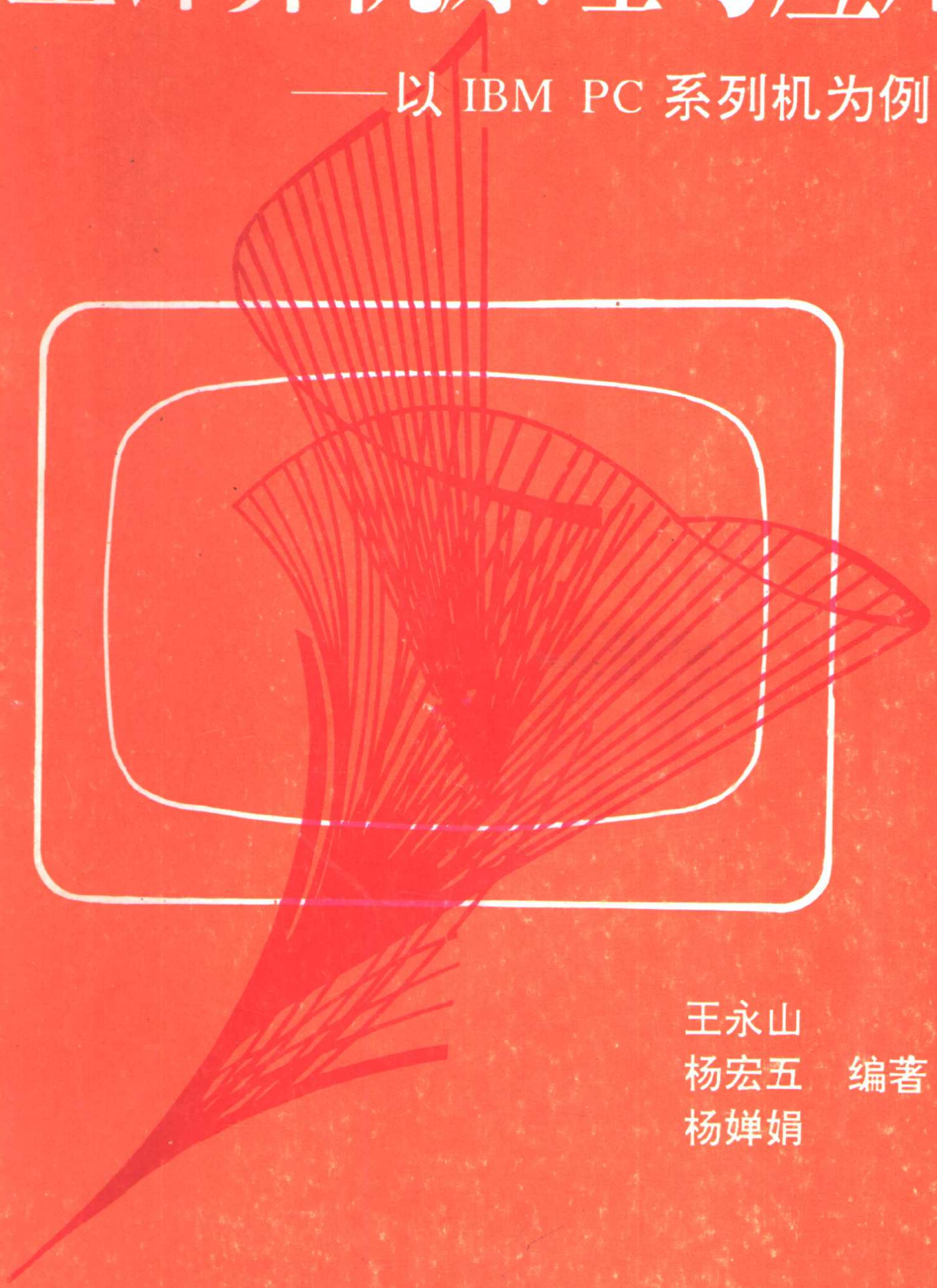
·大专适用·

微型计算机原理与应用

——以 IBM PC 系列机为例



王永山
杨宏五　编著
杨婵娟

西安电子科技大学出版社

微型计算机原理与应用

——以 IBM PC 系列机为例

(大专适用)

王永山　杨宏五　杨婵娟　编著

西安电子科技大学出版社
1995

(陕)新登字 010 号

内容简介

本书是为电子技术应用等专业大专生“微型计算机原理与应用”课程编写的教材。为适应课程内容更新的需要，本书以 8086／8088 微处理器和 IBM PC 系列机为例讨论 16 位微型机。全书共七章，讨论了计算机中数据和信息的表示方法，微处理器组成原理和如何以微处理器为核心组成微机系统的方法，汇编语言程序设计技术和输入输出接口技术等问题。

本书在内容选择、次序安排和叙述方式等多方面，都突出地体现了编者的指导思想：面向教学和面向应用相结合，既便于学生自学，又能直接指导应用。本书也可作为从事微机软硬件开发工作的科技人员的参考用书。

微型计算机原理与应用
——以 IBM PC 系列机为例
王永山　杨宏五　杨婵娟　编著
责任编辑　梁家新

西安电子科技大学出版社出版发行
陕西省富平县印刷厂印刷
新华书店经销
开本 787×1092　1/16　印张 22 3/16　字数 528 千字
1994 年 5 月第 1 版　1995 年 8 月第 2 次印刷　印数 8 001—16 00

ISBN 7－5606－0334－3/TP・0122(课)　　定价：16.50 元

前　言

本书是为电子技术应用等专业大专生“微型计算机原理与应用”课编写的教材。以前电子类专业大专生该课程的内容以 Z80 为代表的 8 位微处理器和微机系统为主，本书则改为以 8086／8088 微处理器为代表的 16 位微处理器和 IBM PC 微机系统为主。这是教材内容的一次大更新。

