

IBM PC系列微机速成入门



白似雪 李 力 段隆振 曹吉青 编著

IBM PC

XI LIE WEIJI

SU CHENG RUMEN

江西高校出版社

IBM PC 系列微机速成入门

白似雪 李力 段隆振 曹吉青 编著

江西高校出版社

(赣)新登字第007号

书 名: IBM PC系列微机速成入门
作 者: 白似雪 李力 段隆振 曹吉青 编著
出 版: 江西高校出版社(南昌市洪都北大道96号)
发 行: 各地新华书店
经 销: 各地新华书店
印 刷: 丰城市印制业股份有限公司
开 本: 787×1092 1/16
印 张: 13.875
字 数: 350千
印 数: 1—5150
版 次: 1997年1月第1版第1次印刷
定 价: 12.00元

ISBN 7—81033—415—8

TP · 4

邮政编码: 330046 电话: 8512093

(江西高校版图书凡属印刷, 装订错误, 请随时向承印厂调换)

前 言

微型计算机的应用已深入到国民经济的各行各业乃至平常人家。了解微机、使用微机几乎成为现代人必备的劳动技能之一。为此,我们编写此书,旨在使那些渴望了解和使用微机的人们极快地领略微机及其应用的奥秘;为他(她)们进一步的学习提供导航和打下必要的基础。

全书语言通俗易懂,内容丰富,取材新颖,具有很强的实用性,既可作为初学者的入门教程,也可作为有一定使用微机经验的读者的案头资料。

参加本书编著的有:白似雪、李力、段隆振、曹吉青(以姓氏笔划为序)同志,该书是由以上同志集体讨论定稿。白似雪编写了第三章和第七章的第二节;李力编写了第一、二章和第七章的第一节;段隆振编写了第四、五章;曹吉青编写了第六章。辜晓波同志审阅了全书并提出了许多有益的意见,在此表示感谢。

由于编写时间仓促,书中错误及不妥之处敬请读者不吝批评指正。

-编者

1994,10

目 录

第一章 微机概论	(1)
§ 1.1 微机发展简史	(1)
§ 1.2 微机硬件的结构	(2)
§ 1.3 微机软件	(6)
§ 1.4 典型微机系统	(9)
第二章 微机操作系统	(14)
§ 2.1 操作系统概念	(14)
§ 2.2 PC-DOS 概述	(16)
§ 2.3 常用 PC-DOS 命令	(22)
2.3.1 列目录、查看文件名	(22)
2.3.2 清洁屏幕	(25)
2.3.3 建立子目录	(25)
2.3.4 改变目录	(27)
2.3.5 撤消目录	(28)
2.3.6 复制文件	(28)
2.3.7 设置日期和时间	(30)
2.3.8 删除文件	(31)
2.3.9 显示文件内容	(31)
2.3.10 更换文件名	(32)
2.3.11 格式化磁盘	(32)
2.3.12 磁盘复制	(35)
2.3.13 显示版本号	(36)
2.3.14 设置提示符	(36)
2.3.15 设磁盘卷标	(37)
2.3.16 显示卷标	(37)
2.3.17 设置文件属性	(38)
2.3.18 复制文件	(38)
2.3.19 设置路径	(41)
§ 2.4 高级 DOS 命令	(42)
2.4.1 分屏显示	(42)
2.4.2 按指定顺序排列内容	(43)
2.4.3 在文本文件中查找指定的字符串	(44)
2.4.4 显示磁盘目录结构	(45)
2.4.5 检查磁盘	(45)
2.4.6 设置数据文件搜索路径	(47)

2.4.7	复制 DOS 程序	(48)
2.4.8	备份磁盘内容	(48)
2.4.9	恢复备份文件	(49)
2.4.10	硬盘分区	(50)
2.4.11	批命令	(54)
第三章 汉字系统		(60)
§ 3.1	汉字系统概述	(60)
§ 3.2	汉字代码	(60)
3.2.1	汉字输入码	(61)
3.2.2	汉字内部码	(61)
3.2.3	汉字地址码	(61)
3.2.4	汉字交换码	(62)
3.2.5	汉字字形码	(63)
§ 3.3	常见汉字系统介绍	(63)
3.3.1	CC-DOS 汉字系统	(63)
3.3.2	WM-DOS 汉字系统	(81)
3.3.3	SUPER-CCDOS 汉字系统	(84)
3.3.4	天汇汉字系统	(85)
§ 3.4	五笔字型汉字输入方法	(86)
3.4.1	五笔字型概述	(86)
3.4.2	汉字的基本笔划	(86)
3.4.3	基本字根、字根键盘	(87)
3.4.4	键名汉字输入	(88)
3.4.5	成字字根及笔划的输入	(88)
3.4.6	汉字的结构	(89)
3.4.7	汉字的字型	(89)
3.4.8	汉字的识别码	(89)
3.4.9	汉字的拆分原则	(90)
3.4.10	万能学习键	(91)
3.4.11	简码输入	(91)
3.4.12	词语输入	(91)
3.4.13	重码与容错码	(91)
3.4.14	小结	(92)
第四章 编辑软件		(93)
§ 4.1	WPS	(93)
4.1.1	WPS 的运行环境	(93)
4.1.2	WPS 的启动和退出	(94)
4.1.3	WPS 的主菜单	(95)
4.1.4	文本编辑	(96)
4.1.5	文件操作	(98)
4.1.6	块操作	(99)

