

TURBO

PASCAL程序设计

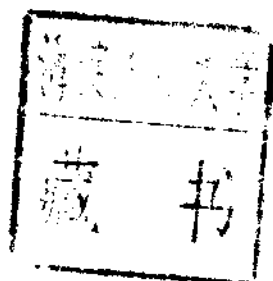
●唱江华 张益铎 主编●

北京航空航天大学出版社

12
23H/1

TURBO PASCAL程序设计

唱江华 张益铎 主编



北京航空航天大学出版社

0021009

(京)新登字166号

内 容 提 要

本书以近年来最畅销的TURBO PASCAL3.0版为背景,从与标准PASCAL兼容部分开始,首先详细地介绍了PASCAL语言的数据描述和程序设计方法,然后对TURBO PASCAL的特殊功能进行了专章介绍,并把部分内容整理成附录,从而它包括了TURBO PASCAL3.0版的全部内容和4.0版的部分内容。每章后都设有习题。为方便读者上机操作,设专章介绍了在PC机和APPLE-II机上的操作方法。书中例题(除示意性例题外),均有实用背景并可直接上机运行,消除了学习内容与上机操作的脱节。因此,本书既可做PASCAL语言的教学用书,也可做为专业技术人员的参考书。

JSSSS/34

TURBO PASCAL程序设计

唱江华 张益铎 主编

责任编辑 樊 毅

*

北京航空航天大学出版社出版 新华书店总店科技发行所发行

各地新华书店经销 北京密云华都印刷厂印刷

开本787×1092 1/16 印张:25.5 字数:653千字

1991年9月第1版 1991年9月第1次印刷 印数:15000册

ISBN 7-81012-263-0/TP·047 定价13.40元

序

如果一般地向读者推荐一本PASCAL语言书几乎已无话可谈,因为PASCAL语言本身的作用越来越被人们所熟悉,它的特点并非鲜为人知。之所以还要说上几句,是由于TURBO PASCAL本身的独特功能和这本书写法上免除了初学者学习标准PASCAL和按某一具体版本上机之间的困难。

M.Bridger在《Byte》杂志1986年第2期上为TURBO PASCAL 3.0版做了一番认真的评价:

首先TURBO PASCAL很小(小于40K),但它集全屏幕编辑、编译、调试及运行时的错误检查于一体,这一友好的用户环境是一般PASCAL版本不可比的。一般PASCAL版本即使不含编辑程序也要比它大5~10倍。其次是它的速度快,它的编译速度是著名的PASCAL MTI的69倍;它有关于处理浮点运算的专门程序;在屏幕编辑方法上,它没有调用PC-DOS的服务程序,而是直接将字符写入屏幕存储器;这样处理的结果,一方面大大提高了编辑速度,另一方面又设法消除了访问内存对于屏幕引起的干扰。

TURBO PASCAL的又一特点是它大大扩充了原标准PASCAL的功能。它丰富的标准过程,直接处理的内存与接口级的预定义类型,以及它对DOS的功能调用,特别是它的4.0以上版本的模块化分离编译等手段,使它能上能下,向上可开发大型软件,向下可改进系统功能,作为得心应手的工具它必将倍受人们的喜爱。

按M.Bridge的观点,虽然TURBO PASCAL并非标准,但事实上,它已经卖出了50万个拷贝,成为近年来最畅销的计算机语言之一,这使它成为了事实上的标准。

目前,已出版了很多以标准PASCAL为蓝本的《PASCAL程序设计》教材,但实际上上机时,读者又必须参考相应的实现版本的说明书,这给初学者带来了许多不便。这本《TURBO PASCAL程序设计》一方面没有丢失“标准”,另一方面把理论说明和具体上机实现相统一。这无论是在教学上还是在一般的读者需求上都是一个值得推荐的方法。对于保持TURBO PASCAL全部内容的完整性同循序渐进地向读者介绍这一矛盾,作者处理得十分得当。

愿这本书能作为广大读者的良师益友,把程序设计水平推向更高的层次。

宋子善

前 言

PASCAL语言自七十年代问世,以首次推出符合结构化程序设计原则而著称,至今它仍是实现软件工程化设计方法的有力工具。同时它又以数据类型丰富、描述数值问题和非数值问题灵活、严谨,程序风格优雅而被广泛应用于科学计算、数据处理、系统软件的设计中。

