



袁淑君 师书恩 林钧礼 编

BASIC 结构程序设计

北京师范大学出版社

BASIC 结 构 程 序 设 计

袁淑君 师书恩 林钧礼 编

北 京 师 范 大 学 出 版 社

内 容 提 要

本书全面系统地介绍了 BASIC 语言及其结构程序设计方法。全书共有九章,包括基本 BASIC 和扩展 BASIC 以及文件处理。第八章集中讨论了字符处理函数及其应用,大大加强了 BASIC 语言字符处理的能力。

本书在介绍语言的过程中,突出地介绍了结构程序中 TOP-DOWN 的设计方法,力求使读者形成良好的正确的程序设计习惯,以提高程序设计技巧和保证程序的质量。

本书是北京师范大学计算中心的教师在校内讲课的基础上编写的,书中具有较多的例题,每章都附有小结和习题,并在最后附有习题的参考答案。全书通俗易懂、深入浅出、内容丰富,适合中等以上文化程度的读者自学,也可作为大专院校学生学习 BASIC 语言的教材以及其它部门的计算机应用工作者、技术人员的参考书。

BASIC 结构程序设计

袁淑君 师书恩 林钧礼 编

北京师范大学出版社出版

新华书店北京发行所发行

北京朝阳展望印刷厂印刷

开本: 787×1092 1/16 印张: 20.5 字数: 507 千

1985 年 5 月第 1 版 1988 年 8 月第 4 次印刷

印数: 67 500—70 500

ISBN7-303-00305-3/O·78

定 价: 3.85 元

出版说明

BASIC语言简单易学，使用方便，不仅能进行科学计算，而且具有很强的数据处理和图形处理功能。对初学计算机语言的人来讲，它是一种能较快入门的算法语言，所以它受到人们的欢迎。但是，掌握BASIC语言仅仅了解BASIC的基本语句是很不够的，重要的是掌握BASIC程序设计方法。本书正是一本在介绍BASIC结构程序设计方法方面很有特色的书。

本书在详细介绍BASIC-PLUS语言的同时，十分强调程序设计方法。书中使用的TOP-DOWN编程方法是结构程序设计中广泛应用的一种方法。使用这种方法编写程序，是对所要解决的实际问题进行分析，把一个大而复杂的问题分成若干独立的子问题，拟出系统流程，并标出数据在系统流程中的流动关系，展示出求解问题的一般轮廓，以此作为流程图的第一级。然后对每一个子问题用划一的方法分别处理，即分析问题，定义数据，发展算法并逐步细化，直到很容易用程序设计语言描述为止。特别值得提出的是，本书在定义数据时使用了数据表概念，用它来描述程序中所使用的数据及变量，增加了程序的易读性。使用这种方法设计出的BASIC程序，结构性强，易读，易修改，从而提高了程序的质量。此外，这种方法对学习其它计算机语言的程序设计也是有帮助的。

本书有一定的通用性。它从程序结构这个角度来讲述BASIC语言的语句，指出某种结构可能使用的语句。书中还包括有大量的事务管理和数值计算方面的例题和习题，并说明如何使用正确的方法定义数据、发展和加细算法，最后给出程序及运行结果。因此，这本书不仅可作为BASIC语言的教科书，而且也可作为广大计算机应用人员的参考书。

前 言

BASIC语言是一种易学、具有人机会话功能的程序设计语言。自从二十世纪六十年代中期第一个BASIC语言文本问世以来，由于这种语言简单明了、易学好懂，受到了人们的普遍欢迎，很快就成为国际通用的计算机算法语言。随着计算机应用的日益广泛，BASIC语言也在不断发展。虽然它的规模仍不大，但其功能大为增强，至今已很适于一般的数值计算和事务管理，在国民经济多种多样的应用中都可以看到它的足迹。加上这种语言便于检查和调整，使用灵活方便，很容易被初学者掌握，所以对计算机语言的教育和普及起了极大作用，在众多的计算机语言中，它被誉为六十年代的“十大功臣”之一。目前，许多计算机系统都配有BASIC语言，特别是在小型机和微型机上，BASIC语言已成为必配语言之一。

