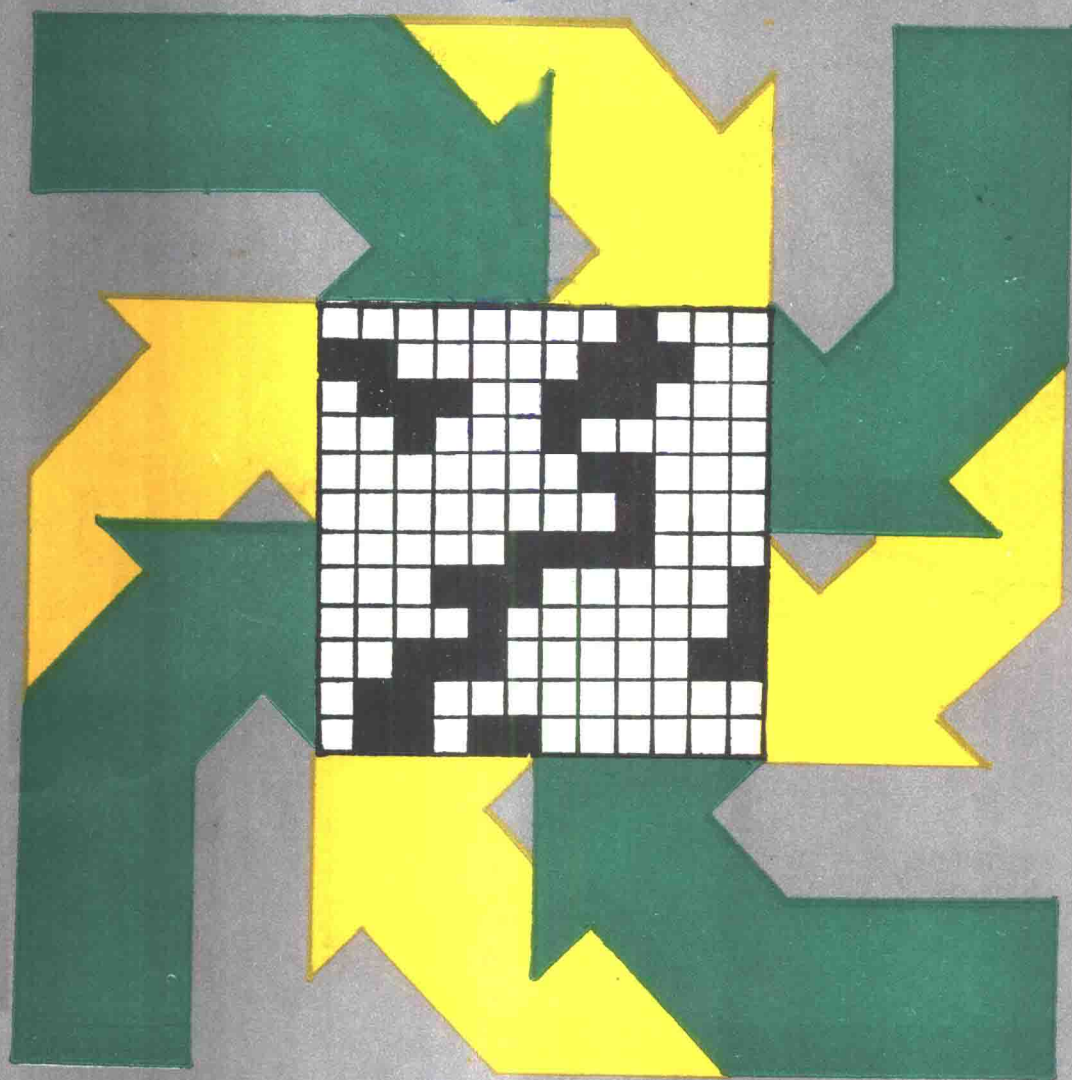


高等学校试用教材

程序设计技术

麦中凡 主编

高等教育出版社



高等学校试用教材

程序设计技术

麦中凡 主编

高等教育出版社

内 容 简 介

本书是为非计算机专业人员学习程序设计技术而编写的。在初步学习了一门程序设计语言之后,自然需要学习一些软件知识以提高实际设计程序的能力,本书是适应这方面需要的入门性教程。

本书的第一部分以软件工程观点介绍近代应用软件的开发方法、程序设计基本原理、程序结构与结构化程序设计、算法设计与数据结构、程序的风格与效率。第二部分围绕程序的测试介绍所必需的软件知识:程序运行环境,程序覆盖与虚存概念,数据管理与数据库。