计算机科学领域中的数据结构、编译技术和操作系统等课程中,用PASCAL语言定义概念,描述算法简明、清晰,所取得的成功已被人们所公认,而它对培养初学者养成良好的程序设计习惯、提高对现实数据对象及算法的描述能力上所起的作用,也越来越被人们认识。有人建议,初学计算机高级语言,以先学PASCAL语言为好,待养成良好习惯后再根据需要进行学习象FORTRAN、COBOL、PROLOG等专业性较强的语言。笔者从多年来PASCAL教学和实践经验中深感这一看法的正确。

近年来,对标准PASCAL的实现版本越来越多,使用起来也十分方便。如果说BASIC语言由于使用方便而被人们喜欢,那么PASCAL新版本的方便灵活将使PASCAL语言更受人们欢迎。一旦PASCAL被人们掌握,它发挥的作用将使BASIC望尘莫及。

美国Borland公司86年推出的TURBO PASCAL 3.0版在符合N.Wirth原报告标准的基础上,扩充了大量的标准过程,包括随机文件处理过程、图形处理过程、操作系统直接调用、硬件调用等,大大加强了原标准的描述能力。尤其它提供的“用户友好”的操作环境,将编辑、编译、调试、运行时的错误检查集于一体,运行时不需要专门的程序库,也不需要链接等操作。一台PC机(或APPLE-II)一张软盘片就可编制、调试大型软件。TURBO PASCAL消除了PASCAL被束之高阁的状态。

TURBO PASCAL不仅小巧、而且玲珑。它的功能,特别是3.0以后的版本使PASCAL许多受指责的地方得到了弥补,提供了模块化分离编译的手段,说明部分可以任意次序和重复进行,变量可以用有类型变量进行初始化,CASE语句ELSE分支、和系统底层有友好的接口方法等等。使它完全可以和N.Wirth本人为同一目的推出的MODULOR-2语言相媲美。我们已看到许多用TURBO PASCAL书写的大型软件。

为避免学习标准PASCAL时,上机还要熟悉一种具体的版本给读者带来的困难,我们直接以TURBO PASCAL语言手册为依据编著了本书,使读者可边学习边上机,三年多的教学试用效果是满意的。

为使读者能循序渐近地学习而又不失TURBO PASCAL的特殊功能,我们首先对TURBO PASCAL中符合标准PASCAL的部分按常规的顺序进行了介绍,并附以大量的例题和习题,使读者能掌握PASCAL语言的基本内容和程序设计风格。把TURBO PASCAL中的特殊功能放在后序章节。在此基础上写了进一步功能一章和整理了附录部分,使本书基本保留了原技术手册中的全部内容。因此,本书既可作为教材又可做为专业技术人员的参考书。

为帮助读者理解书中内容,每章后都附有习题,且专门编写了上机操作一章,介绍了对TURBO PASCAL源程序的编辑、编译、运行调试方法,配合读者掌握TURBO PASCAL的操作环境。书中的例题都经过上机调试,读者可容易地在PC机上实现。

本书第一章介绍了计算机的基础知识,为后续章节的理解做了必要的准备。第二、三章介绍了PASCAL语言源程序的基本结构、基本语法及标准数据类型,目的是使读者能掌握一般的程序设计方法。第四至第九章、第十一章介绍了PASCAL中各种由用户定义的数据类型,并结合实例力图使读者能学会PASCAL语言对数据对象的描述方法和程序设计风格。把递归方法移至第十章介绍,是为避免读者在过程概念方法尚不熟悉的情况下学习递归过程的困难。第十二章总结性地向读者介绍结构化程序设计方法,同时专门介绍了5.0版的模块化方法。第十三章以后及部分附录是TURBO PASCAL的特殊技巧部分,介绍的深入问题可供感兴趣的读者及专业技术人员参考。尤其附录部分可使读者在掌握了PASCAL的一般功能后做速查手册使用。

全书由唱江华、张益铎主编。张益铎编写第一、二章,邓文新编写第三章,王英杰编写第四、七章、李南飞编写第五章和第十三章的§13.6,伊崇信编写第六、十章,唱江华编写八、九、十四章,黄云桂编写第十一、十三章及附录部分,第十二、十五章及第十四章的§14.3、§14.6由迟乐军编写。编写过程中得到了齐齐哈尔轻工学院计算机站的大力支持,这里表示感谢。