4.1.7 查找与替换	(101)
4.1.8 打印控制	(103)
4.1.9 多窗口编辑	(109)
4.1.10 排版与制表	(111)
4.1.11 其它功能	(113)
4.1.12 模拟显示与打印功能	(114)
4.1.13 文件服务功能	(117)
4.1.14 帮助功能	(118)
§ 4.2 行编辑程序 EDLIN	(118)
4.2.1 概述	(118)
4.2.2 EDLIN 常用编辑命令	(121)
第五章 计算机高级语言	(128)
§ 5.1 计算机语言概述	(128)
5.1.1 机器语言	(128)
5.1.2 汇编语言	(128)
5.1.3 高级语言	(129)
5.1.4 高级语言的执行过程	(129)
§ 5.2 BASIC 语言	(130)
5.2.1 BASIC 语言的基础知识	(131)
5.2.2 提供数据语句和打印输出语句	(136)
5.2.3 控制语句	(143)
5.2.4 数组说明语句(DIM)	(153)
5.2.5 字符串处理	(154)
5.2.6 IBM PC BASIC 常用命令及使用	(157)
§ 5.3 C 语言快速入门	(159)
5.3.1 建立第一个 C 程序	(160)
5.3.2 变量、语句和算术表达式	(162)
5.3.3 输出格式的控制	(164)
5.3.4 输入	(164)
5.3.5 条件语句	(165)
5.3.6 另一种循环语句——FOR 语句	(166)
5.3.7 数组	(167)
5.3.8 函数	(168)
5.3.9 符号常数	(169)
5.3.10 再谈命名	(170)
5.3.11 C 语言的头文件和标准库	(171)
5.3.12 TURBO C 集成开发环境简介	(172)
第六章 WINDOWS 的使用方法	(176)
§ 6.1 Windows 概述	(176)
§ 6.2 Windows 的基本操作	(176)
6.2.1 基本概念	(176)

6.2.2	进入和退出 Windows	(178)
6.2.3	窗口的使用	(178)
6.2.4	菜单及对话框的使用	(181)
6.2.5	文档操作	(183)
6.2.6	使用 Program Manager 的方法	(186)
6.2.7	File Manager 的使用方法	(191)
第七章	微机常见故障及处理	(199)
§ 7.1	硬件维护及故障处理	(199)
7.1.1	安装与检验	(199)
7.1.2	启动过程的故障及处理	(201)
7.1.3	设置与配置	(202)
7.1.4	微机硬件的日常维护	(207)
§ 7.2	软件故障	(207)
7.2.1	系统不能正确引导	(207)
7.2.2	死机	(208)
7.2.3	系统不能正常工作	(208)
§ 7.3	计算机病毒	(208)
7.3.1	计算机病毒的检测	(209)
7.3.2	计算机病毒的清除	(210)
7.3.3	计算机病毒的预防	(210)
附录一:ASCII 码表		(212)
附录二:五笔字型键盘字根总图		(213)

第一章 微机概论

§ 1.1 微机发展简史

在当今社会里,微机这个词如同收音机、电视机一样深深地印在了人们的脑海里。同时,微机给人们的工作和生活习惯也带来了巨大的变化,一些人甚至把微机同文化等同起来了,他们把不懂得使用微机的人称做为“机盲”,如同把不识字的人称做为“文盲”一样。毋庸置疑,学会和掌握微机的使用是当今社会发展的必然趋势。

也许正是微机应用领域的广泛性,才使得微机在人们的心目中产生了一种神秘感:它既然“无所不能、无所不在”,那么要掌握和使用它也一定是件不容易的事。自从1946年世界上第一台计算机诞生以来,直到七十年代中期以前,计算机做为一种功能强大的计算工具一直为少数专家们所垄断。虽然这种垄断不是专家们有意而为的,但是当时的情形确是如此。那时要掌握和使用计算机必须经过一段时间的专门训练,而且这种训练还必须具有一定的专业基础知识。因此它也就不为一般人所接受。随着社会的发展,人们对高新技术的需求日益增长,科学家们特地研制出一种微型计算机——也就是我们现在所说的“微机”来顺应人们的这种需求。与这之前的计算机相比,微机不但有体积小,使用方便等特点,而且价钱也便宜得多。也正是因为如此,微机才得以日益广泛的应用起来。因此,学会和掌握微机的使用不应该再是很困难的。

早在1794年,法国政府为了编制一套数学用表,要计算从1到200000的自然数精确到小数第19位的对数和有正弦、正切值及对数以及正弦、正切与它们的弧度之比的对数等等,竟雇佣了数百人,历时两年才完成。其中仅自然数的对数表就包括了近800万个数字。这说明其工作量之大已超出了人工计算的能力了。要满足这种大规模、高精度的计算要求,靠手工计算已显然不能适应需要了。所以人们必须寻求一种新的计算工具以提高工作效率。于是研制出了机械计算机。

尽管机械计算机能够部分地取代人工计算中的那些繁杂枯燥的过程。但它们的计算速度离人们的要求还相差甚远。1906年电子管问世。1946年用电子管为主要材料的第一台电子计算机在美国的宾夕法尼亚大学诞生了,它取名为ENIAC,用了18000多个电子管、重达三十多吨,其计算速度是任何一台机械计算机所不能比的。不仅如此,它的诞生还为电子材料应用于计算工具的制造方面打下了基础。所以它是划时代的产品,也是我们把它称做现代电子计算机之祖的原因。以电子管为材料的计算机在现代计算机史上也就称做第一代计算机。

