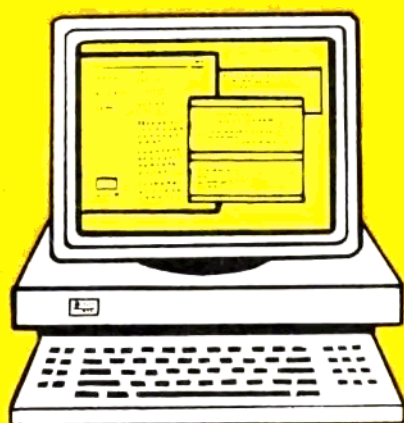


经济数据处理与微机COBOL

潘硕健 主 编

王小鸽 副主编



西 科 学 技 术 出 版 社

前 言

本书主要为高等院校财经类专业学生而编著。我们试图从实践环节着手,按照结构化程序设计思想,由浅入深地介绍一级 COBOL 的基本用法及其理论。书中大量实例是编著者从历年来自行开发的经济应用软件中精选而出,它们紧密联系实际经济业务。因而本书又可作为从事电子计算机在经济中应用工作同志的一本培训教材或参考书。

本书由 4 部分组成。第 1 章为第一部分,阐述微型电子计算机系统的一般构成和必要的程序设计基础知识。第 2~5 章为第二部分,结合经济实务阐述 COBOL 概况以及必须掌握的 COBOL 基本内容。第 6、7 章为第三部分,着重介绍常用的 COBOL 程序设计技巧与手段。第 8 章为第四部分,以中国人民银行某市分行电子同城票据清算系统为例,给出一个完整的数据处理系统的设计方法与步骤。

本书由潘硕健主编,王小鸽副主编。由潘硕健、王小鸽、袁放健、张成虎、应喆、赵黎明共同编写。分工如下:潘硕健编写第五章;王小鸽编写第二章、第三章、第四章前三节;张成虎编写第四章的第四节、第八章;袁放健编写第六章、第七章;应喆编写第一章;赵黎明编写习题与作业;四个附录由王小鸽从各种参考书中选取;在编写过程中,我们参考了大量已出版的同类书籍,在此向各位同仁致谢,作者姓名恕不一一列举。

由于时间紧迫,学识水平有限,书中不妥之处还望读者不吝赐教。

编 著 者

1991 年 5 月

目 录

第1章 电子计算机系统概述	(1)
第1.1节 电子计算机系统组成.....	(1)
1.1.1 硬件的基本组成	(1)
1.1.2 软件及其分类	(3)
*1.1.3 长城微机系统配置简介	(5)
第1.2节 程序设计语言与程序设计规范.....	(6)
1.2.1 机器语言与汇编语言	(6)
1.2.2 高级语言	(7)
1.2.3 流程图(GB1526-79)及其用法	(9)
1.2.4 程序设计规范	(9)
*1.2.5 常用的几种程序设计规范	(11)
第1.3节 数据处理.....	(13)
1.3.1 数据与信息	(13)
1.3.2 数据处理概念及其分类	(13)
1.3.3 经济数据处理	(15)
第1.4节 数据结构及其组织技术.....	(17)
1.4.1 数据结构	(17)
1.4.2 文件与文件管理系统	(19)
第2章 COBOL 语法概要	(23)
第2.1节 COBOL 的历史与现状.....	(23)
2.1.1 COBOL 的产生与发展	(23)
2.1.2 COBOL 的特点	(23)
2.1.3 COBOL 的功能模块划分及其用途	(24)
第2.2节 COBOL 构词法.....	(25)
2.2.1 COBOL 字符集	(25)
2.2.2 COBOL 字	(26)
2.2.3 COBOL 常量 (Literals)	(28)
2.2.4 字符串 (CHARACTER-STRING)	(29)
第2.3节 COBOL 语法规则及其格式约定.....	(29)
2.3.1 COBOL 语法规则	(29)
2.3.2 语法规则的格式约定	(31)
第2.4节 COBOL 源程序的基本结构与书写格式.....	(31)
2.4.1 COBOL 源程序的基本结构	(31)

2.4.2	COBOL 源程序的书写格式	(32)
第 3 章	COBOL 核心模块	(34)
第 3.1 节	标识部与环境部	(34)
3.1.1	一个简单的 COBOL 程序实例	(34)
3.1.2	COBOL 源程序的标识部	(36)
3.1.3	COBOL 源程序的环境部	(37)
第 3.2 节	过程部	(38)
3.2.1	过程部结构	(38)
3.2.2	一般的输入输出语句	(40)
3.2.3	算术语句	(42)
3.2.4	MOVE (传送) 语句	(47)
3.2.5	控制转移语句	(48)
3.2.6	IF (条件) 语句	(56)
第 3.3 节	数据部	(65)
3.3.1	数据部概述	(65)
3.3.2	工作存贮节	(66)
* 3.3.3	屏幕节	(74)
第 4 章	COBOL 顺序输入输出模块	(82)
第 4.1 节	磁盘顺序文件的形成	(82)
4.1.1	顺序文件与磁盘顺序文件	(82)
4.1.2	形成一个磁盘顺序文件	(83)
第 4.2 节	顺序文件更新处理	(97)
4.2.1	顺序文件更新的过程	(97)
4.2.2	顺序文件更新实例	(100)
第 4.3 节	顺序文件更新处理应注意的问题	(110)
4.3.1	顺序文件的排序	(110)
4.3.2	确定文件中记录格式的一般原则	(111)
4.3.3	原始数据的正确性检查及其方法	(112)
* 第 4.4 节	磁带顺序文件	(113)
4.4.1	磁带的特性及记录分块	(113)
4.4.2	COBOL 中有关磁带文件的语法成分	(115)
4.4.3	磁带文件的形成与更新	(118)
第 5 章	表处理模块	(120)
第 5.1 节	表的定义	(120)
5.1.1	表的概念	(120)
5.1.2	OCCURS (出现) 子句	(121)
5.1.3	表元引用	(124)
第 5.2 节	表的建立	(125)