北京航空航天大学宋子善教授审阅了全书并提出了宝贵的意见。

由于编者水平有限,加之时间仓促,书中错误与不妥之处,望读者不吝指正。

编 者

一九九一年三月

目 录

第一章 计算机基础知识

§ 1.1 计算机硬件软件结构	(1)
§ 1.1.1 计算机硬件系统	(2)
§ 1.1.2 计算机软件	(3)
§ 1.1.3 机器语言和高级语言	(4)
§ 1.2 数制、码制、编码	(5)
§ 1.2.1 数制	(5)
§ 1.2.2 码制	(8)
§ 1.2.3 数字编码与字符编码	(10)
习题一	(11)

第二章 PASCAL语言概述

§ 2.1 PASCAL语言特点	(12)
§ 2.2 PASCAL源程序结构	(12)
§ 2.2.1 程序首部	(13)
§ 2.2.2 程序说明部分	(14)
§ 2.2.3 语句部分	(14)
§ 2.3 字符集和符号	(15)
§ 2.3.1 基本字符	(15)
§ 2.3.2 符号	(15)
§ 2.3.3 分隔符	(17)
§ 2.4 数据类型的概念	(17)
§ 2.5 常量和变量	(18)
§ 2.5.1 常量	(18)
§ 2.5.2 常量定义	(18)
§ 2.5.3 变量说明	(19)
§ 2.6 标准数据类型	(20)
§ 2.6.1 整数类型	(20)
§ 2.6.2 实数类型	(22)
§ 2.6.3 字符类型	(24)
§ 2.6.4 布尔类型	(25)
§ 2.7 表达式与赋值语句	(26)
§ 2.7.1 表达式	(26)
§ 2.7.2 赋值语句	(29)
§ 2.8 输入和输出过程	(30)
§ 2.8.1 读语句	(31)
§ 2.8.2 写语句	(33)
§ 2.8.3 字符类型变量的输出	(34)

§ 2.8.4 整数类型变量的输出	(35)
§ 2.8.5 实数类型变量的输出	(36)
§ 2.8.6 布尔类型变量的输出	(37)
§ 2.9 简单程序举例	(38)
习题二	(40)
第三章 控制语句	
§ 3.1 复合语句	(43)
§ 3.2 条件语句	(44)
§ 3.2.1 单分支条件语句	(44)
§ 3.2.2 双分支条件语句	(47)
§ 3.2.3 条件语句嵌套	(49)
§ 3.3 情况语句	(53)
§ 3.4 当语句	(56)
§ 3.5 直到语句	(60)
§ 3.6 循环语句	(64)
§ 3.7 循环嵌套	(68)
习题三	(71)
第四章 枚举类型和子域类型	
§ 4.1 枚举类型	(73)
§ 4.1.1 引入枚举类型的必要性	(73)
§ 4.1.2 枚举类型的定义、运算规则和输入输出方法	(74)
§ 4.2 子域类型	(80)
§ 4.2.1 引入子域类型的必要性	(80)
§ 4.2.2 子域类型的定义和运算规则	(80)
§ 4.3 日历程序	(87)
§ 4.4 类型相容和赋值相容	(91)
习题四	(93)
第五章 过程与函数	
§ 5.1 过程	(95)
§ 5.1.1 无参过程	(95)
§ 5.1.2 带参过程	(100)
§ 5.1.3 值参数和变量参数	(103)
§ 5.2 函数	(107)
§ 5.3 过程嵌套与标识符作用域	(111)
§ 5.3.1 过程和函数嵌套的概念	(111)
§ 5.3.2 标识符的作用域	(113)
§ 5.4 非局部量与副作用	(115)
* § 5.5 过程参数与函数参数	(117)
* § 5.6 外部子程序	(120)
习题五	(122)

第六章 数组类型

§ 6.1 数组的概念	(124)
§ 6.1.1 概述	(124)
§ 6.1.2 数组类型定义和一维数组	(124)
§ 6.1.3 类型的一致性	(132)
§ 6.1.4 下标越界及检查	(132)
§ 6.2 多维数组	(134)
§ 6.2.1 多维数组的类型定义	(134)
§ 6.2.2 多维数组的应用举例	(136)
§ 6.3 字符数组和字符串类型	(141)
§ 6.3.1 字符数组	(141)
§ 6.3.2 字符串类型	(143)
§ 6.4 可调数组参数	(151)
§ 6.4.1 概述	(151)
§ 6.4.2 可调数组参数指明的语法	(152)
§ 6.4.3 例题及应用	(154)
习题六	(155)

