



新编 BASIC 结构化程序设计 教程

汪明霓 主编

上海交通大学出版社

新编 BASIC 结构化程序设计教程

汪明霓 主编

上海交通大学出版社

内 容 提 要

本书以 IBM-PC 微机为标准,同时兼顾 APPLE-Ⅱ 微机,介绍 BASIC 语言程序设计的基本方法。全书编排合理,系统性强,突出基本算法、数据结构、编程技巧及实际应用,程序结构清晰,简明易懂。最后简介了新一代 BASIC 语言——True BASIC。

本书适合大专院校、中等专业学校以及职业学校作为计算机基础教材,也可供从事计算机工作的工程技术人员阅读参考。

(沪)新登字 205 号

新编 BASIC 结构程序设计教程

出版:上海交通大学出版社

(上海市华山路 1954 号·200030)

发行:新华书店上海发行所

印刷:常熟市印刷二厂

开本:787×1092(毫米)1/16

印张:14

字数:339000

版次:1990年8月 第1版

印次:1993年2月 第3次

印数:14501-20700

教目:93-3721

ISBN 7-313-00705-1/TP·39

定价:6.80 元

前 言

全国高等院校计算机基础教育研究会于1989年7月底在山西忻州召开了“BASIC发展与前景研讨会”。会议指出，BASIC语言在我国计算机教育普及当中作出了巨大贡献，尽管传统BASIC存在着版本繁杂、“方言”众多、程序结构性差、运行速度较慢、语言特性过多依赖硬件等缺点，但受我国目前国情，特别是微型机装备情况的制约，BASIC语言还会继续存在下去。而且BASIC语言自身也在不断发展，目前已推出了充分结构化的，完全可以同其他高级语言，诸如PASCAL、FORTRAN等相媲美的True BASIC，Turbo BASIC，Quick BASIC等新版本。在今后较长时间内，BASIC语言还会在我国大专院校和中等学校中作为计算机入门课程。正确的态度应当是不断改进BASIC教学，不断修订BASIC教材，大力推广True BASIC等新版本。本书就是根据上述的会议精神，为适应目前BASIC语言教学需要，而在较短时间内编写出来的。

本书内容丰富，编排系统性强，叙述前后连贯。全书共分为十六章。第一章和第二章分别叙述微型机的基础知识（突出软件）和BASIC语言的基础知识。第三、四两章集中介绍了程序基本结构、条件分支和循环，让读者尽早接触程序设计实例，并进行实际的编程训练。第五章介绍了数组这种BASIC语言中最基本也最重要的数据结构，从而为程序设计方法和技巧的讲述铺平了道路。第六章主要介绍了函数。由于BASIC语言的输出手段比较丰富，我们专门为之设置了一章，即第七章。作图的内容与输出格式紧密相关，故安排在第八章。第九章详细介绍了字符串这种数据类型，包括汉字的使用。从第六章到第九章，我们在介绍基本内容的同时，对程序基本结构的设计进行了反复的深化练习。在这个基础上，第十章和第十一章对程序结构进行了进一步的探讨，即子程序和程序结构化，进而介绍了大型复杂程序的设计方法。从第十二章开始，我们进入到一个较高层次的知识领域，即文件。其中第十二章介绍了上机操作和程序调试，包括程序文件的一些简单操作。这一章的内容，应该适当穿插在以前各章中进行讲解，以指导上机实践。第十三章介绍了顺序文件，其中特别提出了将程序文件作为顺序数据文件处理，从而为顺序文件和字符串处理开辟了一个崭新的应用领域。第十四章介绍了随机文件。第十五章介绍了程序的优化和完善，这一章内容同样可分散到前面的各有关章节中去介绍。最后一章，即第十六章，简单介绍了True BASIC，让读者对这种崭新的BASIC有一个初步的了解，为在不远的将来用True BASIC全面取代传统BASIC打下一定的基础。

本书在编排上有以下特点：

一、在介绍了程序基本结构之后，立即提出数据结构的概念，着重介绍了数组这种最基本的数据结构及其在各个领域中的应用。

二、注重基本算法、程序设计技巧和实际应用。全书不是单调繁琐地罗列各种语句的语法规定和使用细节，而是以程序设计为主线，在介绍设计方法中讲解语句的基本功能。

三、不再区分基本BASIC和扩展BASIC。尽量采用结构性强的语句（IF/THEN/ELSE等），避免使用过多的GOTO语句，力求使程序结构清晰，模块分明，简练易读。

四、例题丰富，取材新颖，而且各章例题前后呼应；习题的形式多，数量也较多，给教

师教学和学生练习以选择的余地。

本书以 IBM-PC 微机及其兼容机为标准,同时兼顾 APPLE-Ⅱ 微机。拥有 APPLE-Ⅱ 或其兼容机的读者,请使用 CP/M 操作系统支持下的 MBASIC 或 GBASIC,其中 MBASIC 不带作图语句,GBASIC 拥有 MBASIC 的全部功能,而且带作图语句,但其解释程序占用较大内存,读者可根据需要自行选用。

顺便指出,CP/M 系统支持下的 MBASIC 或 GBASIC 比 APPLESOFT BASIC 功能更强,使用更加方便,而且它与 IBM-PC BASIC 基本相同(这正是本书以 IBM-PC 微机为主而且可以兼顾 APPLE-Ⅱ 微机的缘由),从发展提高的角度来说,应提倡在 APPLE-Ⅱ 微机上广泛使用 MBASIC 或 GBASIC。