第一代计算机虽然已具有较高的运算速度,但电子管本身的体积太大,用它制造的计算机其结构就不能太复杂,所以它的运算能力也就有限。因此,1948年新的电子材料——晶体管出现后至1958年左右,全部用晶体管组成的计算机制造出来了。它标志着第二代计算机的诞生。第二代计算机不仅比第一代计算机体积小,而且结构设计得更复杂,因此运算能力也提高了很多。

1965年左右,电子工业中推出了一种新技术。其主要特点是把几个至十几个晶体管以及它们之间的连线微缩在一块一平方厘米的金属或陶瓷片上。这样不但大大地压缩了其体积,而

且也提高了可靠性。降低了计算机制造的复杂程度。这就是第三代计算机的特点。

随着科学技术的进步,集成电路技术得到了飞速发展。在基片面积不变的情况下,集成度从最早的几个晶体管迅速提高到几百个、几千个至几万个。根据集成度分为小规模集成电路、中规模集成电路和大规模集成电路。至于集成度在百万个晶体管以上的叫做超大规模集成电路。在七十年代初,计算机开始用中规模以上的集成电路制造,从此计算机进入了第四代。经过二十多年的发展,计算机从第一代上升到第四代。整个的计算机技术也有了很大的进步。体积从大到小、运算速度从慢到快、使用从繁到简、制造从复杂到简单、功能从弱到强。但是无论怎样先进,这段时期的计算机在用途上只有一个那就是数值计算。根据计算机运算速度的快慢和运算能力的大小,计算机又有小型机、中型机、大型机和巨型机之分。

随着计算机技术的日益提高,科学家们发现计算机作为一种先进的计算工具,其利用价值并不是很高。因为在整个的人类社会活动中需要进行复杂的数值计算的时候实在太少。对于一般的人来说,更多的时候是进行一些简单的数值计算。它们的一个共同特点就是:参与运算的数据量大,但计算过程简单。因此对速度的要求不高。而且要求价格便宜、容易操作。这样才能让更多的人使用。正是针对社会的这种需求。微型计算机应运而生了。由于不需要太高的运算速度,所以结构上可以相对简单些,结构简单又使得体积可以大大缩小,加上大规模集成电路技术的应用,使微型计算机的体积大大缩小,而且造价大幅度下降,成了一般的使用单位甚至个人都能买得起的商品。1972年以后,微机开始体现出越来越广泛的优越性,进而推动了全社会使用微机的浪潮。

微机由于体积小、功能相对简单,在使用时每次只能允许一个人操作,不能象小型机以上的计算机那样可以允许多个人同时使用。所以人们又把微机叫做个人计算机。在英文单词里个人计算机是 Personal Computer,习惯上用这两个单词的首字母做为简称,因此微机也叫做 PC 机。

微机发展到今天,其制造材料由当初的大规模集成电路到现在的超大规模集成电路。在体积越来越小的情况下,其运算速度、数据处理量等各方面的性能也越来越高,早已超出了 80 年代初期小型机的性能。然而不论它怎样发展进步,它始终遵循的原则是:安装要简单、使用要方便容易。因此它永远也不能代替其它机种,尤其是一些承担如气象预报、卫星发射等巨大数据运算量的巨型计算机。也正因为如此,微机在发展过程中,更加注意发挥其灵活、方便、操作简单的特点,在非数值计算的领域中越来越体现出它的优势,使它成了计算机家族中重要的一员。

§ 1.2 微机硬件的结构

从计算机的发展历程中我们知道,计算机是做为一种计算工具发明的。做为计算工具,在人类文明社会几千年的历史中,计算机只是许许多多工具中的一种。人们发明和制造这些计算工具其目的当然是为了帮助自己进行复杂和繁琐的数值计算。在人的计算过程中,人们觉得最为困难的事就是记住大量的数字,数值一旦大而且多的话,不仅仅是难以记住的问题,其计算的速度也受到了很大影响。所以不管是哪种计算工具,它们的一个共同特点是要能够记住数字。然而记住数字的形式与用来记住数字的材料密切相关。在计算工具中,计算速度实质上也就是数值形式的变换速度。也就是从一个数字交换到另一个数字。算盘和计算尺之类的工具中这种交换还是依赖于人工。到了机械计算机时代,这种交换则靠的是机械的运动速度。而这种速度与人们所要求的速度还相差得太远。所以当电子以 30 万公里/每秒的运动速度展现在

人们眼前时,人们自然的就想到了用电子材料来制造计算机了。

用电子材料制造计算机的一个问题是如何用电子材料来反映数值。人们在长期的生产活动中,对数的计算总结出了一整套法则,如数的表示,数的运算法则等。其中最为常用的是我们现在称之为十进制的法则。做为电子计算机来说显然也应该按照这种法则来做。但是这里有一个问题,对于一个独立的电子元件如电子管、要让它来体现一个具体的数是没什么问题的,这只要在它通电时测出它的电压或电流值就可以了。但要让它交换数值时就有些为难了,比如原来的电压值为1,现在要加到2,通过加大电压,瞬间就可完成。但是这个过程的变化是连续的,这样我们就很难测到一个准确的值。这在大量的数值运算中就会产生极大的误差从而影响到结果的精确性。这当然是不能允许的。那么这个问题如何解决呢?我们可以注意到,要准确的知道一个电子元件的状态,最简单的方法就是看它是通电还是不通电。比如说我们用来照明的灯泡,当没电时,它是暗的,当打开开关时,它就是亮的,至于它有多亮,我们并不用关心。我们由此可以准确地知道这个电泡的情况。准确性的问题可以这样解决。但这样一来一个电子元件只能表达两个数值,对于更大的数应该如何表达呢?这里就有一个运算法则的问题,这也是电子计算机的核心问题。我们如果弄清了这个问题,也就不难理解计算机为什么能有这么大能力了。

