



21世纪电气信息学科立体化系列教材

C 程序设计教程

主 编 张 蕊 吕 涛
副主编 赵 岚 李支成
盛新福



华中科技大学出版社

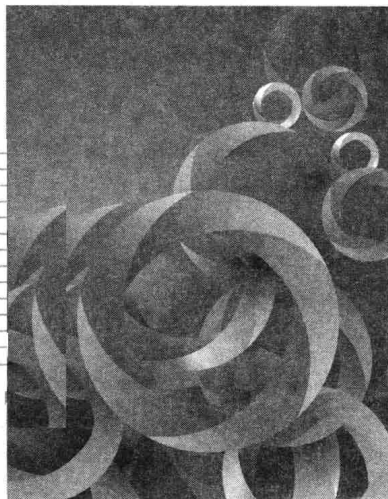
<http://www.hustp.com>



21世纪电气信息学科立体化系列教材

C 程序设计教程

主 编 张 蕊 吕 涛
副主编 赵 岚 李支成 盛新福



华中科技大学出版社

<http://www.hustp.com>

中国·武汉

内 容 简 介

本书是普通高等教育“21 世纪信息学科立体化系列教材”之一。根据国家教育部颁发的“高级语言程序设计”课程的基本要求,本着“加强基础、注重实用、精简内容、创新思维”的原则而编写。内容按进阶方式组织,分为基础篇、进阶篇和高级篇三部分。基础篇包含预备知识、C 语言概述、初识 C 程序、C 程序的基本模块——函数、C 语句以及结构化程序设计等内容;进阶篇包含指针与一维数组、文本文件、结构体与共用体等内容;高级篇包含数据类型、编码及输入输出,多文件大型程序的开发,数组、指针和函数的高级内容,二进制文件,位运算以及结构的高级应用——链表等内容。每章有内容提要、本章小结和练习题。本书适用于高等学校理工科各专业及相关课程的本科教材,也可作为相关专业工程技术人员的阅读和参考书。

图书在版编目(CIP)数据

C 程序设计教程/张 蕊 吕 涛 主编. —武汉:华中科技大学出版社,2012.9
ISBN 978-7-5609-8384-4

I. C… II. ①张… ②吕… III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2012)第 216343 号

C 程序设计教程

张 蕊 吕 涛 主编

策划编辑:王红梅
责任编辑:余 涛
封面设计:秦 茹
责任校对:朱 玢
责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:武汉佳年华科技有限公司
印 刷:湖北新华印务有限公司
开 本:787mm×960mm 1/16
印 张:22.25
字 数:448 千字
版 次:2012 年 9 月第 1 版第 1 次印刷
定 价:38.80 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前言

C 语言作为一门灵活、强大和高效的程序设计语言,受到了工程界和教育界的广泛欢迎。而作为高等院校的学生,学好 C 语言及掌握程序设计的能力,成为一项颇具挑战性的任务。

本书注重学生工程实践和编程能力的培养,并以此为指导思想,在内容编排和编程风格等方面进行了一些探索。我们认为,学习 C 语言并不仅仅等同于掌握 C 语言的语法,就如同仅仅掌握文字并不等同于可以写出优美的文章。因此,在内容编排方面,本书并不完全以语法来划分章节,而是以进阶教程的方式来组织。这样做的好处是,不让学生从一开始就过度地陷入语法细节,而是让其在每阶段的编程能力都能有所精进。

在内容编排上,全书分为基础篇、进阶篇和高级篇三大部分。基础篇按照计算机基础知识、C 语言概述、初识 C 程序、函数、C 语句以及结构化程序设计的顺序来组织。基础篇的目标是让学生可以用简单数据类型进行结构化程序设计。进阶篇则介绍了指针、文本文件和一维数组、结构体和共用体。进阶篇的目标是让学生对广泛使用的构造数据类型指针、一维数组、结构体等进行了解,并可以对文本文件的内容进行读/写,从而能完成相对复杂的编程任务。高级篇则讨论程序开发的一些高级话题,如数据类型、编码和输入/输出,多文件大型程序的开发,数组、指针和函数的高级内容,二进制文件,位运算,链表等。老师可以根据教学的实际情况对高级篇的内容进行取舍。