本书以丰富有趣的例题作导引,并附有大量习题,适于读者深入自学。第三部分除一章专讲数值程序设计外,按软件工程方法演示了一个例子的开发全过程。

本书适用于非计算机专业高年级大学生、大专生、研究生选作教材,也可供计算机软件的初、中级工作人员参考。

高等学校试用教材

程 序 设 计 技 术

麦中凡 主编

高等教育出版社出版

新华书店重庆发行所发行

重庆印制一厂印装

开本 787×1092 1/16 印张 24.25 字数 600 000

1987年10月第 1 版 1987年10月第 1 次印刷

印数 00 001--4 240

ISBN 7-04-000224-8/O·267

书号 13010·01487 定价4.20元

序

本书是作者在北京航空学院研究生院开设“应用程序设计”课程的教材。“应用程序设计”是非计算机专业计算机语言课程的后续课,旨在提高学生的程序设计能力。

1980年开始在大学生中普及计算机程序设计语言时,我们发现,非计算机专业研究生单有“计算机原理”和“程序设计语言”的知识是难以应付毕业设计论文中程序设计内容需要的。他们为调试出500~2500句高级语言的源程序要花掉4~9个月,占一年毕业设计论文时间的1/3~3/4。这样,通过毕业设计“程序设计”倒是入门了,可本专业的课题却没有时间深入。为此,1981年我院为非计算机专业研究生开出“FORTRAN程序设计基础”课程,学时50。其前导课为“计算机原理”、“计算方法”、“程序设计语言FORTRAN”,后续或平行课程为“计算数学”。

这门提高课程没有成熟的教材。对非计算机专业学生他们需要的是独立编制、调试1500~2000句科技计算程序的能力。于是,教材组织可按两种方式:一种是以公共课“计算数学”为背景,结合覆盖面较大的实例讲解1000句左右的FORTRAN程序如何设计和测试。一学期讲5~6个典型问题,学了就能用。另一种是以应用程序需要的有关程序设计知识,举出说明这些系统知识的小型例题(当然不限于数值程序)。讲思想、讲方法,着重基础训练。“FORTRAN程序设计基础”采用的是后一方式,以伪代码的结构化程序设计为主体,并补充FORTRAN 77的有关知识(当时,FORTRAN 77刚开始普及)。

学生们为本教程自顶向下结构化程序设计方法以及大量有一定难度但很有兴趣的例题和习题所吸引,加上本课程十分强调上机实践,所以,尽管本教程没有多少篇幅讲数值程序,但通过一学期学习再去作毕业设计的数值程序,一般没有什么困难。这样,使我们对非计算机专业教学仍从程序设计基本训练入手增强了信心。肯定了本课程的基础性。本课程定为非计算机专业研究生公共基础课后,我们提出的课程目标是:

- 提高程序设计能力。100句高级语言源程序能自如地编写;1000句较复杂的程序能独立地设计和测试。

- 近代应用软件工程入门。了解近代应用软件研制的方法及其涉及的课程。

- 初步能以正确的方法和风格处理程序问题。

通过连续五届学生的教学实践,以上目标基本达到了。应该承认,学生为此付出的代价是比较大的,课内外学时比超过1:2。然而,就程序设计本质而言,你要成为程序设计好手,只要在正规的、正确的思想指导下多做难题、多做大题、多阅读程序范例就能达到。当同学们感到有用而且有兴趣时,他们愿意付出时间。

以后的几年,随着微机在我国普及,语言课上机条件的改善,特别是软件工程的兴起,及各专业计算机应用的发展,程序设计课程单单以FORTRAN为背景讲些实现的算法及编程技巧就不行了。同学们迫切地需要知道在近代计算机上开发程序的知识,诸如,什么是操作系统命令?虚拟存储是怎么回事;数据库解决什么问题;……原先要花很多时间弄懂的难题学

生接受起来已不很费力。个别非计算机专业程序设计语言课程也不再限于教 BASIC 和 FORTRAN 语言, 开始了 PASCAL 语言教学。

面对这些新的变化, 本教程也在不断改进。1984 年增加了程序运行和文件使用的内容, 1985 年增加了操作系统知识, 1986 年完全按软件工程观点改写了全书, 并改名为“应用程序设计”。书中删去了介绍 FORTRAN 77 部分, 除了必须涉及具体语言之处, 一律用伪代码讲述。具体语言一般指 FORTRAN 77, 很少 FORTRAN IV。

总的说来, 改进后的教材在理论性和深度方面浅显了, 删去了循环不变式和程序正确性证明有关章节。在系统性、入门性上作了较大改进。增加了与开发应用程序有关的章节, 程序测试方面给出了较完整的概念。在当今情况下回避用系统论、软件工程的观点去讲述程序设计, 如同只谈树木不谈森林。因而, 本教程以基础面宽为其特征。但是, 不是一般的概论, 而是始终围绕编程实践。旨在学生今后接触到应用软件时不感到陌生和难于理解。当然, 要把软件工程关于程序开发所涉及的各门学科浓缩在一本教程里是不容易的, 难免会有“不深不透”之嫌。为此, 我们在各章后指定了较多的参考书, 以便有兴趣的读者进一步深入学习。

各章的习题量也远远大于实际需要, 两小时课程一般只能布置 3~5 个题。如果是上机交程序清单 (鼓励这样做), 每次只有 1~2 题。多余的题目是为有时间、有兴趣的读者准备的。事实上, 所有习题放在一起足够作为一本习题集。