我们知道十进制数是怎样定义的,现在我们只用两个符号能不能进行数值的运算呢?答案是肯定的,我们可以仿照十进制数的法则来进行。我们用0和1这样两个符号来表示数的值。再用这两个符号排列的位置来表示更大的值,那么符号与位置是怎样一种关系呢?让我们也从右边第一位开始。假设开始时它的值为零,我们表示成“0”,当增加一时,就变成符号“1”,若再加一时,由于没有第三种符号表示了,那么就要进位了,进位后,第一位变成了“0”,左边由于进位成了“1”,这时整个数值的表示应该是“10”。这种表示为数值二,也就是十进制数中的2这个符号。余此类推,三表示成“11”,四表示成“100”,五表示成“101”,……,从中我们可以得出,从右到左,第一位的值就是符号所代表的值,第二位的值是第一位两倍,第三位又是第二位的两倍,……,写成公式就是

$$\cdots \cdots 2^4 2^3 2^2 2^1 2^0$$

当我们写出符号1101时,根据法则我们可以知道它代表的值应该是 $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ 等于十三,用十进制数表示则是13。由此可见,用二种符号也同样可以表示数值。我们把这样一种表示法加上逢二进一这样一种运算法则称之为二进制数。

有了二进制数为基础,用电子元件来设计与制造计算机就变得容易了。比如说我们要让电子元件记一个数如:二十五,我们可以把25这个数先用二进制数的形式表示出来为:11001,那么我们可以用一个相应电路来反映成如图1-1所示:

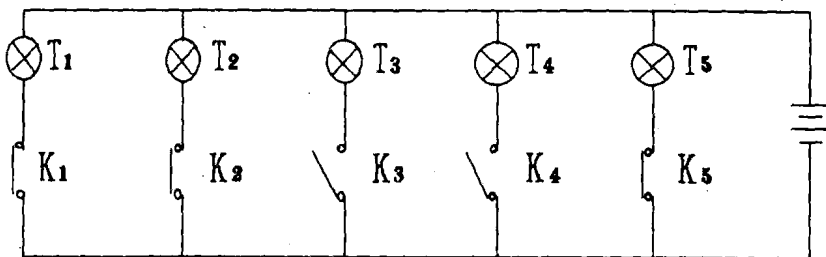


图 1-1

我们把 $T_1 \sim T_2$ 看成是五个灯泡, 当它们亮的时候我们看做是数字 1, 不亮的为数字 0。而把 $K_1 \sim K_2$ 当做是开关, 这样我们只要把 K_1, K_2, K_3 合上, K_4 和 K_5 断开。我们所看到灯泡的现象就是: 亮、亮、暗、暗、亮。这正好就是 11001。如果我们保持这种状态, 那么这五个电路就一直保持有这个数值。 $K_1 \sim K_5$ 的合与断是根据数值的大小来决定的。同样用电路也可以构成数值的计算, 象图 1-2 中这个电路: 在通电的情况下, 我们把开关合上看做是数值 1, 开关断开看做是 0。而 T 的状况看是计算的结果, 那么: 当 $K_1=1, K_2=0$ 或 $K_1=0, K_2=1$ 时, 灯亮结果为 1, 当 $K_1=0, K_2=0$ 时, T 暗结果为 0。另外一种特殊的是: $K_1=K_2=1$ 时, T 也亮。这样就有 $1+0=0+1=1, 0+0=0$ 和 $1+1=1$ 的结果。这与我们常用的运算法则有些不一样。但这并不影响计算机的制造。

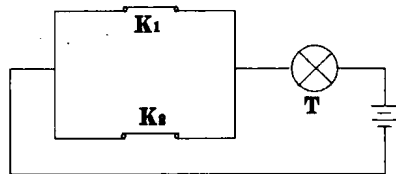


图 1-2

计算机是做为一种计算工具发明的, 因此在计算机的结构中, 一个很关键的组成就是保存数据的部份, 人们希望在整个运算过程中, 所有的数据包括准备计算的和计算过程中产生的中间结果等都由计算机来保存, 这样就解决了人工计算时要记忆大量数据而容易搞错的问题了。所以这一部份在计算机中是很重要的, 这也是计算机与其它计算工具的一个本质上的区别。人们把这一部份叫做计算机的存贮器。存贮器的大小直接关系到一台计算机的运算能力。存贮数据的目的当然是为了对它们进行运算, 那么一个能够对数据进行计算的结构在计算机中是必不可少的。这个结构在计算机中被称做是运算器。运算器是一台计算机的核心部份。要知道它的能力大小主要是看它一次能对多大的数值进行计算。在计算机中所有的数字形式都是用二进制数来表示的。对一个十进制数来说, 它相对应的二进制数形式在位数上要得多得多, 比如十进制数 8 只有一位, 相应二进制数 8 就要写成 1000 共四位。因此虽然人们希望运算器能对任意大的数一次完成运算操作, 但实际上是不可能的, 事实上也没有这种必要, 因为对一些很大的数可以分成几次来计算, 而运算器的运算速度快足以弥补由此造成的损失。