根据编者多年教学实践的体会，确定编写本教材的指导思想是突出面向教学和面向应用。为了实现面向教学和面向应用相结合的指导思想，本书编写时注意了以下几点：

(1) 讲述微处理器原理和如何以微处理器为核心组成微机系统部分时，以 8086 微处理器为背景；在讨论汇编语言程序设计和输入输出接口技术时，则以 IBM PC／XT 系统机为背景。

(2) 在内容的次序安排上，特别注意到便于学生自学阅读。无论整个教材还是各章节都由浅入深，突出重点，前后照应。

(3) 内容的选择和安排充分照顾到学生上机实习环节，并为尽早上机实习创造条件。

(4) 很多章节写入了编者在多年科研实践中的经验和体会，使本书能在学生的以后工作中有指导作用。

全书由王永山主编，并编写其中第一、七章，第二、三、五、六章由杨宏五编写，第四章由杨婵娟编写。

由于编者水平的限制，加之时间仓促，定会有不少缺点和错误，敬请读者批评指正。

编　者

1994 年元月

目　录

第一章　微型计算机系统概述

学习本书的读者，应学习过与本教材关系最密切的两门课程：计算机算法语言和数字电路与系统。在学习算法语言课中，读者已经接触过计算机，而且一般也是微型计算机，能够用 BASIC 语言或 FORTRAN 语言或其它高级语言编出多种功能的程序并在计算机上运行这些程序。可以说，已经具备了一些关于计算机的基本知识。但是，在很多情况下，只具备这些基本知识是不够的，需要对微型计算机的工作原理有更多的知识。

计算机的应用，按其工作特点可分为三类，即科学计算、数据处理和过程控制。

在科学研究，特别是理论研究中，常常会出现这种情况：经过严密地论证和推导，得出复杂的数学方程，需要求得方程的解。如果手工运算，可能要经过数月、数年的时间，有时甚至是手工无法完成的。面对这样的难题，计算机可以发挥强大的威力。计算机在这样的科学计算应用中，一般用高级语言编程。多年来，计算机科学家设计的多种高级语言和开发出的支持各种高级语言的程序(软件)，对计算机在科学计算中应用的贡献是无法估量的。计算机在科学计算中的应用，除了用高级语言编程序以外，与其它两类应用相比较，还有以下两个特点：第一，它没有很强的实时性要求，虽然使用者在运行程序时也希望尽快地得到运算的结果，但对结果产生的时间没有严格的要求，结果产生的迟早不影响结果的有效性。当然，这不包括以时间作变量的过程的仿真或模拟。计算机的高级语言是面向用户的，就是说，用高级语言编程序比较容易和方便，经过短时间的学习和训练，一般人都可以编出功能很复杂的程序。但以程序运行时耗费的时间长短来衡量，用高级语言编程并不是最佳的选择。一般情况下，对于完成相同的功能，用机器码语言(或汇编语言)编出的程序运行起来要比用高级语言编出的程序快得多。第二，在科学计算中，需要输入计算机的数据，一般不是从某种物理现场在线实时采集的，不需要有专用的完成数据采集任务的输入设备；同样，计算的结果，一般也不完成对外界的控制功能，不需要有专用的输出设备与其它系统相连。这就是说，只用于科学计算的计算机，可以是只包括常规输入输出设备的独立的计算机系统。对于一般微型计算机，常规的输入输出设备有键盘(输入)、显示器(输出)、打印机(输出)、磁盘机(输入／输出)。对于小型和大型计算机，常规的输入输出设备除上述几种外，常常还有卡片读入机(输入)、纸带读入机(输入)、纸带穿孔机(输出)等。

现代信息系统的最重要特点是用计算机进行数据处理。数据处理或信息处理总是以某种管理为目的的。例如，在商店的销售柜台用微型计算机进行销售登记、记帐、开发票、列清单等；财务部门的票据处理、帐目处理和结算、打印工资表等；物资仓库进出物资的登记、打印订货清单等；人事部门用计算机建立人事档案等，它们分别属于商业管理、财务管理、物资管理和人事管理。当然，以上所说的数据处理是最简单的情况。还可以举出稍复杂的例子，如一部警戒雷达的数据处理计算机，要完成对飞行目标跟踪的数据处理任务时输入计算机的是雷达检测录取设备录取的飞行目标的一个个独立的点迹数据，即

某一瞬间目标在空中的位置坐标数据和时间等有关参数。计算机要把一个个独立的点迹数据进行相关处理，识别出属于同一批飞行目标的所有点迹，把它们按时间顺序连接起来，从而获得飞行目标的运动轨迹，即航迹。由于雷达系统本身存在测量误差，测量空间存在多种原因形成的干扰，飞行目标飞行状态的变化等，这些因素都是随机的，使实际的航迹处理相当复杂。而且从这个例子可以看出，计算机在数据处理的应用中，有相当多情况是实时性很强的，而且输入数据的采集和结果数据的输出需要专用的设备。这两个特点与前面说明的计算机在科学计算中的应用成为鲜明的对照。这里所说的"实时"有两方面的含义：一方面是数据的采集，如目标点迹数据的录取，是实时的；另一方面是计算机数据处理产生的结果，如对目标点迹处理产生目标的航迹，必须在允许的时间内完成。"在允许的时间内完成"，这是理解数据处理实时性要求的关键。不能用统一的时间长短来定义不同系统的实时性要求。不同系统的"实时"有自己的标准和要求，有自己的允许的时间迟延。如果结果数据产生时间超过了允许的时间迟延，使数据无用了，更甚者使整个系统不能完成规定的任务，这是不允许的。许多数据处理系统实时性要求很高，不能用计算机键盘等常规输入设备输入数据。上例中的雷达检测录取设备可以看作是计算机的专用数据采集设备。雷达航迹处理计算机一般要求有专用的综合显示设备用作输出，有时还要有通信设备与上级指挥机关通信，它们是专用的输出设备。还可以举出许许多多例子来说明计算机在数据处理领域中的应用，它们可能更复杂、功能更强。大的工厂企业单位，高级的政府机关，军事指挥机关等，用计算机完成大量的业务管理任务，甚至协助人们制订计划和长远规划。大量统计资料表明，在早期，计算机的主要应用领域是科学计算。从 60 年代开始，一些发达国家把计算机应用的重点从科学计算转向信息处理，特别是经济管理。

