



世纪高等教育精品大系

Shiji Gaodeng Jiaoyu Jingpin Da Xi

● 主编 胡同森 田贤忠

C程序设计基础

浙江科学技术出版社



世纪高等教育精品大系

Shiji Gaodeng Jiaoyu Jingpin Da Xi

C 程序设计基础

● 主编 胡同森 田贤忠

浙江科学技术出版社

图书在版编目(CIP)数据

C 程序设计基础/ 胡同森, 田贤忠主编. —杭州: 浙江科学技术出版社, 2007. 8

(世纪高等教育精品大系·计算机系列)

ISBN 978-7-5341-3155-4

I. C… II. ①胡…②田… 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(2007)第 129186 号

丛 书 名 世纪高等教育精品大系·计算机系列

书 名 C 程序设计基础

主 编 胡同森 田贤忠

出版发行 浙江科学技术出版社

杭州市体育场路 347 号 邮政编码: 310006

联系电话: 0571-85152486

E-mail: cl@zkpress.com

排 版 杭州兴邦电子印务有限公司

印 刷 杭州印校印务有限公司

经 销 全国各地新华书店

开 本 787×1092 1/16 印张 17

字 数 422 000

版 次 2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷

书 号 ISBN 978-7-5341-3155-4 定价 29.00 元

版权所有 翻印必究

(图书出现倒装、缺页等印装质量问题, 本社负责调换)

丛书策划 郑汉阳 策划组稿 张祝娟

责任编辑 陈 岚 封面设计 孙 菁

责任校对 马 融 责任印务 田 文

前言

有很多同学问过作者：“现在为什么还要学习 C 语言，C 语言不是已经过时了吗？”作为具有多年教学经验和计算机专业工作经历的教师，曾不止一次地回答过这样的问题。

1. 开设程序设计课的目的是以编程语言为平台介绍程序设计的思想和方法。

通过学习，学生不仅要掌握高级程序设计语言的知识，更重要的是在实践中逐步掌握程序设计的思想和方法，培养问题求解和语言的应用能力，为今后进一步学习程序设计打好基础。C 语言具有丰富灵活的控制和数据结构、简洁而高效的语句表达、清晰的程序结构和良好的可移植性，既具有高级语言的优点，又具有低级语言的特点。学好 C 语言可以为今后深入学习 C++、Java 等其他高级语言打下一个良好的基础。

2. 认为 C 语言过时主要有两个理由：它是面向过程设计的，而非面向对象的；它不是可视化编程。

C++是在 C 的基础上开发的，且 Java 与 C++也有千丝万缕的联系，在 Java 环境下就可以直接用 C 程序，因而用 C 语言作为入门语言是较佳选择。由于我们习惯用 Windows 操作系统，于是认为非图形界面即不好，然而若一个问题不用图形界面就可以解决，那又何必做得这么复杂呢？试想，如果一条信息可以告诉你现在的时间是什么，你又何必去翻个闹钟出来看看？

即使是在 Linux 操作系统下，C 语言也大有用武之地，绝大多数的应用软件都是用 C 语言写的，包括图形界面。

3. 由于 C 语言具有低级语言的特点，因此许多涉及硬件底层设备的控制程序都要用 C 语言来完成，如设备驱动程序和操作系统。

本书作为高校学生第一门程序设计课程的教材，力图使学生掌握标准 C 语言的基本知识和 C 程序设计的思想和方法，为今后深入学习 C++、Java 等高级语言打下一个良好的基础。所以，本书在体系上重视理论结合实际，以便于读者低起点、高效率地掌握 C 语言；在内容组织上突出重点而不过度纠缠一些晦涩的语法现象，且注重解题的思路分析和算法的设计；在编排上尽量体现各章节的关联和系统性；在文字叙述上力求条理清晰、概念准确。

为便于学生更好地学习和理解 C 语言，在一些较难理解的部分尽量使用图解的方法，并且在例题中加入了部分代码注释。书中例题都是精选的，不以单纯解释 C 的语法为目的。习题难易适中、覆盖面广，对较难的习题还给出了提示以帮助学生解答。例题和习题强调了编程能力训练的重要性，淡化了部分语法细节。所有教学内容围绕 ANSI C 展开，因考虑到软件系统的流行配置，故重点介绍了在 Visual C++环境中运行调试 ANSI C 程序的方法。

