

# C语言与程序设计

C YUYAN YU CHENGXU SHEJI

刘竞杰 • 主编

合肥工业大学出版社

# C 语言与程序设计

主 编 刘竞杰

副主编 于海霞 许 晟

编 委 (按姓氏笔画排列)

于海霞 王 倩 刘竞杰

许圣洁 李 雪 贺继东

合肥工业大学出版社

## 内 容 提 要

本书以程序设计为主线,以编程任务为驱动,通过具体案例或问题引入教学内容,重点讲解程序设计思想和方法。同时,将知识分散并逐步细化和加深,有利于学生接受和理解,通过必要的提示和简单总结帮助对核心知识的掌握。本书共分14章,其中第1~7章是基本知识和基本编程,包括数据类型、运算符、程序结构和数组等。第10~13章,重点介绍了指针、结构体、文件和函数的组织(递归函数)等。C、C++和Java程序设计语言一脉相承,为了使学生在后续学习中更好地理解面向对象程序设计,在第14章中我们对面向对象、C++和Java做了介绍。

本书可作为大专院校(高职)和计算机培训等教学用书,也可作为C语言自学用书。

### 图书在版编目(CIP)数据

C语言与程序设计/刘竞杰主编. —合肥:合肥工业大学出版社,2009.8

ISBN 978-7-81093-979-9

I. C… II. 刘… 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(2009)第085107号

## C语言与程序设计

主 编 刘竞杰

责任编辑 陆向军

出 版	合肥工业大学出版社	版 次	2009年8月第1版
地 址	合肥市屯溪路193号	印 次	2009年8月第1次印刷
邮 编	230009	开 本	787毫米×1092毫米 1/16
电 话	总编室:0551-2903038 发行部:0551-2903198	印 张	15
网 址	www.hfutpress.com.cn	字 数	365千字
E-mail	press@hfutpress.com.cn	印 刷	合肥学苑印务有限公司
		发 行	全国新华书店

ISBN 978-7-81093-979-9

定价:24.00元

如果有影响阅读的印装质量问题,请与出版社发行部联系调换。

## 前 言

程序设计是高等学校重要的计算机基础课程,它以程序设计语言为平台,介绍程序设计思想和方法。通过该课程学习不仅要掌握程序设计语言的基础知识,更重要的是在学习和实践中逐步掌握程序设计,培养用程序思想解决实际问题以及运用语言的能力。本课程以培养学生程序设计方法和技能为目标,以实践能力培养为重点。

C 语言是广泛使用的程序设计语言之一,由于其既具有高级语言特点,又能直接面向硬件,同时具有丰富灵活的处理能力,使得它既可编写系统程序又可编写应用程序,深受程序设计者喜爱。C 语言已成为众多高校首选程序设计语言。

尽管有很多 C 语言教材,但大多数仍以语言知识点为中心,不利于学生理解程序和设计程序,偏离了以程序设计能力的培养为目标的宗旨。同时,以介绍“语法—程序结构—编写程序”为线,也不符合认知规律和习惯,不适应大学程序设计入门课程的需要。

本教材力求按照教学和学习基本规律以及人们对事物认知习惯来组织。本书以程序设计为主线,以编程任务为驱动,通过具体案例或问题引入教学内容,重点讲解程序设计思想和方法。同时,将知识分散并逐步细化和加深,有利于学生接受和理解知识,通过必要的提示和简单总结帮助对核心知识的掌握。

读者在学习过程中要注意观察和总结,不断理解 C 语言程序设计的特点并提高自己的学习能力。比如:每个语句后一定要有分号(;);函数是 C 程序的基本组织单位;在 C 程序语句中出现的所有双引号均为英文状态下的,即(" "),而非中文状态下的,即(“ ”),单引号也同样为英文状态下的,即(' '),而非中文状态下的,即(‘ ’);自增自减运算(++\-- )也是赋值运算,其前缀和后缀有不同特点;数组是最简单但很有效的数据结构,它常与循环控制结构紧密相连;指针和位运算体现出 C 语言面向内存的重要特色;指针的间接访问特色以及指针和结构体用于构造更加符合实际的数据