编程能力的培养不仅仅是掌握编程语言的语法、结构化的编程思想和常见的算法设计等,也包括遵循通用的编程规范和培养良好的编程风格。C 语言发展到现在,ANSI C 标准得到了广泛的支持,而 C99 标准可以更好地和 C++ 衔接,成为未来的发展方向。因此,全书程序符合 C99 标准,绝大部分也符合 ANSI C 标准。考虑到目前的教学和开发实际,对仅符合 C99 标准的内容进行了标注。

本书由张蕊、吕涛、赵岚、李支成、盛新福编写。张蕊执笔了第 7 章和第 12 章,吕涛执笔了第 5 章、第 6 章和第 15 章,赵岚执笔了第 1~4 章、第 10 章和第 14 章,李支成执笔了第 8 章、第 9 章和第 11 章,盛新福执笔了第 13 章,全书由张蕊统稿。李祥对

本书的结构安排提出了宝贵意见。鄢红国、易国洪等也参与了部分章节内容的编写。华中科技大学出版社对本书的出版提供了很大的帮助。在此对他们的辛勤付出表示感谢。

因水平有限,书中错误在所难免,恳请读者批评指正。

编 者
2012 年

目 录

第一篇 基 础 篇

1	预备知识	(1)
1.1	计算机的发展	(1)
1.2	计算机的系统	(2)
1.3	数制	(4)
1.4	计算机语言	(6)
1.5	本章小结	(9)
	习题 1	(9)
2	C 语言概述	(10)
2.1	C 语言的发展和特点	(10)
2.2	C 程序的结构	(14)
2.3	C 程序开发工具简介	(23)
2.4	本章小结	(26)
	习题 2	(27)
3	初识 C 程序	(29)
3.1	常量与变量	(29)
3.2	数据的运算	(34)
3.3	数据的输入/输出	(39)
3.4	C 程序应用举例	(47)
3.5	高效的运算符	(49)
3.6	宏常量与 const 常量	(51)
3.7	本章小结	(53)
	习题 3	(53)
4	程序的基本模块——函数	(57)
4.1	函数概述	(57)
4.2	库函数	(60)
4.3	自定义函数	(63)
4.4	带参数的宏	(66)
4.5	本章小结	(71)

习题 4	(72)
5 C 语句	(74)
5.1 C 语句综述	(74)
5.2 选择语句	(79)
5.3 循环语句	(98)
5.4 本章小结	(107)
习题 5	(108)
6 结构化程序设计	(115)
6.1 结构化程序设计	(115)
6.2 程序风格与代码格式	(117)
6.3 模块化程序设计	(120)
6.4 简单算法设计	(123)
6.5 小结	(130)
习题 6	(130)

第二篇 进 阶 篇

7 指针与一维数组	(132)
7.1 指针	(132)
7.2 在函数中使用指针	(136)
7.3 数组	(140)
7.4 字符数组——字符串	(161)
7.5 * 内存分配与动态数组	(181)
7.6* 二级指针	(185)
7.7 本章小结	(187)
习题 7	(189)
8 文本文件	(190)
8.1 文件的基本概念	(190)
8.2 文件的基本操作	(192)
8.3 应用实例	(196)
8.4 本章小结	(199)
习题 8	(199)
9 结构体与共用体	(200)
9.1 结构体类型	(200)
9.2 结构体变量	(201)

9.3 结构体数组	(204)
9.4 嵌套的结构体	(208)
9.5 指向结构体的指针	(208)
9.6 向函数传递结构体信息	(211)
9.7 * 复合文字和结构体(C99)	(214)
9.8 * 伸缩型数组成员(C99)	(215)
9.9 * 共用体	(215)
9.10 本章小结	(218)
习题 9	(219)

