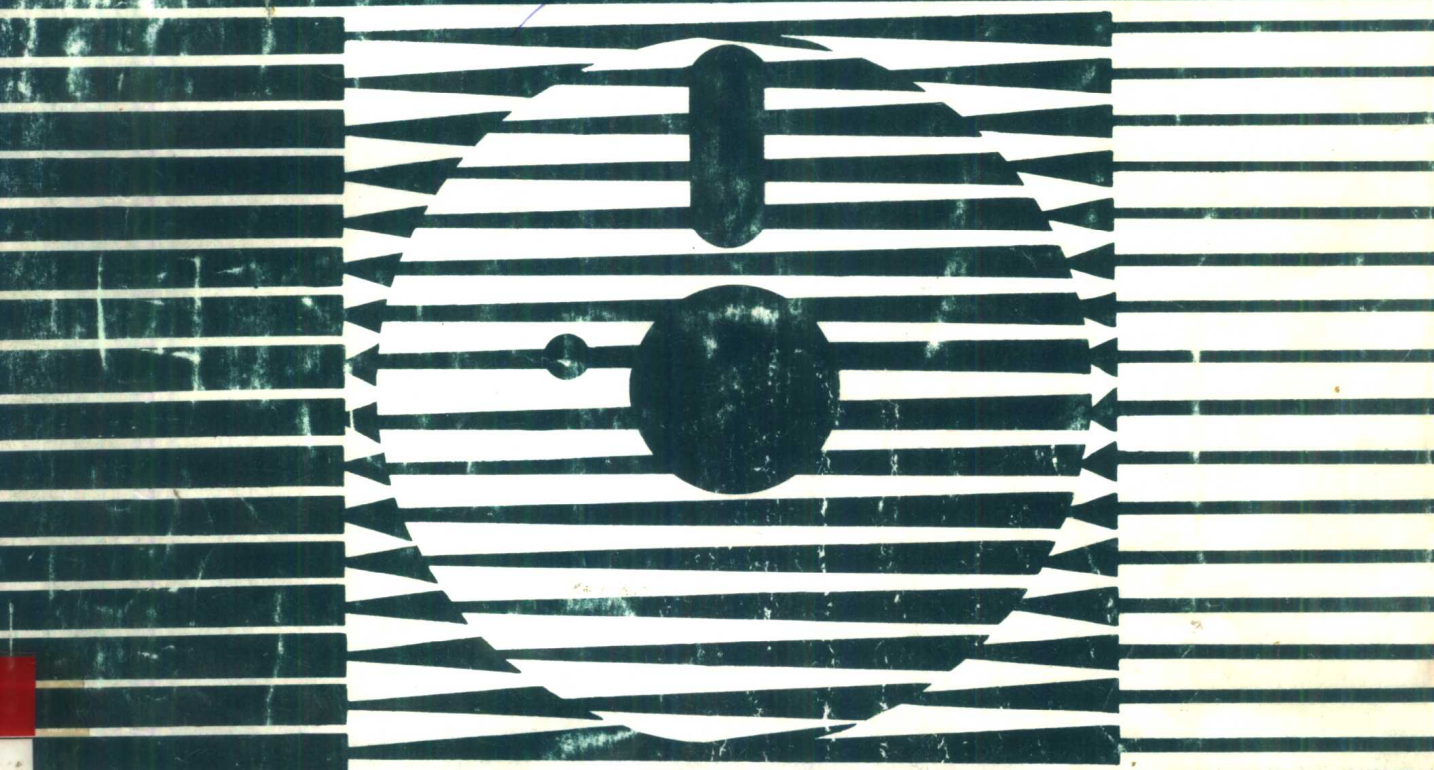


TRUE BASIC

程序设计教程

李世民 潘晓今 杨卫东 刘华红 编著



北京工业大学出版社

True BASIC

程序设计教程

李世民 潘晓今
杨卫东 刘华红 编著

北京工业大学出版社

1990

内 容 简 介

本教程包括 11 章和 7 个附录, 系统全面地介绍 True BASIC 及其结构化程序设计。除基本的程序设计内容如输入和输出、判断和循环、数组和矩阵、子程序和数组外, 还包括递归技术、大型程序模块化设计、“菜单”驱动、交互式动画、文件处理等高级程序设计的概念与技巧, 以及常用的典型算法、汉字应用、程序调试等实用技术。本书在最新推出的 True BASIC 2.03 版本基础上写成, 包括了 1.00 版本所没有的许多新功能和特性。

本书结合教学特点, 叙述由浅入深, 循序渐进; 书中包括大量新颖的程序实例和习题, 便于学习和思考。本书既可作为大专院校计算机基础教学的教材, 也可作为计算机技术人员和爱好者进行 True BASIC 应用程序设计的实用参考书, 还可供工程技术人员和管理干部自学使用。

True BASIC 程序设计教程

李世民 潘晓今 杨卫东 刘华红 编著

特约编辑 罗春元 刘宜河 罗献林

北京工业大学出版社出版

新华书店北京发行所发行

中国有色金属工业总公司 印刷
矿产地质研究院印刷厂

开本 787×1092 1/16 印张 14.6 字数 372 800

1990 年 12 月第 1 版 1990 年 12 月第 1 次印刷

印数: 0001~4500 册

ISBN 7-5639-0119-1/TP·5

定价: 7.00 元

前 言

计算机是当代最重要的科学技术成果之一，它对社会的发展起着越来越重要的作用。计算机软件作为知识产品，是现代信息社会的重要物质基础。计算机作为智力工具，它在教育事业中占有特殊的位置。