第七章 集合类型

§ 7.1 集合类型的定义和运算规则	(157)
§ 7.1.1 集合类型的定义	(157)
§ 7.1.2 集合类型的运算规则	(158)
§ 7.2 键盘操作训练程序	(164)
§ 7.3 趣味程序：谁去破案	(167)
§ 7.4 排课表程序	(169)
习题七	(173)

第八章 记录类型

§ 8.1 记录类型的概念	(173)
§ 8.1.1 记录类型的定义	(175)
§ 8.1.2 记录的嵌套	(176)
§ 8.1.3 记录类型变量及其访问	(177)
§ 8.2 开域语句	(181)
§ 8.2.1 开域语句的定义	(181)
§ 8.2.2 开域语句的嵌套	(182)
§ 8.3 图书借阅管理程序	(185)
§ 8.4 记录的变体	(190)
习题八	(196)

第九章 文件类型

§ 9.1 磁盘文件的逻辑组织	(198)
§ 9.2 随机文件	(200)
§ 9.2.1 随机文件的类型定义	(200)
§ 9.2.2 打开一个随机文件	(201)

§ 9.2.3 随机文件的读写	(202)
§ 9.2.4 关闭文件及文件处理函数	(204)
§ 9.3 随机文件应用举例	(205)
§ 9.4 TEXT文件	(210)
§ 9.5 无类型文件	(213)
• § 9.6 标准文件	(215)
• § 9.7 I/O检查	(218)
习题九	(219)
第十章 递归	
§ 10.1 递归的概念	(221)
§ 10.2 递归过程和函数	(223)
§ 10.2.1 递归过程	(223)
§ 10.2.2 递归函数	(228)
§ 10.3 间接递归和向前引用	(230)
§ 10.3.1 间接递归	(230)
§ 10.3.2 向前引用	(230)
§ 10.3.3 应用举例	(231)
习题十	(234)
第十一章 动态数据结构	
§ 11.1 指针类型与动态变量	(237)
§ 11.1.1 指针类型	(237)
§ 11.1.2 new标准过程与指针变量	(238)
§ 11.1.3 动态变量的访问	(239)
§ 11.1.4 dispose标准过程	(240)
§ 11.2 链表	(241)
§ 11.2.1 链表与递归数据结构	(241)
§ 11.2.2 链表的建立	(243)
§ 11.2.3 链表的插入与删除	(246)
§ 11.2.4 链表的检索	(250)
§ 11.2.5 循环、双向链表	(253)
§ 11.2.6 栈	(255)
§ 11.3 二叉树	(256)
§ 11.3.1 二叉树的概念	(256)
§ 11.3.2 二叉树的遍历	(257)
§ 11.3.3 二叉树的建立与插入	(259)
§ 11.3.4 二叉树的检索	(260)
• § 11.4 TURBO PASCAL 中的指针操作	(260)
§ 11.4.1 指针和地址函数	(261)
§ 11.4.2 堆	(262)
§ 11.4.3 内存动态分配、释放标准过程	(263)
习题十一	(266)

第十二章 结构化程序设计

§ 12.1 问题的提出	(268)
§ 12.2 三种基本结构及结构化流程图	(272)
§ 12.2.1 顺序结构及其结构化流程图	(272)
§ 12.2.2 分支选择结构及其结构化流程图	(273)
§ 12.2.3 循环结构及其结构化流程图	(276)
§ 12.3 自顶向下的设计方法	(278)
§ 12.4 goto语句	(288)
§ 12.5 TURBO PASCAL高版本系统中的模块功能	(291)
习题十二	(303)

第十三章 TURBO PASCAL的作图与音响功能

§ 13.1 屏幕模式控制	(305)
§ 13.1.1 正文模式	(305)
§ 13.1.2 图形模式	(307)
§ 13.2 窗口	(308)
§ 13.2.1 正文窗口	(308)
§ 13.2.2 图形窗口	(309)
§ 13.3 图形功能	(310)
§ 13.3.1 基本图形功能	(310)
§ 13.3.2 扩展图形功能	(310)
§ 13.4 龟作图	(314)
§ 13.5 作图例程	(315)
§ 13.6 音响	(318)