对计算机在过程控制中的应用，也可举出许多例子。例如导弹的发射和制导过程，总是不停地测试当前的飞行参数，经过计算或处理，然后再控制飞行的状态。过程控制应用中的计算机，一般都是组成闭环控制系统中的一个部分，或称为一个环节。所以必需有专用的输入输出设备，使计算机有效地接入控制系统中，而且计算机对于输入信息的处理及结果的输出总是实时进行的。

以上说明了计算机在三类应用中的特点。实际上这三类应用并没有严格的界线，特别是实时数据处理和过程控制，有许多相同的特点，在很多情况下，计算机兼有数据处理和过程控制功能。

我们把计算机的应用按上述分类说明，目的在于说明只学习计算机算法语言课，只会用高级语言编计算程序，在很多应用中，这些知识是不够的。实时数据处理和过程控制要求实时性，希望编制的程序更精练，运行起来更快；专用的输入输出设备与计算机的连接和编程控制(称为接口)，更不是只具有高级语言编程知识所能胜任的。为此，必须对计算机的工作原理有更深入的了解，对计算机的逻辑组成、工作方法、与外界的接口技术以及直接依赖于计算机逻辑结构的机器码语言、汇编语言编程方法等需要进一步地学习。本课程就是基于这一目的而设置的。

本章概述微型计算机系统，勾画出系统组成的轮廓，简要描述微型计算机系统的基本工作方法。过去，很多人在学习微型计算机原理时，常常忽视软件的作用，不把软件作为微机系统的组成部分。本章不仅从道理上强调软件，特别是系统软件是和硬件一样重要的微机系统组成部分，而且专门设置一节内容讨论磁盘操作系统 DOS 的主要功能和常用命令。

1.1 微型计算机系统的组成和工作方法

组成一个计算机系统或微型计算机系统，必需包括硬件(Hardware)和软件(Software)。硬件是组成计算机系统的物理实体，是看得见摸得着的部分。对于微型计算机系统，硬件包括主机箱及其内部的电子器件、机电元件组成的电路、键盘、CRT显示器和打印机等。大型计算机系统相当复杂，组成的硬件常组装在若干个大的机柜中。所谓软件，简单地说就是程序。后面将说明为什么把软件，即程序看得如此重要，竟和硬件一样看作是组成计算机系统的不可缺少的部分。计算机系统中的硬件和软件，是互为基础、互相支持的关系，计算机各种功能正是在硬软件的这种互相支持中形成的。

1.1.1 微型计算机系统的硬件和基本工作方法

一、微型计算机系统的硬件组成

在计算机算法语言课程中，曾把计算机的硬件组成划分为运算器、存贮器、输入输出设备和控制器四个大的组成部分。通过对四个部分的简要描述，说明了计算机是如何仿照人用笔、纸进行计算的操作过程。但是在学算法语言课的时候，还不知道这四大部件是一些什么电路，这四大部分是如何互相配合工作的。

数字电路与系统课程的学习，提供了理解计算机硬件工作原理的基础。让我们回顾一下所学过的数字电路的功能。在讨论组合电路设计时得出这样的结论：对于任何表示因果之间有唯一对应关系的逻辑函数，总可以设计一套组合逻辑电路，使输入输出之间符合因果关系。讨论过的以全加器为核心的算术逻辑单元(ALU)，就是一种这样的组合逻辑电路。ALU在不同控制信号控制下，可以完成两数相加、两数相减等算术运算和逻辑乘、逻辑加、取反等逻辑运算。这就是计算机具有计算能力的基础。我们还学习过时序逻辑电路。以触发器为基本电路形式的寄存器、锁存器和大规模高集成度的存贮器，它们能把二进制代码长时间地寄存和存贮。有了这样有记忆功能又能随时存入和输出的逻辑电路，准备参加运算的数据就可以寄存或存贮起来，经过运算得到的结果数据也可以寄存或存贮起来。例如，算术逻辑单元ALU的输入端由一组寄存器驱动，参加运算的数据先置入输入端的寄存器中。ALU的输出端接到一组寄存器的输入端上，在选通脉冲作用下，可以把ALU产生的结果数据置入寄存器。通过适当的设计把组合电路与时序电路结合起来可以组成控制信号产生逻辑，这种逻辑在时钟脉冲驱动下，可以产生各种运算操作所需要的控制信号序列。如果我们把复杂的运算分解为有先后次序的一系列简单运算，让控制信号产生逻辑按简单运算序列的要求产生控制信号加到ALU和寄存器组成的运算电路上，就可以逐步完成所需要的运算。这就是计算机完成复杂运算的基础。组成计算机硬件的四大部件中的运算器和控制器就是由上述这类逻辑电路组成的。

图1.1.1是更加接近实际的典型的微型计算机的硬件组成框图。图中虽然包括不只四个框，但仍可以找出与四大部件的对应关系。图的右上方存贮器模块就是存贮器部件，用于存贮程序和数据。存贮器模块下方的大容量的存贮装置(主要指磁盘或磁带)框可以看作是存贮器部件，也可以看作输入输出设备。I/O装置框，属于输入输出设备。计算机的