本书共分 11 章,主要包括:

第一章:C 程序设计基础知识。介绍算法的概念与描述工具、计算机语言及其发展、C 语言的发展与特点、C 语言的程序结构,重点介绍 C 程序的上机环境与操作步骤。

第二章:基本数据类型与常用库函数。介绍了一些基本的数据类型、变量与常量的声明与应用,以及常用库函数的简单应用。

第三章:运算符和表达式。介绍 C 的算术、关系、逻辑、条件和逗号等运算符和表达式。

第四章:流程控制。重点介绍分支结构中的 if、switch 语句,循环结构中的 while、do-while、for 语句。

第五章:函数。着重介绍自定义函数的编写与调用,强调函数间参数传递的方式,同时还介绍了递归函数。

第六章:编译预处理与变量的作用域。介绍编译预处理中的文件包含和宏定义,变量的作用域、可见性以及存储类型。

第七章:数组与字符串处理。介绍了一维数组和二维数组的概念、定义、初始化,数组元素的引用和数组作为函数参数的使用,以及字符串、字符串数组的使用。

第八章:指针。介绍指针的概念,用较大篇幅从实用角度介绍了指针变量的应用分类、整理,使之更易于被接受。

第九章:结构体。介绍定义结构体的格式与使用方法,同时还对链表操作的几个典型示例作了分析。

第十章:位运算。主要介绍了位运算的概念以及六种常见的位运算符。

第十一章:文件。介绍文件的概念、文件指针、文件处理的基本过程和相关函数。

本书由浙江工业大学的胡同森教授、田贤忠老师、王英姿老师与浙江工商大学的陶华良老师编写,陈登、陈长军、项方云、余静涛也参加了本书的编写工作。对于浙江工业大学以及信息工程学院领导和浙江科学技术出版社的大力支持,在此表示衷心感谢。

本书可以作为各类大专院校、各类培训与等级考试的教学用书,也可以作为 C 程序设计爱好者的自学用书,还可以作为计算机类硕士考研参考用书。相信通过本书的学习,能为你打下 C 程序设计坚实的基础。

恳请读者提出宝贵意见,可发送至 E-mail:hts@zjut.edu.cn。若需要本书中所有例题的源程序以及习题参考答案与复习题集的电子稿,可以按网址:www.computer.zjut.edu.cn 下载。

编著者

2007 年 6 月

目 录

第一章 C程序设计基础知识	1
第一节 算法的概念与描述工具	1
第二节 计算机语言及其发展	3
第三节 C语言的发展与特点	6
第四节 C语言程序结构	7
第五节 C程序的编译运行	10
第六节 小 结	14
习题一	15
第二章 基本数据类型与常用库函数	17
第一节 基本数据类型	17
第二节 常量与变量	23
第三节 常用标准库函数	28
第四节 小 结	36
习题二	37
第三章 运算符和表达式	40
第一节 表达式的基本概念	40
第二节 算术运算符与算术表达式	41
第三节 赋值运算符与赋值表达式	43
第四节 关系运算符、逻辑运算符与逻辑表达式	45
第五节 条件表达式与逗号表达式	49



第六节 小 结	50
习题三	50
第四章 流程控制	53
第一节 结构化程序设计	53
第二节 选择结构	54
第三节 循环结构	61
第四节 多重循环	76
第五节 小 结	81
习题四	81
第五章 函 数	85
第一节 函数概述	85
第二节 函数嵌套调用	94
第三节 递归函数	95
第四节 小 结	99
习题五	100
第六章 编译预处理与变量的作用域	103
第一节 编译预处理	103
第二节 变量的作用域与可见性	108
第三节 变量的存储类型	111
第四节 小 结	113
习题六	113
第七章 数组与字符串处理	117
第一节 一维数组	117
第二节 二维数组	124
第三节 字符串	130
第四节 字符串数组	137