这本书以功能较强的BASIC-PLUS为基础，全面地介绍BASIC语言，读者可以根据自己的实际情况决定取舍。

本书的特点是：它不只是描述BASIC语言的定义、概念和使用规则，而是在介绍BASIC语言的同时，突出了结构程序设计方法。学习程序设计语言的根本目的在于应用，在于解决各种实际问题。因此，学习程序设计方法是学习语言的重要组成部分。为此，我们在第三章集中地介绍了如何使用逐步加细的程序设计方法解题：分析所要解决的问题，建立数据表；确定算法并逐步加细描述算法，直到很容易用语言描述为止；最后编制出程序。算法描述采用了类似D—流程图的符号，使算法直观、易读、好修改。数据表的应用，增加了程序的易读性，克服了BASIC语言使用变量自由而可能造成程序易错、难读的不足。全书各章中的例题和习题突出了这种方法的应用，以培养解题能力，提高编写程序的质量。

本书的第五章介绍了实现选择结构和循环结构的各种语句。在循环语句中除了一般的FOR-NEXT循环外，还介绍了条件循环语句WHILE-NEXT和UNTIL-NEXT、条件FOR循环以及隐含循环的语句修饰，这些语句增强了BASIC语言的功能，扩大了它的应用范围。第八章集中地讨论了字符串处理函数及应用。大家将会在这一章里看到，BASIC-PLUS所提供的字符串处理函数的数量之多、功能之强是其它版本的BASIC语言少见的，这为扩大BASIC语言的应用领域提供了强有力的工具。此外，在这一章还进一步说明了库函数中的几个标准函数，这一部分也可以移到第二章标准函数部分一起讲述。在第九章中介绍了三种形式的数据文件：ASCII码文件、虚数组文件和Block I/O文件。这些形式的数据文件在数据管理中十分有用，特别是虚数组文件，它不仅可以顺序访问，也可以随机访问，使用简单方便，颇受欢迎。

本书安排了大量的用于数值计算的例题和习题，还有数量更多的用于事务管理的例题和习题。因此，它不仅可以用作BASIC语言的教科书，也可作为广大计算机工作者的参考书。我们希望它在普及计算机应用的洪流中起一定的推动作用。

在使用本书时，需注意以下符号：

1. 下横线 () 表示用户打入的信息。

目 录

第一章 人、语言和计算机	(1)
§1-1 计算机的组成	(1)
一、计算机的处理对象	(1)
二、使用计算机	(1)
三、计算机进行数据处理的基本原理和计算机硬件组成	(2)
四、计算机软件组成	(5)
§1-2 语言、程序和程序设计	(5)
一、低级语言	(6)
二、高级语言	(6)
三、BASIC语言	(7)
四、高级语言执行过程	(7)
五、程序与程序设计	(8)
小结	(9)
第二章 BASIC程序	(11)
§2-1 BASIC程序结构	(11)
一、行号	(11)
二、语句	(12)
§2-2 数据类型	(13)
一、常量	(14)
二、变量	(14)
§2-3 运算符和表达式	(15)
一、算术表达式	(15)
二、关系表达式	(17)
三、布尔表达式	(18)
四、BASIC运算的优先次序	(19)
五、赋值算符	(20)
§2-4 标准函数	(20)
§2-5 BASIC程序的建立、运行和修改	(21)
一、进入和退出RSTS/E系统	(21)
二、几个编辑命令	(22)
§2-6 获得清单的命令	(25)
一、列出文件目录(CATALOG命令)	(25)
二、显示内存中程序长度(LENGTH命令)	(25)
小结	(25)
习题	(26)