第十四章 TURBO PASCAL的特殊功能

§ 14.1 包含文件	(321)
§ 14.2 覆盖系统	(322)
§ 14.2.1 覆盖系统的原理	(323)
§ 14.2.2 覆盖过程的建立	(323)
§ 14.2.3 覆盖的管理	(325)
§ 14.2.4 覆盖的限制	(326)
§ 14.3 编译器指示	(326)
§ 14.4 TURBO PASCAL中的特殊类型和运算	(332)
§ 14.4.1 Byte类型	(332)
§ 14.4.2 特殊运算	(333)
§ 14.5 与内存绝对地址的联系	(334)
§ 14.5.1 绝对地址变量	(334)
§ 14.5.2 绝对地址函数	(335)
§ 14.5.3 预定义数组	(335)
§ 14.6 有类型常量	(336)
§ 14.6.1 非构造型有类型常量	(336)

§ 14.6.2 构造型有类型常量	(336)
§ 14.7 与系统接口程序的设计	(341)
§ 14.7.1 插入机器代码	(341)
§ 14.7.2 DOS的功能调用	(342)
§ 14.7.3 用户编写I/O驱动程序	(343)
第十五章 TURBO PASCAL上机操作	
§ 15.1 运行环境	(345)
§ 15.2 TURBO编辑器	(346)
§ 15.2.1 启动TURBO PASCAL系统	(346)
§ 15.2.2 TURBO编辑器命令及使用	(350)
§ 15.3 TURBO编译器选择项	(356)
附录一 TURBO PASCAL快速参考手册 (V3.0)	(361)
(一) 标准过程和函数	
(二) TURBO预定义常量和变量	
(三) TURBO PASCAL的扩展图形过程和函数	
(四) TURBO PASCAL的各种运算	
(五) TURBO PASCAL编辑程序键盘操作指南	
附录二 TURBO PASCAL与标准PASCAL的对比	(371)
附录三 编译器错误信息	(372)
附录四 运行过程中的错误信息	(375)
附录五 I/O错误信息	(376)
附录六 TURBO语法	(378)
附录七 ASCII码表	(387)
附录八 键盘返回码	(388)
附录九 TURBO PASCAL的内部数据格式与内存分配	(391)
1 基本数据类型	
2 数据结构	
3 参数	
4 函数结果	
5 堆和栈	
6 内存分配	
参考书目	(396)

第一章 计算机基础知识

我们通常所说的计算机是指电子数字计算机，它是用电子线路实现数字和逻辑运算的工具。

计算机自1946年问世以来，已经经历了电子管计算机、晶体管计算机、集成电路(IC)计算机、大规模集成电路(LSI)计算机四个发展阶段。现在又开始了对所谓第五代“智能”计算机的研究，并已取得很大进展。随着计算机技术的发展，计算机的应用也越来越广泛。目前，计算机主要用于以下几个方面：(1) 科学计算；(2) 数据处理；(3) 生产过程自动控制；(4) 计算机辅助管理；(5) 计算机辅助设计；(6) 人工智能等等。

计算机主要特点是运算速度快（目前已有每秒运算数十亿次的计算机）、计算精度高、具有很强的记忆功能和逻辑判断功能，而且能自动连续地高速运算。

§ 1.1 计算机硬件软件结构

计算机是由电子元、器件构成的机器，它所以能脱离人的直接干预，自动地进行计算，是由于人把实现这个计算的一步步操作，用命令的形式——即一条条指令预先输入到计算机能存储信息的部件（称为存储器）中，执行时机器把这些指令一条条地取出来，加以翻译和执行。

对两个数相加这一最简单的运算来说，就需要以下几步（假定要运算的数已在存储器中）：

- 第一步：把第一个数从存储器中取出来，送至计算机的运算部件（称为运算器）；
- 第二步：把第二个数从存储器中取出来，送至运算器；
- 第三步：相加；
- 第四步：把加完的结果送至存储器；
- 第五步：用打印机打印出结果；
- 第六步：停机。

所有这些取数、送数、相加、存数和打印等都是一种操作，我们把要求计算机执行的各种操作用命令的形式写下来，这就是机器指令。通常一条指令对应一种基本操作。这些机器指令按一定顺序排列起来组成完成解题任务的序列称为程序。

机器指令必须满足两个条件。一是机器指令的形式计算机能理解（为此机器指令通常用数字编码形式表示）；二是机器指令规定的操作是计算机能执行的。计算机所能执行的全部指令，就是计算机的指令系统，它是由设计人员设计计算机时用相应的电子线路保证的，这是计算机固有的，不同的计算机有不同的指令系统。

从上面的叙述中可以看出，计算机需有若干电子线路如存储器和运算器等，还有一些机电装置如打印机等，它们通常被称为计算机的硬件。另外还需要由指令组成的程序，通常被称为计算机软件。