第五节 三维数组应用示例	140
第六节 小 结	143
习题七	144
第八章 指 针	149
第一节 指针的基本概念	149
第二节 多级指针	154
第三节 指针数组	155
第四节 指针变量的应用	158
第五节 小 结	171
习题八	172
第九章 结构体	177
第一节 结构体类型与结构体类型数据	177
第二节 结构体与函数	185
第三节 链 表	190
第四节 共用体	202
第五节 小 结	203
习题九	204
第十章 位运算	207
第一节 二进制位	207
第二节 位运算	208
第三节 位运算应用举例	213
第四节 小 结	216
习题十	216
第十一章 文 件	218
第一节 文件概述	218
第二节 文件的打开与关闭	221





第三节 文本文件的顺序读写	222
第四节 文件的定位与随机读写简介	234
第五节 小 结	236
习题十一	237
附录 I 字符与ASC II 码对照表	239
附录 II 运算符优先级	240
附录 III 常用C库函数	241
附录 IV C程序设计样卷	244
附录 V 综合测试试卷(150分)	250

第一章 C程序设计基础知识



一个完整的计算机系统是由硬件和软件两大部分组成的。计算机硬件是指计算机系统的物理装置本身,即我们看得见也摸得着的东西,如主板、处理器、内存等各种设备。但是如果没有软件,它们只是一堆废铜烂铁。计算机软件是指计算机程序以及与程序有关的资料和说明的总称,如Windows XP、Word和各种应用程序。简单地说,软件就是计算机执行的程序。

程序是为解决某一问题而设计的一系列指令,这些指令是由某种计算机语言来描述的。程序设计是指根据所提出的任务,用计算机语言编制一个能正确完成任务的计算机程序。我们学习C语言的目的就是用C语言来设计程序,从而帮助我们解决工作中的实际问题。

本章首先简单介绍算法的概念及描述工具,然后介绍计算机语言的发展过程和C语言的发展与特点,最后通过简单的C程序实例使读者对C程序的结构,以及C程序的调试运行环境有一个初步的了解。

第一节 算法的概念与描述工具

用计算机处理各种不同的问题时,首先要对各种问题进行分析,确定解决问题的方法和步骤,然后根据它编写出计算机程序,再让计算机执行这个程序,最后得出结果。所以,确定解题的方法和步骤是编程的前提。

一、算法的概念

著名计算机科学家Wirth提出了一个著名的公式“数据结构+算法=程序”,是说一个程序应包括以下两方面内容:

- (1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式,即数据结构。
- (2) 对操作的描述。即算法,亦即程序描述了为解决一个问题而采取的方法和步骤。

其实,做任何事都要按一定的步骤。例如,计算 $1+2+3+4+5$ 的值,一般是先将1和2相加得3,再将3加3得6,再将6加4得10,最后再将10加5得15。无论手算、心算或用计算机计算,都要经过有限的、事先设计好的步骤才能得出结果。这种为解决一个问题而采取的方法和步骤就是算法,或者说,算法是解题方法的精确描述。



算法一般分为两大类:数值算法和非数值算法。

数值算法主要用于求数值解的问题,例如求若干数之和、求方程的根、求两个数的最大公约数等。

非数值算法主要用于解决需要用分析推理、逻辑推理才能解决的问题,例如将若干个人名按字母排序、图书情报检索、人工智能中的许多问题等。

目前,数值算法比较成熟,对各类数值计算问题都有成熟的算法可供选用。

一个算法应具有以下特点:

- (1) 有穷性。算法包含的操作步骤是有限的,否则计算机将永远运行下去,永无结果。
- (2) 确定性。算法中每一个步骤应是确定的,不能是含糊、模棱两可的。例如,“吃完饭来老师办公室”这句话是不明确的,因为这句话是指吃完早饭、午饭还是晚饭?除非在特定的条件下,如两人正在吃午饭,否则该要求很难执行。
- (3) 有多个或零个输入。即按算法执行过程中可输入数据,也可以不输入。
- (4) 有一个或多个输出。算法的目的是求“解”,“解”就是结果,就是输出。如“求两个正整数的最大公约数”,输出的是最大公约数,不给出输出的算法是没有意义的。
- (5) 有效性。算法中每一个步骤都应当能有效地执行,并得到确定的结果。例如一个数被零除,就不能有效地执行。

