)
6.2.3 一维数组的初始化.....	(85)
6.2.4 一维数组程序举例.....	(86)
6.3 二维数组.....	(88)
6.3.1 二维数组的定义.....	(88)
6.3.2 二维数组的引用.....	(89)
6.3.3 二维数组的初始化.....	(89)
6.3.4 二维数组程序举例.....	(90)
6.4 字符数组.....	(91)
6.4.1 字符数组的定义.....	(91)

6.4.2	字符数组与字符串的关系·····	(91)
6.4.3	字符数组的初始化·····	(92)
6.4.4	字符数组的引用·····	(92)
6.4.5	字符数组的输入和输出·····	(93)
6.4.6	字符串处理函数·····	(96)
6.4.7	字符数组应用举例·····	(99)
6.5	小结·····	(101)
	习题·····	(102)

第7章 函数

7.1	概述·····	(104)
7.2	函数的定义·····	(105)
7.2.1	函数定义的一般形式·····	(105)
7.2.2	函数的参数与函数返回值·····	(106)
7.3	函数的调用·····	(108)
7.3.1	函数调用的方式·····	(108)
7.3.2	对被调用函数的声明·····	(109)
7.4	函数的嵌套调用和递归调用·····	(112)
7.4.1	函数的嵌套调用·····	(112)
7.4.2	函数的递归调用·····	(113)
7.5	数组名作为函数参数·····	(116)
7.5.1	数组名作为函数参数·····	(116)
7.5.2	多维数组名作为函数参数·····	(119)
7.6	局部变量和全局变量·····	(120)
7.6.1	局部变量·····	(120)
7.6.2	全局变量·····	(121)
7.7	变量的存储类别·····	(122)
7.7.1	自动变量·····	(122)
7.7.2	寄存器变量·····	(123)
7.7.3	外部变量·····	(123)
7.7.4	静态变量·····	(125)
7.8	内部函数和外部函数·····	(127)
7.8.1	内部函数·····	(127)
7.8.2	外部函数·····	(127)
7.9	结构化程序设计方法简介·····	(129)
7.9.1	自顶向下程序设计·····	(129)
7.9.2	运行一个多源文件组成的C程序·····	(129)
7.10	小结·····	(130)
	习题·····	(131)

第8章 指针

8.1 概述	(133)
8.2 指针的概念	(133)
8.3 指针变量	(134)
8.3.1 指针变量的定义	(134)
8.3.2 指针变量的应用	(135)
8.4 指向数组的指针变量及应用	(138)
8.4.1 指向数组元素的指针	(138)
8.4.2 指向多维数组的指针	(141)
8.5 指向字符串的指针变量及应用	(147)
8.5.1 指向字符串的指针	(147)
8.5.2 字符串指针做函数参数	(149)
8.6 指针数组和指向指针的指针	(151)
8.6.1 指针数组及应用	(151)
8.6.2 指向指针的指针变量	(153)
8.6.3 main 函数的参数	(154)
8.7 指向函数的指针变量及应用	(157)
8.7.1 指向函数的指针变量	(157)
8.7.2 返回指针值的函数	(159)
8.8 小结	(161)
习题	(162)

第9章 结构体与共用体

9.1 概述	(167)
9.2 结构体类型变量的定义及引用	(167)
9.2.1 结构体类型的定义	(167)
9.2.2 结构体类型变量的定义	(168)
9.2.3 结构体类型变量的引用	(169)
9.2.4 结构体变量的初始化	(170)
9.3 结构体数组	(170)
9.3.1 结构体数组的定义及初始化	(171)
9.3.2 结构体数组的引用	(172)
9.4 指向结构体类型数据的指针	(172)
9.4.1 指向结构体变量的指针	(172)
9.4.2 指向结构体数组的指针	(172)
9.4.3 用结构体作函数参数	(173)
9.5 链表及其处理	(175)
9.5.1 链表概述	(175)

9.5.2 动态链表的处理	(176)
9.6 共用体	(181)
9.6.1 共用体的概念	(181)
9.6.2 共用体变量的引用方式	(183)
9.6.3 共用体类型数据的特点	(183)
9.7 枚举类型	(186)
9.8 用 typedef 定义类型	(187)
9.9 预处理命令	(188)
9.9.1 宏定义	(188)
9.9.2 文件包含	(190)
9.9.3 条件编译	(190)
9.10 小结	(191)
习题	(193)

第 10 章 文件

10.1 C 文件概念	(200)
10.2 文件类型与文件指针	(201)
10.3 文件的打开与关闭	(202)
10.3.1 文件打开与 fopen 函数	(202)
10.3.2 文件关闭与 fclose 函数	(204)
10.4 文件的读写	(204)
10.4.1 单字符(字节)方式读写	(204)
10.4.2 字节块数据项读写与 fread 和 fwrite 函数	(207)
10.4.3 文件的格式读写	(210)
10.4.4 文件处理的字符串读写函数	(212)
10.5 文件读写指针定位	(213)
10.5.1 rewind 函数	(213)
10.5.2 随机读写和 fseek 函数	(213)
10.5.3 文件位置指针的当前位置与 ftell 函数	(214)
10.6 文件检测	(214)
10.6.1 文件操作出错检测	(214)
10.6.2 文件是否结束检测	(215)
10.7 小 结	(215)
习题	(216)