实际上，计算机是一个很复杂的系统，下面分别介绍一下它的两个组成部分：硬件系统

和软件系统。

§1.1.1 计算机硬件系统

计算机的硬件系统是由电子线路和机电装置组成的。它是计算机系统的物质基础。硬件系统的基本功能就是能够执行预先设计好的指令系统中的各种指令，即能够运行由这些指令编制的各种程序。

其基本结构如图1-1所示，包括如下几个部分：

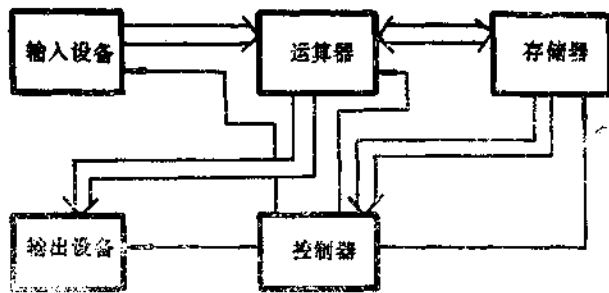


图1-1 计算机的结构图

1. 存储器

存储器的主要功能是保存大量的信息，它的作用类似一台录音机，能把记录的内容保存下来。使用时，可以取出原记录内容，而不破坏其信息；也可以把原来保存的内容抹去，记录新的内容。

在计算机内部设有一个内存储器，简称内存。计算机计算之前，程序和数据通过输入设备送入内存，计算开始后，内存不仅要为其他部件提供必需的信息，也要保存运算中间结果及最后结果，总之它要和各个部件打交道，进行数据传送。

内存由许多存储单元组成，每个存储单元可以存一个数据代码，该代码可以是指令，也可以是数据。为区分不同的存储单元，就把内存中全部存储单元按一定顺序统一编号，习惯上把这种编号称为地址编码，简称地址。计算机对内存采用按地址存取的方式。当计算机要把一个信息代码存入某存储单元或从某存储单元中取出时，首先要给出该存储单元的地址，然后按地址“查找”对应的存储单元，查到后，才能进行数据的存取。这和我们去大楼里找人，要按照他的地址（房间号）寻找他的过程类似。

由于价格和技术方面的原因，内存的存储容量受到限制。为了存储数量更大的信息，就需要采用辅助存储器。它们设置在计算机主机外部，又称外存。

外存用来存放使用不频繁的程序和数据。外存容量比内存大得多，通常存储在外存上的数据要调入内存才能被计算机使用。它存取信息的速度比内存慢。计算机对外存不是按单个数据进行存取，是以成批数据进行处理。

外存种类较多，如磁带、软磁盘和硬磁盘存储器。它们也归属计算机外部设备。

2. 运算器

运算器完成数据的运算。“运算”这个词在计算机学科中的含义比通常理解的更为广泛，它不仅是指加、减、乘、除等基本运算，还包括逻辑判断、逻辑比较以及其它基本逻辑运算。这些都是简单的基本运算。复杂的问题在计算机中只能用简单的运算一步一步实现。然

而，计算机的运算速度却非常快，因此它的能力是用高速赢得的。

运算器中的数据取自内存，运算结果又送往内存，但在内存中存取数和在运算器中进行运算，都是在计算机的指挥机构——控制器控制下进行的。

3. 控制器

控制器的主要作用是使整个计算机能自动地执行程序。控制器控制着整机各部件协调地工作、执行内存中存储的程序。

控制器要从内存中按顺序取出各条指令，每取一条指令，就分析这条指令，然后根据指令的功能向各部件发出控制命令，控制它们执行这条指令中规定的任务。图 1-2 中各虚线便是控制命令信息的通路，实线是数据代码信息的通路。当各部件执行完控制器发出的命令之后，会发出汇报命令执行情况的“反馈信息”。当控制器得知一条指令执行完后，会自动顺序地去取下一条要执行的指令，重复上面的工作过程。只是对不同的指令，发出不同的控制命令而已。这些命令是在计算机中由控制器按一定的时序发出的各种电信号来实现的。

微型计算机系统中，通常把运算器和控制器合在一起称为中央处理器，简称 CPU (Central Processing Unit)。中央处理器、内存和输入输出接口组成计算机主机。