随着计算机应用与教育的深入，从计算机软件生产和教学中发展起来的软件规范化、程序设计工程化的技术思想，已经为广大计算机工作者和教师普遍接受和积极推崇。Pascal 程序设计语言作为结构化程序设计语言的代表，在计算机专业教育中已经确立了它的重要地位。但是，对于更为广大的非计算机专业的计算机程序设计教学而言，选择一种既与已经广泛应用的程序设计语言相联系，吸取它们的优点，又符合结构化设计理论要求的新的程序设计语言作为学习的对象，当是最恰当的。由美国计算机专家 John G.Kemeny 和 Thomas E.Kurtz 严格按照美国国家标准局制定的 BASIC 标准设计，于 1985 年发表的《True BASIC》，即是这样一种理想的程序设计语言。它为具有极其广泛应用基础的 BASIC 开辟了新纪元。

True BASIC 保持了传统 BASIC 简明灵活的特点，并吸收了 FORTRAN、Pascal 等语言的长处，具有程序模块结构清晰，语法表达自然等优点。不但程序的解释执行与程序的编译结合得很好，而且程序的编辑与连接也十分方便。因此，无论是在工程科学计算、数据处理还是图形绘制方面，都能很好地满足计算机程序设计基础教学，以及科学技术和管理工作的需要。

True BASIC 于 1986 年初在我国一出现，就引起了我们的重视，很快即引入我院的计算机基础教学。1987 年初，为给桂林冶金地质学院地质矿产勘查专业硕士研究生讲授程序设计课程，由李世民编写了《True BASIC 程序设计》简明讲义。之后，经李世民、潘晓今、杨卫东等反复修改、补充，继续在研究生和本科生各专业的“程序设计”教学中试用了 3 年，并先后作了 4 次修订重印，逐步形成了一套以结构化设计方法结合典型算法为特点的《True BASIC 程序设计》教材，内容和程序例题方面体现了 True BASIC 最新版本的性能。本教程即为我们多年来所用教材全面系统整理的结果。

本书内容涉及面较广。除基本的程序设计内容，如：输入、处理、输出、判断和循环、数组和矩阵、子程序和函数过程外，还叙述了一些具有启发性的程序模块化设计、“菜单”驱动、交互式动画、声音音乐处理、文件处理、递归技术等高级程序设计的概念与技巧，以及常用的典型算法、汉字应用、程序调试等实用技术。

本书共分 11 章和 7 个附录。第 1 章由李世民、潘晓今编写，第 2 章由刘华红编写，第 3~8 章由李世民编写，第 9、第 10 章由潘晓今编写，第 11 章及附录 1~3 由杨卫东编写，附录 4~7 及全书的习题由刘华红编写，最后由李世民统修定稿。

书中带 * 号的章节, 例如模块、图画、记录文件、字节文件及一些例题等, 有一定的难度, 但对高级程序设计有帮助。当教学时数偏紧或对于初学者, 可以跳过这些内容, 亦可将其作为课程设计的参考材料。

中南工业大学陈松乔教授详细审阅了书稿并提出了宝贵意见, 在此谨表诚挚谢意! 对支持和组织本书出版的桂林冶金地质学院教务处、学报编辑部、教材科、计算机教研室等部门表示衷心的感谢; 对帮助整理书稿的尹晟等老师表示谢意。

由于编著者水平所限, 书中难免存在缺点错误, 敬请专家和广大读者指正。

编 著 者

于桂林冶金地质学院

1990 年 7 月

目 录

前 言	
第 1 章 电子计算机概述	
1.1 计算机的发展和应用	(1)
1.2 计算机基本结构及其功能	(2)
1.3 计算机软件的概念	(4)
1.4 程序设计的概念	(6)
习题 1	(9)
第 2 章 True BASIC 初步	
2.1 True BASIC 的基本语句与程序结构	(10)
2.2 True BASIC 的基本操作	(14)
习题 2	(18)
第 3 章 量与赋值	
3.1 数值常量与字符串常量	(19)
3.2 简单变量与下标变量	(21)
3.3 赋 值	(23)
习题 3	(26)
第 4 章 函数和表达式	
4.1 系统函数	(28)
4.2 表达式	(36)
习题 4	(39)
第 5 章 输入和输出	
5.1 INPUT 语句	(40)
5.2 PRINT 语句	(42)
5.3 READ / DATA 语句与 RESTORE 语句	(46)
习题 5	(49)
第 6 章 判断结构	
6.1 算法及其逻辑框图表示	(51)
6.2 IF 型判断结构	(52)
6.3 SELECT CASE 判断结构	(56)
6.4 程序举例	(58)
习题 6	(62)
第 7 章 循环结构	
7.1 FOR / NEXT 循环和 EXIT FOR 语句	(63)
7.2 DO / LOOP 循环和 EXIT DO 语句	(67)
7.3 程序举例	(70)
习题 7	(75)
第 8 章 数组与矩阵	
8.1 数组说明与数组遍历	(77)