本书适合大专院校、中等专业学校以及职业学校作为计算机基础教材,也可供从事计算机工作的工程技术人员阅读参考。

本书在作为教材时,可以视不同专业、不同层次和学员的不同水平,安排 40 到 60 课时的课堂教学时间,并对教材内容作适当取舍。在整个教学过程中,要重视上机实践环节。有条件的学校,应当给学生相当于授课学时 1/2 以上的上机时间。

本书由汪明霓主编。周长坤、周秋平、姜之骏、钱中一、汤明望、薛昌存担任副主编。编委(按姓氏笔划排列)有:马新民、王建一、王树亮、左纪武、宋小美、张璩、张友良、张维勇、陈宝贤、吴秣陵、周树杰、周凯歌、高清泉、崔来堂、梅雅君、蒋外文等。

在本书编写过程中,参考了王国体、黄新蓉、何明等人的有关著作,在此一并表示感谢。书中不免存在各种欠缺和错误,恳请广大读者批评指正。

编 者

1990 年 5 月

目 录

第一章 绪论	(1)	第二节 数组的应用	(71)
第一节 二进制数	(1)	第三节 数据处理技术	(74)
第二节 微型机的结构与 工作原理	(2)	第四节 矩阵	(82)
第三节 微型机的软件	(7)	练习题	(84)
第四节 应用计算机解题 的一般过程	(11)	第六章 函数	(87)
练习题	(12)	第一节 三角函数和 反三角函数	(87)
第二章 BASIC 语言基础	(13)	第二节 算术函数	(88)
第一节 字符	(13)	第三节 随机函数	(90)
第二节 数据	(14)	第四节 自定义函数	(96)
第三节 表达式	(16)	练习题	(98)
第四节 语句和程序	(18)	第七章 输出格式	(100)
第五节 程序的直接形式 和间接形式	(18)	第一节 TAB 函数	(100)
练习题	(19)	第二节 行式打印法和 行式作图法	(102)
第三章 简单程序	(20)	第三节 全屏幕打印	(104)
第一节 变量赋值	(20)	第四节 PRINT USING 语句	(106)
第二节 显示和打印	(22)	练习题	(108)
第三节 数据的键盘输入	(24)	第八章 绘图	(111)
第四节 跳转	(27)	第一节 屏幕显示状态和高 分辨率作图的原理	(111)
第五节 置数与读数	(28)	第二节 作图语句	(112)
第六节 程序的暂停、结 束与注释	(32)	第三节 作图技巧	(116)
练习题	(34)	第四节 在 APPLE- I 机 上作图	(120)
第四章 程序控制	(36)	练习题	(122)
第一节 程序流程图	(36)	第九章 字符串	(125)
第二节 数控循环	(38)	第一节 字符和 ASCII 码	(125)
第三节 条件分支	(44)	第二节 字符串数据	(126)
第四节 条件控制的循环	(51)	第三节 字符串函数	(129)
第五节 循环的嵌套	(56)	第四节 汉字	(133)
练习题	(61)	练习题	(135)
第五章 数组	(66)	第十章 子程序	(137)
第一节 数组的概念和操作	(66)		

第一节	子程序的设计规则·····	(137)			
第二节	子程序的嵌套·····	(140)			
第三节	子程序的应用·····	(142)			
	练习题·····	(144)			
第十一章	程序结构化·····	(146)			
第一节	程序模块化与 菜单技术·····	(146)			
第二节	逐步求精的程序 设计方法·····	(149)			
	练习题·····	(153)			
第十二章	上机操作与程序调试·····	(154)			
第一节	磁盘与文件·····	(154)			
第二节	IBM-PC 机的 上机操作·····	(155)			
第三节	APPLE-Ⅰ 机的 上机操作·····	(160)			
第四节	汉字输入方法·····	(164)			
第五节	程序调试·····	(166)			
第十三章	顺序文件·····	(170)			
第一节	顺序数据文件·····	(170)			
第二节	源程序文件的操作·····	(176)			
第三节	把源程序文件当作 数据文件处理·····	(178)			
	练习题·····	(181)			
第十四章	随机文件·····	(182)			
第一节	随机文件的操作·····	(182)			
第二节	随机文件操作中 的函数·····	(186)			
	练习题·····	(189)			
第十五章	程序的优化与完善·····	(190)			
第一节	提高效率的措施·····	(190)			
第二节	节省内存的措施·····	(193)			
第三节	程序的完善·····	(194)			
第十六章	True BASIC 简介·····	(197)			
第一节	True BASIC 基础·····	(197)			
第二节	程序控制·····	(202)			
第三节	过程结构·····	(205)			
附 录					
	一、ASCII 码表·····	(212)			
	二、保留字表·····	(213)			
	三、操作系统状态下 的键操作·····	(214)			
	四、常用操作系统命令·····	(215)			
	五、错误信息表·····	(217)			