4. 输入设备

单是主机仍无法工作，因为程序和数据无法送到计算机内存中存储起来。因此，还须有输入设备，输入设备的作用是把要输入的程序和数据信息通过输入接口顺序地送往计算机内存中。通常配置的输入设备有磁带输入机、键盘、磁盘输入机等。

5. 输出设备

计算机还必须配置输出设备，以便把运算结果按人们需要的形式送出。计算机主机通过输出接口把数据送至输出设备。通常配置的输出设备有打印机、显示器等。

输入设备和输出设备都是计算机的外部设备。前面曾提到的外存也归属于外部设备。由磁带和磁盘上保存的信息，可以直接送到内存，已储存在内存中的信息也可以输出记录到磁带和磁盘上。因此，磁带机、磁盘机也可以作为输出设备。

§ 1.1.2 计算机软件

从上面的讨论中我们知道，仅有硬件计算机还无法工作。通常称计算机硬件系统为裸机。要计算机运行起来并解决各种问题，必须给它编制由一条条指令组成的各种程序。随着计算机技术的发展，不仅有用户自己编制的计算程序，还要有专为计算机配置的具有特殊功能的各种程序，这些程序通称为软件。计算机系统的软件内容非常丰富，大致可分四层。如图 1-3 所示，外层可以利用内层提供的手段。

操作系统是对计算机系统资源（包括硬件和软件等）进行管理和控制的程序，使计算机系统能协调地、高效地对数据进行处理，从而使用户方便的完成各项计算任务。如目前在微

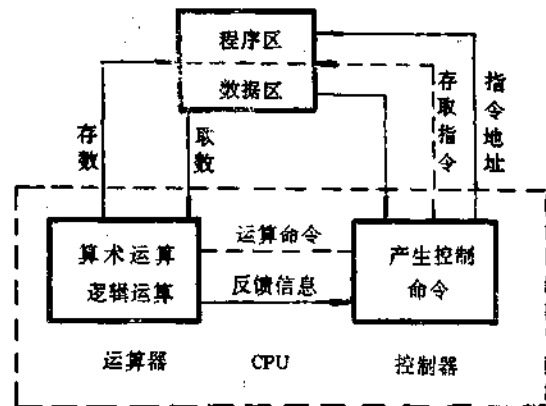


图 1-2 主机结构

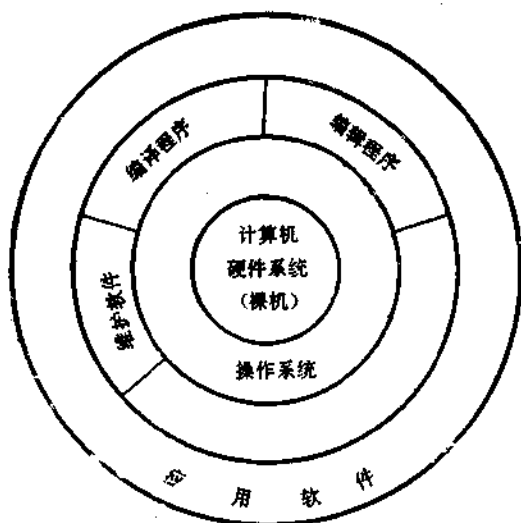


图1-3 计算机软件示意图

具，它也属于计算机软件范畴。

每一台计算机都有它自己的一套指令系统，这是设计计算机时所确定的，是计算机所固有的。指令系统中的每一条指令称为机器指令。一般来说，机器指令是由告诉计算机进行什么操作的操作码和告诉计算机到哪里去找操作数的地址码这两部分构成。通常一条指令对应着一种基本操作。机器指令的形式是由0和1组成的二进制编码。所谓机器语言是指机器指令的集合。机器语言是面向机器的，用机器语言编写程序，就是根据要解决的具体问题写出一条条机器指令组成的程序。用机器语言书写程序十分不方便，只能由计算机专业人员编写。而且每一种机器都有自己特定的机器指令系统，互不通用。

由于用二进制编码来编写程序难于阅读，后来出现了“汇编语言”。它用特定的助记符号，即用帮助人们记忆的符号来代表二进制编码（机器指令），它和机器指令是一一对应的。用汇编语言编写的程序称为汇编语言源程序。但计算机对它不能直接执行，只有经过汇编程序（Assembler）翻译成机器语言的程序才能执行。这个翻译过程就称为汇编。如图1-4所示。