第三篇 高级篇

10 整数类型、编码及输入/输出	(220)
10.1 整数的表示	(220)
10.2 实数的表示	(224)
10.3 字符的表示	(226)
10.4 枚举类型	(227)
10.5 布尔类型(C99)	(229)
10.6 自定义数据类型	(231)
10.7 sizeof 运算符和 size_t 类型	(232)
10.8 字符、字符串和整数、实数间的转换	(233)
10.9 本章小结	(238)
习题 10	(238)
11 多文件大型程序的开发	(240)
11.1 多文件程序的架构	(240)
11.2 常用预处理命令	(241)
11.3 局部变量和全局变量	(249)
11.4 变量的存储类别	(254)
11.5 类型限定词	(257)
11.6 静态函数	(259)
11.7 本章小结	(260)
习题 11	(261)
12 数组、指针和函数的高级内容	(263)
12.1 多维数组	(263)
12.2 二维数组与指针	(267)

12.3	二维数组与指针数组	(275)
12.4	函数指针	(279)
12.5	命令行参数	(282)
12.6	可变参数函数	(284)
12.7	数组型的复合文字(C99)	(286)
12.8	变长数组(C99)	(288)
12.9	内联函数(C99)	(290)
12.10	本章小结	(291)
	习题 12	(291)
13	二进制文件	(293)
13.1	二进制文件的定义	(293)
13.2	基本操作	(294)
13.3	应用实例	(298)
13.4	本章小结	(300)
	习题 13	(301)
14	位运算	(302)
14.1	位运算符	(302)
14.2	位域类型	(306)
14.3	本章小结	(309)
	习题 14	(310)
15	结构的高级应用——链表	(312)
15.1	链表的基本概念	(312)
15.2	单链表	(313)
15.3	循环链表	(321)
15.4	双向链表	(323)
15.5	本章小结	(325)
	习题 15	(325)
	附录	(329)
	附录 A C 语言关键字	(329)
	附录 B 运算符的优先级与结合性	(330)
	附录 C 常用字符与 ASCII 值对照表	(331)
	附录 D ANSI C 标准库	(332)
	参考文献	(348)

第一篇 基础篇

1

预备知识

计算机是现代科技发展的一个产物。计算机在发展过程中经历了五代,一个完整的计算机系统包括硬件系统和软件系统,本章简要介绍了计算机的发展,计算机的硬件系统和软件系统,以及在软件系统中,计算机的数制表述形式,计算机语言发展的过程。

1.1 计算机的发展

公元前 5 世纪,中国人发明了算盘,广泛应用于商业贸易中,算盘被认为是最早的计算机,一直被沿用至今。算盘在某些方面的运算能力(如加法、减法等)要超过现在的计算机。算盘的出现是人类历史上计算设备的第一次重大突破。直到 17 世纪,计算设备才有了第二次重大发展。1645 年,法国人 Blaise Pascal(1623—1662 年)发明了自动进位加法器,称为 Pascalene。1694 年,德国数学家 Gottfried Wilhelm von Leibniz(1646—1716 年)改进了 Pascaline,使之可以计算乘法。后来,法国人 Charles Xavier Thomas de Colmar 发明了可以进行四则运算的计算器。

之前发展的计算机都是基于机械运动方式,世界上第一台真正的电子数字计算机是在 1946 年由宾夕法尼亚大学研制成功的,取名为 ENIAC (Electronic Numerical Integrator and Computer),该计算机重 30 吨,有 18000 个电子管,功率为 25 千瓦,主要用于计算弹道和氢弹的研制。从现在的眼光来看,这台计算机耗费巨大,性能又不完善,但它却是科学史上一次划时代的创新,它奠定了电子计算机的基础。自从这台计算机问世 70 多年以来,从使用的器件角度来说,计算机的发展大致经历了五代的变化。