结构等。

本书共分 14 章,其中第 1~7 章是基本知识和基本编程,包括数据类型、运算符、程序结构和数组等。第 10~13 章,重点介绍了指针、结构体、文件和函数的组织(递归函数)等内容,旨在培养学生解决复杂问题的能力。为了加强对程序设计和算法的理解和实践,除了在第 1 章序言中介绍算法之外,第 8 章(算法概述)和第 9 章(基础知识综合应用示例)分别给予一定程度的强化。C、C++ 和 Java 程序设计语言一脉相承,为了使学生在后续学习中更好地理解面向对象程序设计,在第 14 章中我们对面向对象、C++ 和 Java 做了介绍。

本书第 1、12 章由贺继东编写,第 2、4 章由刘竞杰编写,第 3、5、11 由于海霞编写,第 6 章由李雪编写,第 7、13 章由黄兵编写,第 8、9、14 由王倩编写,第 10 章由许圣洁编写。本书由刘竞杰担任主编,负责统稿,于海霞、许晟担任副主编,负责部分统稿工作。

我们还编写了相应的实践教程,每个实验和课程设计都精心设计。读者可以通过“模仿—改写—编写”的实践过程,在循序渐进的引导中逐渐熟悉编程环境,理解和掌握程序设计思想、方法和技巧,不断提高程序设计能力。

由于我们水平有限,错误之处还请读者批评指正。

编 者

2009 年 8 月

---

---

# 目 录

第 1 章 序 言 .....	(1)
1.1 计算机工作原理概述 .....	(1)
1.2 计算机程序与程序语言 .....	(2)
1.3 程序设计与算法 .....	(5)
1.4 C 语言的发展与特点 .....	(6)
第 2 章 用 C 语言编写程序 .....	(9)
2.1 输入输出程序示例 .....	(9)
2.2 包含控制语句的程序示例 .....	(12)
2.3 包含函数的程序示例 .....	(13)
2.4 C 语言程序结构与基本语法 .....	(14)
第 3 章 变量、数据类型与分支结构 .....	(19)
3.1 变量及变量的作用 .....	(19)
3.2 数据类型 .....	(21)
3.3 分支结构程序设计 .....	(23)
第 4 章 常量、运算符、表达式与输入输出 .....	(34)
4.1 常 量 .....	(34)
4.2 运算符和表达式 .....	(37)
4.3 数据的输入输出 .....	(47)
第 5 章 循环结构程序设计 .....	(56)
5.1 while 语句 .....	(56)
5.2 do while 语句 .....	(58)
5.3 for 语句 .....	(61)
5.4 循环结构的嵌套 .....	(65)
5.5 break 语句和 continue 语句 .....	(67)

---

<b>第 6 章 数 组</b> .....	(74)
6.1 一维数组 .....	(74)
6.2 二维数组 .....	(77)
6.3 字符数组和字符串 .....	(81)
<b>第 7 章 函 数</b> .....	(91)
7.1 函数的定义、调用及声明 .....	(91)
7.2 函数参数及参数传递 .....	(96)
7.3 全局变量和局部变量 .....	(98)
<b>第 8 章 算法概述</b> .....	(100)
8.1 算法概念 .....	(100)
8.2 算法表示 .....	(101)
8.3 结构化程序设计思想 .....	(104)
<b>第 9 章 基础知识综合应用示例</b> .....	(107)
9.1 冒泡排序 .....	(107)
9.2 选择排序 .....	(109)
9.3 矩阵相加 .....	(112)
9.4 字符串查找 .....	(114)
<b>第 10 章 指 针</b> .....	(117)
10.1 有关指针的程序示例 .....	(117)
10.2 指针的运算 .....	(119)
10.3 指针与数组 .....	(120)
10.4 指针和数组作为函数的参数 .....	(131)
<b>第 11 章 结构体与共用体</b> .....	(141)
11.1 结构体 .....	(141)
11.2 共用体、枚举和用户定义类型 .....	(154)
11.3 链 表 .....	(156)