8.2	数组的应用	(82)
8.3	矩阵语句	(86)
8.4	内部数组和矩阵函数	(93)
8.5	程序举例	(97)
	习题 8	(105)
第 9 章 子程序、函数和库		
9.1	子程序	(107)
9.2	函数	(115)
9.3	库	(121)
9.4	递归	(123)
* 9.5	公用变量、共享变量和模块	(126)
9.6	程序举例	(132)
	习题 9	(138)
第 10 章 图形		
10.1	屏幕模式与窗口设置	(140)
10.2	基本绘图语句	(142)
10.3	图形着色	(147)
10.4	BOX 语句与动画设计	(152)
* 10.5	图画	(154)
* 10.6	坐标输入、单键输入和多重窗口	(158)
* 10.7	声音与音乐	(164)
* 10.8	程序举例	(166)
	习题 10	(182)
第 11 章 文件		
11.1	概述	(184)
11.2	文件的公共操作	(184)
11.3	正文文件	(187)
* 11.4	记录文件	(193)
* 11.5	字节文件	(197)
11.6	打印机的使用	(198)
11.7	高级编辑	(199)
	习题 11	(201)
附录 1	出错处理	(203)
附录 2	程序的设计与调试技术	(206)
附录 3	True BASIC 的运行环境	(209)
附录 4	命令总结	(212)
附录 5	语法补充	(216)
附录 6	出错信息	(220)
附录 7	ASCII 码表及 GET KEY 键码表	(225)
参考文献		

第1章 电子计算机概述

本章介绍计算机的工作原理、计算机的硬件和软件、计算机系统组成及程序设计概念等内容,为更好地掌握程序设计技术和进一步学习计算机科学打下基础。

1.1 计算机的发展和应用

1.1.1 计算机发展概况

现代科学技术的进步,于1946年诞生了世界上第一台电子计算机--ENIAC,从而使计算工具发生了质的跃变。40多年来,电子计算机技术性能的不断提 高,应用范围的不断扩大,极大地推动了社会生产、社会管理、社会服务各个领域 中科学技术的迅速发展,使计算机科学成为科技进步的带头学科和先导技术。它正在促使今天的世界由工业化社会向信息化社会转化。

自第一台电子计算机诞生以来的40多年中,由于对计算机结构原理的研究不断发展和微电子技术不断进步,使电子计算机大约每10年就发生一次换代更新:

1946年~1956年——电子管计算机;

1957年~1963年——晶体管计算机;

1964年~1974年——集成电路计算机;

1975年以后——大规模集成电路计算机。

每次换代,计算机的性能都比上一代优越,运算速度成十倍地提高,而体积迅速缩小,成本大幅度下降。目前,计算机正向两极发展:计算机的微型化,促进了计算机应用的普及与社会化;而计算机的巨型化,则促进了尖端科技的研究与发展。计算机的发展水平已经成为一个国家应用技术水平和科学技术发达程度的重要标志。

1981年10月在日本东京召开的第五代计算机国际会议的情况表明,第五代计算机将是超大容量、超高速、具有新的结构体系的智能化计算机。这一领域已经成为世界科技竞争的主要目标之一。

1.1.2 计算机的应用

由于计算机具有高速度运算、大容量存贮、能进行逻辑判断和自动运行等特点,今天电子计算机的应用已经遍及到社会生活的各个领域。

(1) 科学计算 高速度、高精度的数值计算是计算机在科学、工程、技术方面最基本和经常的应用。数学、物理、天文、地理、气象、水文、土木工程中许多繁杂的运算工作只有电子计算机才能胜任。

(2) 数据处理与信息加工 对航天技术、地质勘测、银行帐目、资源管理、人口统计、网络通讯、情报资料检索中大批量的数据传递、分析和处理,高速度、大容量计算机可以应付自如。

(3) 自动控制 卫星、导弹、飞船、自动生产线和复杂的空中及地上交通管理,都需要实时的自动控制。计算机在这个领域中的应用,使其技术质量与效率显著提高。

(4) 计算机辅助设计和辅助教育 CAD(计算机辅助设计)用于电子、机械、航空、造船、化工、建筑等部门,使得复杂产品的设计过程能自动产生产品的造型,分析产品的性

能, 对设计方案作出选择与优化。因而可以提高设计质量, 缩短设计周期, 使设计标准化。而 CAI (计算机辅助测试) 在学习和培训方面的应用, 可以有效地提高教育质量和水平。

(5) 人工智能 其研究的领域包括: 模式识别、语言识别、定理自动证明、自动程序设计、专家系统和机器人等。目前最有代表性和实用意义的是专家系统和机器人。

另外, 办公自动化和计算机艺术造型, 也是近年来迅速发展的计算机应用领域。

1.2 计算机基本结构及其功能

现今使用的计算机绝大部分都是冯·诺伊曼式计算机 (Von Neumann 是 20 世纪著名数学家)。这种计算机主要由 3 类功能不同但相互关联的部件构成: 存贮器 (Memory); 中央处理器 (CPU); 输入/输出设备 (I/O Units)。