5.2.1	在工作存储节对表赋值	(125)
5.2.2	在过程部中用 READ 语句或 ACCEPT 语句对表赋值	(126)
第 5.3 节	表处理	(129)
5.3.1	SET 语句与表处理	(129)
5.3.2	SEARCH 语句与表处理	(131)
5.3.3	PERFORM 语句与表处理	(138)
第 6 章	索引文件与相对文件	(144)
第 6.1 节	问题的引入	(144)
6.1.1	实时处理	(144)
6.1.2	低活动文件更新	(144)
6.1.3	多文件的更新	(144)
6.1.4	多重查找	(145)
6.1.5	要求文件记录有序排列	(145)
第 6.2 节	形成索引文件	(145)
6.2.1	索引文件的基本概念	(145)
6.2.2	建立索引文件	(147)
6.2.3	显示索引文件记录	(153)
6.2.4	增加、删除索引文件记录	(162)
6.2.5	修改索引文件记录	(170)
6.2.6	打印索引文件记录	(174)
6.2.7	索引文件小结	(181)
* 第 6.3 节	编程中几个特殊问题的处理	(182)
6.3.1	在程序内部设置打印的行间距	(182)
6.3.2	在程序内部选择汉字打印的字型	(185)
6.3.3	怎样避免执行 OPEN 语句时的非正常出口	(187)
6.3.4	绘图处理	(190)
第 6.4 节	实时更新处理	(193)
6.4.1	问题及示例程序 E611	(193)
6.4.2	示例要点说明	(209)
第 6.5 节	相对文件	(211)
6.5.1	相对文件的基本概念	(211)
6.5.2	与相对文件有关的语句、子句	(212)
6.5.3	形成相对文件	(213)
6.5.4	批处理更新相对文件	(224)
第 7 章	程序间通信	(238)
第 7.1 节	概述	(238)
7.1.1	主程序与子程序	(238)
7.1.2	调用执行子程序的过程	(238)

7.1.3	使用子程序的好处	(239)
7.1.4	主程序子程序的编译连接方法	(239)
第7.2节	无数据传送的程序间调用	(240)
7.2.1	问题及示例	(240)
7.2.2	有关问题的几点说明	(243)
第7.3节	有数据传送的程序间调用	(245)
7.3.1	基本概念	(245)
7.3.2	问题及示例程序	(246)
7.3.3	阅读编写示例程序 E702 及 E7021 的几点说明	(250)
第7.4节	程序间链接 (CHAIN)	(252)
7.4.1	链接 (CHAIN) 与调用 (CALL) 的区别	(252)
7.4.2	有关用法说明	(253)
*第7.5节	COBOL 调用汇编语言子程序	(256)
第8章	经济数据处理系统设计案例研究	(260)
第8.1节	使用软件工程方法开发经济数据处理系统	(261)
8.1.1	软件危机与软件工程	(261)
8.1.2	软件生存周期	(261)
8.1.3	结构化方法	(263)
8.1.4	工作成果成文	(264)
8.1.5	技术审查和管理复审	(265)
8.1.6	主要的图形工具	(265)
第8.2节	银行同城票据清算系统开发实例研究	(267)
8.2.1	问题定义	(267)
8.2.2	电子化同城票据清算系统可行性研究	(269)
8.2.3	电子化同城票据清算系统需求分析	(276)
8.2.4	系统设计	(281)
习题与作业		(293)
附录A	COBOL 保留字	(306)
附录B	COBOL 错误信息列表	(309)
附录C	ASCII 代码表摘要	(318)
附录D	ISO COBOL-78 语言格式一览表	(320)
主要参考文献		(335)

第1章 电子计算机系统概述

电子计算机是自动、高速、精确、大量地进行计算和信息处理的电子机器。电子计算机的出现是20世纪科学技术最卓越的成就之一，是科学技术和生产力高度发展的必然产物，是人类智慧的结晶。世界上第一台电子计算机是1946年研制成功的。40多年来，随着微电子技术的飞速发展，计算机技术突飞猛进，使人类进入了一个新时代——信息时代。

随着人类社会的发展，电子计算机已不是单纯的科学计算工具，它还可以完成对各种信息的收集、存贮和加工处理。巨型、大型计算机的研制与应用水平是衡量一个国家科学技术发展水平和经济实力的重要标志。因此学习和应用计算机技术是大势所趋。本章将介绍微型计算机系统组成、程序设计语言、程序设计规范和数据处理等内容。