---

---

第 12 章 文 件 .....	(163)
12.1 文件概念 .....	(163)
12.2 文件的打开与关闭 .....	(166)
12.3 文件的读写 .....	(168)
12.4 文件的定位与查找 .....	(175)
第 13 章 函数与程序结构 .....	(180)
13.1 函数的组织 .....	(180)
13.2 函数递归 .....	(183)
13.3 函数与指针 .....	(185)
13.4 变量的存储类型 .....	(188)
13.5 宏定义 .....	(192)
13.6 编译预处理 .....	(193)
第 14 章 面向对象的程序设计基础 .....	(195)
14.1 面向对象的基本概念 .....	(195)
14.2 C++程序设计语言 .....	(196)
14.3 Java 程序设计语言 .....	(207)
附 录 .....	(213)
1. Turbo C 语言开发环境介绍 .....	(213)
2. 常用字符与 ASCII 代码对照表 .....	(218)
3. C 语言运算符的优先级与结合性 .....	(219)
4. Turbo C 2.0 常用库函数 .....	(220)
5. 常见错误信息表 .....	(227)
6. 常用词汇中英文对照表 .....	(230)
参考文献 .....	(232)



# 第1章 序 言

## 【本章要点】

- 计算机基本工作原理是什么？
- 什么是计算机程序？什么是程序语言？
- 程序和算法的关系是什么？
- 计算机语言具有哪些功能和特点？
- 如何使用 C 语言集成开发环境？

对于初学计算机和计算机程序设计的读者来说,了解和理解计算机工作原理和程序设计及设计语言的基本概念是十分必要的。当然,更为重要的是要通过不断地实践,逐步领会程序设计的基本思想和方法,提高程序设计能力。

## 1.1 计算机工作原理概述

计算机是 20 世纪人类最伟大的发明之一。它通过自动、连续地运行程序,能够代替人类完成各种复杂的计算和实现对各类信息的处理。

一个计算机系统由硬件和软件两大部分组成。硬件是指看得见、摸得着的设备实体,包括运算器、控制器、存储器、输入设备、输出设备等,如图 1-1 所示。软件则不能直接触摸,比如程序、文档等。构造硬件的基本思想是处理功能逻辑化,即用逻辑电路构造各种功能部件,如用门电路、触发器来构造运算器、控制器、存储器等。在硬件基础上,可以根据需要配置各种软件,如操作系统、编程语言、各种支撑软件等。硬件与软件按层次结构组成复杂的计算机系统。

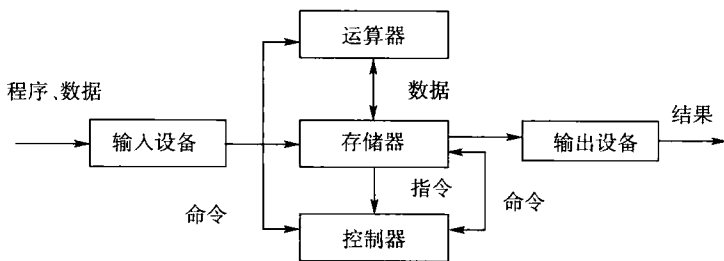


图 1-1 计算机组成示意图

一个计算机系统是如何工作的呢? 不管做一次复杂的数学计算,还是对大量的数据进行查询,或者对一个过程实现自动控制,用户都必须按照处理的步骤,用编程语言事先编写程序,然后通过输入设备(如键盘)将程序和需要处理的数据送入计算机,存放在存储器中。用户编写的程序称为源程序,是不能被计算机直接执行的。计算机只能执行机器指令,即要求计算机完成某种操作的命令,简称指令,如执行加法操作的加法指令、执行乘法操作的乘

法指令、执行传送操作的传送指令等等。因此,计算机在运行程序之前,必须将源程序转换为指令序列,并将这些指令按一定顺序存放在存储器的若干单元中。每个单元都有一个固定的编号,称为地址。只要给出某个地址,就能访问相应的存储单元,对该单元的内容进行读/写操作。