存贮器是保存信息并能方便地存取信息的装置; 中央处理器是对信息进行运算和处理的装置; 输入/输出设备是实现计算机与外部环境交换信息的装置。

冯·诺伊曼式计算机的基本结构如图 1-1 所示。常见微型计算机系统如图 1-2。

组成计算机的物理器件实体称为计算机的硬件, 硬件分为主机和外部设备两大类。主机主要由中央处理器和内部存贮器组成, 它是计算机的中心; 主机以外的各种设备 (如外部存贮器、输入设备、输出设备等) 统称为外部设备。

计算机的整体性能是由主机和各外部设备的性能决定的。计算机的基本性能包括运算速度、内存容量、外存容量、输入/输出设备的数量和种类等。

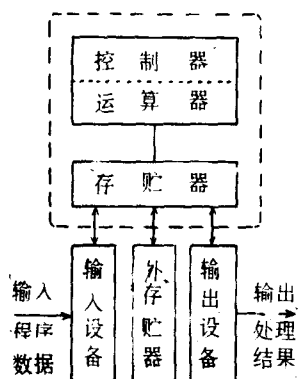


图 1-1 计算机基本结构

1.2.1 存贮器

利用存贮器存取各种信息是现代计算机的重要特征之一。为了既方便又大量地存贮和使用信息, 计算机的存贮器分为外部存贮器 (简称外存或辅存) 和内部存贮器 (简称内存或主存) 两类。

外部存贮器是独立的外部设备之一, 它的主要作用是长期、大量地保存信息。各种备用的程序和数据通常保存在外部存贮器中。计算机工作时, 可以根据需要随时将这些程序和数据调入内存使用。目前常用的外部存贮器包括磁盘机、磁带机、光盘机等。

内部存贮器位于主机内部, 通常由能够高速存取信息的集成电路芯片组成。它与中央处理器紧密联接在一起。内部存贮器的特点是信息读写的速度极快。中央处理器利用它来存取当前工作过程中频繁使用的信息, 以提高计算机的工作效率。内部存贮器又细分为只读存贮器 (ROM) 和随机存贮器 (RAM) 两种。计算机启动和运行时必须使用但又不允许修改的信息通常被固化在 ROM 内; 而 RAM 中的信息则既可读出又可写入, 但是如果关闭计算机, RAM 中的信息亦将随之消失。

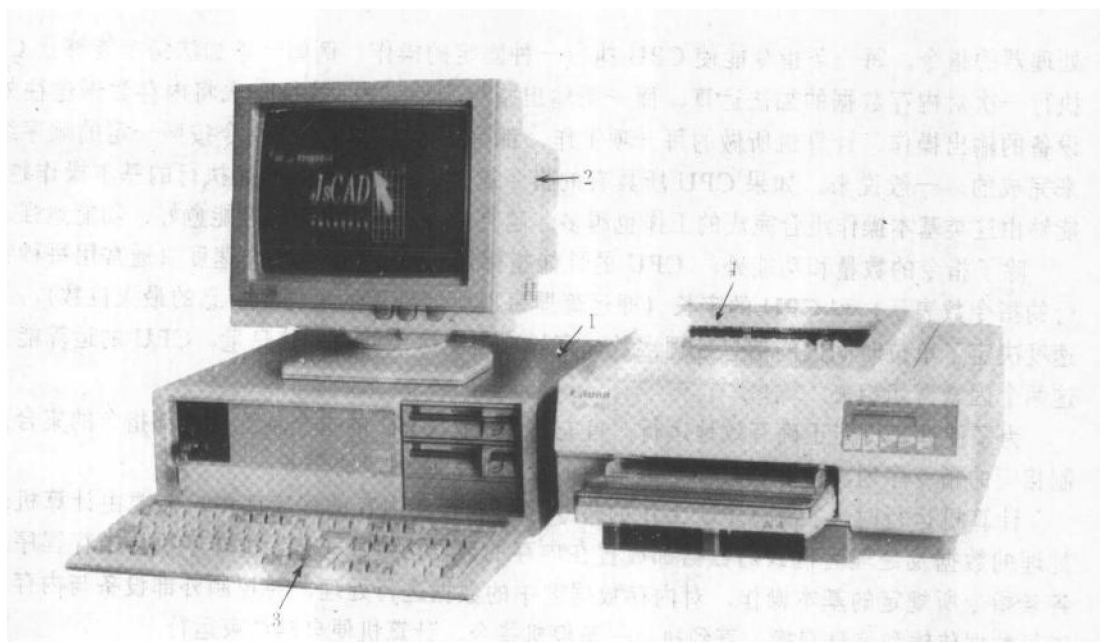


图 1-2 常见微型计算机系统

1-主机；2-显示器；3-键盘；4-打印机

为了能够迅速准确地存取大量信息，存贮器被划分成许多存贮单元。存贮单元具有如下性质：

- (1) 每个存贮单元都有一个唯一的编号，通常称为地址。
- (2) 每个存贮单元可容纳一个固定位数的二进制信息。二进制信息由二进制数表示。二进制数是由数码“0”和“1”构成的序列，它的每一位 (bit) 是计算机表示信息的最小单位。每 8 位二进制数为一个字节 (byte)，一个存贮单元中的信息可由多个字节组成。
- (3) 存贮单元中的内容一般既可读出又可写入。对存贮单元进行“读”操作时，仅拷贝其中内容，单元内容保持不变；而对存贮单元进行“写”操作时，是将该单元中原有的内容改写成新的内容，原有信息即被消除。