第1.1节 电子计算机系统组成

计算机之所以能对大量的数值或文字进行准确高速的自动处理，是由于它按照存放在计算机内部预先设计好的称之为程序的处理步骤，对计算机各部分发出控制信号，使各个部分统一协调地工作。这就是所谓“存贮程序”概念，即把处理步骤和原始数据存放在计算机内部，使之依照这些步骤来控制全机自动协调地完成处理任务。

计算机内部的程序和数据，是以二进制代码的形式进行传送和处理的。所谓二进制代码，它的形式是由“0”和“1”二个数码组成的代码串。二进制的优点是：“逢二进一”，计算机之所以选择二进制形式去工作，是因为该形式运算简单，物理实现容易，设备投资最节省。

从使用者的角度来看，程序是计算机系统极为重要的组成部分，它与使用者所能看到的电子机械装置，即所谓“硬件”相对应，被称之为“软件”。硬件只有与软件相配合，构成计算机系统才能发挥作用。如果没有硬件，软件就失去了工作的物质基础。同样道理，只有硬件而没有软件系统，硬件则很难甚至无法发挥作用。这里需要强调，现代的计算机硬件系统和软件系统之间的分界线并不明显，某些硬件功能可以由软件加以实现，而某些软件功能又可以由硬件来实现，即某些硬件可以软化，某些软件可以硬化。

1.1.1 硬件的基本组成

计算机硬件系统基本组成如图1.1所示。

1. 控制器

控制器的作用是依据程序发出相应的控制命令（程序事先存放在存贮器中，它规定了操作的步骤）。控制器依次从存贮器中取出指令（指令是指程序中的一个或一类操作

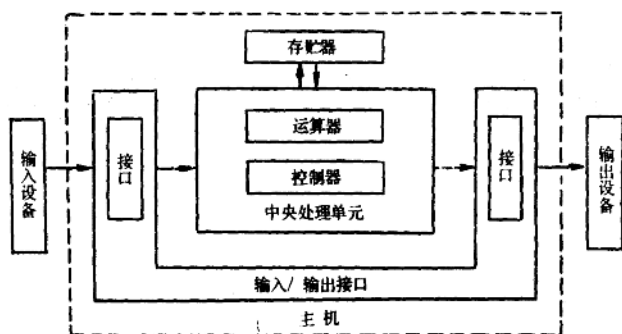


图 1-1 计算机硬件系统框图

命令），对指令进行分析，然后发出适当的控制信号，使计算机的各部分执行相应的操作。在此应当说明，控制器本身不进行运算，运算是由运算器来完成的。

2. 运算器

运算器是进行算术和逻辑运算的单元，在需要进行某项运算时，由控制器发出命令，将存放在存储器中的数据送到运算器，然后由控制器再发出相应的运算命令（如 A 和 B 相加），运算完成之后，将结果送回存储器保存起来，以便下次运算使用或输出。

控制器和运算器合起来称中央处理单元（Central Processor Unit，简称 CPU），它是计算机的核心部分。

3. 存储器

存储器是计算机的记忆装置，用来存放程序和数据。计算机中数据的存贮和运算均以二进制形式进行。

存储器由许多存贮单元组成。每个单元保存 m 位二进制信息，并且都有一个编号（称之为地址）。整个存储器就好像收发室的一排信箱，每个信箱都有一个号码。一个存贮单元有 m 位二进制信息， m 可能是 8 位、16 位、32 位或更多位。一般每 8 个二进制位编为一个字节（Byte）。存储器包含的字节数，总是 1024（即 2^{10} ）的倍数，常用“1k”表示。IBM PC/XT 内存为 512k，即它包含 $512 \times 1024 = 524\,288$ 个字节。

存储器一般分为内存贮器和外存贮器二级，内存贮器（简称内存或主存）通常直接与运算器、控制器相联系，用来存放当前运行所需要的程序和数据，以便向中央处理单元高速地提供信息。外存贮器（简称外存）用来存放当前不参加运行的程序和数据，在需要时可与内存成批交换信息，其特点是存贮容量大，但速度低。目前计算机内存一般使用半导体集成电路存贮芯片来组成。外存多采用磁表面存贮器，如磁盘、磁带。这类外存介质可以反复使用，记录的信息可以长期保存。一般来讲，内存的信息在断电后会丢失。因此，用户应当经常把内存信息周期性转存到外存中去。

通常，把中央处理单元与内存合在一起，称为计算机主机。

4. 输入输出接口

它是计算机主机与输入输出设备之间的控制逻辑部件。主要作用有：实现中央处理

单元和输入输出设备之间的通讯联络控制；实现数据缓冲以达到主机与输入输出设备之间的速度匹配以及数据格式的变换和交换状态信息。

5. 输入输出设备

它是计算机与外界交换信息的设备。常用输入输出设备有：键盘、显示器、打印机、绘图机、光笔、图形输入板、磁带机和磁盘机等。输入输出设备亦称外围设备。

1.1.2 软件及其分类

软件是计算机使用者与硬件系统进行信息交换、通讯对话、按使用者之思维控制与管理计算机的“程序系统”。软件可以扩大计算机的功能和用途，提高计算机的使用效率，使编制程序和使用计算机变得更容易更方便。使用者是通过软件与硬件发生关系的。这种关系如图 1.2 所示。