汇编语言虽然比机器语言前进了一步，但仍比较烦琐，无通用性。

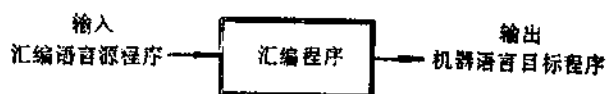


图1-4 汇编过程示意图

五十年代末，发明了“程序设计语言”即高级语言。它是由各种类似自然语言的具有不同意义的“符号”（在PASCAL语言中称为保留字和标识符）和“表达式”，按照一定的语法规则组成的。用高级语言编写的程序同人们习惯用的自然语言（英语）、数学公式相似。高级语言的语句功能强，一个语句往往相当于许多条机器指令，从而使用高级语言编写程序更方便容易。一般的说，高级语言是独立于机器的，它是面向过程的。在编写程序时人们不需要对机器的指令系统有深入的了解，而且一个用高级语言编写的程序在不同型号的机

型计算机广泛使用的DOS操作系统、UNIX操作系统、CP/M操作系统等。

公用程序：包括各种编辑、服务程序。例如文本编辑程序、屏幕编辑程序，还有常用的数学函数库，软件包等；

编译程序：例如BASIC、FORTRAN、ALGOL、COBOL、PASCAL等语言的编译程序；

应用程序：用户为了解决某些具体科技的计算或信息处理而编写的程序。应用程序大多是用某种高级语言编写的。

§1.1.3 机器语言和高级语言

语言是人们交流思想的工具，计算机语言是人与计算机之间进行“对话”的工具。

器上都可以使用，或只做小小的改动。

当然，计算机并不能直接接受和执行高级语言编写的程序，运行前需要“翻译”，把用高级语言编写的程序（称为高级语言源程序）先翻译成机器指令程序，然后再让计算机执行机器指令。这个“翻译”不是由人担任的，而是有一个起“翻译”作用的“编译程序”。事先将它放到计算机内，然后由它对高级语言源程序进行翻译。这个翻译过程称为编译。如图1-5所示。

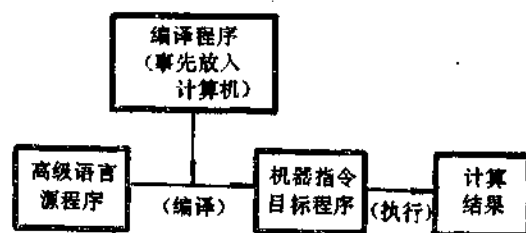


图1-5 编译过程示意图

由此可见，用高级语言编写的源程序送入计算机执行时，应分两步进行。第一步是进行编译，调用编译程序，将源程序翻译成机器指令组成的目标程序，如果编译无错误，就可进行第二步。第二步是运行，运行的程序把接收进来的数据进行处理加工，然后输出结果。这里强调一点，高级语言源程序的语句并没有和机器指令一一对应，这和汇编语言源程序不同。

目前国内外比较流行的高级语言有几十种，如FORTRAN、ALGOL60、COBOL、BASIC、PASCAL等。

每一种高级语言都有相应的“编译程序”，例如用FORTRAN编写的源程序，只有用FORTRAN编译程序才能把它翻译成机器指令程序。用PASCAL语言编写的源程序只能用PASCAL编译程序翻译成机器指令程序。还需指出，同一种高级语言，对于不同的计算机其编译程序也是不同的，因为每种机器有不同的指令系统。在计算机出厂时，厂方配备一种或几种语言的编译程序（存在软磁盘上）提供给用户。当我们使用某一计算机时，不能随便选用编译程序，而只能用该计算机随机配备的与相应语言对应的编译程序，方可对源程序进行编译。

另外还有一种处理高级语言的方法。如BASIC语言解释程序，在执行BASIC语言编写的源程序时，一条源程序由BASIC解释程序“翻译”成机器指令后执行，接着再进行下一条，这样一条一条地“解释”和执行。这种方法，程序运行速度比编译方法慢。

§ 1.2 数制、码制、编码

为了物理上便于实现和节省状态，计算机采用二进制数字系统。计算机处理的所有数、所有字母和符号，都要用二进制编码来表示。

§ 1.2.1 数制

这里所讲的数制是指进位计数制。计算机采用的二进制和我们习惯的十进制，都是采用进位计数制。除此以外，还有八进制和十六进制。

观察各种进制的计数方法，可以发现它们有两个共同点。一是采用进位计算方式，例如逢十进一；逢二进一；逢八进一；逢十六进一。而十、二、八、十六恰是各种进制中表示一