运算器和控制器都是由数字逻辑电路组成的，人们常把它们合起来称为中心处理部件，缩写为 CPU(Central Processing Unit)。在微型计算机系统中采用的微处理器，就是其全部功能集成于超大规模集成电路中的 CPU。所以，微处理器(或者包括某些辅助逻辑)实际上完成着运算器和控制器的功能。

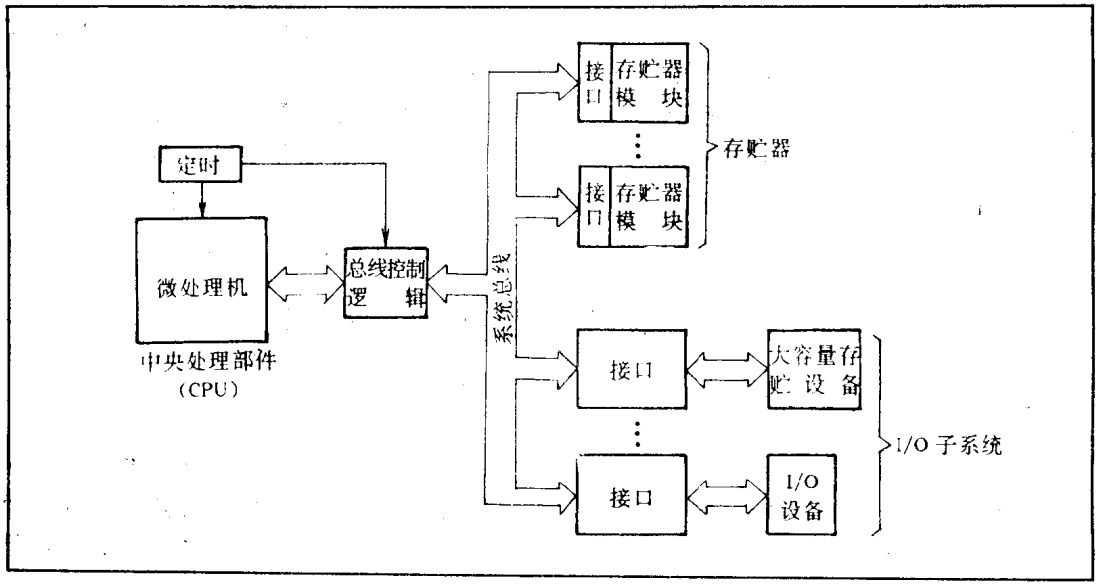


图 1.1.1 微型计算机组成框图

微型计算机系统体系结构特点之一是采用总线(Bus)结构。总线结构，是主体部分与其它部分连接的一种策略。其基本思想是，主体部分与其它多个不同部分都通过同一组精心设置的连线相连接。在微型计算机系统中，有两个级别(或层次)的总线，即微处理级总线和系统级总线。微处理器级总线的主体部分是微处理器。微处理器通过微处理器级总线与其它逻辑部分连接组成微机系统的基本部分——主机板。系统级总线的主体部分是主机板。主机板通过系统级总线与其它部件连接组成微机系统。为什么采用总线结构呢？首先来看微处理器级总线结构的必要性：在早期的由分立元件(电子管或晶体管)或小规模集成电路组成的计算机中，可以根据逻辑需要在任何点之间直接用导线相接，所以计算机各部件之间的连线常常可以结扎成很粗的捆。但在微型计算机系统中不允许这样，一个微处理器集成电路芯片只有几平方厘米，它与外界的连接点数受尺寸限制最多也不过几十个连接点。这就迫使设计者必须精心安排这些接点上的信号，使这些接点成为与外界交换信息的最有效的公用通路，这就是微处理器级总线。系统级总线是以微处理器级总线为基础经过一些逻辑处理(图 1.1.1 中的总线控制逻辑)形成的。第五章将对系统级总线进行较深入讨论。

在当前的系统微机(不是单板机)中，一般采用这样的组织方法：把微处理器、总线控制逻辑、定时逻辑、部分存贮器等构成基本系统所必要的逻辑组装在一块较大的印刷电路板上。该印刷电路板称为系统板或主机板，它固定在机箱内。系统板上还装有多个多接点的长形插座(IBM PC 机的插座有 62 个接点)，每个插座都可以插入另一块印刷电路板与系统板垂直。所谓系统级总线就是系统板上接到这些插座接点上的全部信号线。各插座是并联的，各插座上信号的排列是相同的。构成完善的微机系统所包括的外部设备，如

CRT显示器、打印机、磁盘存贮器等接口逻辑和扩充容量的存贮器，都装在印刷电路板上，插入系统板的插座里，通过系统级总线与系统板内逻辑相接。在这里，"总线"的含义更容易理解了，任何新增加的外部设备，只要遵照插座接点的信号约定设计接口逻辑电路，都可以"挂"在"总线"上，成为微机系统的一部分。

二、微型计算机的基本工作方法

在学习高级语言编程时，我们并不知道计算机是如何执行一条条语句的，想像不出如全加器、寄存器等逻辑电路是如何完成求 sin(x)等复杂函数语句的。现在可以回答这个问题了。理解这个问题涉及到指令、程序、程序的存贮和执行等几个基本概念。

1. 指令和程序

前面说过，以全加器为核心的算术逻辑单元 ALU，在控制信号控制下，可以完成两数的相加、相减、逻辑加，以及一个数的取反、取负等算术或逻辑运算。任何微处理器芯片中，都包括 ALU 逻辑，除上述基本算术逻辑运算外，微处理器还可以完成数据在寄存器之间、寄存器与存贮器之间、寄存器与外部设备之间的传送操作。我们知道乘法是连加运算，除法的商可在连减的过程中产生，sin(x)等许多函数可以展开成只含加、减、乘、除基本运算的级数，总之无论多么复杂的运算都可以分解为一系列基本运算。计算机执行高级语言的任何一条语句，都是在执行了一系列基本运算后完成的。