一般将软件分为“系统软件”和“应用软件”两大类。

1. 系统软件

系统软件（或称为系统程序）用于计算机硬件的运行管理和维护。它包括与用户有关的程序的装入、翻译、管理、维护、控制和运行等多种程序。系统软件由三个部分组成：操作系统、语言处理系统和服务程序。

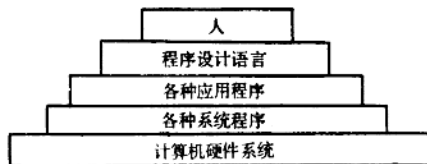


图 1.2 人与软件和硬件的关系

(1) 操作系统 (Operating System)：在早期计算机系统中，一般只包括与编写应用程序有关的语言处理程序以及一些简单的管理外围设备的管理程序。语言处理程序是把计算机不能直接接受的程序设计语言编写的程序，加工翻译成机器能够接受的程序。管理程序主要用于协调主机与外围设备之间的相互联系。随着计算机技术的发展，计算机的使用方法与信息处理方法也日益多样化。例如，把几个独立的程序和数据汇集在一起输入计算机进行“成批处理”；用一台计算机处理许多用户从各自终端装置同时发来的信息和数据的“分时处理”；在银行临柜业务中要求计算机立即响应的“实时处理”等。另外，计算机硬件结构也越来越复杂，运算速度、处理能力也越来越强。为了有效地使用计算机资源，提高计算机的使用效率，扩大处理能力，在系统软件中发展出了操作系统。

操作系统可以全面有效地管理计算机资源，自动调度用户的程序，合理组织计算机工作流程，成了用户与计算机之间的必不可少的中介。

(2) 语言处理系统：用户为了解决某一具体问题，通常要采用适当的高级程序设计语言编制程序，该程序称之为“源程序”。计算机对源程序再处理，产生出一个自身能接受并立即可投入运行的程序（即“目标程序”）。完成源程序到目标程序转换的工具是语言处理程序。第 1.2 节将详细讨论这一问题。

(3) 服务程序：包括各种诊断程序、检查程序、分类/合并程序、链接、编辑程序以及系统应用程序等辅助性程序。它们为更方便地使用计算机创造了条件。

2. 应用软件及其应用领域

应用软件是为了解决各行各业特定问题所编制的各种程序的总称。随着计算机功能的扩大和应用范围的扩展, 应用软件的研制发展速度很快。为了编制出合格的应用软件, 我们应当熟悉本专业知 识, 做到不仅能读懂程序, 而且可以修改或独立编写这类程序, 这样才能使应用软件更好地发挥作用。总括起来, 应用软件的应用领域大致有以下几个方面:

(1) 科学计算: 科学计算也称为数值计算。这是计算机最基本的应用方面。例如, 人造地球卫星、宇宙飞船轨道的计算; 原子能物理学、天体力学、空气动力学、流体力学、高层建筑的结构力学分析等学科的复杂计算; 国民经济计划平衡方面的大量计算等等。这些问题往往计算量大、难度高, 利用计算机可大大缩短计算周期, 节省大量人力和物力。

(2) 信息处理: 当今社会是信息化的社会, 大量信息的收集和加工, 单靠人工是不行的。计算机可作为信息处理的有力工具, 例如, 用于企业管理、会计、统计、资料档案管理和图书馆书刊检索等方面的工作。信息处理又称为: “数据处理”, 即对信息或数据进行综合分析和加工整理, 其特点是被处理的信息量大且有一定的时间要求。例如, 银行帐务处理, 就要把当天的经营情况及时汇总、分类、结算、统计和制表, 用人工进行处理不但费时费力, 而且容易出错, 而用计算机处理既及时又准确。

(3) 自动控制: 利用计算机实现某个生产过程的自动控制, 不仅可以大大提高劳动生产率, 减轻劳动强度, 而且可以提高产品质量, 降低生产成本。因此, 近年来在机械、冶金、石油、化工、电力、建筑以及轻纺工业等生产部门已广泛使用计算机去控制生产过程, 并且获得了很好的效果。

(4) 计算机辅助设计、辅助制造和辅助测试: 所谓计算机辅助设计 (CAD), 就是用计算机来帮助设计人员进行工程设计。例如, 在电子计算机硬件的研制过程中, 就可以利用 CAD 技术进行体系模拟、逻辑模拟和自动布线等, 从而大大提高了设计工作的效率。

计算机辅助制造 (CAM) 是用计算机来进行生产设备的管理、控制产品加工过程。例如, 在产品的制造过程中, 应用计算机来控制机器的运行, 处理生产过程中所产生的各种数据, 控制和处理原材料的流动以及对产品质量进行检验等。在生产过程中, 使用 CAM 技术能提高产品质量, 降低成本, 缩短生产周期以及劳动条件。

计算机辅助测试 (CAT) 就是利用计算机进行产品测试。例如, 在生产大规模集成电路的过程中, 由于电路复杂, 用人工测试往往比较困难, 不但效率低, 而且容易损坏元器件, 而利用计算机辅助测试技术, 就可以自动地测试集成电路的各种交、直流参数和逻辑关系, 并且可以实现产品的分类和筛选。