第三章 BASIC语言的结构程序设计	(28)
§3-1 程序设计要点	(28)
一、定义数据	(28)
二、问题的算法	(29)
三、流程图	(30)
§3-2 结构程序设计的控制结构	(31)
§3-3 选择和循环	(33)
一、选择结构	(33)
二、循环结构	(35)
§3-4 调整BASIC程序	(37)
小结	(38)
习题	(39)
第四章 输入和输出	(41)
§4-1 提供数据语句	(41)
一、赋值语句	(41)
二、交互式的数据输入语句(INPUT语句)	(43)
三、读数据语句(READ), 数据语句(DATA), 和数据恢复语句(RESTORE)	(46)
四、三种提供数据语句的比较	(50)
§4-2 打印语句(PRINT语句)	(52)
一、PRINT语句的用途	(53)
二、PRINT语句输出格式	(55)
三、PRINT语句和LET语句的比较	(58)
小结	(58)
习题	(59)
第五章 选择结构和循环结构	(61)
§5-1 选择结构	(61)
一、无条件转移语句(GOTO语句)	(61)
二、条件转移语句	(63)
三、多重选择结构	(68)
四、散转语句(ON-GOTO语句)	(74)
§5-2 循环结构	(75)
一、FOR循环结构——计数器控制循环	(75)
二、WHILE/UNTIL循环结构——条件循环结构	(81)
§5-3 条件循环结构的使用	(85)
一、使用标记值终止循环	(85)
二、使用计算结果控制循环	(88)
§5-4 有条件地结束FOR循环	(91)
§5-5 语句修饰符	(93)
一、修饰符IF和UNLESS	(94)
二、FOR修饰符	(95)
三、UNTIL和WHILE修饰符	(95)

四、嵌套修饰符	(96)
§5-6 嵌套循环	(97)
§5-7 错误处理	(102)
一、错误转向控制语句(ON ERROR GOTO)	(102)
二、退出错误处理子程序(RESUME)	(104)
§5-8 暂停程序运行(SLEEP和WAIT)	(105)
§5-9 停止程序运行	(106)
一、结束语句(END)	(106)
二、暂停语句(STOP)	(106)
小结	(106)
习题	(107)
第六章 数组和下标变量	(111)
§6-1 数组说明	(111)
§6-2 下标、下标的值和下标变量	(112)
§6-3 数组操作	(113)
一、数组的初始化	(113)
二、数组元素的读入和印出	(114)
§6-4 部分地使用数组元素	(119)
§6-5 数组的排序和检索	(124)
一、冒泡排序	(124)
二、冒泡排序的BASIC程序	(125)
三、检索数组	(127)
§6-6 矩阵操作	(131)
一、矩阵的读入和打印	(131)
二、矩阵初始化	(132)
三、矩阵运算	(134)
小结	(138)
习题	(139)
第七章 子程序和自定义函数	(142)
§7-1 子程序	(142)
一、转子语句和返回语句	(142)
二、子程序嵌套	(149)
三、条件转向语句(ON—GOSUB)	(149)
§7-2 自定义函数	(150)
一、库函数	(150)
二、自定义函数	(155)
三、函数参数和全局变量	(164)
§7-3 程序举例——简单统计问题	(167)
小结	(174)
习题	(175)
第八章 字符串处理	(177)

§8-1 字符串的长度——串长度函数(LEN)	(177)
§8-2 子字符串	(178)
一、子字符串标记	(178)
二、字符串函数	(178)
三、查找子字符串	(180)
§8-3 字符串操作	(181)
一、字符串连接	(181)
二、字符串表达式和字符串比较	(181)
三、数字字符串运算函数	(187)
§8-4 转换函数	(187)
一、数字字符串和数的相互转换	(187)
二、ASCII字符串/数字表转换语句(CHANGE)	(189)
三、字符串及其ASCII码值间的转换	(190)
四、字符串到字符串的转换	(191)
§8-5 产生特殊字符串的函数	(193)
一、SPACE\$函数	(193)
二、STRING\$函数	(193)
§8-6 日期和时间函数(DATE\$和TIME\$)	(194)
一、DATE\$函数	(194)
二、TIME\$函数	(194)
§8-7 应用举例	(195)
一、密码生成程序	(195)
二、扫描FOR循环头程序	(197)
三、本文编辑问题	(201)
小结	(202)
习题	(203)
第九章 格式输出和文件	(206)
§9-1 格式输出	(206)
一、格式输出语句(PRINT USING)	(206)
二、打印数字	(206)
三、打印字符串	(211)
四、程序举例	(212)
§9-2 文件处理	(214)
一、文件和设备	(214)
二、设备名、文件名和通道	(215)
三、文件组织	(216)
四、程序访问文件或设备的过程	(216)
五、文件操作	(219)
§9-3 ASCII码文件	(220)
一、打开和关闭ASCII码文件	(220)
二、读/写数据操作	(220)