除了运算器和存贮器外, 计算机还有另一个重要的组成部份, 这个部份要负责按照人们的要求, 从存贮器取出要进行运算的数据送到运算器中, 然后要指示运算器是做哪一种运算, 最后还要负责把运算的结果存回到存贮器中。这一部份在计算机中叫做控制器。有了控制器、运算器和存贮器, 一台计算机就可以进行计算工作了。因此这三个部份的组合就叫做计算机的主机。

细心的读者可能已经发现, 有了计算机的主机, 那数据是怎样放到存贮器中去的呢? 计算出结果后我们怎样知道是多少呢? 这就引出了计算机的另一个组成部份, 就是数据的输入和输出部份。有了输入部份, 我们就可以把数据和我们对这些数据要做什么运算的指示送到计算机的存贮器和控制器中去。这一部份的任务实质上就是把数据和我们的要求都变成二进制数形式的电信号送给计算机。因此这一部份在制造上可以用任何材料而不局限于电子元件, 所以在原理和外形上有各种不同的输入机器, 我们把这些统称为计算机的输入设备。相应的输出部份也是同样, 它与输入设备的目的相反, 它的目的是把计算机中二进制数形式的电信号转换成我们能认识的其它的信号形式。所以它也有多种设备, 因此统称为计算机的输出设备。

由此我们可以看到: 一台计算机应该有必不可少的五个部份组成, 它们之间的关系如图 1-3 所示。

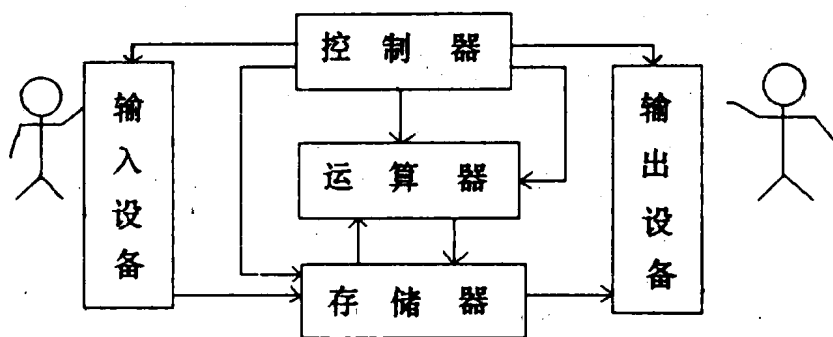


图 1-3 计算的组成

在这五个部份中,真正与我们打交道的只有输入和输出设备两个部份。在早期的计算机中,由于当时的条件所限,这两个部份,尤其是输入部份在使用时是很麻烦的,人们要把要运算的数据先手工转换成相应的二进制数形式,再一位一位的在输入设备上排好,然后再按下相应的开关,才能完成一次输入数据的过程。所以计算机的使用就不是那么容易的事了。随着科学技术的进步,现在的计算机输入、输出设备变得越来越符合我们日常生活的习惯了。比如输入设备,现在通用的是一种标有我们日常所熟悉的文字符号的键盘,我们只要轻轻地按一下键盘上有相应符号的键,这个符号就会由键盘内部的电路自动转换成相应的二进制数形式的电信号,并送到存贮器中去。这样一来人人都会使用计算机就可能了。

在计算机的组成部份中,还有值得一提的一个部份,它就是计算机的辅助存贮器部份。我们已经知道电子计算机是为了完成高速运算目的而选用了电子元件来制造的,它的重要组成部份之一——存贮器当然也不例外。然而人们发现,用电子元件组成的存贮器虽然解决了高速存取数据的问题,但它本身却存在一个难以克服的缺陷,这就是它所存贮的数据不能长期保存。它能够存贮数据的先决条件是要有电,一旦在工作过程中出现突然断电的情况,那么它所保存的所有数据都会丢失。另外,一台计算机的存贮器受体积的限制也不可能做得很大,以至能长期保存那些人们需要反复使用的大量数据。为了能够长期保存数据,专家们特地为计算机设计制造了另一种存贮器,这种存贮器不用电子元件来制造的,它能够长期保存数据,但它的存取速度比电子元件组成的存贮器要慢得多,因此它不能取而代之。为了区别,我们把这种类型的存贮器叫做辅助存贮器或外部存贮器,而把电子元件的存贮器叫做主存贮器或内部存贮器。

外部存贮器主要的一种是用磁材料来制造的,磁电之间很容易转换。磁材料一旦被磁化,只要不受强磁力的干扰,它的状态就可以稳定相当长的一段时间。与有电无电没有直接关系。这种磁材料的存贮器有很多种,其中最多的是磁带和磁盘两种形式,它们在接收和送出数据时都是依赖相应的机械设备,所以它们的速度也就受到了限制。但由于它们可以和计算机完全脱离开来制造和保管,因此它们一方面可以做成很大的存贮量,另一方面可以方便的携带与保管。所以它越来越受到人们的重视,成了现代计算机中一个不可忽视的组成部份。

如果我们把一台计算机比做人的话,它的运算器、控制器和主存贮器相当于它的大脑,而输入设备可以看做是它的眼睛和耳朵;输出设备则是它的手和嘴,辅助存贮器只是它的一本笔记本,用来记录哪些需要长期记住的信息。由于辅助存贮器可与计算机的制造分开,在使用时也只是有选择地连结,具一定的独立性,这样它的存贮量就可以是无限的。就象你的笔记本一样,可以用完一本再用新的一本,所以辅助存贮器又叫海量存贮量。