第一章 绪 论

从世界上第一台电子计算机 ENIAC 于 1946 年问世以来, 计算机技术取得了突飞猛进的进步, 电子计算机已经经历了四代。

第一代, 1947~1957 年, 电子管计算机。

第二代, 1958~1964 年, 晶体管计算机。

第三代, 1965~1971 年, 中小规模集成电路计算机。

第四代, 1972~今, 大规模超大规模集成电路计算机。

据统计, 每 5~8 年, 计算机的运算速度就提高 10 倍, 而体积却缩小 10 倍, 成本也降低 10 倍。

随着电路集成化的提高, 近年来出现了微型电子计算机, 简称微型机。由于它具有体积小、重量轻、功耗小、成本低、灵活方便、可靠性强等优点, 因而很快地渗透到国民经济、社会生活各个部门和各个领域, 应用范围极为广泛。许多产品由于采用微型机控制, 而得以升级换代, 功能增加而成本降低。许多企事业单位由于采用微型机进行管理, 工作效率和经济效益大为提高。微型机还在计算机辅助教学 (CAI)、辅助设计 (CAD)、人工智能 (AI) 等许多领域中成了人们的得力助手。微型机的大力推广和应用正在改变我们的社会生产和生活方式。随着计算机技术的发展, 这种改变将会越来越明显地表现出来。

本书中, 我们将以目前最广泛流行的微型机 IBM-PC 为对象 (适当照顾 APPLE- I 型微机), 阐述微型机的构造和工作原理, 学习程序设计方法, 并掌握上机操作要领。

第一节 二进制数

计算机采用二进制来计数和运算。什么是二进制呢?

二进制其实并不神秘。在日常生活中, 二进制处处皆是。举个最简单的例子, 两只筷子称为一双, 这就是二进制的“逢二进一”。如果我们把两双称为一把的话, 则“只一双一把”就构成了一个二进制系统。表 1-1 列出了筷子数的十进制表示和二进制表示之间的对应关系。

表 1-1

筷子数(十进制)	把位	双位	只位	筷子数(二进制)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	10
3	0	1	1	11
4	1	0	0	100
5	1	0	1	101
6	1	1	0	110
7	1	1	1	111

二进制数只 0 和 1 两个数字, 从表 1-1 可以看到: 两只筷子可以表示成一双, 即 $10_{(2)}$; 3 只筷子可以表示成 1 双 1 只, 即 $11_{(2)}$; 6 只筷子表示成 1 把 1 双, 即 $110_{(2)}$ (这里下标 $_{(2)}$ 表示二进制)。实际上, 任何十进制数都可以用二进制数来表达, 一个数的十进制表示和二进制表示之间可以进行相互转换。

十进制数转换为二进制数一般采用“除二取余法”。例如，将 47 化为二进制数。

从竖式可得 $47 = 101111_{(2)}$ 。注意除要除到商为 0 为止。将所有的余数按反序写出，即得所求的二进制数。

二进制数转换成十进制一般采用“乘幂相加法”。例如：

$$101111_{(2)} = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ = 47$$

这里，第一个数字 1 的后面有 5 位，则表示 1×2^5 ；第二个数字 0 的后面有 4 位，则表示 0×2^4 ；依此类推。将各位数字表示的乘幂相加，则得所求的十进制数。

为了便于换算，现将常用的 2 的各次幂列表如下（表 1-2）。

二进制数和十进制数一样，也可以进行加、减、乘、除各种运算，而且运算规则十分简单。以加法为例，它的规则只有三条：

$$0+0=0, \quad 0+1=1, \quad 1+1=10$$

最后一条即是“逢二进一”。

例如： $1001_{(2)} + 11100_{(2)} = 100101_{(2)}$ ，写成竖式为：

$$\begin{array}{r} 1001 \\ +) 11100 \\ \hline 100101 \end{array}$$

即十进制的 $9 + 28 = 37$ 。

二进制数乘法规则也只有三条：

$$0 \times 0 = 0, \quad 0 \times 1 = 0, \quad 1 \times 1 = 1$$

例如， $1001_{(2)} \times 101_{(2)} = 101101_{(2)}$ ，写成竖式为：

$$\begin{array}{r} 1001 \\ \times) 101 \\ \hline 1001 \\ 0000 \\ 1001 \\ \hline 101101 \end{array}$$

即十进制的 $9 \times 5 = 45$ 。

在计算机中，为什么要采用二进制呢？这是因为计算机是用电信号来进行工作。二进制只有 0 和 1 两个数，如用高电平表示 1，低电平表示 0，则很自然、方便和准确；而如用十进制，要用不同的电平表示 0、1、2、……、9 等十个数字，就显得十分困难。计算机中采用二进制，使得电路结构大为简化，运算速度和可靠性大为提高。

2	47	
2	23	……1 ↑
2	11	……1
2	5	……1
2	2	……1
2	1	……0
0	……1	

表 1-2

N	2 ^N
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
10	1024
16	65536

第二节 微型机的结构与工作原理

要了解微型机的结构和工作原理，可以从计算器入手。

图 1-1 是计算器的外形图。它的下半部是键盘，键面上刻有数字和各种符号，上半部是显示屏，它显示键入的数据和运算结果。如果打开后盖，可以看到一个集成块，别看它只有指甲那么大，内部却有成千上万个晶体管，它是计算器的心脏，叫做运算器。

计算器的功能结构，可以用图 1-2 表示出来。键盘称为输入设备，而显示屏称为输出设备。计算器虽然运算得很快，但是它的功能还是非常有限的。举例来说，如果在火车站行李房

工作,给旅客称行李、算运费。按规定,每人可以免费携带行李 20kg,如果超过 20kg,超过的部分按 0.15 元/kg 计算,则计算过程如下图(图 1-3) 所示。或者写成:

第一步: 键入重量 P;

第二步: 判断 P 是否大于 20,如果是,则转第四步, 否则, 到第三步;

第三步: 运费 F 为 0, 跳到第五步;

第四步: 运费 $F = (P - 20) \times 0.15$;

第五步: 显示运费 F, 跳回第一步, 接待下一个旅客。

这里, 我们把解决问题的过程分成了一系列的步骤,这些步骤组成了一个程序,我们将在下一节中详细地讨论程序。

象图 1-3 这种表示解题步骤及其执行顺序的图称为程序流程图, 又叫程序框图。

考察一下用计算器算运费的过程,除了第四步以外,其它步骤它都是无能为力的,整个程序都必须记在人的脑子里,由人进行判断和指挥操作。如果程序复杂,长时间工作下来,人脑的疲劳是可想而知的。

显然,人们希望能把程序本身交给机器,让机器根据程序去自动进行判断和运算,从而把人们从繁琐重复的脑力劳动中解放出来。

这就得求助于电子计算机。计算机与计算器的根本区别,就在于计算机能够存储人们事先设计好的程序,然后根据程序自动地进行判断和运算。

可以想象,计算机比计算器复杂得多。

图 1-4 是微型机的结构示意图,图中中间部分称为主机,主机由中央处理器(CPU)和内部存储器(内存)两部分组成。

内存用来存放程序和数据。程序无论多么复杂,都可以分解为一些最基本的机器操作,例如:加、减、比较、移位、存数、取数等等,这些最基本的机器操作,称为机器指令,简称指令。

指令也用二进制数进行编码,它们也和数据一样存放在内存里,所以,在内存中程序和数据的形式是统一的。

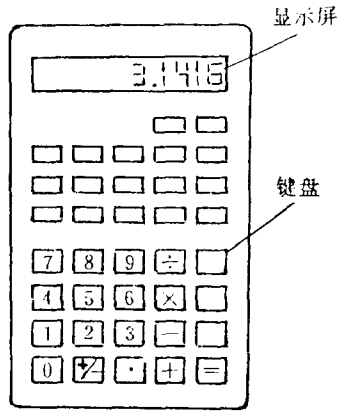


图 1-1

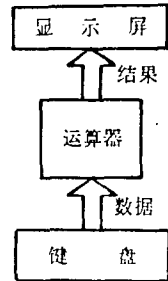


图 1-2

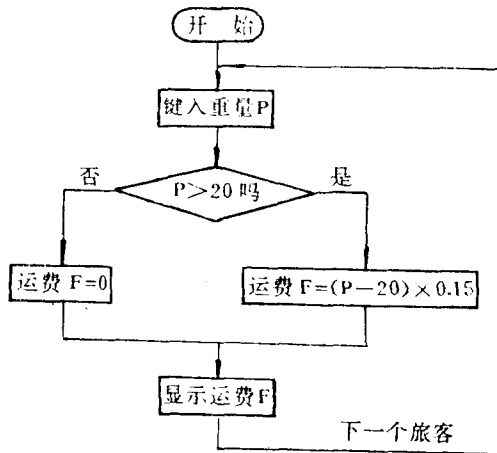


图 1-3

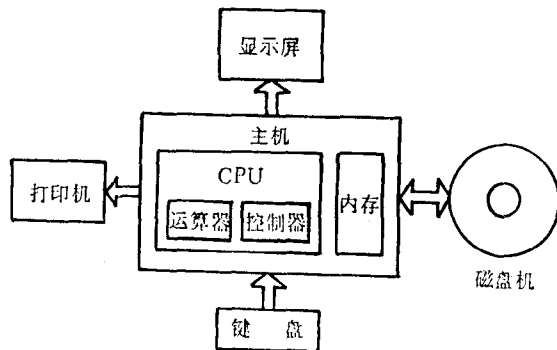


图 1-4

内存大小是衡量计算机性能的一个很重要的指标。内存越大，能存放的程序就越大，能同时处理的数据就越多。内存的大小一般用字节(Byte)来表示，一个字节就是8位二进制码。微型机的内存一般为48K或64K字节(1K=2¹⁰=1024)，高档微型机，内存可达256K以上。

内存的工作框图可见图1-5，内存的每个存储单元(字节)都编有地址，可以由地址信号选中任一个存储单元，读/写控制线控制读/写的时机，通过数据线将数据写入选中的存储单元，或者从选中的单元中将数据读出送上数据线。

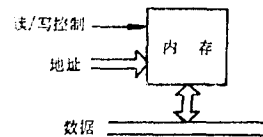


图1-5

CPU由运算器和控制器所组成。这里的运算器相当于计算器中的运算器，但它除了可以进行算术运算外，还可以进行比较判断等逻辑运算。运算器的工作框图可由图1-6来表示。图中的控制信号用来控制运算的种类和运算时机，即进行何种运算和何时进行运算。