当计算机启动运行后,控制器将某个地址送往存储器,从该地址单元取回一条指令。控制器根据这条指令的含义,发出相应的操作命令,控制该指令的执行。比如执行一条加法指令,先要从存储单元或寄存器中取出操作数,送入运算器,再将两个操作数相加,并将运算处理的结果送回存储单元或寄存器存放。如果用户需要了解处理结果,则计算机通过输出设备(如显示器、打印机等),将结果显示在屏幕上,或打印在纸上。图 1-2 表示出了计算机的简单工作流程。

从以上的描述可以看出,计算机作为一个处理信息的工具,首先需要解决两个最基本的问题:第一,信息如何表示才能被计算机识别;第二,采用什么工作方式才能使计算机自动地对信息进行处理。对这两个问题的解决做出杰出贡献,并且产生深远影响的是一位美籍匈牙利数学家冯·诺依曼。他在 1945 年提出的 EDVAC 计算机设计方案中,总结了计算机的设计思想,被称为诺依曼思想,采用这一思想体制的计算机就称为诺依曼机。

诺依曼体制的主要思想包括:

- 采用二进制代码形式表示信息(数据、指令);
- 采用存储程序工作方式(这是诺依曼思想最核心的概念);
- 计算机硬件系统由五大部件(存储器、运算器、控制器、输入设备和输出设备)组成。

最简单地理解为:存储程序和程序控制。这些思想奠定了现代计算机的基本结构,并且开创了程序设计的新时代。

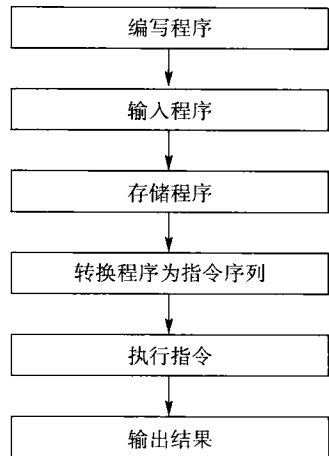


图 1-2 计算机工作流程

## 1.2 计算机程序与程序语言

### 1.2.1 程序

计算机程序(Program)是人们为了解决某种问题用计算机可以识别的代码编写的一系列加工步骤。计算机能严格地按照这些步骤去执行,程序执行的过程实际上是计算机对程序表达的数据进行处理的过程(计算机就是用来帮助人们进行信息处理)。一方面,计算机语言提供了一种表达数据和处理数据的能力;另一方面程序必须按照程序语言的规范来进行编写。

### 1.2.2 程序设计语言

我们都知道语言是交流的工具,中国人之间用汉语交流,英国人之间用英文交流等等。

同样计算机语言是计算机和编程人员之间的交流工具,一方面,计算机能执行计算机语言编写的代码;另一方面,程序设计人员要能编写和识别程序代码。语言都有一定的规则,计算机语言也不例外,但计算机语言也有自己和自然语言不通的地方。计算机语言(Computer Language)指用于人与计算机之间通讯的语言,是人与计算机之间传递信息的媒介。

在计算机发展的过程中,程序设计语言的发展经历了三个阶段:机器语言、汇编语言和高级语言。

### (1) 机器语言

机器语言是用二进制表示的,计算机能直接识别和执行的语言。机器语言执行速度最快。由于用二进制表示,因而程序可读性差,非专业人员难于掌握。机器语言中的每一条语句(称为指令)由操作码和操作数组成,反映了操作的对象和性质。

### (2) 汇编语言

为解决机器语言难于学习和掌握的问题,将二进制指令通过符号化表示就形成了汇编语言。汇编语言采用助记符号表示机器语言中的指令和数据,即用助记符号代替了二进制表示的机器指令,例如用 ADD(Addition)表示做加法的助记符,每条汇编语言的指令对应一条机器语言的指令。