(5) 人工智能: 又称为智能模拟, 是根据信息的反馈和周围环境来确定或选择输出结果, 是一门探索模拟人的感觉和思维规律的科学。计算机可以模拟人的感觉, 从事逻辑判断, 定理证明等高级思维活动。具有一定“思维”能力的机器人的出现就是人工智能研究的可喜成果。机器人在宇宙空间探索、深海打捞、处理放射性污染物等方面大有作为, 在某些领域中从事人类所无法直接从事的工作。

*1.1.3 长城微机系统配置简介

下面,以长城 0520 (与 IBM PC/XT 兼容) 系列微机为主,对其硬件系统和软件系统配置作一简要介绍。

1. 长城微型机硬件系统

(1) 中央处理单元:长城 0520 系列机的中央处理单元使用 Intel8088 微处理器芯片。主要辅助控制器有: Intel8087 协处理器 (处理浮点数)、8253 时钟控制器 (处理定时信号)、8259 中断控制器 (处理中断信号)、8237A DMA 控制器 (处理内存与外存数据交换)。

(2) 存储器

①内存。使用半导体集成电路存储器,分为可以随机进行数据存取的随机存储器 (RAM) 和在工作过程中只能读,不能存写数据的只读存储器 (ROM)。随机存储器容量一般为 512k 或 640k 字节,只读存储器容量为 40k 字节。

②外存。外存采用可更换存储介质的软磁盘和不可更换存储介质的硬磁盘。一般配置二个软盘驱动器和一个硬磁盘机。

(3) 输入输出接口:常用输入输出接口卡 (即适配器电路板) 有:彩色/图形监视器接口卡,单色显示器与并行打印机接口卡,并行打印机接口卡,5 吋软盘驱动器接口卡,硬盘机接口卡,键盘接口卡 (在系统板上), RS232C 异步通信接口卡等。另外还有一些选件,供用户选用。

(4) 输入输出设备:键盘,光笔,图形输入板,显示器,软磁盘驱动器和硬盘机,打印机及绘图仪等。

2. 系统软件

目前,可在长城 0520 微机上运行的操作系统有: PC-DOS、MS-DOS、CCDOS、UCDOS、CP/M86、并发 CP/M86、UCSDP-system 以及在长城 286、386 中运行的 XENIX 操作系统等。

(1) PC-DOS 是 IBM PC 系列微机的主要操作系统,原由 Microsoft 公司研制成功,命名为 MS-DOS。后来 IBM 公司与 Microsoft 公司经过协商后,确定采用 MS-DOS 作为单用户微机标准操作系统,并命名为 PC-DOS。

(2) CCDOS 是我国机械电子工业部第六研究所率先在 PC-DOS 1.0 版本上增加汉字输入输出功能,推出了 CCDOS 1.0。不久又推出 CCDOS 2.0,目前已有 4.0 版本问世。

(3) CP/M86 是在 Intel 8086/8088 上使用的 CP/M 簇操作系统,并发 CP/M86 是单用户多任务操作系统。

(4) UCSDP-system 是一个“与处理机无关”的操作系统。它几乎全是用 PASCAL 语言写成的。

(5) XENIX 是最有前途的 16 位微机操作系统 UNIX 的变种。UNIX 操作系统第一版是美国贝尔实验室的肯·汤普逊在 PDP-11 小型机上实现的,现已成为 M68000、Z8000、Intel8086、80286、80386 以及 80486 等高档微机的主要操作系统。

3. 应用软件

可在 IBM PC 系列微机上运行的应用软件多达几千种,其中绝大多数不经修改就可以在长城微机上运行。根据长城微机的不同应用领域,列出几种主要应用软件如下。

(1) 中小企业管理类:该类软件主要应用于企业财务、订货、仓库、计划、销售、市场预测以及文件档案等方面的管理。例如:总帐系统,应收帐系统,库存系统,订货系统,工资系统,人事管理系统和固定资产管理系统等。

(2) 办公室自动化类:该类软件主要用于文件处理、文字处理、档案管理以及自动信件处理。例如,编辑打印、拼写检查和邮件归并等软件。

(3) 工程人员工作站类:主要用作计算机辅助设计(CAD),进行工程设计和计算。主要软件有:MGI/CAD、AUTO-CAD、CAPS 逻辑电路辅助设计软件,PLAN 建筑机械辅助设计软件等。

(4) 教育类:长城微机可运行的教育软件分为初等教育、中等教育、一般教育及培训等。

应用软件有:操作系统培训,M-BASIC 培训,Word Star(字处理)培训,化工实验辅助教学软件以及英语试题库等系统。

(5) 娱乐游戏类:数量较多,且已普及,举例从略。

(6) 通信及网络类:长城微机的功能有一定的局限性。在企业管理和办公室自动化等应用领域内,一台微机往往被用作某一网络的工作站或作为某个系统的一个外围机。此时,长城机需配上一定的通讯和网络软件。通讯软件一般有:CROSS-TALK、MOVE-IT、PROCOMM 等。局域网络软件有:3COM 公司的 3*网、Novell 公司的 NETWARE 网、D-LINK 网以及 LAN Data store 等网络软件。