三、ASCII码文件举例	(223)
§9-4 虚数组文件	(228)
一、虚数组说明(DIM#语句)	(228)
二、打开和关闭虚数组文件	(229)
三、访问虚数组文件	(229)
四、程序举例	(235)
§9-5 Block I/O文件	(237)
一、打开和关闭Block I/O文件	(237)
二、读/写数据操作	(238)
三、访问I/O缓冲区	(239)
四、数据转换(CVT函数)	(242)
五、程序举例	(243)
小结	(246)
习题	(247)
习题参考答案	(248)
附录	(306)
附录一: ASCII码表	(306)
附录二: BASIC-PLUS的保留关键字	(307)
附录三: PDP-11 BASIC-PLUS错误信息	(309)

第一章 人、语言和计算机

掌握一种程序设计语言，是使用计算机的前提条件。这意味着，如果你能用“语言”去编写计算机能运行的程序，计算机就会协助你完成各项工作，行使事务管理、科学运算、生产过程控制或者完成信息管理等，使人们从繁琐的劳动中解脱出来，节省大量的时间去从事创造性工作。

我们在这一章向读者介绍计算机的基本组成部分，进而明确人与计算机是借助于用“语言”编写的程序联系起来；接着讲述程序设计语言是什么，程序与程序设计又包括什么内容。

§1-1 计算机的组成

一、计算机的处理对象

在没有涉及到计算机的构成前，先介绍一下计算机的处理对象是什么。计算机无论用在哪个领域，无论协助人完成什么任务，计算机所能处理的是数据，即它的处理对象是数据。

什么是数据呢？一提到数据有人会马上想到数值或数字。当然，数字、数值是数据，可数据的定义比起数字与数值更广泛。数据可以这样定义：凡是能被计算机接受，并能被处理的文字、数字、符号等及其各种不同组合。

二、使用计算机

电子数字计算机是一种能自动、高速、精确地完成各种数据存贮与处理、数字运算和控制功能的电子设备。它的基本功能是数字计算。组成它的物质基础主要是电子器件，所以称它为电子数字计算机，或简称计算机。当今计算机在信息社会中更显示处理信息的威力，它已深入到各个经济领域，企业、事务的管理和人们的日常生活中，它接受各式各样的数据，进行存贮、处理并输出处理结果，因此有人认为，信息社会是以计算机为核心的。

随着计算机应用领域的扩大，计算机本身有了很大发展，它的发展经历了：电子管、晶体管、集成电路和大规模集成电路四代，现在正向第五代计算机过渡。与此同时软件也有了相

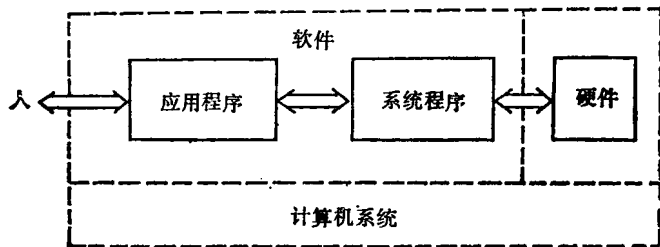


图1-1 人和计算机系统之间的联系

应的发展, 因此通常说计算机实际是指计算机系统。任何计算机系统无论大小、功能强弱如何, 都由图1-1所示的两大部分: 硬件和软件组成。

使用计算机, 人与计算机系统必然要发生某种联系。人们使用的硬件是输入输出设备, 终端机等, 将数据输入并处理, 把结果输出来等都是通过应用程序和系统程序等软件来完成的, 计算机硬件是靠系统程序指挥自动运行的。人们为了让计算机按自己的意志处理各种问题, 是在计算机自动运行的情况下, 执行事先编好的应用程序, 因此图1-1在某种程度上也展示人与计算机系统之间的关系。

三、计算机进行数据处理的基本原理和计算机硬件组成

计算机是一种数据处理机器, 它能进行数值计算、逻辑运算、自动控制、辅助设计、智能模拟等等。这些数据处理是各式各样的, 各有其特殊性, 但计算机处理它们的过程是基本一致的, 处理的算法也总是归结为一些基本的算术和逻辑操作。也就是说, 计算机进行各种数据处理的基本原理是相同的。所以只要弄清了具有代表性的计算机进行数值计算的基本原理, 就可推知计算的基本组成。