计算机处理信息，既要求较大的存贮容量，又要求快捷的存取速度。因此存贮容量和存取速度是衡量计算机存贮器性能的两项重要指标。存贮容量常用字节数表示，每 1024 个字节 (即 2^{10}) 简称为 1KB (即 1024 Byte)。存取速度用读 / 写 1 个字节信息的时间表示，时间单位为 ms (毫秒， 10^{-3} 秒)， μs (微秒， 10^{-6} 秒) 或 ns (毫微秒， 10^{-9} 秒)。

1.2.2 中央处理器

中央处理器 (简称为 CPU，即 Central Processing Unit) 是执行程序、实现信息处理的部件，由运算器 (ALU)、控制器 (CONT) 和若干个寄存器组成。CPU 位于主机内部，它是计算机的核心机构。其中，运算器的主要功能是完成算术或逻辑运算；控制器的作用则是根据程序指令的性质发出各种控制信号，对计算机的工作过程进行控制，以保证计算机的各个部件相互协调地完成对信息的处理；寄存器用于存放当前使用的运算信息和控制信息，它与内存相比具有更快的信息存取速度。

CPU 能够执行若干基本的操作。规定 CPU 执行这些基本操作的二进制信息，称为中央

处理器的指令。每一条指令能使 CPU 执行一种固定的操作。例如一条加法指令能够让 CPU 执行一次对内存数据的加法运算，而一条输出指令则使 CPU 执行一次将内存数据送往外部设备的输出操作。计算机所做的每一项工作，都是由这类基本操作指令按照一定的顺序组合来完成的。一般说来，如果 CPU 所具有的指令越多，说明 CPU 所能执行的基本操作越多，能够由这类基本操作组合完成的工作也越多、越复杂，因而 CPU 的性能愈好、功能愈强。

除了指令的数量和功能外，CPU 的性能指标还包括 CPU 的工作速度（通常用每秒钟执行的指令数表示）和 CPU 的字长（即运算器每次所能处理的二进制信息的最大位数）。工作速度决定了单位时间内的信息处理次数，字长决定了每次处理的信息量，CPU 的运算能力与这两个因素密切相关。

为了使计算机能正确有效地运行，首先需要根据 CPU 的指令系统（全部指令的集合）编制相应的指令序列，使计算机在这组指令的控制下完成预定的任务。

计算机运行时，控制 CPU 工作的指令序列放置在内存的程序区中；需要由计算机进行处理的数据及处理后得到的数据都放置在内存的数据区中。CPU 按顺序执行内存程序区中各条指令所规定的基本操作，对内存数据区中的数据进行处理，并控制外部设备与内存之间进行数据传输和信息交换。直到执行一条停机指令，计算机便自动结束运行。

1.2.3 输入/输出设备

输入设备和输出设备（简称为 I/O 设备，即 Input/Output 设备）是计算机与外部进行信息交换的装置。在 CPU 的控制下，输入设备能够接收来自计算机外部的各种形式的信息（如文字、数字、图形、声音、电讯号等等）。输入设备将这些信息转换成二进制形式，并传输到内存，以便由 CPU 进行处理。同样，输出设备能够在 CPU 的控制下，把计算机内部预先存有或经处理后得到的二进制形式的信息，还原或转换成所需要的信息形式（如文字、数字、图形、声音、电讯号等等），并传输到计算机外部。因此，I/O 设备是实现人及外部环境与计算机相互交流信息的装置。

I/O 设备的配置因计算机不同而异。一台计算机根据需要常常配有多台 I/O 设备。常用的输入设备有：键盘、鼠标器、光笔、数字化仪、图形文字扫描仪、摄像仪、音频输入装置等。常用的输出设备有：屏幕显示器、打印机、X-Y 绘图仪、投影仪、音频输出装置等。另外，计算机之间也能够通过 I/O 设备相互连接，形成一定范围的计算机通讯网络。I/O 设备的种类和性能，是决定该计算机应用范围和功能强弱的重要因素。

1.3 计算机软件的概念

1.3.1 指令与指令系统

计算机的工作过程，就是硬件执行预先设计好的指令序列的过程。一个完整的计算机系统，应当包括计算机硬件和能使硬件有效运行的计算机指令。

不同类型的中央处理器其指令系统是不相同的，但基本包括以下类型的指令：

数据传输指令——将数据从一个部件传输到另一部件；

算术逻辑指令——对数据进行算术或逻辑运算。数据的来源及处理后的存放位置既可是某寄存器，也可以是某存储单元；

程序转移指令——改变指令的执行顺序；

处理机控制指令——对中央处理器本身进行控制。

在每种类型的指令中，根据特定的操作要求又包含了许多具体指令，因此一台计算机往往包含几十或上百条指令。一台计算机所有指令的集合，称为该计算机的指令系统。

按照一定的规则组成，可完成规定功能的指令序列称为程序。构造该指令序列的过程称为程序设计。程序以及设计、使用和维护该程序的有关文件档案，统称为计算机软件。

1.3.2 软件的种类

计算机软件按用途可分为3类。

(1) 面向计算机管理和操作的软件 这类软件包括操作系统、分时系统、网络系统等。它们的作用是：当计算机工作时，自动对计算机的硬件设备及有关软件进行有效管理，使其进入正常的工作状态。例如，让计算机的键盘等待输入，以便接收计算机用户输入的信息；让计算机的显示器等待从主机传送过来的信息，以便及时输出到屏幕上；让作为外存使用的磁盘机等待中央处理器的控制信号，以便和内存交换信息等等。一台比较简单的计算机只需要一般的机器管理程序对其进行管理。而一台配置齐全、功能完善的计算机，就需要使用操作系统。