第一代为 1946—1957 年,电子管计算机。计算机运算速度为每秒几千次至几万次,体积庞大,成本很高,可靠性较低。但在此期间形成了计算机的基本体系,确定了

程序设计的基本方法,数据处理机开始得到应用。

第二代为 1958—1964 年,晶体管计算机。运算速度提高到几万次到几十万次,可靠性提高,体积缩小,成本降低。在此期间,工业控制机得到应用。

第三代为 1965—1971 年,中小规模集成电路计算机。可靠性进一步提高,体积进一步缩小,成本进一步下降,运算速度提高到几十万次至几百万次。在此期间形成机种多样化,生产系列化,使用系统化,小型计算机开始出现。

第四代为 1972—1990 年,大规模和超大规模集成电路计算机。可靠性更进一步提高,体积更进一步缩小,成本更进一步降低,运算速度提高到每秒 1000 万次至 1 亿次。由几片大规模集成电路组成的微型计算机开始出现。

第五代为 1991 年开始的巨大规模集成电路计算机,运算速度提高到每秒 10 亿次。由一片巨大规模集成电路实现的单片机开始出现。

总之,计算机从 1946 年诞生以来,大约每隔 5 年运算速度提高 10 倍,可靠性提高 10 倍,成本降低 10 倍,体积缩小为原来的 $1/10$ 。而 20 世纪 70 年代以来,计算机的生产数量每年以 25% 的速度递增。

1.2 计算机的系统

1.2.1 硬件系统

一个完整的计算机系统由硬件系统和软件系统组成。计算机的硬件系统是指构成计算机系统的实体和装置(即用手能摸得着的计算机实物),属于物理系统。为了方便大家了解硬件系统,我们可以先看一个简单的例子:当要进行一个运算时,首先需要一支笔将要解决的问题记录在一张纸或一个本子上,然后按照一定的规则和步骤,借助于一定的运算工具将问题解答出来,最后记录在这张纸或本子上的将是最终的答案,它可以展示给所需要的人(把这步称为结果的输出)。在这里把笔称为记录的工具,它主要完成数据的输入和记录工作,还可以把结果写出来告诉需要者;纸或者本子是存储的工具,它来进行所有的数据、运算过程的存储工作;运算工具来完成中间或最终运算结果的计算;最后还要有一个参加运算的人,来进行整项工作的调控。

计算机的解题过程完全和上面的例子中的解题情况完全相似,也需要有运算工具、解题步骤和原始数据的输入与存储、运算结果的输出,以及整个运算过程的调度、控制,只不过这些全部都是由电子设备和其他设备自动进行的。在计算机里,相当于“运算工具”的部件称之为运算器;相当于“纸”那样具有“记忆”功能的部件,称之为存储器;相当于“笔”那样,把原始信息传送到计算机或把结果显示出来的部件,称之为输入/输出设备;而相当于“人”能够控制整个运算过程的部件,称之为控制器。

运算器:是计算机中进行数据加工的部件,其主要功能包括以下几方面。

(1) 执行数值数据的算术加减乘除等运算,执行逻辑数据的与或非等逻辑运算,由一个被称为 ALU 的电路部件完成。

(2) 暂时存放参加运算的数据和中间结果,由多个通用寄存器来承担。

(3) 运算器通常也是数据传输的通路。

控制器:是计算机中控制执行指令的部件,其主要功能包括以下几方面。

(1) 正确执行每条指令(取指令、分析指令、执行指令)。

(2) 保证指令按规定序列自动连续地执行。

(3) 对各种异常情况和请求及时响应和处理。

说到底,控制器要向计算机各功能部件提供每一时刻协同运行所需要的控制信号。

存储器:在现在的计算机系统中,计算机的存储器是由高速缓冲存储器、主存储器、外存储器所组成的多级(层)存储器系统,是计算机中用于存储程序和数据的子系统。这三级存储器所用的存储介质的工作原理和特性各不相同。

在存储器中,一般把能够存储一个数据信息(即多位二进制代码)的单元称为存储单元,存储器是由许多存储单元组成的,每个存储单元都有一个相应的编号,这个编号就是存储单元的地址,在后面需要进行存储单元访问的时候,就通过存储单元的地址查找,来访问相应的存储单元,如图 1-1 所示。