(一) 计算机进行数值计算的原理和硬件组成

为了了解计算机进行数值计算的基本原理, 先看看人用简单的计算工具如算盘或计算器等是怎样进行数值计算的, 也就是先来描述一下由人、计算器及纸和笔组成的系统进行计算的过程。

假设给出 x 值, 计算 e^x ($e=2.71828$), 即对 x 值按算法 e^x 进行处理。计算过程可描述如下:

(1) 变换算法使之适合于数值计算方法的马克劳林公式

$$e^x = \sum_{n=0}^R \frac{x^n}{n!} \quad (\text{根据精度要求, } R \text{ 取某一正整数})$$

即
$$e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^R}{R!}$$

其中 $0! = 1$, $1! = 1$, $2! = 1 \times 2$, $3! = 1 \times 2 \times 3$, $\cdots R! = 1 \times 2 \times 3 \cdots \times R$, 由这个算法可得到计算 e^x 的数值解。

(2) 编制数值运算的操作步骤: 假若给定 $x=2$, 且若上述数值解中 R 取3能满足精度要求。于是

$$e^x \doteq 1 + 2 + \frac{2^2}{1 \times 2} + \frac{2^3}{1 \times 2 \times 3}$$

第一步 计算 $1+2$

第二步 计算 2×2

第三步 计算 1×2

第四步 计算 $\frac{(2 \times 2)}{1 \times 2}$

第五步 计算 $(2 \times 2) \times 2$

第六步 计算 $(1 \times 2) \times 3$

第七步 计算 $\frac{(2 \times 2 \times 2)}{(1 \times 2 \times 3)}$

第八步 计算 $(1+2) + \frac{2^2}{1 \times 2}$

第九步 计算 $\left(1+2+\frac{2^2}{1 \times 2}\right) + \frac{2^3}{1 \times 2 \times 3}$

这个数值计算操作步骤由人用笔记录在纸上。

(3) 人依次读出运算操作步骤，操纵计算器，实行具体运算操作，并将每次运算获得的结果记录在纸上。于是可得如下操作过程表：

表 1-1

	读 出 操 作 步 容	执 行 操 作	记 录 结 果
(1)	$1+2$	加	3
(2)	2×2	乘	4
(3)	1×2	乘	2
(4)	$\frac{(2 \times 2)}{(1 \times 2)}$	除	2
(5)	$(2 \times 2) \times 2$	乘	8
(6)	$(1 \times 2) \times 3$	乘	6
(7)	$\frac{(2 \times 2 \times 2)}{(1 \times 2 \times 3)}$	除	1.333
(8)	$(1+2) + \left(\frac{2^2}{1 \times 2}\right)$	加	5
(9)	$\left(1+2+\frac{2}{1 \times 2}\right) + \left(\frac{2^3}{1 \times 2 \times 3}\right)$	加	6.333

(4) 操作步骤执行完最后一步所得的结果是最后结果(注意，这里是为了说明计算过程， R 取3，所得结果精度很差)。

对于这样一个人、计算器、纸和笔的计算系统如果能用相应的电子设备来代替就可以组成一台计算机。即用一个称为控制器的部件代替人的脑和手的作用，它是计算机的“神经中枢”，由它统一指挥和控制计算机各部分的联系和操作；用一个称为运算器的部件来代替计算器的功能；用一个称为存贮器的部件来代替纸和笔的作用。此外，由于上述过程(1)要求数值解，(2)编制数值计算的步骤一般只能是由人完成的，所以还需要一个把编制好的计算

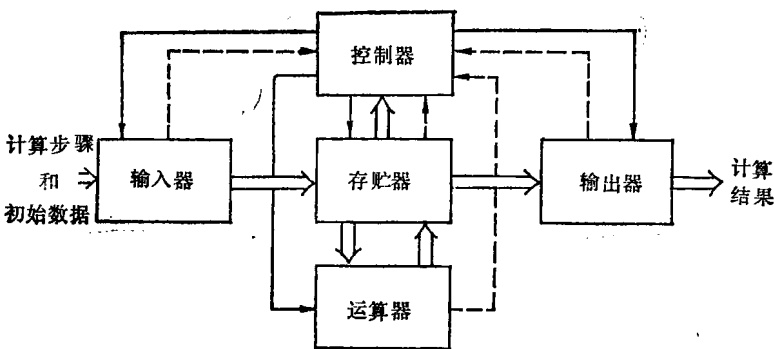


图1-2 计算机基本原理框图

步骤和初始数据告诉给机器的输入器；上述过程(4)机器得到最终结果要回答给人，这还需要一个称为输出器的部件。把控制器、运算器、存贮器、输入器、输出器按上述人、计算机及纸和笔的计算系统的工作过程连接起来就组成了计算机的硬件整体（组成计算机系统的各种实际设备，总称为硬件）。如图1-2所示