二、算法的描述工具

在设计程序时,通常使用专门的算法表达工具对算法进行描述,如自然语言、流程图、N-S图、伪码等。下面通过简单例子来重点介绍前两种。

1. 用自然语言表示算法,通俗易懂,适合求解简单问题

【例1-1】 设有两个整数A和B,要求将它们互换。

第一步:定义一个新的变量C,将A的值赋给C;

第二步:将B的值赋给A;

第三步:将C的值赋给B。

以上程序可以简写为:

S1: $C \leftarrow A$;

S2: $A \leftarrow B$;

S3: $B \leftarrow C$ 。

【例1-2】 输入一个数,求其绝对值并输出。

S1: 输入x;

S2: 判断x的值,若x小于零,则将x取反(即: $x \leftarrow -x$);

S3: 输出x。

【例1-3】 求1~100之间所有整数的和。

S1: $sum \leftarrow 0, t \leftarrow 1$;

S2: $sum \leftarrow sum + t$;

S3: $t \leftarrow t + 1$;

S4: 若 $t \leq 100$,则转到S2,否则顺序执行S5;

S5:输出sum,结束。

2. 用流程图表示算法,直观、易懂
流程图中的基本符号如图1-1所示。

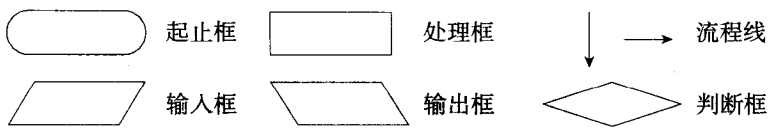


图1-1 流程图的基本符号及其含义

如图1-2(a)、(b)、(c)所示为前面3个例子的流程图。

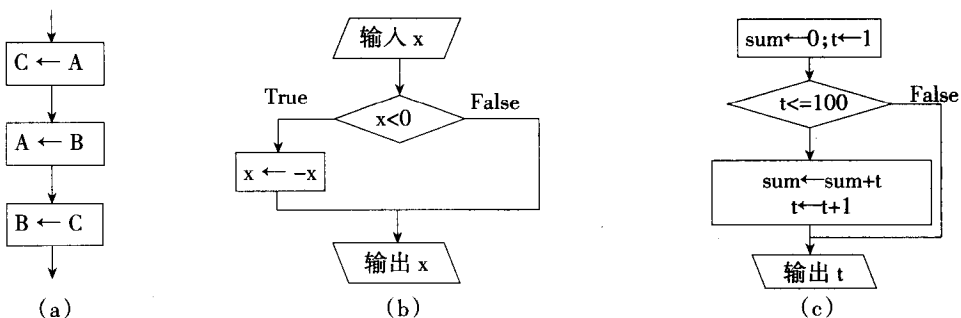


图1-2 例1-1~例1-3的流程图

第二节 计算机语言及其发展

有了算法后,如果用计算机来实现,必须要用某种计算机语言把它描述出来,变成计算机能识别的程序。

程序就是为解决某一问题而设计的一系列指令。安排人们日常生活、工作的程序,可以用汉语、英语、日语等不同语言描述;为使计算机能按照人们的意志工作的程序,则要用计算机能够理解和识别的语言,即计算机语言来描述。计算机语言的发展,经历了机器语言、汇编语言、高级语言这几个阶段。

一、机器语言

计算机能够直接执行的程序,是机器语言程序。

机器语言程序用二进制编码表示的机器指令编写,一台计算机能够执行哪些机器指令,取决于其硬件设计。计算机能够执行的不同指令的集合,称为该计算机的指令系统,计算机的指令系统中一般都有数百条指令。

早期的计算机都只能执行机器指令程序,指令系统是计算机唯一能够直接识别和执行的语言,因此又称作机器语言。

计算机之所以能够自动完成指定的各种运算,是由于用来描述这些运算的指令、运算所需要的数据已事先被存储在内存的指定地址中,计算机是从程序区的首地址开始执行程序的,故能自动地执行这些指令。