操作系统的主要功能是管理计算机硬件的中央处理机、存储器、外部设备以及外存储器上的软件等。它能最大限度地发挥硬件和软件的作用，使用户能够有效地利用和控制计算机。

对于连接了多个用户终端的计算机，分时系统能够让主机的工作时间合理地分配给各个终端使用。如果建立了计算机网络，那么在网络系统的支持下，便可以实现计算机之间的信息通讯。面向计算机管理和操作的软件与硬件的关系密切，是根据硬件的结构和工作方式来设计的。只有在这些系统软件的支持下，计算机才能够发挥效能，各种其他软件也才能够有效地运行。

(2) 面向计算机维护的软件 这类软件包括机器故障检查和诊断程序、自动纠错程序、硬件性能测试程序、软件测试程序等。在进行硬件或软件安装、调试、检测、修理时，或者是在计算机运行过程中，它们被用来检查硬件或软件故障，纠正由计算机故障所引起的运算错误。

(3) 面向用户的软件 这类软件包括计算机语言的处理程序、图形文字等各类信息数据的编辑程序、数据库管理程序、汇集了常用数学处理方法的数学库程序、解决各类应用问题的各种专用或通用的应用程序等。最基本的是各种计算机语言的处理程序。面向用户的软件使计算机的用户能够方便有效地进行程序设计，为用户提供各种运算处理功能，从而有利于解决各种计算机开发应用问题。

计算机语言处理程序包括编译程序和解释程序两种形式。编译程序的作用是将用计算机语言写的源程序全部翻译成计算机能够执行的机器指令代码目标程序。解释程序对源程序的翻译过程是边翻译边执行，不生成目标程序。目标程序运行速度较快，而解释方式有利于人机对话。有了编译程序或解释程序，用户便能够利用易于描述解题过程的计算机语言编制各种应用程序，由编译程序或解释程序将它们转换成计算机能够执行的指令，从而提高程序设计的效率。

各类软件中，有的软件用于保证计算机系统正常工作，发挥和提高计算机系统的运行效率，为计算机系统提供通用的服务功能，这类软件称为系统软件；有的软件则是在计算机应

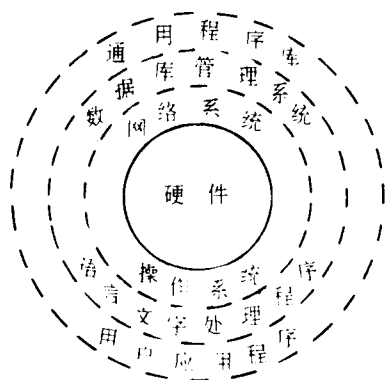


图 1-3 硬件—软件层次结构关系

用领域为解决某些具体的计算或信息处理问题而编制的，这样的软件称为应用软件。不同的计算机可以根据其工作性质和应用需要配置不同的系统软件和应用软件。

1.3.3 软件与硬件的关系

由图 1-3 可见，计算机系统内的硬件和软件以及软件的各部分之间是一种层次结构关系。软件是在硬件基础上对硬件性能的扩充和完善，软件之间则是外层软件需要内层软件为其提供运行的条件。在硬件外每加一层软件，计算机的功能就提高一层。通常把这种功能更强的计算机称为“逻辑机”。

1.4 程序设计的概念

1.4.1 基本概念

进行程序设计，就是用计算机能识别和理解的符号来描述计算机的运行步骤。程序设计的符号表达规则，称为计算机语言。计算机语言可以大致分为 3 个层次：机器语言，汇编语言，高级语言。在机器语言层次上，直接使用具体的二进制指令编程；在汇编语言层次上，使用易于理解和记忆的指令助记符号编程；在高级语言层次上，则使用接近于自然语言或某种工程语言的表达方式编程。

进行程序设计时，首先需要针对实际问题找出有效的解题方案，即确定解题的“算法”。在计算机科学中，算法往往用自然语言、数学公式、图形符号等易于理解的方式来描述。

算法设计是程序设计的关键。它涉及到合理使用计算机的各种运算操作、合理安排数据和程序在计算机存储器中的存贮等问题。

为了说明三种不同层次语言的程序设计，这里引入一台简单的模型计算机。

计算机的机器指令的一般格式为：

操作码 (O)

操作数 (D)

一条机器指令是一个二进制数，分为操作码和操作数两部份。其中，操作码 (O) 规定该指令的操作性质（即做什么），操作数 (D) 则给出参加运算的数或参加操作的数之内存地址。