现在我们来考察控制器(图1-7)。

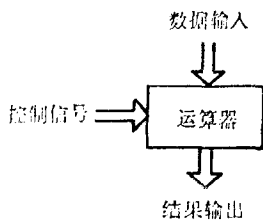


图1-6

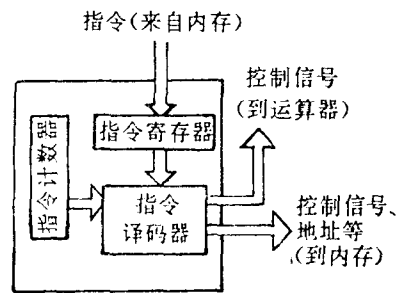


图1-7

控制器主要由以下几部分组成：指令计数器、指令寄存器、指令译码器。

指令计数器，顾名思义，具有计数的功能，即自动加1的功能，它控制程序执行的全过程。指令寄存器存放从内存中取来的指令。指令译码器具有很复杂的逻辑电路，但它的功能是十分机械的，给它一定的输入信号，则得到确定的输出信号，就象翻译电报码一样。现在假设指令译码器有8根输入线和8根输出线，如8根输入线信号是10101010(1为高，0为低)则8根输出线给出信号11110000，又比如输入为00000001，输出为11111101。也就是说，输入信号就是指令，而输出就是送往运算器和内存的控制信号。

由此可见，指令确是微型机的基本操作单元，一台微型机性能的优劣，指令丰富与否是标志之一。微型机一般有几十条到几百条不同的指令。

微型机只向人们提供了这些最基本的操作，即指令，下面的事就得由人去做了，这就是把一条条指令有机地组合起来，形成一个程序，并把它送到内存中去，让微型机执行。

下面我们来具体考察一下一个程序的执行过程。假设内存中已有一个程序，指令编号为1, 2, 3, ……。起动后，指令计数器按一定的节拍工作，它从1开始计数，第一条指令被取入指令寄存器，指令译码器立即对其进行译码，发出控制信号去指挥运算器或其他部件进行动作。然后，指令计数器计数到2，第二条指令被取入指令寄存器，由指令译码器译码发出控制信号。依次下去，直到程序执行完毕为止。由此可见，微型机当中真正自动的部分只有一个，那就是指令计数器。其它的部分，象指令译码器、内存、运算器等都是死板的、机械

的。微型机只提供了一些最基本的操作，即指令，需要人们去把这些个别的、离散的指令有机地组合起来，交给计算机去执行，离开了人们的指挥，没有人们编制的程序，微型机什么也干不了。

控制器和运算器一般做在一个集成块上，它是微型机的核心，所以我们把它称为中央处理器（CPU）。CPU和内存一起，构成了微型机的主机（图1-8）。

微型机的操作速度是很快的，每秒可以执行几十万到几百万条指令；而且因为数字构成非常简单，不是0就是1，所以很难出错。这就是微型机能算得既迅速又准确的奥秘所在。

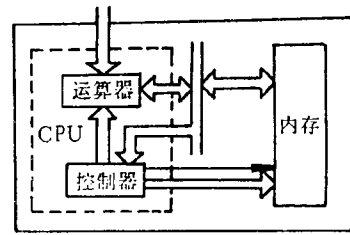


图1-8

除了主机以外，微型机还得配备一些外部设备，那么微型机需要哪些外部设备呢？

首先，我们得把指令和数据输给微型机，这通常是用键盘来输入的。这种键盘比计算器的键盘复杂，和普通英文打字机键盘差不多，上面刻有26个英文字母、0~9等十个阿拉伯数字、各种标点符号和一些功能键。这种键盘可以很方便输入英文，但不能直接输入汉字。怎样输入汉字呢？这个问题我们将在第十二章中介绍。

其次，我们要知道微型机运算的结果，就需要给微型机配上输出设备，这通常是显示器和打印机。显示器相当一个电视机的显象管，你从键盘上键入的字符和计算机的运算结果都象电视一样显示于屏幕之上，看起来十分清楚。通常整个屏幕可显示24行，每行80个字符。要把程序或运算结果打印在纸上，则需要打印机。

还有一种十分重要的外部设备，叫做外部存储器，它有很多种类，如磁带机、磁盘机等。我们知道，程序和数据是以电信号保存在内存里的。当你关机断电时，这些电信号就会消失。那么，怎么办呢？解决的办法也很简单。你听收音机时，要想把某一段音乐永久地保存下来，你可以用录音机把它录在磁带上，也就是用磁信号保存这些音符。在微型机上，我们也可以用磁带来“录音”，把程序和数据录在磁带上。这样录成的磁带，如果用录音机放出来，会发出一种类似鸟叫的声音。然而，微型机上配得最多的“录音”设备还是磁盘机和磁盘。磁盘分软磁盘和硬磁盘两种，软磁盘就象一张塑料唱片（图1-9），封在一个黑色纸套中，中央有一个大圆孔。磁盘机中有一个小马达，小马达的轴就穿过这个大圆孔带动磁盘转。磁盘的纸套上有一个长圆孔，磁盘机中的磁头就通过这个长圆孔，对磁盘进行读写（即“录音”和“放音”）。磁盘机工作时磁盘转得很快，磁头能方便地移到磁盘上的任何地方，而录音磁带只能顺序地从一头转到另一头，所以使用磁盘要比磁带方便得多，速度快得多。好一点的微型机都配磁盘机，只有低档的微型机才配磁带机，即录音机，因为它很便宜。