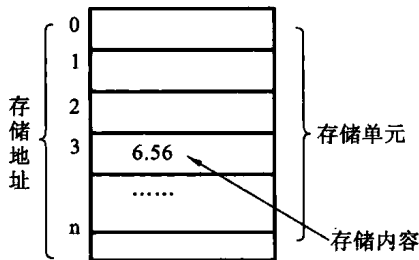


图 1-1 存储器示例图

输入/输出设备:输入设备是向计算机中送入程序和数据的有一定独立功能的设备,通过接口和总线与计算机主机连通,用于人机交互联系,如计算机键盘、鼠标等;输出设备是计算机中用于送出计算机内部信息的设备,如打印机、显示器等。

除了上述各部件以外,计算机硬件系统中还必须有总线。系统总线是构成计算机硬件系统的骨架,是多个系统部件之间进行数据传送的公共通路。借助于这条通路,计算机在各系统部件之间实现传送地址、数据和控制信息的操作。

1.2.2 软件系统

1.2.1 节介绍了计算机是由运算器、存储器、控制器、总线、输入/输出设备组成的。这些部件或设备都是由元器件构成的有形物体,称之为硬件或硬设备。但是在解决问题的时候,还要遵循一定的运算规则和运算步骤,计算机也是如此,如果仅有硬件,计算机是无法完成各种运算的。要完成相应的运算,就要把一些规则和步骤书写

出来,这就是计算程序。因为它是无形的东西,所以称之为软件或软设备。

事实上,在利用计算机进行计算、控制或其他工作的时候,需要有各种用途的程序。因此,凡是应用于一台计算机的各种程序,统称为这台计算机的程序或软件系统。计算机软件系统一般分为两大类:一类叫系统程序;另一类叫应用程序。

系统程序:用来简化程序设计,简化使用方法,提高计算机的使用效率,发挥和扩大计算机的功能及用途。它包括服务性程序、语言处理程序、数据库管理程序、操作系统四种类型。

应用程序:是用户利用计算机解决某些问题而编制的程序,如工程设计程序、数据库管理程序、网络软件、应用软件、自动控制程序等。随着计算机的广泛应用,这类程序的种类越来越多。

计算机的软件系统就是用计算机的程序开发语言来实现的,预计计算机软件的发展,将开发更高级的计算机语言。这是因为目前所使用的高级语言编写程序时程序比较复杂,开发成本较高。计算机语言发展的方向是标准化、积木化、产品化,最终是向自然语言发展,它们能够自动生成程序。

1.3 数制

数制也称计数制,是用一组固定的符号和统一的规则来表示数值的方法。虽然计算机能极快地进行运算,但其内部进行运算的并不是人类在实际生活中使用的十进制,而是使用只包含 0 和 1 两个数值的二进制(这是由计算机硬件系统的组成器件所决定的)。当人们把信息用十进制的形式输入到计算机内部后,输入计算机的十进制将被转换成二进制进行计算,计算后的结果又由二进制转换成十进制,这都由操作系统自动完成,并不需要人们手工去做。下面就介绍一下二进制数制、八进制数制、十六进制数制。