假定这台模型计算机具有表 1-1 所示指令系统。其中，Acc 表示中央处理器中用于放数据和进行算术逻辑运算的寄存器（称为累加器），(Acc) 表示该寄存器中的内容；k 表示某个存贮单元的地址，(k) 表示该存贮单元中的内容。

1.4.2 计算机语言程序

设要求编写计算分段函数

$$y = \begin{cases} x^2 - 3, & (x \leq 10) \\ x + 2, & (x > 10) \end{cases}$$

的模型机器语言程序。

在分段函数的计算过程中， x 是输入的变量，2, 3, 10 是常量， y 是计算结果的输出量。

(1) 机器语言程序 用具体机器所能识别和执行的机器指令，去描述某个要计算机求解或处理的问题，这样的指令序列就是一个机器语言程序。

为了让计算机能自动按照分段函数的要求，根据输入的 x 求解 y 的数值，需要先将 3 个常数存入内存（例如，将它们分别存入内存的 00001, 00010, 00011 二进制单元），分配 00100 单元（4 号单元）给变量 x ，00101 单元（5 号单元）给变量 y 。然后，将按照分段函数的计算要求，用指令编出反映具体计算过程的机器语言程序，把它按指令顺序逐条逐条地存放在从 01010 起的若干个存贮单元之中（每个单元存放一条指令）。内存情况如表 1-2 所示。

(2) 汇编语言程序 由表 1-2 可见，即使是计算这样一个不太复杂的分段函数，用机器指令所编写的指令代码程序竟如此繁琐。全由“0”和“1”所表示的数据及程序，既容易弄错又不直观。要用这种形式求解一个复杂问题，会存在许多困难。为此，人们创造了汇编语言。汇编语言为计算机的机器语言指令的操作性质安排了助记符号，汇编语言采用这些助记符来写程序；存贮单元可用变量表示，数据可用十进制数表示。计算分段函数的汇编语言程序如下：

INP	x	;输入 x
LOA	x	;取 x 到 Acc
SUB	10	;作 $x-10$
CJP	L1	;如果 $(x-10) < 0$ 则转标号 L1
LOA	x	;取 x 到 Acc
ADD	2	;作 $x+2$
STO	Y	;将 $x+2$ 存入 Y
JMP	L2	;无条件转向标号 L2
L1:	LOA	x ;取 x 到 Acc
	MUL	x ;将 x 与 x 相乘
	SUB	3 ;从乘积中减去 3
	STO	Y ;将 x^2-3 存入 Y
L2:	OUT	Y ;输出 Y
	HLT	;停机

显然，汇编语言比机器语言前进了一步：指令符号比较容易记忆，数字、变量表示直观，有利于编制程序，程序的可读性明显地比机器语言的好。

但是，计算机并不能直接执行这种程序。它必须被翻译为机器语言程序后，方能由计算机执行。这个翻译工作就是由一个叫做“汇编语言编译器”的系统软件来完成的。汇编语言所写的源程序，经过“编译器”的翻译，产生一个机器语言程序（又称为目标程序），由计算机执行这个目标程序得到计算结果。

(3) 高级语言程序 汇编语言虽然比机器语言前进了一大步，但它与特定的机器和指令系统密切相关，程序的通用性较差（当然，这一点将随着硬件的标准化而不断改善）。而且程序设计中需要详细地了解机器的内部特性，才能写出有效的汇编语言程序。为了提高程序设计

表 1-1 一台模型计算机的指令系统

操作码	助记符	操作名	操作内容	指令类型
0001	ADD	加法	$Acc \leftarrow (Acc) + (K)$	算术逻辑指令
0010	SUB	减法	$Acc \leftarrow (Acc) - (K)$	算术逻辑指令
0011	MUL	乘法	$Acc \leftarrow (Acc) * (K)$	算术逻辑指令
0100	DIV	除法	$Acc \leftarrow (Acc) / (K)$	算术逻辑指令
0101	LOA	取数	$Acc \leftarrow (K)$	数据传输指令
0110	STO	存数	$K \leftarrow (Acc)$	数据传输指令
0111	JMP	无条件转移	转向第 K 单元	数据转移指令
1000				
1001				
1010	CJP	条件转移	若 $(Acc) \leq 0$ 则转 K, 否则转下一条指令	程序转移指令
1011	INP	输入	把输入设备上的数送入 K	数据传输指令
1100	OUT	输出	把 K 单元的内容输出外部	数据传输指令
1101	HLT	停机	停止计算结束程序	处理机控制指令