第 11 章 常见错误和程序调试

11.1 常见错误分析	(220)
11.2 程序调试	(229)
11.3 检查和分析错误原因	(231)

11.4 Turbo C 环境下编辑、调试和运行 C 程序的方法	(231)
附录 I 常用字符与 ASCII 代码对照表	(237)
附录 II C 语言位运算	(239)
II.1 位运算符和位运算	(239)
II.2 位段	(241)
附录 III 运算符和结合性	(242)
附录 IV Turbo C 常用库函数	(244)
附录 V 常见的编译出错信息	(250)
参考文献	(254)



第1章

概述

1.1 程序设计语言的发展

计算机被称为是 20 世纪人类最伟大的发明之一。在计算机问世以来的几十年中,随着计算机硬件的不断升级换代和计算机应用领域的迅速扩大,计算机程序设计语言也随之有了很大的发展。在程序设计语言的发展过程中,先后产生了机器语言、汇编语言、高级语言、直到目前流行的面向对象语言和可视化编程语言,经历了从低级向高级的发展过程。

1.1.1 机器语言

计算机程序设计语言是人与计算机进行信息交流的工具,它用于描述控制计算机工作的各种指令和数据。在机器语言中,这些指令(如加法指令、减法指令、数据传输指令等)和参加运算的数据、地址,都用一定形式编码的二进制数表示。由于机器语言指令的表示形式与它在计算机中存放的形式一致,计算机可以直接识别,因而用机器语言编写的程序,计算机不仅可以直接执行,而且运行速度很快。不过,人们使用机器语言编写程序却很困难,它不仅难学、难记、难理解、难维护,而且容易出错。

1.1.2 汇编语言

由于用机器语言编写程序非常困难,随之而来的就出现了符号汇编语言。汇编语言把用二进制数表示的机器指令,用一些“助记符”表示,例如用英文缩写 ADD 表示加法运算, SUB 表示减法运算,用一些其他形式的数字和符号表示数值、存储单元的地址等。汇编语言指令与机器指令基本上是一一对应的,它便于记忆,便于使用,因而编写程序比较容易。但是由于汇编指令计算机不能直接识别与执行,所以用汇编语言编写的程序,执行前需要先翻译成机器指令,然后才能执行。这个翻译过程称作“汇编”,专门用于进行这种翻译工作的程序称作“汇编程序”。

汇编语言虽然比机器语言前进了一大步,但仍保留了机器语言的一些特点。例如,它仍是面向机器的语言(因此汇编语言和机器语言被称作低级语言),用它编写程序不仅要了解计算机内部结构有一定的了解,而且不同类型机器使用的汇编语言也不相同。尽管如此,汇编语言仍具有程序短、运行速度快的特点,因此在某些特殊应用(如实时控制)中仍然使用。

1.1.3 高级语言

从 20 世纪 50 年代中期以来陆续产生了许多“高级算法语言”,这些高级语言不依赖于具

体的机器,可以用接近于数学语言和自然语言的方式来描述解决问题的方法和步骤。高级语言由于独立于机器,编程者在不了解机器内部构造和特点的情况下,也可以编写出解决实际问题的程序;高级语言由于具有通用性,用高级语言编写的程序,可以在不同类型的计算机上运行;高级语言由于易学易用,因此迅速得以推广和使用。显然,使用高级语言可以提高编程效率,提高程序的通用性,极大地推动了计算机的普及与应用。

尽管高级语言具有诸多优点,但是用高级语言编写的程序,计算机仍然不能直接识别和执行,需要将它翻译成机器指令后才能在计算机上执行。一般将用高级语言编写的程序称为“源程序”,把翻译后生成的机器语言程序称为“目标程序”。这种翻译通常有两种做法,即“编译”方式和“解释”方式。所谓编译方式是先用编译程序把高级语言源程序翻译成目标程序,然后再执行生成的目标程序;而解释方式是将源程序由解释程序逐句翻译,解释一句执行一句。

迄今为止虽然已有上千种高级语言,可新的计算机语言仍在不断出现。尤其是面向对象语言和可视化编程语言,以其显著的优点和强大的功能,已经成为当前主要的软件开发工具。

面向对象语言是比面向过程语言更高级的程序设计语言,面向对象语言的出现改变了编程者的思维方式,使程序设计的出发点由着眼于解决问题的过程,转向着眼于问题中的对象及其相互关系,面向对象语言更接近于自然语言,是人们对于客观事物更高层次的抽象。

可视化编程语言具有强大的图形用户界面(GUI)开发功能,它不需要编写大量的程序代码,只需使用一些预先建立的控件,就可以很容易地设计出各种美观实用的图形用户界面,是 Windows 环境下理想的软件开发工具。

目前在各领域中经常使用的高级语言,主要有以下几种:

- Fortran 语言是第一个出现的高级语言,它在 1954 年提出,1956 年实现。它特别适用于数值计算,是科学计算的主流语言,目前还在使用。

- Basic 语言是初学者语言,简单易学。现今 Basic 已有许多高级版本,尤其 Windows 环境下的 Visual Basic 是一个功能强大的可视化软件开发工具。

- Pascal 语言是结构化程序设计语言,适用于教学、数值计算、数据处理和系统软件开发等。目前十分流行的 Delphi,就是由它发展而来的。

- C 语言程序简练、功能强大,适用于系统软件开发、数值计算、数据处理等,目前已成为高级语言中使用最多的语言之一。C++ 是在 C 语言的基础上发展起来的,是对 C 语言的扩充。它是一个功能强大的面向对象程序设计语言,是目前主要的软件开发工具,其中具有可视化编程环境、功能强大的 Visual C++ 最为流行。

- Java 语言是一种跨平台分布式程序设计语言,它是一种基于 C++ 的新语言,具有简单易学、安全稳定、面向对象、与平台无关、支持多线程等特点。它也是目前流行的主要软件开发工具之一。

1.2 C 语言的历史与特点

C 语言是国际上广泛流行的计算机高级语言。由于它既具有高级语言的特性,又具有某些低级语言的特性,因而被称为中级语言。它不仅可用于编写系统软件,也可编写应用软件,所以它仍然是目前最流行、最受欢迎的计算机语言之一。

1.2.1 C 语言的发展历史

C 语言最初是作为设计计算机操作系统的语言研制的,它是以 1967 年英国剑桥大学 Martin Richards 推出的 BCPL 语言等为基础发展而来的。1970 年美国贝尔实验室的肯·汤普森(Ken Thompson)在开发 UNIX 操作系统的过程中,以 BCPL 语言为基础设计出了 B 语

言。但由于B语言过于简单,功能有限,于是1972年贝尔实验室的丹尼斯·里奇(Dennis Ritchie)又在B语言的基础上编写出了C语言。C语言不仅保持了BCPL和B语言的优点,同时还克服了它们的缺点。1973年肯·汤普森和丹尼斯·里奇合作,重新用C语言编写了UNIX操作系统。其后,随着UNIX操作系统开发获得成功,并且被日益广泛地应用,C语言也被越来越多的人接受。于是,C语言逐渐发展成为一种通用的程序设计语言,风靡全世界。

C语言在产生的初期没有统一的标准,直到Brian W. Kernighan和Dennis M. Ritchie以UNIX第七版中的C编译程序为基础,合作编写了影响深远的名著《The C Programming Language》。该书中介绍的C语言成为后来各种C语言版本的基础,即现在所谓的“经典C”或“K&RC”。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语言的发展和扩充制定了新的标准,称为ANSI C。其后,于1987年ANSI又公布了ANSI C的新标准,1990年ANSI又与国际标准化组织(ISO)合作在全球范围内将C语言标准化。目前流行的各种C语言编译系统虽有一些不同,但都以这个标准为基础,其基本部分是相同的。本书的叙述也基本上以ANSI C为基础。

1.2.2 C语言的特点

C语言能够成为目前应用最广泛的高级语言之一,是由其语言特点决定的。C语言的特点可大致归纳如下:

(1) 语言简洁、紧凑,使用方便、灵活。C语言一共只有32个关键字,9种控制语句,程序书写形式自由。

(2) 运算符丰富,表达能力强。C运算符包含的范围很广泛,共有34种运算符。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他高级语言中难以实现的运算。

(3) 数据结构丰富,具有现代化语言的各种数据结构。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数据结构的运算,尤其是指针类型数据使用起来更为灵活多样。

(4) 具有结构化的控制语句(如if... else语句、while语句、do... while语句、switch语句、for语句),用函数作为程序模块以实现程序的模块化。因此是理想的结构化语言,符合现代编程风格的要求。

(5) 语法限制不太严格,程序设计自由度大。如对数组下标越界不做检查,由编程者自己保证程序的正确性。对变量类型使用比较灵活,如整型数据与字符型数据以及逻辑型数据可以通用。一般高级语言的语法检查比较严格,几乎能检查出所有的语法错误。而C语言允许编程者有较大的自由度,因此放宽了语法检查。这样程序设计人员就有了更大的自主性,就能设计出更加灵活的程序。但是它对程序设计人员的要求却更高了,因此初学者一般较难掌握。

(6) C语言允许直接访问物理地址,能进行位运算,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此C语言既有高级语言的功能,又具有低级语言的许多功能,可以用来写系统软件。C语言的这种双重性,使它既是成功的系统描述语言,又是通用的程序设计语言,因而有人把它称为“中级语言”。

(7) 生成的目标代码质量高,因此程序执行效率高。一般只比汇编程序生成的目标代码效率低10%~20%,可代替汇编语言编写系统软件。不过在某些对程序执行效率有特殊要求的编程中(如操作系统的核心部分),有时还是需要使用汇编语言编程。

(8) 用C语言写的程序可移植性好(与汇编语言相比),基本不做修改就能用于各种型号的计算机和各种操作系统。

C语言的以上特点,读者现在也许还不能深刻理解,待学完以后再回顾一下就会有比较深