计算机硬件只能识别机器指令。要执行汇编语言编写的程序,必须先用一个程序将汇编语言的程序翻译成机器语言的程序,用于翻译的程序称为汇编程序。

### (3) 高级语言

机器语言和汇编语言都是面向机器的,所谓的面向机器是指 CPU 不同,其机器语言和汇编语言都有可能不同。从 20 世纪 50 年代中期开始,逐步发展了面向问题(面向过程)和面向对象的程序设计语言,称为高级语言。高级语言表达方式接近被描述的问题,接近于自然语言和数学表达式,易于人们接受和掌握。随着计算机技术的发展,形成了为数众多的计算机高级语言,得到广泛应用的有十几种,每种高级语言都有其最适合的应用领域。

如:FORTRAN 语言适用于数值计算;BASIC 语言简单易学,适合于初学者学习;Pascal 语言适用于教学;C 语言适用于系统软件和应用软件的开发;Java 语言既是通用的程序设计语言,又是可以用于 Internet 开发的面向对象的程序设计语言。

由于计算机只能识别机器语言,因此,用高级语言编写的程序(或称源程序)都要通过专用的程序将高级语言翻译成机器语言程序(或称目标程序)后才能被计算机执行。将高级语言翻译成机器语言的方式有两种:一种是将高级语言程序一次性翻译成机器语言,再通过连接程序将系统库连接到程序中形成可执行程序,然后运行,称为编译型语言(如图 1-3 所示)。另一种是翻译一句执行一句,称为解释型语言。

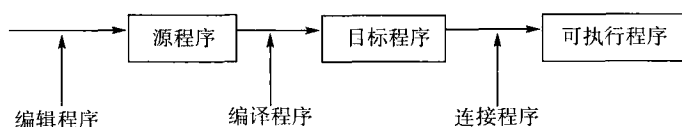


图 1-3 编译型语言的操作过程

## 1.2.3 程序设计语言的功能

程序设计语言是编程人员用来编写程序的,是编程人员与计算机交流的语言。编程人

员为了让计算机按照自己的意愿处理数据,那么程序语言必须具备数据表达和数据处理的能力,同时和自然语言一样,程序设计语言必须按照严格的语法规则来编写程序(如自然语言由词组成句子,由句子组成段落)。

### (1) 数据表达

在信息世界中数据是多种多样的,而语言本身的描述能力是有限的。为了使程序语言能充分表达各种各样的数据,一般将数据抽象为若干种类型。数据类型就是对某些具有共同特点的数据集合的总称。如人们常说的整数、实数就是数据类型的例子。数据类型涉及两方面内容:一方面该数据类型代表的数据是什么(即数据类型的定义域)?另一方面能在这些写数据上执行什么样的操作(即操作,又称运算)?如我们所熟悉的整数: $\{\dots, 0, 1, 2, 3, \dots\}$ ,能执行的操作有 $+$ 、 $-$ 、 $*$ 、 $/$ 等。

在程序设计语言中,一般都要事先定义几种基本数据类型供程序员直接使用,如整型、浮点型(实型)、字符型等。这些基本数据类型在程序中的基本对象主要是两种形式:常量和变量。常量在程序中是不变的,如 100 是整型常量,13.8 是浮点型常量,'A'是字符型常量等。变量则它的值可以改变,可以对它进行相关的操作,如在 C 语言中可以通过 `int i;` 来定义一个整型变量 `i`,`i=20` 是对变量 `i` 赋值为 20。

然而,在现实中有许多复杂的数据类型,无法用程序设计语言的某种基本数据类型来表示,但程序设计语言提供了构造新的具体数据类型的功能,如数组(Array)、结构体(Structure)、文件(File)、指针(Pointer)等。

总之,程序设计语言提供的基本数据类型和构造复杂数据类型的功能,为有限能力的程序设计语言表达客观世界中的多种多样的数据提供了良好的基础。

### (2) 数据处理

数据处理即流程控制,程序设计语言除了能表达各种各样的数据外,还必须提供一种手段来表达数据处理过程(程序流程控制)。程序的控制过程是通过程序的语句来实现的。

当要解决的问题比较复杂时,程序控制过程也会变得十分复杂。一种比较典型的程序设计方法:将复杂程序划分为若干个相互独立的模块,使完成每个独立的模块变得简单容易实现,在设计时不受其他模块的干扰。同时,通过现有模块积木式的扩展就可以形成复杂的、更加庞大的程序模块。这种程序设计方法称之为结构化程序设计。C 语言就是支持这种设计方法的典型语言。