图中： $\Rightarrow$ 表示数字化的计算步骤中，输入的初始数据，中间结果及最终结果，即数据流的方向。

$\rightarrow$ 表示控制器发出的控制命令信息。

$\rightarrow$ 表示被控部分状态反馈信息。

对于这样一台计算机进行数值计算的基本原理可简述如下：当要计算的数学问题已经找到相应的数值逼近方法，并编制成数值计算步骤(步骤也被数字化)之后，计算机的控制器就命令输入器将计算步骤和初始数据记录到存贮器中。然后控制器从存贮器依次读出第一个计算步骤，根据读出步骤的规定控制运算器对步骤中包含的数据执行确定的运算操作，控制器从反馈信号得知运算操作完，把所得中间结果记回存贮器，然后经反馈信号通知控制器取第二个计算步骤，重复上述第一个步骤的执行过程。如果依次读出所有步骤并执行之，直至最后一个步骤执行完，将最终结果存于存贮器内。整个运算一经完成，控制器就命令输出器把存放于存贮器的最终结果打印或显示出来。

归结起来就是：输入器在控制器控制下输入计算步骤和初始数据，控制器从存贮器读出计算步骤并分析步骤，指挥运算器进行处理，最后命令输出器以各种方式输出最终结果。这一切工作是由控制器控制的，而控制器赖以控制的依据则是存放在存贮器中的计算步骤。

(二)计算机硬件组成框图

如上所述，一台计算机通常由输入器、存贮器、运算器、控制器和输出器五个基本部分组成。由于存储器、运算器、控制器三部分起运算操作(数据加工处理)的主要作用，所以把它们合称为主机，而输入器、输出器及外存贮器则合称为“外部设备”。又因为运算器和控制器大都仅由单一的电子线路构成，特别是由大规模和超大规模集成电路组成的计算机中，控制器和运算器不论在逻辑关系上还是在结构上都有十分紧密的联系，多装配在十分紧密的结构中，甚至干脆组装在一起。同时它们又是数据加工处理的中心部件，所以把它们合称为“中央处理机”(即所谓的CPU)。另外，运算器的基本功能是完成算术、逻辑运算操作。于是有人把它叫做“算术逻辑部件”(所谓的ALU)，这样中央处理机(CPU)有算术逻辑部件和控制部件两部分。此外，由于CPU是高速工作部件，外部设备是低速工作部件，所以为了解决这个矛盾，通常一台计算机要配备多种多台外部设备，要求输入、输出的控制电路能使主机与外部设备同时工作，各外部设备能够同时工作。这个功能的扩大使得计算机的输入、输出控制电路发展为计算机的外部设备通道(即所谓的I/O通道或I/O交换器或接口)。在接口控制下，可以实现多台外部设备与主机成批快速交换数据。

按照这样演变的结果，一台典型计算机是由中央处理机(CPU)，内存贮器(也称主存MAIN MEMORY)，输入/输出(I/O)通道(接口)及其设备等三部分组成它的硬件实体，其组成简框图示于图1-3。

细心的读者也许已看到，人和计算机系统之间的联系，从硬件来看，是通过输入/输出设备——输入器和输出器实现的。它在人和计算机系统之间起着数据变换的作用，把人们所熟悉的数据表达形式变为机器内部所能接受和识别的数据形式(输入时)或进行相反的变换