一条机器指令由操作码、若干个地址码组成。操作码表示该指令执行何种操作,地址码标识作为操作对象的数据或指令的地址。

为便于描述机器语言程序编制、执行的过程,我们虚拟了一台模型机,其指令系统总共才10条指令,仅包括最基本的操作。为书写方便,模型机的指令格式用八进制而不是用二进制来表示,其中 a_1a_2 为操作码, $d_1d_2d_3d_4$ 为地址码。表1-1为模型机的指令格式和指令系统。

表1-1 模型机的指令格式和指令系统

操作码	操作内容
00	停机
01	寄存器中的数与 $d_1d_2d_3d_4$ 中的数相加,结果存于寄存器
02	寄存器中的数与 $d_1d_2d_3d_4$ 中的数相减,结果存于寄存器
03	寄存器中的数与 $d_1d_2d_3d_4$ 中的数相乘,结果存于寄存器
04	寄存器中的数与 $d_1d_2d_3d_4$ 中的数相除,结果存于寄存器
05	从 $d_1d_2d_3d_4$ 单元取数,存于寄存器
06	把寄存器中的数送到以 $d_1d_2d_3d_4$ 为地址的存储单元
07	无条件转去执行 $d_1d_2d_3d_4$ 单元中的指令
10	如果寄存器中的数小于0,则执行 $d_1d_2d_3d_4$ 中的指令,否则顺序执行
11	输出 $d_1d_2d_3d_4$ 单元中的数

为使读者了解机器语言程序的编制、调试和运行过程,我们用一个简单的例子来说明。

【例1-4】 用模型机的指令系统编制程序,求 $S=99+97+95+\cdots+3+1$ 的和,并输出结果。

(1) 在内存的下列单元中送入数据:

- 0001单元送入0(该单元存放各次累加的和S,初值为0);
- 0002单元送入2(表示相邻两项之差);
- 0003单元送入八进制数143(即十进制数99,为被加项的初值,记为a)。

(2) 从0010单元起,送入如表1-2所示中的指令。

表1-2 例1-4程序中的指令及说明

指令存放地址	指令		操作说明
	操作码	地址码	
0010	05	0001	取s(0001单元中的数)到寄存器
0011	01	0003	$s \leftarrow s+a$, 寄存器中的数加a(0003中的数)后送回0001单元
0012	06	0001	
0013	05	0003	取a到寄存器
0014	02	0002	$a \leftarrow a-2$, 寄存器中的数减2后送回a(即0003单元),此时寄存器与a中的数相等
0015	06	0003	
0016	10	0020	若寄存器中的数小于0,则执行0020中的指令,否则执行下一条指令
0017	07	0010	无条件转去执行0010中的指令
0020	11	0001	输出0001中的数(即所求的和)
0021	00	0000	停止执行

(3) 从程序首地址0010启动,计算机将自动执行该程序。在完成了99、97直至1的累加后,当a(0003中的数)等于“-1”时,计算机将执行输出和停止指令。从执行过程可知,除非遇到控制指令如无条件转移、条件转移指令,否则计算机将按指令在内存中存放的地址逐条、顺序地执行。

用机器语言编写程序,存在下列几个主要问题:

(1) 冗长繁琐,难记、难写、难读。编写机器语言程序,一定要熟悉指令系统,而指令系统包含数百条指令。机器指令只能完成一些基本操作,程序的每个步骤往往被分解为若干条机器指令,因此经常为解决一个简单问题而编制出相当复杂的机器语言程序。这样的程序即便是编程者本人也是难以看懂的,许多人宁可自己重新编制程序,也不愿去消化、改写别人的程序。

(2) 难以修改。在编程处理复杂问题时,错漏总是难免的,往往要经过多次修改后才能完成程序的编制过程。修改机器语言程序是非常棘手的,仅举一例:机器语言程序中,要写出每一条指令的地址,如果漏写了一条指令,插入它后,后续的各条指令的地址都要重新改写。

(3) 由于上述原因,用机器语言开发应用程序的周期很长。如果用于科研,为了物理模型、数学模型的每一次变化而修改程序,那要付出很大的代价,也许当你的程序调试好了,这一科研课题也过时了。