计算机能直接完成的两数加、减、逻辑乘、逻辑或以及数的取反、取负、传输等等许多基本运算或操作，每种基本运算或操作称为一条指令。在学习微机原理和汇编语言编程时，指令是最基本最重要的概念。如何理解指令和程序的概念？我们应掌握以下要点：

(1) 一个微处理器所能执行的全部指令，就是这个微处理器的指令系统(Instruction Set)。一个微处理器的指令系统是设计微处理器时决定的，成为微处理器固有的功能。指令及指令系统所能完成的功能的强弱，是这种微处理器功能强弱的具体体现。

(2) 指令在微处理器内是以代码形式出现和施展控制的，任何一条指令都用与其它指令不同的代码表示。假设微处理器内有寄存器 A 和寄存器 B，又假设"寄存器 A 中的数与寄存器 B 中的数相加，其和存入寄存器 A"是一条指令，并假设其指令码为 00000001，那么微处理器内一旦出现了指令码 00000001，就会按指令功能的规定执行寄存器 A 与 B 内容相加，和存于 A 的操作。不同的指令有不同的指令码。指令码对操作的控制方法也是不难理解的：微处理器内有专用的寄存器寄存指令码，这个寄存器称为指令寄存器。它的输出控制专门设计的组合网络，这个组合网络能在指令码控制下，把时钟脉冲变换成完成该指令操作所需要的控制信号序列，加到 ALU 等执行逻辑，完成指令包含的一系列微细操作。

(3) 按运算功能要求把指令排列起来，这就是程序，指令是构成程序的基本单元。对于不同型号的微处理器或由不同型号微处理器组成的微型计算机，可能编出具有相同运算功能的程序，例如都有计算 cos(x)的功能，但编程用的指令和指令序列可能是不同的。这是因为不同型号微处理器有不同的指令系统，而且编程序时不能排入指令系统中不存在的主观臆造的"指令"。可见，这里所说的程序与高级语言编的程序有很大差别，高级语言编程用的语句绝大多数是不随微机型号而改变的，而这里所说的编程序用的指令则完全依赖于微处理器的型号。高级语言易于学习掌握，因为其语句形式很接近自然语言，而这里所

说的指令所指明的操作是计算机内的基本操作，只有那些具备计算机原理知识的人才能理解，所以也只能由这样的人编写程序。

直接用表示指令的二进制代码编程，称为用机器码语言编程。为了便于记忆和书写，每条指令的二进制代码可用一组字母或符号表示，用字母或符号表示的指令编程称为用汇编语言编程。用汇编语言编的程序最终必须变换成机器码语言程序才能在计算机内执行。在计算机内，任何信息都必须以二进制代码形式存在。不难想像，高级语言程序中的任何一条语句的功能都是靠若干条指令的程序段完成的。

(4) 机器码指令排出的程序在准备执行时，必须存贮于存贮器中。程序存贮于存贮器中，而不是临时由人工把一条条指令输入计算机，这一点成为计算机所以有很强功能的关键之一。这是因为以电子速度从存贮器中取出指令要比人工输入指令快得很多很多。大多数计算器没有存贮程序的功能，在计算一个长的算术式时，要人工一步一步打入"指令"，这就是计算器与计算机的重要区别之一。

2. 计算机的基本操作过程

建立了指令和程序的概念之后，不难总结出计算机的基本工作方法。计算机的工作就是运行程序，未运行程序的计算机就是未工作、"未上班"的计算机。在正常情况下，一台微机的电源一接通，就开始运行某种程序，这是应该牢牢记住的概念。所谓运行程序就是这样一个连续的过程：逐条地从存贮器中取出程序中的指令并执行指令规定的操作。

现在回顾前面讨论的微型机硬件组成，不难看出计算机的硬件是实现上述过程的基础。存贮器既存贮程序又存贮数据；微处理器逐条地从存贮器中取出程序中的指令码，把指令码变换成控制信号序列，并把控制信号发向有关的部件，控制完成指令规定的操作；指令的操作可能是某种运算，也可能是从存贮器的数据区取来某个数据或向某个存贮单元存入一个数据，也可能是与某个外部设备之间传送数据。

在将要结束本节讨论的时候，我们就微处理器、微型计算机的"微"字作些介绍。

首先，微处理器、微型计算机的"微"字来源于微电子学的微字。集成电路技术是微电子学的核心，微处理器是超大规模集成电路(VLSI)，是微电子学发展的结果，微型计算机是以微处理器为核心的计算机，其体积小。

其次，从计算机分类来说，计算机的出现比集成电路早得多。早在复杂的微处理器出现之前，根据功能的强弱和体积的大小，就有大型、中型和小型计算机之分。大约 70 年代后期微型计算机出现，体积比小型计算机小得多，但这并不意味着微型机的功能也比小型机弱，今天的 IBM PC 微型计算机比 70 年代的中型计算机的功能还要强。

科学技术的发展并不受人为的定义的限制，微电子学集成电路技术不可能仅限制用于生产微处理器和微型计算机。新的大型、中型和小型计算机中采用了大规模和超大规模集成电路器件后，功能已经远远超过了 70 年代的大型计算机。