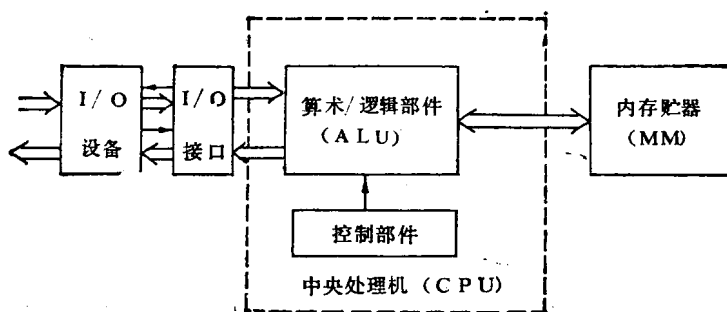


图1-3 计算机组成简框图

(输出时)，但是它还不能解释与人相联系的数据的含义和用法规则，这部分是由下面所要描述的软件来完成的。

四、计算机软件组成

前边我们已讲过，计算机系统除了硬件就是软件，计算机系统是硬件和软件结合的有机整体。软件是相对硬件而言的，也有人称软件为软设备。软件是计算机运行所需要的各种程序及其有关文件与资料，象汇编程序、各种高级语言的编译程序、操作系统、数据库、程序库程序、各种应用程序、机器维护诊断程序、与各类程序相应的维护使用手册、程序说明和框图等。我们可以借助软件，了解和掌握计算机的使用，扩大计算机的功能，提高计算机的效率。

§1-2 语言、程序与程序设计

首先要问，什么是语言呢？语言专家(Webster)曾把语言定义成：“为相当大的团体的人所晓得并使用的字以及组合这些字的方法的统一体。”还有人把语言说成是人与人之间交流思想的工具。这些定义在社会科学中也可能说的过去，然而，从自然科学领域去研究语言学，这类定义有不定因素。如相当大的团体是多大呢？交流思想的工具有多少种，哪一种是语言呢？自然科学最常用的是从数学角度准确描述某一领域。若是用数学理论去分析研究语言，则语言可定义为：

语言是一个字符表(集)上的句子的任何集合。

字符表是字(母)、数字、符号的任何有穷集合。

句子是由字(母)、数字、符号组成的有限长度的集合。

严格讲，计算机所使用的各类语言，无论是汇编语言、机器语言、解释语言，还是编译语言都是由一个有限的字符集，并由字符集构成语句与数据，按一定语法规则组成的任何集合。

一提到计算机上所使用的语言，说法多种多样。如将ALGOL语言称之为面向数学语言，在另一种场合又将ALGOL、FORTRAN、COBOL语言叫做面向过程语言或编译语言，有时把FORTRAN、ALGOL、BASIC语言称为算法语言或高级语言。其说法不一的原因有两个：其

一是语言是在发展过程中逐步形成的；其二是以语言针对性产生不同的定义。

从计算机使用的语言发展进程看，可将计算机的程序设计语言分为两大类：低级语言和高级语言。

一、低级语言

低级语言中包括机器语言或称计算机语言和汇编语言。

机器语言(machine language)

计算机是按二进制原理设计的，故计算机只能识别二进制数。由二进制码0, 1的数字代码组成一套指令去控制计算机完成各种操作。这种二进制的指令是面向计算机的，因此得名机器指令。它是计算机最原始指令，这些机器指令的集合称为机器语言。

一条机器指令用来控制计算机进行一种操作。如一条运算指令告诉计算机应当进行什么运算，哪些数据参加运算，参加运算的数据存在什么地方(到哪里去取数)，计算结果应送到什么地方去等等。对于不同的计算机有不同的指令种类和指令结构。一台计算机所拥有的指令种类及指令结构称为这台计算机的指令系统。计算机问世的最初十年(1946~1955)人们就是用这种与机器紧密相关的机器语言编写程序的。人们为了学会编程就必须掌握和记住所用的指令系统。编写程序时，无论是内存分配，指令执行顺序，输入/输出组织，都得由程序员一条一条指令进行精心安排。因为机器指令是数字化的，程序的直观性差，编写繁琐枯燥，而且很难记忆，容易出错，检查和排除错误也比较困难。又因每台计算机的指令系统各不相同，编写出来的程序互不通用，只能限于同种机器上运行，给程序移植带来不易克服的障碍。

汇编语言(assembly language)