和计算机家族中其它机型一样,微机的组成也是由以上几个部份构造的。不过它的各部份

的内部构造与其它机型有所不同。这是因为当初设计微机时主要是考虑怎样使它的体积更小，制造时怎样更容易。尽管有大规模集成电路技术，但还远远没有达到现在这样的程度，因此，微机中各部份的内部结构相对更简单，在计算功能方面如运算器的能力，运算速度、存贮量等都不如小型机以上的机型。在微机中运算器和控制器被集成在一块基片中，它被叫做中央处理器，其英文单词为：Central Processing Unit 所以又叫 CPU，它的输入设备通常是用键盘，上面是标有 A、B、C、D……26 个英文字母和 0~9 十个数字以及一些象逗号、引号等印刷符号的按键组成的。输出设备通常是一个显示器。存贮器(内存)与 CPU 是做在一块电路板上，它是微机的主板。CPU、内存、输入输出设备之间的联系是用一组公共的线路，这线路在微机中叫做总线。总线是数据、控制信号等信号的通道。一台典型的微机结构如图 1-4

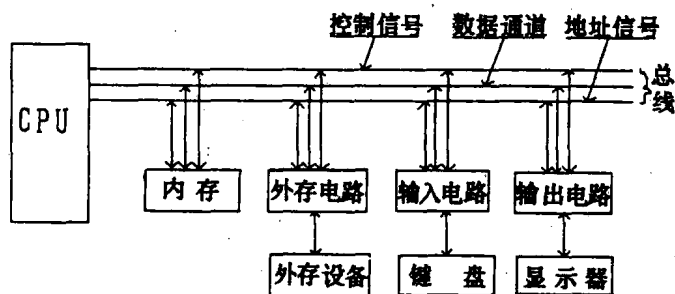


图 1-4 微型机结构

由于采用了这种总线结构，使得微机具有很好的扩充性能，人们可以设计各种专门的电路接在微机的总线上，使得微机具有这种专门用途的能力。这也是微机得以广泛应用的原因之一。所有组成和构造出计算机的材料统称为计算机硬件。

§ 1.3 微机软件

一台计算机造出来之后，原则上只要它一通电就可以开始工作了。但是这种想法可能过于简单了些。我们已经知道，计算机的硬件其实都是一些电子元件构成的，它们的工作方式实质上就象是一堆电子开关那样开开合合。那么在我们把数据输入给计算机时，这些开关应该怎样动作呢？也就是说，哪些开关应该开着，哪些应该合上？才能保证数据送到指定位置呢？等等等等，如果不解决这些问题，那么数据将无法为计算机所接受。计算机也就谈不上可以工作了。

怎样能让一台计算机进行工作呢？我们大致要做好这样几件事情：

1. 设置计算机的初始状态，比如通电后，各个部份应该打开哪些开关？合上哪些开关？以便能够开始工作。
2. 把要做的事情安排好并输入给计算机，比如说进行数值计算；那么要算哪些数据？它们放在存贮器中什么位置？第一步做什么？第二步做什么？做完之后又怎样？等等等等。
3. 让计算机开始工作。根据我们的需要，让它或者进行计算、或者接受数据，或者停止工作等等。

这几件事情，我们不但要在使用计算机前先想好，而且最终还得用计算机所能接受的形式也就是二进制数的形式表达出来，这样计算机才可能明白我们要让它干什么？怎么干？比如说：我们打算让计算机做的运算是加法，那么我们就要写成 11001101 这样的形式，而停止工作要写成 01110110，如此等等。这样的一种表达形式我们称之为计算机的指令。实际上这种指令形式是根据计算机的硬件构造来编制的一种符号。对一台计算机来说，根据它的结构和每个部份

具体的构造,对应的有一整套这种指令,以便它的每一个具体的动作都能够受到我们的指挥。这一套指令就叫做计算机的指令系统。不同结构的计算机它们的指令系统是不同的。我们要让计算机做的事情的每一步细节用指令的形式写出来这个过程叫做编写计算机程序。所以说要使用计算机就要先熟悉它的指令系统,并会用指令系统编写程序。

指令系统和用指令系统编写的计算机程序这样一种能够让计算机工作起来的形式相对计算机硬件而言称之为计算机软件。之所以把它们叫做软件,一方面是说明它们是计算机的一个不可缺少的组成部份,另一方面是说它们不象计算机硬件部分那样是一种固态的,有形的物质,而仅仅是符号形式。与计算机的硬件一样,计算机软件也是有一个发展提高的过程的。正如我们刚才所说的,指令系统的表示方法是二进制数的形式,这种形式一般的人很难学会并记住。就是专门从事计算机工作的专家们也同样困难。另外不同结构的计算机的指令系统也不同,这更限制了大部份人的使用。为了让使用计算机的人们能更容易的记住指令系统,科学家们想出了用一种人们更熟悉的符号——文字的形式来对应每一条指令的方法,比如说:对前面假设的加指令 11001101,对应用文字 ADD 来帮助记忆,而 ADD 实际上就是英文加的意思,在编写程序时就用这种形式来写。这种符号的形式在计算机软件中叫做汇编语言,用汇编语言编写的程序叫做汇编语言源程序。有了汇编语言,使用计算机就要方便多了。但是计算机并不能直接理解接受这样一种形式。作为一种兼顾两方面的做法,就是用指令编写一个专门把汇编语言的符号转换成相应二进制数形式的程序并事先输入到计算机中,当人们把汇编语言源程序输入到计算机中后,由这个转换程序把它转换成相应的指令程序。这个过程叫做汇编,而这个转换程序也就叫汇编程序。整个过程如图 1-5 所示:

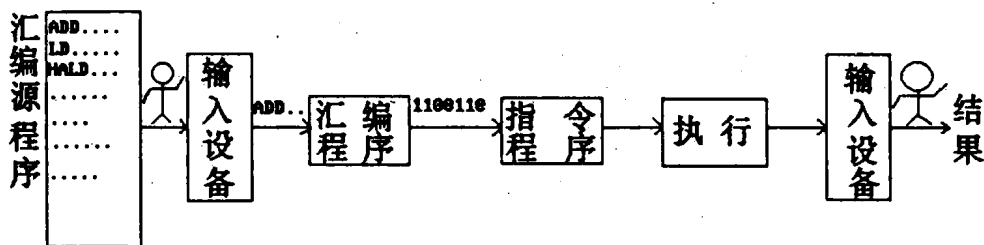


图 1-5 汇编程序执行过程

有了汇编语言,对使用计算机时编写程序的工作是方便多了,但是整个的工作量相对来说却加大了。比如说:在源程序输入到计算机之前,汇编程序必须先输入到存贮器中,源程序输入到存贮器中时,要考虑把它放在哪个具体位置以至不会和汇编程序发生冲突?汇编程序把源程序转换成的指令程序也有一个放在哪个具体位置的问题。另外还有诸如:输入设备正在工作时,存贮器应该做哪些准备?当输出设备正在送出数据时,又有数据要立即送出怎么办?等等,……这些问题总归起来就是如何安排好计算机中各部份的工作的问题。使得它们之间既能够互相配合、互相协调、又不致于由此使得它们本身的功能受到浪费而造成整体效果的下降。解决这个问题的方法自然还是要我们在熟悉所使用的计算机硬件的情况下,把它们的各个部份在什么时候应该做什么、怎么做等等安排好,再用二进制数形式的指令表达出来并在输入汇编语言源程序之前把它们输入到计算机中,接着让计算机按照这个安排先做起来,最后才能让计算机做实际的运算工作。由此可见,这个过程既复杂、工作量又大,它甚至要比我们要让计算机做的计算的工作量还大,然而它又是使用计算机必不可少的工作过程之一。这个过程使得一般的计算机使用者无法真正的使用计算机,而是要通过专门的计算机操作人员协助才能做到。这就使得使用计算机的实际效果受到了影响。这显然不是计算机设计者们所愿意得到的结果。

为了让不熟悉计算机硬件结构的大多数人也能直接使用计算机,就要让计算机中各组成部份都做好相应地准备,并且在使用中能够随时调整它们之间的关系,以保证数据和程序的正常运行。比如说:当你的计算程序需要输入一组新的数据时,就要让键盘或其它的输入设备处于接受状态;当在计算过程中有一些中间数据产生时,就要让它们暂时存放在存储器中的某一个位置,而且在需要时又能够从这个位置中取出来。这种类似的操作对每一个使用某一台计算机的人来说要求都是一样的。这种操作过程因为是根据计算程序的提出而进行的,要掌握它不仅仅是要对计算机硬件结构相当熟悉,而且要求交换的时间与计算程序的运行时间一致。因此别说是一般的使用者,就是专门的操作人员也难以胜任。最可行的办法还是让计算机自己来处理了。当然所谓的让它自己来处理,实际还是要专家们用指令编写好一个程序,让计算机按照这个程序来做这些事,这样的程序在计算机专业词汇中叫做操作系统。

有了操作系统,使用计算机的人们就完全不必去了解这台计算机的硬件是怎样组成的,它们中各部份又是怎样协调的,只要你的计算程序中提出要哪部份工作,操作系统就会把这部份的状态调整好,以满足你的要求,所以操作系统实际上就是计算机硬件的大管家。当然,操作系统在形式上也是一个用指令系统编写好的程序,只不过这个程序是一定要在计算机刚通电开始工作时就要先运行的,而且在每个计算程序工作时它都要处在工作状态。它首先对计算机硬件的各部份进行检查,如果都没有故障则把各部份都置成准备工作状态,然后到指定的外部存储器中把操作系统程序其它部份输入到内存中并让它运行。对计算机进行检查的程序较小,而且因为它要在通电后自动的进行工作,所以它不能放在外部存储器中,而是存在另一种特殊的存储器中,它与内存不一样的是,它不能像内存那样可以按需要随时地往里面存入或取出数据。它只能往里面存入数据一次,以后就只能从里面读出数据而再也不能存入了。存入到里面的数据不受供电的影响。也就是说:没有电它里面的内容也不会消失。这种存储器因此而叫做只读存储器,英文缩写为 ROM。相应的内存就有另外一个名称叫做为随机访问(存取)存储器,英文缩写为 RAM。存储在 ROM 中的这一部份操作系统程序也叫做引导程序或自检程序。