IBM PC，其 IBM 是 IBM 公司的缩写，而 PC 就是个人计算机的意思，前几年，微型计算机前常加"个人"两字。"个人"是与"多人公用"意思相对立的。微型计算机以前的计算机一般是安装在专用机房、有多个终端、供多个用户使用的，自然价格昂贵。而个人微型机一般是放在普通的办公室或家庭，同时只一个人操作使用，由于价格便宜很多个人都可拥有。但是，还是那句话，科学技术的发展并不受人为定义的限制。一台带有多个终端、同时多个人使用、运行多道程序的微型计算机已经出现，而且价格也并不昂贵。微型

计算机不一定是"个人"计算机了。

1.1.2 如何理解软件是微型计算机系统的组成部分

前面已说过，软件是程序。但是，这样说并没有表示出软件的重要性，也没有说明为什么把软件看作是微型计算机系统的组成部分。执行由指令组成的程序是计算机的工作，没有执行程序的计算机就是未工作的计算机。或者从另一意义上说，计算机的任何功能，看起来很简单，例如从键盘上输入一个字符，在 CRT 显示器上显示一个字符或在打印机上打印一个字符等，如果用基本指令排出程序来都需要有数十条或数百条指令，都是比较复杂的。用户不希望使用这样的计算机，因为太难使用了。

为了使计算机强大的功能潜力得以发挥，为了使计算机用户感到使用容易和方便，许多从事计算机科学的专业人员编出了一些程序，计算机借助运行这些程序的功能来"接待"用户。由于这些程序是专门为用户使用计算机编制的，它们的功能大大方便了用户对计算机的使用。大家熟悉的 BASIC 语言程序就是这类程序的一个例子。BASIC 语言解释程序是专业人员编制的。在运行 BASIC 语言解释程序时，计算机"接待"用户并不再要求用户用指令编程，而是用 BASIC 语言的语句编程。它等待并接受用户从键盘输入的 BASIC 语言的语句，可以修改这些语句，可以把程序的所有语句作为一个文件存入磁盘中，可以从磁盘中读出存入的程序，而且可以运行编好的 BASIC 语言程序。可不可以说由于 BASIC 语言解释程序"上班工作"，使用户感到面对的不再是以指令编程的计算机，而是另外一台以 BASIC 语言编程的计算机呢？从使用的意义上可以这么说。

在日常生活中也能找到可以类比的例子。一个青年克服了原来的缺点，学习了很多知识，增长了工作的才干，人们会评论说"他变了，完全变成了另外一个人"。实际上他的身躯并没有变。但人们这时评论他，是以人品和才干的尺度去衡量他，人品和才干是人与人交往时人的特征的主要方面。

对于计算机的评论也是这样，从用户使用的立场出发，代表计算机特征的是它的功能。由于专业技术人员编制的具有特定功能的程序在计算机上运行时(或者说，计算机在这个程序支持之下)，使计算机有了新的功能，人们有理由说它变成了另一台计算机。这就是软件的基本概念。软件是程序，而且主要指那些由专业人员编制的，在计算机上运行时，增强了计算机功能的程序。

组成计算机的硬件可以影响计算机的功能；同样，计算机所具有的软件也可以影响计算机的功能。从对计算机功能影响的意义上，硬件和软件的作用是相同的。而且，现今市场销售的计算机系统，没有一台是不带任何软件的"计算机"。所以，软件是组成微型计算机系统不可缺少的组成部分。

计算机的软件可以分成两大类：系统软件和用户(或应用)软件。系统软件是这样的一些程序，计算机在运行这些程序时为其它程序的开发、调试、运行等建立一个良好的环境：能方便地输入程序、做好执行前的准备处理以及可靠地运行程序。系统软件一般是计算机厂商提供的。用户软件是系统的用户为解决自己的特定问题的需要而设计的程序或购买的程序。

操作系统(Operating System)是最基本的系统软件。现代的微型计算机系统都是这样

的：电源一接通，就要使操作系统处于运行状态，操作系统总是第一个"上班工作"。只有操作系统"上班工作"，才能"接待"其它软件或用户的命令操作。所以，操作系统是用户在使用计算机时最先"打交道"的软件，是人和计算机的接口或界面(Interface)。下一节我们将进一步介绍操作系统。

下列程序也属于系统软件，它们都必须在操作系统支持下运行。

高级语言的编译程序(Compiler)或解释程序(Interpreter)。这是在计算机算法语言课中已经遇到过的。BASIC 语言可以在 BASIC 解释程序下输入程序和边解释边执行，也可以像 FORTRAN 或其它高级语言那样，先经过编译和连接，然后在机器上运行。

文本编辑程序(Text Editor)。运行这种程序时，用户可以通过键盘输入用各种语言写的源程序，对源程序进行编辑修改，把源程序作为文件存贮在磁盘上。在算法语言课上机实习时，已经用过这类软件。本书附录 D 介绍的 EDLIN 程序是 IBM PC 机操作系统支持的文本编辑程序，是上机实习要用到的软件。

连接程序(Linker)。在用各种语言编源程序时，用户可能把一个完整的程序任务分成多个模块进行编写，并对各个模块分别编译产生各自的目的模块。连接程序的任务是把目的模块连接成可执行的程序文件。附录 F 介绍的 LINK 程序是 IBM PC 机操作系统支持的连接程序，也是上机实习要用到的软件。

调试程序(Debug)。它又称为查错程序，是汇编语言编程不可缺少的工具。运行调试程序可以检查出被调试程序在逻辑功能上的错误。附录 G 给出的 DEBUG 程序是 IBM PC 机操作系统支持的调试程序，要特别重视这一软件的作用。