(4) 依赖于机器,难以移植。由于指令系统是面向机器的,不同种类的CPU对应的指令系统之间往往相差很大,因此,机器语言程序的可移植性较差。

在早期,用机器语言编程是经过严格训练的专业技术人员的工作,普通程序员一般难以胜任。现在,几乎没有人用机器语言编写应用程序了。

二、汇编语言

虽然用机器语言编程有许多不便且对它有很高的要求,但程序的执行效率高,CPU严格按照程序中的指令序列去执行,没有多余的操作。在保留“程序执行效率高”的前提下,人们开始着手研究一种能大大改善程序可读性的编程工具:选用了一些能反映机器指令功能的单词或词组来代表该机器指令,把CPU内部的各种资源符号化,用符号名来引用相应资源。

这样,令人难懂的二进制机器指令就可以用通俗易懂的符号指令来表示了,这些具有一定含义的符号称为助记符,用指令助记符、符号地址等组成的符号指令称为汇编指令。

用汇编指令编写的程序称为汇编语言程序,其可读性大大提高了,但失去了CPU能直接识别的特性。例如指令“MOV AX, BX”用来表示将BX中的数据赋值到AX中,而CPU却不能直接识别并执行它。为解决这个问题,就需要执行一个翻译程序,它能把汇编语言编写的源程序自动翻译成CPU能识别的机器指令序列,如图1-3所示,该翻译程序被称为汇编程序。

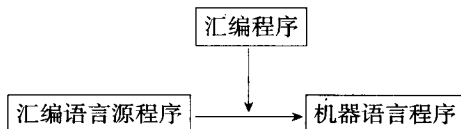


图1-3 用汇编程序处理汇编语言源程序,生成机器语言程序



尽管采用指令符号化和伪地址(用符号表示存储单元地址)的方法,汇编语言程序的可读性和编程难度有所降低了,但是从本质上讲,它还是面向机器的。

三、高级语言

在长期遭受低级语言(机器语言、汇编语言)编程低效率的困扰后,人们逐渐意识到应设计一种这样的语言,它接近于数学语言或人的自然语言,但又不依赖于计算机硬件,编出的程序能在所有机器上通用。第一个完全脱离机器硬件的高级语言——FORTRAN语言诞生于1954年,此后又出现了数百种高级语言,其中有重要意义的有几十种。

20世纪60年代中后期,随着计算机应用的普及,所开发的应用程序越来越多、规模也越来越大,由于缺乏科学、规范的系统规划与测试以及评估标准,其恶果是许多耗巨资开发的程序由于含有错误而无法使用,甚至带来巨大损失,给人的感觉是越来越不可靠。这一“软件危机”使人们认识到,大型程序的编制是一项新的技术,应像处理工程一样处理软件研制的全过程,程序的设计应易于保证其正确性、便于验证其正确性。于是,1969年结构化程序设计方法被提出来了,1970年第一个结构化程序设计语言——Pascal语言的诞生标志着结构化程序设计时期的开始。

20世纪80年代初开始,在软件的设计思想上又产生了一次革命,其结果是面向对象的程序设计的开始。正如汇编程序支持用汇编指令编程一样,FORTRAN和C语言的较低版本的系统支持结构化程序设计。在支持面向对象的语言开发工具中,Visual Basic及Visual C++是广为人知的较成功的代表作。

高级语言的下一个发展目标是面向应用,也就是说,在该语言环境中,只要正确表达你需要做什么,就能自动生成算法以及目标程序,这就是非过程化的程序语言。

第三节 C语言的发展与特点

C语言是国际上最著名的高级程序设计语言之一,也是使用范围最广的计算机编程语言之一。它不仅编写如操作系统、数据库之类的系统软件,而且还可以用来编写各种应用软件。C语言可以运行在各种平台上,从个人电脑到巨型机,从DOS到UNIX系统。

第一个C语言是贝尔实验室的Dennis Ritchie于1972年在B语言的基础上开发的,起初C语言是作为UNIX操作系统的开发语言。在此后的近20年中,C语言得到了广泛的应用,导致了許多不同版本的C语言的产生,对希望所开发的程序能够在不同计算机系统上运行的程序开发者来说,迫切需要制定一种标准的C语言版本。