表 1-2 分段函数机器语言程序

地址	RAM 存贮情况	注 释
00001	00000 00010	常数 2
00010	00000 00011	常数 3
00011	00000 01010	常数 10
00100		存放从外设输入的变量 x 的数值
00101		存放分段函数的函数值 y
00110		
	(Θ) (D)	
01010	01011 00100	从外设输入 x 的值存入 4 号单元
01011	00101 00100	从 4 号单元取数到 Acc
01100	00010 00100	将 3 号单元中的数与 Acc 中的数相减 (即 $x-10$)
01101	01010 10010	如果 $(Acc) \leq 0$, 则转 18 号单元, 否则做下一指令
01110	00101 00100	从 4 号单元取数到 Acc (即 $Acc \leftarrow x$)
01111	00001 00001	将 1 号单元中的数与 Acc 中的数相加 (即 $Acc \leftarrow x+2$)
10000	00110 00101	将 Acc 中的数送入 5 号单元 (即 $Y \leftarrow x+2$)
10001	00111 10110	无条件转向第 22 号单元
10010	00101 00100	从 4 号单元取数送到 Acc (即 $Acc \leftarrow x$)
10011	00011 00100	将 4 号单元中的数与 Acc 中的数相乘 (即 $Acc \leftarrow x$)
10100	00010 00010	将 2 号单元中的数与 Acc 中的数相减 (即 $Acc \leftarrow x^2-3$)
10101	00110 00101	将 Acc 中的数送入 5 号单元 (即 $Y \leftarrow x^2-3$)
10110	01100 00101	向外设输出 5 号单元中的数
10111	01101 00000	停机
11000		

的效率, 便于程序的修改、调试, 人们进一步创造了高级程序设计语言。例如 50 年代的

FORTRAN, 60年代的ALGOL-60、BASIC、COBOL, 70年代的PASCAL、C, 80年代的True BASIC语言等等。

用True BASIC高级程序设计语言来描述上述分段函数的计算过程, 就十分简便:

INPUT x	!输入 x
IF x<10 then	!如果 x 小于等于 10 则
LET y=x*x-3	!按 y=x*x-3 计算
ELSE	!否则
LET v=x+2	!按 y=x+2 计算
END IF	!结束判断处理
PRINT y	!输出 Y
END	!结束程序

高级语言是独立于具体机器, 便于描述算法的程序设计语言。用高级语言所编写的程序与汇编语言程序一样, 机器也是不能直接执行的。但通过一个对应于该高级语言的编译程序或解释程序, 能够将其翻译成机器语言的目标程序, 再由机器执行。

由上述3种层次的程序设计可以看出, 机器语言程序和汇编语言程序是描述计算机内部的具体操作过程的指令程序, 因此必须根据具体计算机的指令系统和助记符来进行程序设计。而高级语言程序允许使用类似于自然语言和数学公式的语句序列来描述解题过程, 它与具体计算机的指令无关, 因此不仅便于描述复杂的算法, 而且程序不受具体机器的限制。只要该计算机配有相应高级语言的翻译软件, 就可以在指令系统不同的各种计算机上运行高级语言程序。

习 题 1

- (1) 计算机硬件由哪几个基本部件组成, 各自的主要功能是什么?
- (2) 试解释下列概念, 并简要说明其相互关系: ①软件, ②程序, ③算法。
- (3) 高级语言与机器语言有什么区别, 各有什么优缺点?
- (4) 解释程序和编译程序有何异同点, 它们在计算机系统中起什么作用?
- (5) 操作系统在计算机系统中起什么作用?

第 2 章 True BASIC 初步

BASIC 语言自 60 年代创立至今, 人们用它解决了大量实际问题, 已成为世界上使用得最广泛的高级程序设计语言之一。True BASIC 是 BASIC 的创始人于 1985 年发表的, 集其他高级语言之大成, 为计算机程序设计提供了又一有效的工具。它具有如下特点:

- (1) 结构功能完善, 语法表达自然, 程序编辑灵活;
- (2) 与 BASIC 兼容性良好, 在 IBM PC 上可使用全内存;
- (3) 系统函数丰富, 具有高级图形和动画设计功能;
- (4) 解释方式下工作简明方便, 编译方式与解释方式并存;
- (5) 便于移植, 便于与汇编语言程序连接;
- (6) 符合美国国家 BASIC 标准;
- (7) 支持汉字。

从这一章起将详细讲述 True BASIC 语言及其程序设计。本章主要介绍 True BASIC 程序的整体概貌及 True BASIC 的上机基本操作。

2.1 True BASIC 的基本语句与程序结构

True BASIC 程序由 True BASIC 语句构成, 每个 True BASIC 语句规定计算机完成一项特定的工作(或称为操作)。True BASIC 语句分为简单语句和结构控制语句两种。简单语句使计算机完成一项简明的操作任务, 而结构控制语句可以将简单语句组织成各种序列, 对程序的运行过程进行控制。

2.1.1 三个简单语句

一般情况下, 无论是简单程序或者是复杂程序, 都应具备输入数据、运算处理、输出结果这 3 项基本功能。在 True BASIC 语言中, 这 3 项基本功能可以运用以下 3 个简单语句来实现。

键盘输入语句: INPUT Var1, Var2, ..., VarN

运算赋值语句: LET Var1 = expr

屏幕输出语句: PRINT expr1, expr2, ..., exprN

其中, Vari 为变量, expri 为表达式(详见第 3、第 4 章)。

运用这 3 个最基本的 True BASIC 语句, 便能够进行初步的 True BASIC 程序设计。

下面是简单的 True BASIC 程序设计的例子。

例 1 从键盘上输入圆的半径值, 计算并显示圆的面积。

若用 R 表示圆的半径, S 表示圆的面积, 由几何定理有

$$S = \pi \times R \times R$$

根据此数学公式, 编制程序如下:

INPUT r	! 输入圆半径 R 的值
LET s = 3.14159 * r * r	! 计算圆面积的值并赋给变量 S
PRINT s	! 输出 S 的值
END	! 结束程序