(7) 其它方面的应用软件:这些软件具有特殊用途,是针对某一特定行业的特殊要求而设计的,具有很强的专用性。如在经济领域里:ESP(计量经济学软件包),RATS(时间序列分析软件包),LINDO(交互式线性整型规划软件包)等。

第 1.2 节 程序设计语言与程序设计规范

1.2.1 机器语言与汇编语言

1. 机器语言

人和计算机交流,必须解决一个“语言”沟通问题。计算机并不能理解和执行人类相互交流所使用的自然语言,只能接受和执行二进制的指令。每一条二进制的指令使计算机执行一个操作。例如在 IBM PC/XT 计算机中以 0000001111110101 表示“进行一次加法”,以 0010101011011001 表示“进行一次减法”。计算机能够直接识别和执行的这种指令,称为“机器指令”。不同的设计师为每一种不同类型的计算机都规定了只有该类机器才可以执行的若干指令,这种指令的集合就是机器语言“指令系统”,简称“机器语言”。例如,IBM PC/XT 及其兼容机所配置的 Intel 8088/8086 微处理器的指令系统包含 104 种指令。为了解决某一特定的问题,需要有选择地使用指令系统中

的某些指令，这些被选中并编排在一起的指令就组成一个“程序”（Program），也就是说，一个程序是完成某一特定任务的一组指令序列。

机器语言随不同类型的机器而异。有的机器以 8 个二进制位作为一条指令的长度，有的机器则以 16 个二进制位来表示一个指令。显然，为了进行一次加法，在不同的计算机上所使用的指令可能完全不同。用机器语言编写程序工作量大，容易出错，程序直观性差，不易阅读，给使用者带来很大的不便。这也给计算机的普及推广造成很大的障碍。因此，在机器语言基础上发展演变出“符号语言”，即“汇编语言”。

2. 汇编语言

汇编语言是用“助记符号”来代替那些难懂难记的二进制代码。所以，汇编语言亦称为“符号语言”。例如，上述二进制代码加法指令用汇编语言可写为：

ADD SI, BP

其中，ADD 的英文含义为相加，这条指令代表将 SI 寄存器中的数和 BP 寄存器中数相加（寄存器是运算器中实现运算的单元），并把结果存入 SI 中。显然，这种表示形式要比机器语言容易理解和记忆。相同功能的汇编语言指令和机器语言指令基本上是一一对应的。汇编语言的出现，可使程序设计人员摆脱对枯燥冗长的二进制代码的记忆，集中精力去考虑程序设计方法。

用汇编语言编写的程序称为“源程序”（Source Program）。这种程序不能直接在机器上运行，必须先把它们逐条翻译成机器指令，然后交给计算机执行。这个“翻译”工作称之为“汇编”，是由一种专门的“汇编程序”来完成。经过转换后形成的可以由计算机直接执行的机器指令程序称为：“目标程序”（Object Program）。图 1.3 表示汇编语言源程序的执行过程。

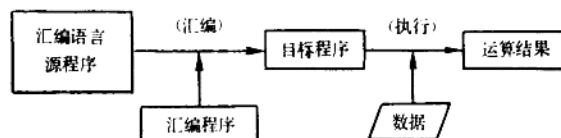


图 1.3 汇编方式

汇编语言和机器语言都是针对特定的计算机系统的，不同类型计算机所配置的汇编语言是不同的。因此，我们称汇编语言和机器语言为“面向机器的语言”，它们也被称为“低级语言”。使用汇编语言编程，对程序编写人员的要求比较高，即要求他了解计算机内部结构，因而它仍然不易为广大的非计算机专业的人员所掌握，它与自然语言和数学语言还有很大差距，这就促成了高级语言的产生。

1.2.2 高级语言

为了减轻人们的编程难度、提高编程效率与质量以及推广计算机的应用领域，50 年代中，出现了第一个高级语言——FORTRAN 语言；60 年代，又相继出现了 COBOL、ALGOL 以及 BASIC 等语言。相对机器语言和汇编语言而言，这些语言被称为“高级语言”。它们的特点是：用一种接近于自然语言和数学语言的专用语言来表示算法，它们基本上是由英文单词和数学上的运算符号所构成。它们的出现，可以使程序设计者摆

脱对机器内部结构的了解,集中精力考虑解题的逻辑和对运算过程的描述。只要有一定数学和英语知识,在短时间即可学会使用计算机。高级语言基本上不依赖于机器,易学、易记、易于检查和修改,通用性强。

显然,计算机是不能直接执行用高级语言编写的源程序的,必须要有“翻译”,把高级语言源程序翻译成可由计算机直接执行的目标程序。这种翻译工作由被称为“编译程序”的专用软件来完成。每一种高级语言都有对应于自己的编译程序。根据翻译过程不同,有“编译方式”和“解释方式”两种翻译方法。

编译方式是这样一个过程,事先把编译程序装入计算机内,编译程序把用高级语言编写的源程序整个读入计算机中,进行有关检查和处理,然后将这个源程序整个地翻译成目标程序,最后执行目标程序。用 FORTRAN、ALGOL、COBOL、PASCAL、C 等语言编写的源程序等都采用编译方式。