在操作系统的帮助下,一般人使用计算机在操作时已经不是很困难的事了,反过来编写一个计算程序的工作显得过于困难了。虽然有了汇编语言,人们不必去记那些很难记住的指令,但毕竟汇编语言中的符号只是每一条指令的注解,而且还是用单词缩写和简写的形式表示,在使用时还要和用指令一样具体安排每一个数据的动态,比如某个数具体从哪来,到哪去等等。仍然摆脱不了要对计算机硬件组成进行了解的麻烦。人们希望能够用更为简单的方法来使用计算机,也就是希望在不了解计算机硬件的情况下编写出一个计算机能够理解并运行的程序来。而要实现这个愿望,最容易的方法就是用自己所熟悉的符号形式来表达解决某一个问题的想法,也就是说用人类的自然语言符号来编写程序。但是这不大可能,因为人们的自然语言词汇量大,语法复杂,要让计算机认识这些词汇和理解这些语法语义,就要象汇编语言一样由专家编写一个这样的程序来把这些词汇转换成相应的二进制数的形式,那么这个程序将是一个很大很大的程序。专家采用了一种办法:就是围绕着某种特定的需要,从自然语言中抽出那些能够表达出这种需要的最少的词汇量和最简单的语法,定义成一种新的语言形式,这样的一种语言形式就叫做计算机语言,为了与汇编语言区别开来,又叫计算机高级语言。当然对这样的语言,也要有一个对应的程序来充当语言的词汇和语法与二进制数形式的指令之间的转换。这种转换程序也叫做编译程序或解释程序。

计算机高级语言解决了使用计算机时编写程序的难题。但是做为程序设计也就是想解决某一个具体问题的方案这个过程仍不是一件容易的事,一方面要对这个问题所涉及的各个

领域非常熟悉;另一方面也要对所选用计算机高级语言十分的理解,毕竟它不能像我们习惯的自然语言可以随心所欲的使用,这是它的词汇量和语法有限造成的。所以不是每一个人不经一定的专门训练就能编写程序的。要真正做到人人都会使用计算机,就要求计算机的软件设计专家们为每一个可能要解决的问题都编一个专门的程序,这样的程序只需要使用者告诉操作系统,把它存到内存中去,然后再从键盘上输入相应的数据,就可以得到满意的结果。这种程序叫做应用程序,有了应用程序,人们就可以真正做到不需要专门的训练就会使用计算机了。

计算机的硬件和软件总称为计算机系统。硬件有各种机型,软件则分为操作系统、系统软件(包括编译、解释和汇编等程序),应用软件等等。它们之间的关系如图 1-6。对于一个普通的使用者来说,你所接触的只是操作系统而已,当你要解决某个具体问题时,如果有相应的应用软件。你只要通知操作系统把它们运行起来,然后再给出一些数据就可以了。如果你觉得现有的应用软件不能圆满的解决你的问题,那么你可以自己编写一个计算机程序以达到你的目的,这样一来,你就得花些时间去学会一种或几种计算机高级语言和设计程序的方法,通过系统软件最后写出一个你所希望的应用程序就行了。当然现有的系统软件还无法使你满足,你也可以用汇编语言,高级语言甚至指令系统自己设计一个相应的系统软件。这样一来,你还得学习计算机硬件的有关知识,这就需要较多的时间了。所以说要学会使用计算机是要难也难,要容易也相当容易,这就要看你准备用计算机干什么而已。

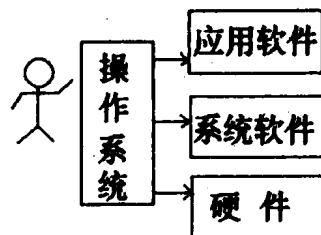


图 1-6 计算机系统

§ 1.4 典型微机系统

做为一套微机系统,它的软件与其它机型的软件基本上是一致的,而它的硬件部份就有它自己的一些特点,因此我们着重来看看它的硬件部份。

前面我们已知道,微机的产生得力于大规模集成电路技术的应用,随着集成电路技术的发展,超大规模集成电路在体积不变的情况下能把几万甚至上百万个晶体管集成起来,微机的功能也就随之增强。运算器的能力从四位发展到八位、十六位、三十二位。现在的微机在功能上已超过了八十年代初期小型机。甚至达到了当时中型机的水平。

由于微机一问世,就展现出广阔的应用前景。因此企业界一些有能力的公司争相研制、推出自己的产品。这些公司中有些是专门生产 CPU 这类微机中的集成片,有些则是用这些集成片来构成一套完整的微机。在生产集成片的公司中,最有代表性的是美国的英特尔公司(Intel)、莫托洛拉公司(Motolola)、和齐洛格公司(Zilog)。他们根据自己的技术力量设计出具有自己的特点的 CPU。并逐步形成自己的系列产品。如英特尔公司的新产品是以 Intel80××为系列的,以最早的八位 CPU8080、到十六位的 8088、8086、80286、到现在的三十二位的 80386、和 80486 为序列号,而莫托洛拉公司的产品有 M6800、M68000 等型号,Zilog 公司则是 Z80、Z8000 等产品。这些公司的产品虽然都是微机中的 CPU,但它们内部的具体设计是不同的,因此它们各自有自己的指令系统。对一家公司来说,为了保证自己的产品在使用上的连续性,他们就要设法保留以前的产品的使用方式,在 CPU 这样一种产品中,使用方式就是它的指令系统。也就是说:当 CPU 本身的结构性能提高以后,新指令系统除了增加新的指令外仍能要保留旧的 CPU 所使用的指令。这样一种方式称之为兼容。一般地来说各公司之间的 CPU 指令是不兼容的。