为了克服机器语言读写困难，人们创造了汇编语言。在汇编语言中，采用了一些助记符号代替原机器指令码。用符号名代替各类地址，包括存放各种量的单元的地址和转移指令中所用的地址。因汇编语言符号代码指令仍然是与特定的计算机或某一类系列机的机器指令一一对应，仍属于一种面向机器的语言，在读写指令时比机器指令有了一定的改进。但编写程序繁琐、枯燥、工作量太大，易出错、不易修改、没有通用性等，仍和机器语言一样。因此给计算机的推广使用造成很大障碍。用汇编语言编写的源程序上机执行时，计算机先将源程序通过已存在计算机中的汇编程序译成机器指令，程序才能运行。

二、高级语言

在使用机器的实践中，人们很快认识到用低级语言编制程序和检查错误的繁重劳动可以在很大程度上交给计算机去做。就是在这样一个背景下，于50年代(1957)诞生了第一个高级语言——FORTRAN语言。

自高级语言出现之后，尤如雨后春笋，发展极快。世界上现在已用的高级语言有几百种之多，常用的有以下几种：FORTRAN、ALGOL、COBOL、BASIC、PL/1、LISP、PASCAL、ADA等。

高级语言的共同特点是：脱离特定的机器，是一种类似于自然语言和数字描述语言的程

序设计语言。在用高级语言设计程序时，不再是一条条指令排序，而是用各种各样的语句，每种语句的功能隐含一串指令。用高级语言的语句描述计算机处理过程或算法，所以对高级语言有面向过程语言和面向算法语言的叫法。

三、BASIC语言

BASIC 语言是高级语言的一种。BASIC 是 Beginner's All-purpose Symbolic Instruction Code的缩写。

BASIC语言的关键字，其含意与自然语言的意义接近，所用的运算符和运算式子和我们日常用的数学式子也相差不多。例如，用BASIC语言，想要得到 $3 \times 8 \sin(\pi/2)$ 的计算结果，只需输入语句PRINT 3 * 8 * SIN(PI/2)，计算机即可计算出结果，并打印出来。

BASIC语言是1964年由美国Dartmouth College的V.G. Kemeng和Thomas F. Kurty 两位教授以FORTRAN为蓝本创造的，是FORTRAN的一个子集。1964年首先在General Electric 225计算机上实现，1965年发表第一个版本。

1978年，基本BASIC建立了美国国家标准。1980年基本BASIC建立了国际标准。由于BASIC语言简单易学，调试方便，功能较全，因此，它不仅可以用于数值计算，而且还具有一定的数据处理能力。特别是BASIC—PLUS增加了字符处理，自选打印格式的功能，使它的应用范围得以扩大。因此，BASIC语言成为应用广泛的世界通用语言，它在计算机语言的教育和普及等方面都起到了极大作用。所以被人誉为六十年代的“十大功臣”之一。

随着计算机的广泛应用，BASIC语言也在不断发展，功能不断加强，目前各种机器上通常都配有BASIC语言，特别是在微型机上，它已成了“打不败的对手”。

四、高级语言执行过程

尽管我们可以用高级语言编写程序，但是计算机只“懂得”机器语言程序。换言之，计算机不能直接接受和执行用高级语言编写的程序，中间得经过“翻译”将高级语言程序译成机器语言程序。通常把用高级语言编写的程序称为源程序，翻译成能被机器执行的程序称为目标程序或称目的程序。承担翻译工作的程序因高级语言结构不同而异。象FORTRAN、COBOL、ALGOL、PASCAL等为编译结构，承担翻译工作的程序为编译程序。大多数BASIC语言为解释结构，承担翻译工作的为解释程序。与每一种高级语言相对应的翻译程序是事先已存放在计算机里的机器代码程序。当用某一种高级语言时，调用相应的翻译程序担任翻译工作。

解释程序的翻译，象自然语言的口译者。口译者每遇到一句话便翻译一句，使听者接受；解释程序每遇到BASIC源程序的一个语句，就将它翻译成机器语言并让机器执行，在程序执行完后输出处理结果。其过程如图1-4所示。

解释程序对源程序中重复执行的语句(例如循环体中的语句)需要重复地解释执行，浪费时间，因而效率低。但解释执行方式便于实现人机对话，容易修改，这是编译型程序所不及的。

编译程序象自然语言的笔译者一样。笔译者将一篇外文资料译成中文让人阅读，一次译完后可被阅读任意次；编译程序将一份源程序从头至尾翻译成机器语言，让机器接受，然后执行之，并允许重复执行任意次，将处理结果输出出来。其过程如图1-5所示。