解释方式是这样一个过程,事先将“解释程序”装入计算机内,“解释程序”对正在进入计算机中的高级语言程序进行逐条翻译,即解释完一条高级语言的指令,就立即执行这条指令所要求的操作,直至高级语言程序也被执行结束。就是说源程序的执行是边解释边执行。BASIC 语言大都采用解释方式(除少数采用编译方式外)。

目前,高级语言种类很多,其适用的范围也不同。比较通用的有:

FORTRAN (FORMula TRANslator)

ALGOL (ALGOrithmic Language)

PL/1 (Programing Language)

COBOL (COMMON Business Oriented Language)

BASIC (Beginner's All-purpose Symbolic Instruction Code)

PASCAL, Turbo PASCAL

C, Turbo C

其中, FORTRAN、ALGOL、PASCAL、Turbo PASCAL 是面向科技运算的算法语言, COBOL 是面向数据处理(适用于商业、经济管理)的商用语言。PL/1 兼有科技运算和数据处理功能,它是适应性较强的算法语言。BASIC 具有人机会话功能,适合于初学者使用。而 C、Turbo C 主要用于编写系统软件。

上述的高级语言均是“面向过程的高级语言”,它们被称之为过程化语言。即该类语言在解决某个具体问题时,必须编写出相应的可由计算机来完成的解决这种问题的具体步骤及处理办法。

随着计算机的发展,人们进一步希望出现这样的语言,只要指出让计算机“做什么”,而不必具体指出“如何做”,再由计算机自己去解决“如何做”的问题。这就是“非过程化语言”所应达到的功能,即不用给出解决问题的过程。目前已经问世的非过程化语言有:

FORTH (人工智能语言)

dBASE (小型关系数据库语言)

QBE (关系数据库语言)

ORACLE (关系数据库语言)

例如,使用数据库语言,只需向计算机中输入“将全厂(或某车间)职工中年龄大

于 40 岁的职工姓名、职务、部门打印出来”这样一条命令，计算机就会自动执行，并打印出所需结果。显然，这又是一个飞跃，为更多的人更方便地使用计算机创造了极为有利的条件。非过程化语言是比高级语言功能更强的计算机语言。显然，计算机语言越高级，人们使用它就越方便，但对硬件的具体构造的了解就越困难。

目前，国内外大多数计算机上运行的程序，多是采用面向过程的语言编写的。因此我们应当熟练地掌握用该类语言编写程序的方法和技巧。

1.2.3 流程图 (GB1526-79) 及其用法

学习计算机语言的目的是利用该语言设计出可供计算机运行的程序。在拿到一个实际问题之后，怎样才能编写出程序呢？一般情况下应按图 1.4 所示步骤进行。

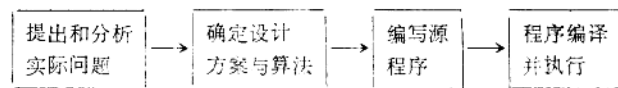


图 1.4 利用计算机解决实际问题步骤

一般来说，程序设计工作最关键的步骤就是确定设计方案或算法。所谓方案就是为解决问题而采取的方法和步骤，而算法则是解题方法的精确描述。解决一个问题的过程就是一个方案或算法的实现过程。在程序设计中，一般采用流程图来描述方案或算法。我国国家标准局公布《流程图图形符号标准》中常用的图形符号如图 1.5 所示。

同时，该标准中还列出指明具体输入/输出方式的图形符号。如图 1.6 所示。

流程图中流程方向，一般是从左到右，自上而下，要用箭头指出方向。

流程图的主要优点是逻辑描述清楚；形象化；直观化和便于交流。

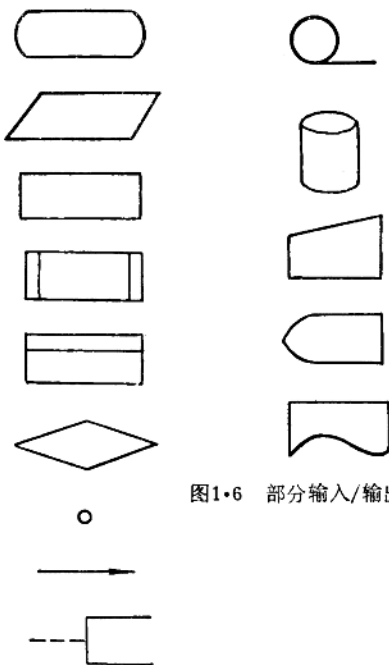


图 1.6 部分输入/输出图形符号

图 1.5 常用流程图图形符号

1.2.4 程序设计规范

人们对现代计算机系统的巨大功能和惊人速度赞叹不已的同时，也对计算机软件系统所存在的严重问题表示关注。由于早期计算机的运行速度低，存储容量小，因此，

在评价一个程序的优劣时，总是把效率放在首位，即一个好的程序应该节省运行时间与存储空间。早期的程序设计以追求设计的技巧为主要目标，往往不注意程序的易读性与易修改性；程序设计无章可循，使编制出的程序晦涩难懂，给程序交流、调试和维护增加了困难。这样编制出来的程序可靠性较差。随着计算机技术的发展，计算机的运算速度和内存容量迅速提高，硬件的制造成本不断降低，程序的效率已不是主要矛盾。又由于程序的规模日益扩大，程序的可靠性问题日益突出。编制、维护和修改程序的费用急剧上升。客观现实要求软件的生产方式从“个体单干”转向按照“工程”的方式来组织，即按照一定的规范，一定的步骤由多人分工协同来进行程序设计。