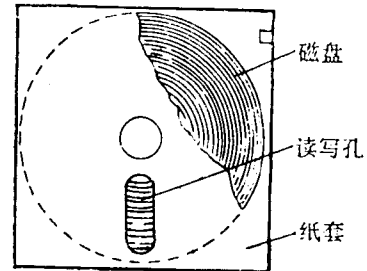


图1-9

外存的容量可以很大。一张磁盘，薄薄的，直径不过十几厘米，却可以存放几百万个字符，相当于厚厚一大本书，而且一张磁盘存满了可以另换一块磁盘。硬盘的存储量就更大了，可达几十兆。所以外存的存储量是没有限制的。相比之下，内存存储量就有限了。

外存的动作比内存起来，要慢一些，因为外存要靠机械运转。而内存就在主机当中，而且全都是电信号传输，所以速度非常快。

综合上述，内存的特点是速度快，但容量有限，断电后信息丢失；外存的特点是容量大，

断电后能长时间保存信息，但速度较慢。内存和外存，都是微型机不可缺少的组成部分，它们取长补短，相互配合，协调工作。

现在，我们回过头来再来看图 1-4。可以看到，整个微型机是一个由许多部件组合而成的系统，各个部件之间的箭头表示信息流通的方向，信息从键盘进入主机，由主机流到显示屏和打印机，主机和磁盘机之间的箭头是双向的，这表示我们既可以把信息从主机存入磁盘，也可以从磁盘中将信息读到主机中来（放在内存里）。CPU 是整个微型机的核心，所有的信息流通一般都得经过它。

微型机种类繁多，有高档的，如 IBM-PC（图 1-10），中档的，如 APPLE-Ⅰ，还有很多低档的品种，比如 PC-1500 等等。

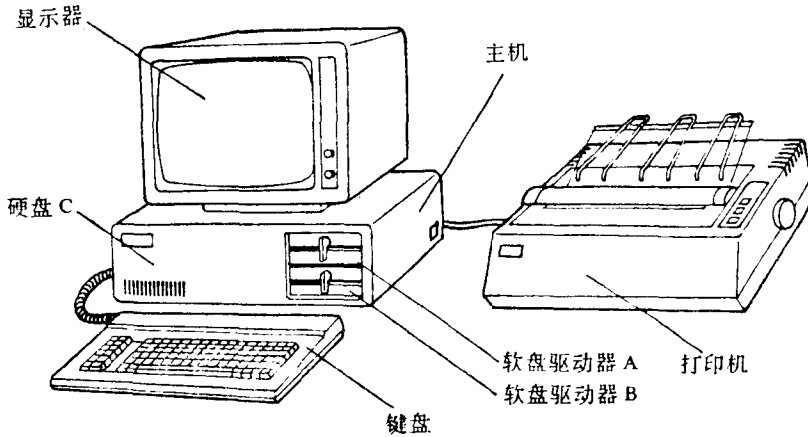
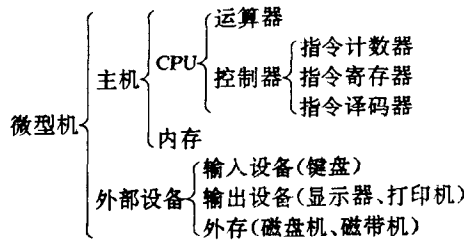


图 1-10

现在，让我们把本节内容回顾一下。

1. 微型机的结构



2. 微型机的工作原理

微型机的复杂的电子线路只提供了“指令”这样的基本操作，需要人们把指令编成程序，放入内存。然后，指令计数器自动加 1 计数（就跟时钟一样），指令依次执行，完成预定的任务。所以微型机完全是忠实地执行人的意志，离开了人，微型机什么也做不成。把微型机称为“微电脑”，其实是不确切的，充其量它只能算一个机械的、死板的“脑子”。然而，微型机比起计算器来又是一个飞跃。计算器不能存储程序，算题的步骤完全是使用者临时设计的，按一下它动一下，从这一点上说，计算器和算盘本质上没有区别。

3. 微型机的“记忆”

微型机靠内存和外存来“记忆”大量的信息，这两种存储器各有长处和短处，内存容量

小，但速度快；外存容量大，但速度慢。所以，当前需要运行的程序和需要的数据一般总是放在内存里，而暂不用的程序和数据都要及时地存放到外存当中去。

第三节 微型机的软件

一、软件的概念

硬件和软件，是两个常用的计算机术语。我们先来讲讲什么是硬件。

所谓硬件，就是看得见、摸得着的实实在在的东西。微型机的各种部件，象 CPU、内存、主机、各种外部设备等，都称为微型机的硬件。

那末，软件是什么呢？软件就是看不见、摸不着的东西，它是各种程序及资料文档的总称。它在我们大脑里形成，存在计算机内存和外存里，也可以显示在屏幕上、打印在纸上。

为了深入了解硬件和软件的概念，我们来举几个例子。

算盘，大家都很熟悉。买来一个算盘，就是有了“硬件”。但光有算盘还不行，你还必须学会珠算口诀和各种算法，才能进行计算，这些口诀和算法就是“软件”。硬件是重要的，但软件更为重要。同样一个算盘，摆在会打算盘的人和不会打算盘的人面前，效果完全不一样。不会打算盘的人只看到一个框和一串串的珠子，而会打算盘的人却可以将算盘用“活”，成为他的得力工具。

买来录音机和空白磁带，这是“硬件”；空白磁带录上了音乐或外语，这磁带就带上“软件”了。

硬件和软件的概念现在已扩展到各个领域。甚至可以说，人本身也是“硬件”和“软件”的结合。人生下来，身体发育长大，这是你的“硬件”；而在成长的过程中，你学习和掌握各种知识和技能，逐步树立起你的世界观，这就是你的“软件”。