1990年,一种在发展中确定了C语言版本被ISO组织批准为国际标准,称为ANSI C或标准C(以下简称C)。由于C集中了高级语言、低级语言的长处,所以它已迅速普及并成为当今最有发展前途的计算机高级语言之一。

C语言有以下的特点:

- (1) 简洁紧凑、灵活方便。一共只有32个关键字、9种控制语句,程序书写自由。
- (2) 运算符丰富。C的运算符包含的范围很广,共有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他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数据类型的运算。指针概念的引入,使程序执行效率比以前高。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,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,这些函数可方便地调用,并具有多种循环语句、条件语句来控制程序的流向,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格,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语言允许程序直接访问物理地址,可以直接对硬件进行操作,因此它兼有高级语言和低级语言的许多功能。C语言能够像汇编语言一样对位、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,可以用来编写系统软件。

(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。其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。

第四节 C语言程序结构

一、简单C程序示例

对于C程序设计的初学者来说,没有比自己编制程序、并在计算机上运行得出正确结果更使人愉悦的事了。为使大家有一个感性的认识,我们先看几个简单的C程序。

【例1-5】 一个简单的加法计算程序,求两个整数的和。

```
#include <stdio.h>           /* 编译预处理 */
void main()
{
    int x,y,sum;              /* 声明变量 */
    x=5;                      /* 给变量x赋值 */
    y=100;                    /* 给变量y赋值 */
    sum=x+y;                  /* 计算和 */
    printf("和是:%d\n", sum); /* 输出结果 */
}
```

 **运行结果:** 和是:105

✱ 程序说明: 编译预处理命令“#include <stdio.h>”将“stdio.h”文件嵌入到程序中,使输入、输出能正常执行。

main是函数名(后面必须有一对圆括号),用一对花括号括住的部分是函数体,花括号



内是声明语句、执行语句。

“int x,y,sum;”是变量声明语句。变量是内存中的存储单元,能够存储供程序使用的数据。“sum=x+y;”是计算两个数的和,并把结果存放到变量sum中。程序中“/*”和“*/”之间的文本是注释文字,在程序执行中不起任何作用,只是增加程序的可读性。

【例1-6】 编程,输入两个变量后,交换他们的值。

```
#include <stdio.h>           /* 编译预处理 */
void main()
{ int a, b, c;               /* 声明3个int类型变量 */
  printf("请输入A,B的值:\n"); /* 输出提示信息 */
  scanf("%d%d",&a,&b);        /* 输入a,b的值 */
  c=a; a=b; b=c;             /* 将a,b互换 */
  printf("A=%d B=%d\n",a,b); /* 输出a和b交换后的结果 */
}
```

运行结果: 请输入A,B的值:

```
3 5      加下划线表示输入以换行结束的1行字符,下同。
A=5 B=3
```

※ 程序说明: 整个程序由一个主函数main组成。语句“int a,b,c;”声明3个变量用于存放3个整数,“scanf("%d%d",&a,&b);”用于从键盘输入两个整数放入变量a、b中。

“c=a; a=b; b=c;”这三条语句用于交换变量a、b的值,首先用临时变量c记下变量a中的值,然后把变量b中的值赋给变量a,最后把c中的值(就是原来a中的值)赋给变量b,这样就交换了两个变量中的值了。

【例1-7】 输入3个数,求其中的最小数。

```
#include<stdio.h>
float min(float x, float y) /* 定义min函数 */
{ float m;                 /* 定义变量m */
  if(x<y)                   /* 比较两数x和y,将较小者赋值给m */
    m=x;
  else
    m=y;
  return m;                /* 返回m的值 */
}
void main()
{ float a,b,c,t,mindata;
  printf("请输入三个数:\n");
  scanf("%f%f%f",&a,&b,&c);
  t=min(a,b);              /* 调用函数min,求a,b两数中的较小者 */
  mindata=min(t,c);        /* 调用函数min,求t,c两数中的较小者 */
  printf("最小数:%f\n",mindata); /* 输出mindata */
}
```