1.3.1 基本概念

1. 数码

数码是数制中表示基本数值大小的不同数字符号。例如,十进制有十个数码 0、1、2、3、4、5、6、7、8、9;二进制有两个数码 0、1。

2. 基数

基数是数制中所使用的数码的个数。例如,二进制的基数为 2;十进制的基数为 10。

3. 位权

位权是数制中某一位上的 1 所表示数值的大小(所处位置的价值)。例如,十进制

数 123,1 的位权是 100,2 的位权是 10,3 的位权是 1。

4. 数制

数制是计数的规则。在人们使用最多的进位计数制中,表示数的符号在不同的位置上时所代表的数的值是不同的。

1.3.2 十进制数

十进制是人们最为习惯采用的一种数制。十进制使用 0、1、2、3、4、5、6、7、8、9 十个符号(称为数码)按照一定规律排列起来表示的数。10 是这种数制的基数。向高位进位的规则是“逢十进一”,给低位借位的规则是“借一当十”,数码处于不同的位置,它所表达的数量的含义是不一样的,例如,86.4,8 处于十位,代表的数是 80;6 处于个位,代表的数是 6;4 处于十分位,代表的数是 $4/10$ 。不同位置的数码代表不同的数的表示方法称为位置计数法,把表示某一数位上单位有效数字所代表的实际数值称为“位权”,简称“权”。十进制的权是以 10 为底的幂。任意一个十进制数都可以用一个加权系数展开式来表示。例如,把十进制数 $(5786.115)_{10}$ 表示成加权系数展开式为: $5 \times 10^3 + 7 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 + 1 \times 10^{-1} + 1 \times 10^{-2} + 5 \times 10^{-3}$ 。

1.3.3 二进制数

二进制是计算机技术中广泛采用的一种数制。二进制数是用 0 和 1 两个数码来表示的数。它的基数为 2,进位规则是“逢二进一”,借位规则是“借一当二”。二进制也是采用位置计数法,它的位权是以 2 为底的幂,例如,二进制数 $(111.01)_2$,其权的大小顺序为 2^2 、 2^1 、 2^0 、 2^{-1} 、 2^{-2} ,二进制数一般可写为: $(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_2$ 。

二进制数码和十进制数码所对应的关系见表 1-1。

表 1-1 十进制与二进制数码对应关系表

十进制	0	1	2	3	4	5	6	7	8	9	10
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

1.3.4 八进制数

为了方便计数,减少数值的位数,在计算机技术中还采用八进制数制。八进制数是用 0、1、2、3、4、5、6 和 7 八个数码来表示的数。它的基数为 8,进位规则是“逢八进一”,借位规则是“借一当八”。八进制也是采用位置计数法,它的位权是以 8 为底的幂。

因为 8 为 2^3 ,所以八进制数的数码可以用 3 位二进制数码表示,所对应的关系见表 1-2。

表 1-2 八进制与二进制数码对应关系表

八进制	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111

八进制数一般可写为： $(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_8$ 。

1.3.5 十六进制数

与八进制数一样,另一种在计算机技术中经常被使用的数是十六进制数。十六进制数是用 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F(或 a、b、c、d、e、f)十六个字符来表示的数。它的基数为 16,进位规则是“逢十六进一”,借位规则是“借一当十六”。十六进制也是采用位置计数法,它的位权是以 16 为底的幂。

因为 16 为 2^4 ,所以十六进制数的数值可以用 4 位二进制数码表示,所对应的数值关系见表 1-3。

表 1-3 十六进制与二进制数码对应关系表

十六进制数	0	1	2	3	4	5	6	7
二进制数	0000	0001	0010	0011	0100	0101	0110	0111
十六进制数	8	9	A	B	C	D	E	F
二进制数	1000	1001	1010	1011	1100	1101	1110	1111

十六进制数一般可写为： $(a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m})_{16}$ 。

1.4 计算机语言

1.4.1 计算机低级语言

计算机每完成一次动作,实现一个功能都是按照已经用计算机语言编好的程序来执行,程序是计算机要执行的指令的集合,而程序全部都是用计算机语言来编写的。所以人们要控制计算机一定要通过计算机语言向计算机发出相应的命令。

1. 机器语言

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,由于软、硬件技术条件的限制,人们只能用计算机的语言去命令计算机完成相关的工作,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,这种计算机能够认识的语言,就是机器语言。

当时的程序就是一个一个的二进制文件。一条机器语言成为一条指令。指令是不

可分割的最小功能单元。而且,由于每台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,造成了重复工作。但由于使用的是针对特定型号计算机的语言,故而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,它是属于直接面向机器的语言。

2. 汇编语言

为了减少机器语言编程带来的问题,人们对使用的语言进行了改进:用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串,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中相应的含义,使得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,即第二代计算机语言。然而计算机是不认识这些符号的,这就需要一个专门的程序,专门负责将这些符号翻译成二进制数的机器语言,这种翻译程序被称为汇编程序。