按照结构化程序设计的观点,任何程序都可以将模块通过三种基本的控制结构实现。这三种控制结构是:顺序、分支、循环。

### (3) 语法规则

程序员用程序设计语言编写程序处理相应的问题。在程序中,一般要表达数据,包括定义用于存储数据的变量;还要描述数据处理的过程,包括语句级的控制和单位级的控制。为了让计算机能理解程序员在程序中所描述的这些工作,用程序设计语言所编写的程序必须符合相应的语法规则(Grammar)。

编写程序就像用自然语言来写文章,首先语法要通,即要符合语言所规定的语法规则。当然,语法通了并不意味着你的文章就符合要求了,你有可能词不达意、离题万里。后者就是在程序调试时需要发现的事,即找出程序中的错误(非语法错误)。这是一个需要耐心和经验的过程,而语法错误的检查相对容易多。

### 1.3 程序设计与算法

一个程序应包括以下两个方面内容:

- (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。即操作步骤,也就是算法。

数据是操作的对象,操作的目的是对数据进行加工处理,以得到期望的结果。作为程序设计人员,必须认真考虑和设计数据结构和操作步骤。著名的瑞士计算机科学家 Niklaus Wirth 提出了一个公式:程序=数据结构+算法。

广义地讲算法是解决问题的逻辑步骤,是对特定问题求解步骤的一种描述。简单地说,任何解决问题的过程都是由一定的步骤组成的,把解决问题的确定方法和有限步骤称为算法。只有通过算法能够描述出来的问题,才能够通过计算机求解。对同一个问题,可以有不同的解题方法和步骤,也就有不同的算法。

计算机算法:是用程序解决问题的逻辑步骤,是指令的有限序列。

#### 1. 算法的基本特征

算法是一个有穷规则的集合,这些规则确定了解决某类问题的一个运算序列。对于该类问题的任何初始输入值,它都能机械地一步一步地执行计算,经过有限步骤后终止计算并产生输出结果。归纳起来,算法具有以下基本特征:

- (1)有穷性:一个算法应包含有限的操作步骤而不能是无限的。
- (2)确定性:算法中每一个步骤应当是确定的,而不能是含糊、模棱两可的。
- (3)有效性:算法中每一个步骤应当能有效地执行,并得到确定的结果。
- (4)输入:有零个或多个数据的输入。
- (5)输出:有一个或多个数据的输出。

#### 2. 算法的表示

原则上说,算法可以用任何形式的语言和符号来描述,通常有自然语言、伪代码、流程图、N-S图、PAD图、程序语言等。现在简单地介绍一下自然语言和流程图(更详细介绍参照第八章)。

##### (1)用自然语言表示

自然语言可以是中文、英文、数学表达式等。用自然语言表示通俗易懂,缺点是可能文字过长,不太严格,表达分支和循环结构不方便。

**【例 1-1】** 如求数列  $1+2+\dots+m$  的值  $N$ ,当  $N>10000$  时结束。

算法可表示如下:

- ①  $n=0$ ;
- ②  $m=0$ ;
- ③  $m$  加 1;
- ④  $N$  加  $m$ ;
- ⑤ 判  $N$  是否大于 10000,如果满足关系结束;不满足关系继续执行③④。