现在我们还是回到计算机上来。我们可以这样说：硬件只提供了一些最基本的操作功能（指令），而必须用软件来把这些基本的操作功能组成一个有机的整体，这样才能解决各种复杂的问题。硬件是计算机的“物质基础”，软件是计算机的“上层建筑”；硬件是计算机的躯体，而软件才是计算机的灵魂。

软件是脑力劳动的成果，它和硬件一样，是有价值的，可以以商品的形式出售。一块磁盘，是空白盘还是带有软件的磁盘，外表都一样，而价值要相差几到几百倍。这就跟空白录音带和原声带价格不一样一个道理。现在，随着微型机的日益广泛的应用，对软件的需求量越来越大，需要大批的人来编制软件，软件已逐渐成为一门新兴的产业。软件当然主要由软件工程人员来编制，但对于广大计算机用户来说，懂一点软件和程序的基本知识，进而会编一些短小的程序，也是很有必要的。

二、程序

在上一节中，我们已经知道了程序就是解题的步骤。完整而确切地说，所谓程序，就是为完成某一任务而设计的有限多的步骤所组成的一个有机的序列。

在日常生活中，我们到处可以碰到程序：

做早操：第一节，伸展运动；第二节，扩胸运动；……

电台播音：六点，开始曲；六点零五分，天气预报；六点十分，音乐；……

做数学题：例如 $D - (A + B) \cdot H / 2$ ，先做 $A + B$ ，再把和与 H 相乘，然后除以 2 得到一

个商，最后从 D 中减去这个商。

从这许多例子中，我们再来体会一下程序的特性。

第一，目的性。程序必须有一个明确的目的，即为了解决什么问题。

第二，分步性。程序是分为许多步骤的。不可能一步就解决问题。

第三，有限性。解决问题的步骤不可能是无穷多的。如果有无穷多个步骤，在计算机上就实现不了。

第四，有序性。这是最重要的一条，解题步骤不能杂乱无章地堆积在一起，而是要按一定的顺序排列。但是程序的这种有序也不能理解为仅仅是简单的顺序：一、二、三、四、五、……，它还可以有一个很重要的形式——分支，即根据条件的不同，走不同的路，用两套不同的步骤来解决问题。例如，电台在每周一、三、五的七点十五分播放曲艺，而每周二、四、六、日的七点十五分却播音乐；等等。

正是因为程序可以有分支，才使程序能有千变万化，去灵活地解决各种复杂的问题。

下面，我们来具体考察几个用计算机解题的程序。

例 1-1 计算正方形的面积。

第一步：输入正方形的边长 A ；

第二步：用公式 $S=A^2$ 计算面积 S ；

第三步：在屏幕上显示面积 S ；

第四步：结束。

例 1-2 计算行李运费。这个例子在上一节中已经提到了，有必要再提出来分析一下。

第一步：键入重量 P ；

第二步：判断 P 是否大于 20，如果是，则转第四步，否则，到第三步；

第三步：运费 F 为 0，跳到第五步；

第四步：运费 $F=(P-20) \times 0.15$ ；

第五步：显示运费 F ，跳回第一步。

在例 1-1 中，几个步骤按一、二、三、四的简单顺序执行；而在例 1-2 中，第二步有个条件分支，即根据重量 P 是否大于 20，决定走哪条路；如果 P 大于 20，则走一、二、四、五、一；如果不大于 20，则走一、二、三、五、一（参见图 1-3）。在复杂的程序中，这样的条件分支结构是很多的。

例 1-3 计算 3 个正方形的面积。

第一步：从第二步做到第四步，重复做 3 遍；

第二步：输入正方形的边长 A ；

第三步：用公式 $S=A^2$ 计算面积 S ；

第四步：在屏幕上显示面积 S ；

第五步：结束。

这个程序的执行步骤是一、二、三、四、二、三、四、二、三、四、五。这里，出现了另一种重要的程序结构——循环，即反复做第二步到第四步这一段程序。和条件分支一样，循环也是程序中经常使用的结构。

一个程序，无论其多么长，多么复杂，都是由上述的顺序、分支和循环加上过程等四种基本结构所构成的。在以后的章节时，我们将详细地探讨怎样用这几种基本结构来设计程序。

三、语言

语言是人们交流思想、传达信息的工具。中国人使用汉语，英国人、美国人使用英语，这些语言称为自然语言。

除了自然语言之外，还有许多专门语言。例如乐谱是音乐家的语言，图纸是工程师的语言，等等。专门语言有一个重要的特点，就是语言规则十分严密，不象自然语言那样随便，即使有一点错误也不影响意思的理解。

我们和计算机打交道，也得使用语言。但是一般的计算机还不够聪明，不能直接听懂我们的话，不理解我们的自然语言。为此，人们特地设计了一些专门语言，即计算机语言，用来给计算机编程序，和计算机进行对话。

最简单的计算机语言是由 0 和 1 组成的机器码语言。例如：如果 3 是 00000011，5 是 00000101，加法是 00101001，那么 $3+5$ 就可以写成 000000110010101100000101。这种语言计算机最欢迎，1 代表高电平，0 代表低电平，它可以立即照办。但人看起来实在太吃力了，我们把这种机器码语言叫做低级语言。