汇编语言虽然在理解性和阅读性方面有了一定的改进,但它同样十分依赖于机器硬件,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,程序精练而质量高,所以至今仍是一种常用而强有力的软件开发工具。

由于机器语言和汇编语言都是属于面向机器指令编程的语言,它们都对机器硬件有一定的依赖性,编写出来的程序可移植性不好,不利于纠错和维护,但运行效率很高。这两种语言被称为低级语言。

1.4.2 计算机高级语言

20 世纪 60 年代中后期,随着计算机软、硬件技术的发展,计算机的使用范围越来越广泛,对软件的需求越来越多,并且软件的规模也越来越大。用汇编语言开发出来的程序,给软件的发展造成了一定的阻碍。大家在寻求一种更接近于数学语言或人类自然语言,同时不依赖于开发机器,又能够在计算机上运行的语言。经过大家的努力,在 20 世纪 60 年代末期,结构化程序设计方法被提出,随之产生了结构化程序设计语言,告别了用低级语言编写程序的时代,由于这种语言容易被理解和阅读,并具有一定的可移植性,人们称之为高级语言。这也是计算机的第三代语言。

最早出现的高级语言为 FORTRAN 语言,随后又出现了面向过程的高级语言,如 ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C 等;20 世纪 80 年代初期,面向对象程序设计思想形成并发展,又涌现出一批面向对象程序设计语言,如 C++、Visual Basic、Delphi、JAVA 等就是典型代表。C 语言就是在此期间产生并发展起来的,它介于高级语言和低级语言之间,即简化了程序的编写,又在某种程度上能够实现对计算机硬件的直接操作。

与低级语言不同,高级语言编写的程序易学、易读、易修改、通用性好,还可以在不

同的计算机系统中运行,这个特性大大减轻了程序员的负担,使他们可以将精力放在应用系统的逻辑上。但高级语言编写的程序不能在操作系统上直接运行,执行时,需要根据计算机系统的不同,将程序代码翻译成计算机系统可以直接识别的计算机语言。这个工作,由高级语言系统自动进行处理。

1.4.3 C 程序的开发过程

当最初拿到一个问题,要用计算机把它解决出来,并得到预期的结果,这个过程就是程序开发的过程。程序的开发是一门技术,需要相应的理论、技术、方法和工具来支持。为了让大家对编程有一个概括的了解,把 C 程序的开发过程分为以下几个步骤。

(1) 对问题进行功能分析,定义程序要实现的目标。

开始时,需要很明确地知道程序要完成什么功能,在实现功能的过程中需要哪些信息,经过哪些步骤,以及最终向用户反馈什么样的结果等。在这个阶段,只是用考虑一般问题的方法来解决这个问题,而不用具体的计算机语言术语。

(2) 设计程序。

在第一步对问题有了一个概念性的认识之后,现在就要考虑怎么样用程序来实现它,列出实现它的步骤(这就是算法)。还需要确定在程序中如何表示数据,以及如何处理数据。当接触的程序越来越复杂的时候,大家就会体会到一个好的方式来表示信息可以使程序中的数据处理和程序设计更容易。

(3) 编写源代码。

程序有了清晰的设计后,就可以通过编写代码来实现它。选择一种高级语言(如 C),将你的程序设计表述出来(即程序的编写)。

(4) 编译。

也就是编译源代码。通过编译器,将源代码转换成为可执行代码,可执行代码是用计算机的本机语言或机器语言所表示的程序代码,C 编译器用来将 C 语言转换成特定的机器语言。C 编译器还最终从 C 库中向最终程序加入代码,编译的最终结果是生成一个包含计算机可以理解的并让用户可以执行的可执行文件。

编译器在编译的过程中还可以检查程序中的错误,并向用户反馈错误报告。

(5) 运行程序。

上一步形成的可执行文件,是一个可以运行的程序,在很多公用的环境下,只要键入相应的文件名即可。

(6) 测试和调试。

程序在运行的过程中,可能得到的结果并不是预期结果,这时需要检查程序当中的错误(称为 bug),并进行修改,这个过程称为调试。调试就是为了发现并修改错误。