##### (2)用流程图表示

流程图是用一些图框来表示各种操作。优点是直观形象,简单易于理解,便于修改和交

流。ANSI 规定了一些常用的符号(图 1-4)。

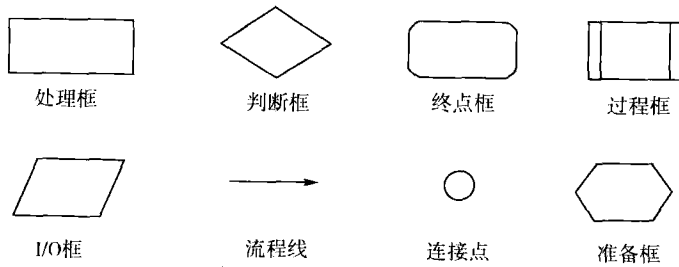


图 1-4 标准流程图符号

【例 1-2】 求  $1+2+3+\dots+10$ 。

用流程图表示如图 1-5。

计算机无法识别自然语言和流程图,只有用计算机语言编写的程序才能被计算机执行。因此在使用自然语言和流程图描述出算法后,还要将它转换成计算机语言程序。

如【例 1-2】用 C 语言表示如下:

```
#include<stdio. h>
void main()
{
    int i, s;
    s=0;
    for(i=1;i<=10;i++)
        s=s+i;
    printf("%d", s);
}
```

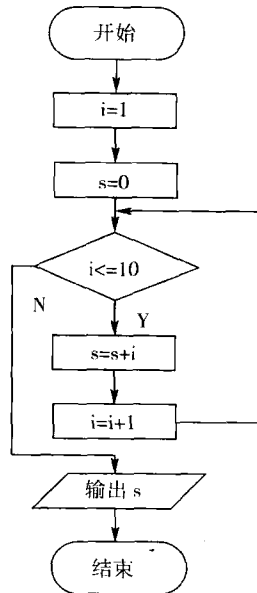


图 1-5

## 1.4 C 语言的发展与特点

### 1.4.1 C 语言的发展简史

C 语言是目前国际上广泛流行、很有发展前途的计算机高级语言。适合作为系统描述语言,既可以编写系统软件(如操作系统),也可以编写应用软件。

早先的操作系统主要用汇编语言编写。汇编语言依赖于硬件,程序的可读性和可移植性差。人们一直在寻找一种很接近计算机硬件的高级语言,C 语言就是在这种情况下应运而生的。C 语言是一种既具有一般高级语言特性,又具有低级语言特性的语言,集它们的优点于一身。

C 是在 B 的基础上发展起来的。它的前身是 ALGOL 60(1960 年,面向问题的高级语言,离硬件比较远,不宜用来编写系统程序)。1963 剑桥大学推出了 CPL(复合程序设计语

言),CPL 语言接近硬件一些,但规模较大,难以实现。1967 年剑桥大学的马丁·理查德对 CPL 进行了简化,推出了 BCPL(基本复合程序设计语言)。1970 年美国贝尔实验室的肯·汤普逊对 BCPL 进行了进一步的简化,突出敢硬件处理能力,并取了“BCPL”的第一个字母“B”作为新语言的名称,并且用 B 语言编写了第一个 UNIX 操作系统。但“B”语言过于简单,功能有限,1972 年贝尔实验室的布朗·W. 卡尼汉和丹尼斯·M. 利奇对 B 语言进行了完善和扩充,在 B 语言的基础上设计出了 C 语言。C 语言保持了 BCPL 语言和 B 语言精练、接近硬件的优点,又克服了它们过于简单、数据无类型、功能有限的缺点。后来多次改进,在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版的突出优点才引起人们普遍注意。

1977 年,为了让 C 语言脱离 UNIX 操作系统,成为在任何计算机上都能运行的通用计算机语言,布朗·W. 卡尼汉和丹尼斯·M. 利奇合著了著名的“C 程序设计语言”一书,对 C 语言的语法进行了规范化的描述,成为当时的标准,通常简称为《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,后来由美国国家标准协会(American National Standards Institute)在此基础上制定了一个 C 语言标准,于 1983 年发表。通常称之为 ANSI C。

现在使用的各种 C 语言编译系统虽然基本部分是相同的,但也有一些不同。在微型机上使用的有 Microsoft C, Turbo C, Quick C, Borland C 等,它们的不同版本又略有差异。因此,在使用一个系统之前,要了解所用的计算机系统配置的 C 编译系统的特点和规定。

#### 1.4.2 C 语言的特点

C 语言同时具有汇编语言和高级语言的优势,它把高级语言面向过程和低级语言与硬件关系密切的优点有机地结合起来,具有许多显著的特点:

- (1) 语言简洁、紧凑,使用方便、灵活。ANSI C 一共只有 32 个关键字,9 种控制语句(if ~ else、for、while、do ~ while、continue、break、switch、goto、return)。
- (2) 程序书写自由。主要用小写字母表示,压缩了一切不必要的成分。
- (3) 运算符极其丰富。C 语言共有 34 种运算符。并把括号、赋值、逗号等都作为运算符处理。从而使 C 的运算类型极为丰富,可以实现其他高级语言难以实现的运算。
- (4) 数据结构类型丰富。基本类型、结构类型(数组、结构体、联合体、枚举型)、指针、空类型等,为处理各种复杂数据类型提供了实用的手段。
- (5) 具有结构化的控制语句。C 语言程序的主要成分是函数,用函数作为程序模块,实现了模块化。另外 C 语言具有多种结构化的控制语句(if ~ else、for、while、do ~ while 等),可以很容易地实现结构化的各种基本结构,用来设计结构化程序。
- (6) 生成的目标代码质量高,程序执行效率高。
- (7) 可移植性好(较之汇编语言)。
- (8) C 语言允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。

#### 1.4.3 程序编译与编程环境

##### 1. 程序的编译

计算机硬件能理解的只有计算机的指令,用程序设计语言编写的程序不被计算机直接

执行,这就需要有一个软件将相应的程序转换成计算机能直接理解的指令序列。对于 C 语言等高级程序设计语言来说,这种软件就是编译程序,即编译器(Compiler)。编译器首先对源程序进行词法分析,然后进行语法和语义分析,最后生成可执行代码。如果程序中有语法错误,编译器会直接指出其中的语法错误。但是,编译器能生成可执行代码并不意味着程序就没有错误了,它只说明没有语法错误。对于程序中的逻辑错误,编译器发现不了,必须通过调试才能发现。

## 2. 编程环境

编写一个程序要做很多工作,包括编辑、编译和调试等过程。所以,许多程序设计语言都有相应的编程环境。程序员可以直接在该环境中完成上述工作,以提高效率。具体 Turbo C 的开发环境见附录一。

## 习 题

1. 简述计算机工作原理。
2. 简述程序语言的发展阶段以及高级程序语言的功能。
3. 什么是算法,算法与程序的关系?
4. 用流程图表示  $1+2+3+4+\dots+100$ 。



## 第2章 用C语言编写程序

### 【本章要点】

- 如何编写简单C语言程序?
- C语言有哪些控制结构?
- C语言程序基本组成如何?
- C语言有哪些最基本语法?

在第一章中,我们对程序和程序设计语言有了一些初步认识,了解了计算机的工作原理和程序的功能,在实际生活中,无论多么复杂的计算机系统无一不运行着程序。但无论多么复杂的系统,它都是一个个程序在运行着,其设计者也都是从最简单的程序设计开始的。

无论从编写应用软件还是编写系统软件来说,C语言无疑都是最实用的程序设计语言。这里,我们首先从编写简单C程序开始。

### 2.1 输入输出程序示例

#### 2.1.1 简单C程序解析

【例2-1】 在显示屏幕上输出“Hello,C Program!”。

问题求解:这个问题很简单,直接使用C语言输出函数将字符串输出。

求解步骤(略)。

程序清单:

```
1 #include<stdio. h>
2 void main() /* 定义主函数 main() */
3 {
4 printf("Hello,C Program! \n"); /* 使用 printf()函数输出字符串 */
5 }
```

运行结果:

Hello,C Program!

现在分析上述程序,初步熟悉C程序:

第1行,#include<stdio. h>是C程序编译预处理之“文件包含”(在后续章节中有专门学习)。第2行,定义主函数main(),任何一个C程序有且只有一个主函数main(),无论主函数main()位置在那里,当程序运行时,总是首先从主函数main()开始执行。第4行,C语言的输出函数printf()调用语句,输出字符串,其中'\n'不是字符串部分而是换行符,它表示字符串输出后换行(我们称之为转义字符,见后续章节)。

第3、5行,为对称的大括号{ },大括号把函数的语句括起来,称为函数体。