不同类型的计算机，它们的机器码是不同的，所以低级语言的通用性很差。

为了让人使用方便，人们又设计出许多种接近自然语言的计算机语言。本书所要介绍的 BASIC 语言，就是其中的一种。这些语言，容易阅读，也容易学习和掌握，而且通用性很强，我们称它们为高级语言。现在，我们把例 1-1 和 1-2 用 BASIC 语言写出来。

例 1-1 计算正方形面积。

```
10 INPUT A
20 S=A^2
30 PRINT S
40 END
```

例 1-2 计算行李运费。

```
10 INPUT P
20 IF P>20 THEN 40
30 F=0:GOTO 50
40 F=(P-20)*0.15
50 PRINT F:GOTO 10
```

稍微懂一点英语的人马上就能看懂这两段程序。注意例 1-1 里 20 句中间的符号 (^) 是乘方，例 1-2 里 40 句中的符号 (*) 是乘号。

高级语言种类很多，常见的除 BASIC 外，还有 FORTRAN、ALGOL、PASCAL、COBOL 等等。高级语言的通用性很强，同一种语言，既可以在这种机子上用，也可以在那种机子上用。即使有区别，也是很微小的，所以它们的应用越来越广泛。

计算机并不能直接懂得高级语言，因此，需要有一个翻译，来把高级语言翻译成机器码；反过来，机器运算的结果也必须翻译成人们能够认识的形式。这种翻译工作是由计算机自己完成的，这需要很复杂的翻译程序。程序翻译有两种形式，一种是翻译一句机器就执行一句，这种翻译形式叫做解释，象一般的 BASIC 语言就是解释执行的；另一种是整个程序全部翻译成目的码即机器码，然后一次执行，这种翻译形式叫编译，FORTRAN、PASCAL 等语言采用的是这种编译形式。

我们可以用图来表示“人—高级语言—翻译程序—计算机”这样的层次分明的相互关系

(图 1-11)。人们看到的只是用高级语言写的应用程序，再里层的过程人们是看不见的。

随着计算机科学的发展，计算机变得越来越聪明，正在向着智能型的方向发展，向理解自然语言的目标挺进。

四、操作系统

要弄清什么是操作系统，可从简单的监控程序讲起。

让我们回忆一下微型机的总体结构（请参见图 1-4）。现在假设我们要送一个字符到内存某一单元去，那么，CPU 有下面这些事情要做：

- (1) CPU 必须反复地（比如每隔 0.01 秒）询问键盘，有没有字符键入；
- (2) 判别键入的是什么字符；
- (3) 将字符取入 CPU；
- (4) 将字符存入指定的内存单元。

同样，如要从内存中取一个字符送到显示屏上去显示，则 CPU 有下面这些事情要做：

- (1) 从内存的指定单元中将字符取入 CPU；
- (2) 确定这个字符应该显示在屏幕上的什么位置（这通常依赖于上一个字符的位置）；
- (3) 将字符送到显示器去显示。

通过这两个例子，我们可以看出，即使是一个很小的操作，微型机都要把它分为许多步骤去做，也就是说，都必须由一小段程序来完成。

很明显，为了使整个微型机的主机能运转起来，能完成象接收字符、发送字符、存放数据、读取数据等等许多基本功能，必须有一个控制指挥程序，这个控制程序叫做监控程序。它以固化的形式存在内存里，停电以后不会消失。一旦开机，CPU 就在这个监控程序的指挥下开始工作。

懂得了监控程序的概念，我们再来回顾一下内存。内存可以分成两部分（图 1-12），一部分是固化形式的，只能到里面去读数据，我们称之为只读存储器（ROM）；另一部分就是以前介绍的，既可以从里面读数据，也可以往里面存数据，即可以反复读写，断电后内容会消失，我们称之为随机存储器（RAM）。

ROM 存放监控程序，出厂时就预先固化好了，而 RAM 存放其它程序，包括用户程序和数据。

下面我们回到“操作系统”。操作系统是管理协调计算机各个部分（包括各种外部设备）进行工作的一套程序。从某种意义上讲，它是监控程序的扩充。操作系统分成许多档次，有单机单用户操作系统，单机多用户操作系统，直到网络型的多机操作系统。下面只简要地叙述单机单用户操作系统。

单机单用户操作系统，主要是对磁盘进行管理，所以又叫磁盘操作系统。我们知道，编制的或买来的程序都是存在磁盘上的，这些存在磁盘上的程序称为文件，一批数据存在磁盘上也成为文件。盘上的文件很多，因此就产生了文件目录登记、存放的地址安排、文件的保护、文件的更改、文件的删除、删除后存储空间的回收、文件调入内存或从内存将文件写入磁盘等等许多有关文件的操作。这些操作的细节如果都要由人去管理，那是极其繁琐的，而且容易出错。所以磁盘文件必须由计算机自己来管理，这个管理程序就称为磁盘操作系统

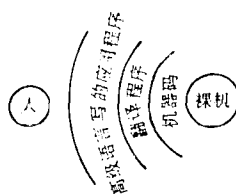


图 1-11

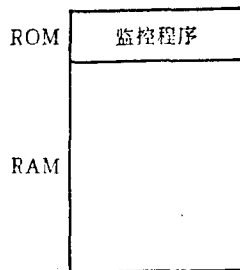


图 1-12