程序设计常用的规范化模式是“结构程序设计”。所谓结构程序设计，目前还没有一个为人们普遍接受的定义，比较流行的看法是：结构程序设计是一种程序设计技术，它采用自顶向下，逐步求精和单入口、单出口的控制结构。

结构程序一般由三种基本结构组合而成：顺序结构、选择结构和循环结构。

1. 顺序结构

见图1·7，在这种结构中各框（例如A和B）是被顺序执行的，每框可以表示一种操作或一个完整的模块。顺序结构是最简单的一种基本结构。如果把其它两种结构从整体上分别看作是一个模块的话，整个程序可被看作是由一系列的顺序结构所组成。

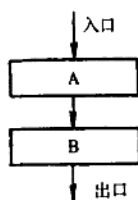


图1·7 顺序结构

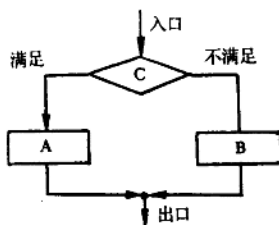


图1·8 选择结构

2. 选择结构

如图1·8所示，此结构中有一个判断框，并有两个分支，根据条件（C）是否满足而分别执行A框或B框。其中B框可以不含任何操作（即省略B框）。

3. 循环结构

又称重复结构，有两类。其一是当型（while）循环结构（见图1·9），当条件满足时，重复执行A框的操作，一旦条件不满足就退出A框的操作，继续往下执行。如果一开始进入条件判断（C1）时，条件就不满足，则A框的操作一次也不执行。其二是

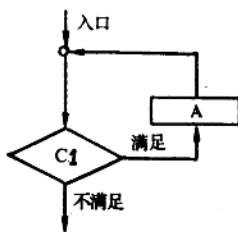


图1·9 当型循环结构

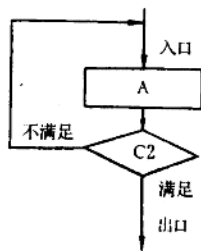


图1·10 直到型循环结构

直到型 (Until) 循环结构(见图1·10)。与前者不同的是, A 框至少执行一次。然后对条件进行判断, 如果条件不满足, 继续循环处理 A 框, 直到条件满足, 再退出循环, 执行下面一个基本结构。

请注意, 以上两种循环结构的区别是:

当型循环结构是“先判断后执行”, 而直到型循环结构是“先执行后判断”。

这三种基本结构有以下共同特点:

第一, 只有一个入口, 只有一个出口。

第二, 结构中每部分都可能执行到。也就是说, 对每一框来说, 都有且仅有一条从入口到出口的路径通过它。

第三, 结构内没有“死循环”(无休止的循环)。

计算机科学家已经证明, 由上述三种基本结构组成的流程图可以分解任何复杂的问题, 由它们所构成的算法称为结构化算法。

怎样编写程序呢? 有两种截然不同的方法, 其一是“自顶向下”的方法, 其二是“自下而上”的方法。举例说, 用第一种方法编写一部著作, 往往先确定主题, 然后分成若干章, 定出各章题目, 再把每章分为若干节, 再细分为若干段等等。即由粗到细, 逐步求精, 直到写出文章为止。这种方法是先有全局, 进行整体设计, 然后再进行下一层的设计, 逐步细化。结构程序设计采用这种方法的好处是可以提高程序设计的成功率和生产率。过去有不少程序设计人员往往使用第二种称之为“自下而上”的方法, 即拿到题目后未作充分考虑就一个语句一个语句地写程序, 写出的程序通常漏洞百出, 质量较差, 可读性低。

结构程序设计方法的三个要点是: 自顶向下、逐步求精、模块化。所谓模块化, 是将一个大任务分成若干个较小的部分, 每部分承担一定的功能, 称为“功能模块”, 各模块可以分别由不同的人编写调试, 这就使得一个大系统能在较短的时间内由许多人分工协作完成, 因此其实质是工程化。

结构程序设计是程序设计规范常用的一种模式, 但并不是唯一的模式, 下面再介绍几种常用的程序设计规范。

• 1·2·5 常用的几种程序设计规范

1. Jackson 方法

Jackson 方法由英国的 M·Jackson 提出, 此方法在欧洲颇为流行, 它特别适用于设计企事业管理类的数据处理系统。

Jackson 方法的某些基本思想同结构程序设计方法是一致的, 如模块化、由顶向下、逐步细化、程序的结构与问题相对应等。但是 Jackson 方法的一个显著特点是在数据结构的基础上建立程序结构。

该方法主张程序结构与问题的内在数据结构相对应。许多经济管理类系统所处理的数据具有层次结构, 如文件是由记录组成, 记录又由数据项组成。数据结构不但影响程序的结构而且影响程序处理过程, 重复出现的数据通常由具有循环结构的程序来处理, 选择数据要用选择结构的程序来处理。数据组织层次通常和使用这些数据的程序的层次