还有些程序属于系统软件，随着软件资源的不断丰富，还会出现新的系统软件，这里不作更多讨论。

1.2 磁盘操作系统 DOS 的基本功能

如果在学习算法语言课时曾在苹果机或 IBM PC 机上实习过，那就已经使用过磁盘操作系统 DOS 了。磁盘操作系统 DOS(Disk Operating System)是一组程序，它存贮在磁盘上，是在购买计算机时随计算机带来的。每次加电启动机器时，总是首先把磁盘上的 DOS 程序装入内存贮器并使 DOS 程序运行，然后计算机才能接受人在键盘上输入的命令或运行其它程序。所以，DOS 是使用计算机时首先接触的软件，其它任何软件都必须在 DOS 的直接或间接支持下运行。DOS 是人和计算机硬件联系的纽带。用计算机术语说，DOS 是人与计算机之间的基本接口。

本节介绍 IBM PC 机磁盘操作系统的基本功能。除了理解操作系统的基本功能外，还有两个目的：第一，在本课程开始时就理解系统软件的重要性；第二，尽早具有在机上操作的基本知识，为上机实习打下基础。

1.2.1 磁盘结构和文件介绍

因为一般情况下操作系统所包括的程序是存贮在软磁盘上的，而且我们自己编的程序和其它信息也需要存贮于软磁盘上。因此，有必要先介绍一下软磁盘是如何存贮信息的。

软磁盘的结构如图 1.2.1 所示，看上去像一张小的唱片，有磁性物质涂在薄圆盘的表面上。向磁盘写入信息或从磁盘读出信息都是在磁盘以圆心为轴转动情况下进行的，所以信息是按称为磁道的同心圆存贮的。在 IBM PC 系列机，操作系统 DOS 在 2.0 或 2.0 以上版本情况下，软磁盘分为两面，称为面 0 和面 1。每面有 40 个磁道，分别称磁道 0～39。每个磁道又分为 9 个扇区，每个扇区有 512 个字节(每字节 8 位)的位置。所以每个软盘的最大容量为 368 640 字节。在读写信息时，除磁盘面圆周转动外，还有磁头作径向移动，以便指向任意磁道的位置。信息通过磁头向磁盘上写和读。磁盘面的转动和磁头的径向移动以及读或写操作都是受程序控制进行的。图 1.2.1 上画的磁盘的保护外皮、磁头读写窗口、写保护缺口以及标牌等，对照实际磁盘看一看就清楚了。

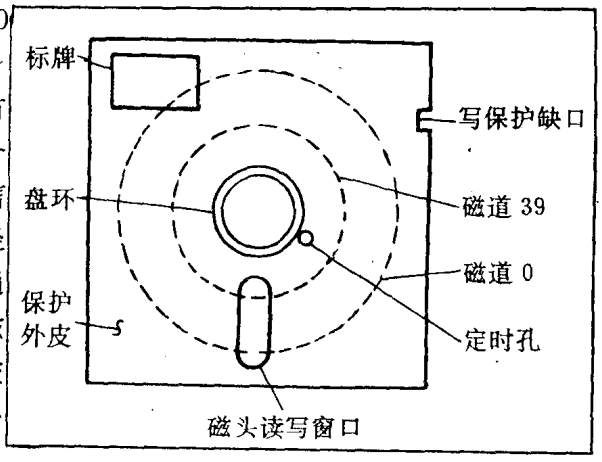


图 1.2.1 典型的软磁盘结构

图 1.2.1 所示的软磁盘可以置入磁盘驱动器，这时可以向磁盘上写入信息或从磁盘上读出信息，也可以更换另一张磁盘置入磁盘驱动器。所以，对微机系统而言，软磁盘提供的存贮容量是无限制的。还有一种磁盘是固定的，称为硬盘。硬盘的结构与软盘的结构是不同的，这里不准备详细介绍，但从信息读写的意义上，可以认为硬盘只是比软盘有更多的磁道，有更多的存贮容量。硬盘的存贮容量高达 10～200 MB。操作系统的程序也可以从软盘复制到硬盘上，这时微机可用硬盘上的操作系统启动工作。

文件(file)是一个重要的术语，信息是以文件为整体存贮在磁盘上的。每个文件都以自己的文件名作为文件的标志，包括若干相关的信息。文件在存贮时命名，根据文件名去查找和读出。

磁盘上的文件可以分为两类：程序文件和数据文件。程序文件存贮的是程序，可能是输入的源程序、目的程序(高级语言源程序经过编译产生的)或可执行的程序。程序文件都是由属于语言处理的系统软件处理形成和存贮的。例如，用 FORTRAN 语言编程时，借助前面介绍的文本编辑程序输入源程序并在磁盘上建立源程序文件；借助 FORTRAN 编译程序把源程序编译成目的程序并在磁盘上建立目的程序文件；借助于连接程序把目的程序连接成可执行程序并在磁盘上建立可执行的程序文件。数据文件可以借助文本编辑程序建立。但一般情况下，数据文件是在应用程序运行中由程序控制建立和产生的。这就是说，程序可以把某些数据组合起来起一个文件名字并存贮于磁盘中。同样，程序也可以按名字查找磁盘上的文件、读出文件和修改文件。

文件名是由字符(包括字母、数字和某些符号)组成的。完整的文件名包括两部分：开头是基本部分，字符数最少 1 个，最多 8 个；后面部分称为扩展名，字符数为 1～3 个。程序文件的扩展名是必须有的，而且遵循确定的约定，以后会讲到。但数据文件的扩展名不是必须有的，可有可无(可有可无的情况以后还会经常遇到，称为可选的，这已成为通用的术语)。基本名和扩展名之间要有一个黑点。选择文件名的字符组时，应尽量以某种关系表示文件的特征，以便记忆。

由于一台微机常常不只有一个磁盘驱动器，所以在读、写磁盘文件时，可在文件名前置入磁盘驱动器的标志字符 A:、B: 或 C:。例如:

A: TEMP.ASM

B: TEMP.EXE

C: WANG.DAT

磁盘上存贮文件采用树形目录结构。这里需要理解“目录”和“树形”两个词的含义。任何文件的文件名都成为目录中的一项，就是说，目录是文件的目录。“树形”是指目录是多层次的，如同树一样。树有根，根上有枝，枝上可以再有枝也可以有叶。与此相似，如图 1.2.2 那样，文件在磁盘里的组织有根目录，根目录中的各项是名字，这名字可能是文件名(相当于树叶)，也可能是子目录名字(相当于树枝)。同样，子目录中的项可能是文件名和更低层的子目录名。图 1.2.2 中以 [ ] 表示文件，以 [| |] 表示子目录。

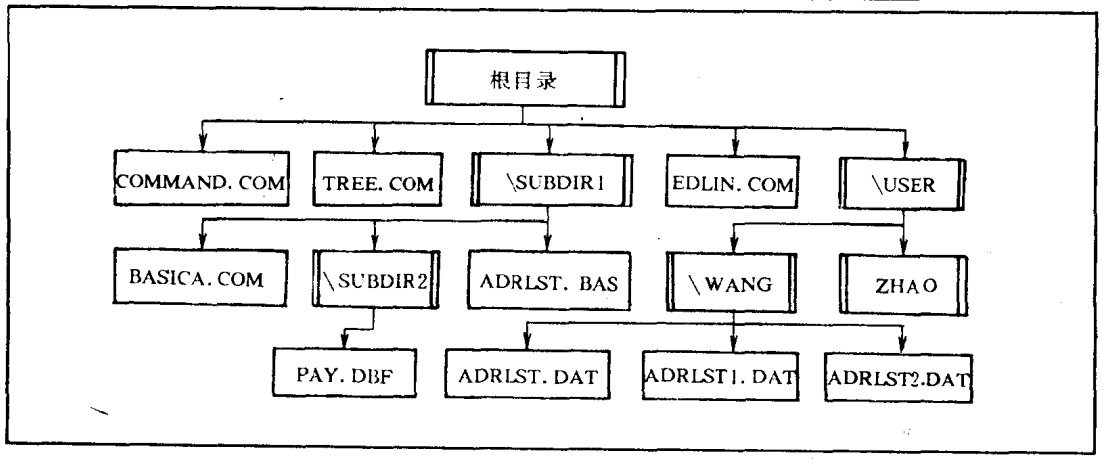


图 1.2.2 磁盘文件树形目录结构

对于这种树形目录结构，需要引入路径(path)概念。为了对文件进行操作，例如建立一个文件或查找一个文件，不仅要知道文件所在的磁盘驱动器的盘符和文件名，还要知道文件所在的目录名，甚至是多级目录名。如果该文件就在当前目录(当前目录的含义后面说明)中，则仅指出文件名即可，这时操作系统将自动在当前目录中查找该文件。如果文件不在当前目录中，则必须指出从当前目录(或根目录)到文件所在目录所经过的所有目录名，这就是路径。

以书面表示或在键盘上输入时，路径用由反斜杠隔开的一组目录名表示。有两种情况：一种是从根目录开始到文件所在目录的路径，称为绝对路径，表示时以反斜杠“\”开始。例如:

\SUBDIR1

\SUBDIR1\SUBDIR2

\SUBDIR1\SUBDIR2\PAY.DBF

另一种是从当前目录开始到文件所在目录的路径，称为相对路径，表示时不以反斜杠开始。例如:

PAY.DBF

..\SUBDIR2\PAY.DBF

第一个例子表示文件 PAY.DBF 就在当前目录。第二个例子中，“..”表示当前目录的上级目录。这时，文件 PAY.DBF 不在当前目录中，而在 SUBDIR2 子目录中。从当前目录到文件 PAY.DBF，要先回到当前目录的上级目录，SUBDIR2 是这个上级目录中的一个子目录。

当系统初始启动之后，当前目录即为根目录。此后，通过使用“改变当前目录”命令(CD)，可使当前目录改变为指定目录。

这样一来，无论为什么目的要表示一个文件时，都可用一组完整的文件表示字符组(file specification 或简写为 filespec)表示。这种字符组包括下列部分：

[d:][path\]filename

其中：d: ——磁盘驱动器符；

path——[\][dirname[\dirname[...]]

dirname——子目录名；

filename——fname[.ext]；

fname——文件名基本部分；

ext——文件名扩展部分。

文件表示字符组是本节讨论磁盘文件得到的重要结果，很多 DOS 命令涉及磁盘文件时，都要求给出文件表示字符组，所以应记住它的格式。

顺便介绍三个常用的目录管理 DOS 命令。

(1) 建立子目录命令 MKDIR(或简写为 MD)。MD 命令可以在根目录建立一个子目录，也可以在子目录下再建一个子目录。例如打入命令

MD C:\Sales

将在硬盘 C 上建立一个 Sales 子目录。若打入命令

MD C:\Sales\Wan

将在子目录 Sales 下再建一个 Wan 子目录。

(2) 删除子目录命令 RMDIR(或简写为 RD)。每次只能删除一个子目录，且这个子目录必须是空时才能删除。根目录和当前目录不能删除。例如打入命令

RD C:\Sales\Wan

将把子目录 Wan 删除，条件是子目录 Wan 是空的，而且不是当前目录。

(3) 改变当前目录命令 CHDIR(或简写为 CD)。这个命令用来告诉 DOS 把哪个目录路径记作为当前目录。假设当前目录是根目录，打入命令

CD C:\Sales

之后，C 盘上的 Sales 子目录即为 DOS 认定的当前目录。在没有新的 CD 命令时，Sales 一直是当前目录。打入命令

CD\

将把根目录恢复为当前目录。