本教程力图回避“计算数学”和“离散数学”背景, 起点并不高。读者只要学过一门高级程序设计语言, 有集合和逻辑表达式的概念, 以及一般微积分的基础, 并有独立上机编制 4~5 个 30 句左右的小题目的经验, 就可以学习本书。因而, 本书可作为非计算机专业大学生的教材。但是, 在有限的学时里讲如此多的内容最好是不再去补语言和上机操作的知识。我们的办法是第一次课作语言水平测试。基本零分的学生 (20 分以下) 补作 8 道上机小题也可以跟班听课。事实证明, 这部分学生里面最后也有学得不错的。对于只会 BASIC, ALGOL 60 语言的学生, 我们并不鼓励他们改用 FORTRAN 做题, 相反, 我们认为结构化 BASIC 和 ALGOL 60 比 FORTRAN IV 更好, 只要上机条件允许就应坚持。至于 PASCAL 和 C 语言就更不用说了。

随着我国计算机基础教育的普及, 大学毕业论文使用程序的题目日益增多, 本教材可供选作高年级大学生的提高课教材。对于大学生除了习题和作业的要求可降低而外, 教程内容基本不变。因为本教程仅仅是建立软件工程时代程序设计的完整概念。如果上机条件好, 完全可以达到本课程的项目标。为此, 我们在本次修订中又删去了并行程序设计一章; 尽可能更换了以 IBM PC 微机为背景的例题; 补充了一批较为容易的习题; 对于难度稍大的题目均作了提示; 对于较为理论化的章节加注了“*”号, 教学中介绍其中一点概念就可跳过去。因为不同的读者可以, 也应当有不同的理解。本教程还没有一个难度较大的理论章节是贯彻始终的, 因而不影响其他部分的教学。

本书在历次修改过程中得到北京航空学院研究生院大力的支持。计算机系黄燕斌同志参与了 1986 年讲义修订工作。他提供了第二篇的一批例题资料, 起草了九、十、十一、十二章 (原来章节) 的初稿。但由于其他原因, 他不能参加本次修订工作, 我非常惋惜。本次修订工作中计算机系姜静波同志更换了大部分 IBM PC 例题并改写了第十四章, 同时为审题校订也付出了大量劳动。

本书得以出版要感谢全国高校计算机基础教育研究会谭浩强副教授, 教材编审组史济民、

金茂忠副教授。没有他们的支持和鼓励本书出版难于实现。还要感谢北京联合大学自动化工程学院田淑清副教授，她在百忙中审阅了全书并提出了宝贵的意见。同时，感谢在成书和历次修改中帮助过我的人：北京航空学院计算机系李伟琴副教授，飞行力学系李宁同志，计算中心刘江滨同志，以及刘虹同学。

袁中凡

1986年9月于北航

目 录

第0章 绪论	1	2.1.4 抽象与程序方法学	31
0.1 信息社会与计算机文化	1	2.2 子目标	32
0.2 电子数字计算机系统概述	2	2.2.1 嵌套子目标	32
0.2.1 计算机系统	2	2.2.2 并行子目标	33
0.2.2 计算机系统的特点	3	2.3 模块性	33
0.2.3 通用机与专用机	4	2.3.1 用户界面上的模块	33
0.2.4 计算机系统发展概览	5	2.3.2 模块划分	34
0.3 计算机应用的发展	6	2.4 局部化与信息隐藏	35
0.3.1 早期的应用	6	2.5 一致性、完整性、可验证性	36
0.3.2 当前应用方向	9	习 题	37
0.3.3 近期的研究与发展	11	参考文献	40
0.4 关于本书的说明	13	第三章 程序设计的步骤	41
习 题	15	3.1 软件生存期	41
参考文献	15	3.2 定义问题	42
		3.2.1 规格说明	42
		3.2.2 计算模型	43
		3.3 设 计	47
		3.3.1 总体设计	47
		3.3.2 详细设计	49
		3.3.3 设计的表示	49
		3.4 编 码	55
		3.5 测 试	55
		习 题	56
		参考文献	60
		第四章 程序结构与结构化程序设计	61
		4.1 程序控制	61
		4.1.1 语句级控制	61
		4.1.2 块级控制	63
		4.2 关于GO TO语句	64
		4.3 逻辑结构与形式结构	65
		4.4 块结构与局部化	65
		4.5 结构化程序设计	67
		4.5.1* 结构化程序的理论基础	67
第一篇 程 序 设 计			
第一章 程序设计导论	16		
1.1 程序设计的概念	16		
1.2 程序设计的特点	16		
1.3 程序设计概念的范畴	18		
1.4 程序设计准则	19		
1.4.1 正确性	19		
1.4.2 可靠性	20		
1.4.3 简明性	20		
1.4.4 有效性	21		
1.4.5 可维护性	22		
1.4.6 适应性	22		
1.5 程序质量	23		
习 题	23		
参考文献	26		
第二章 程序设计的基本原理	27		
2.1 抽 象	27		
2.1.1 数据的抽象	29		
2.1.2 控制的抽象	30		
2.1.3 抽象与模型	30		

4.5.2 伪代码	72	6.7.3 用数组代栈	174
4.5.3* GO TO语句的消除	76	习 题	175
4.5.4 结构化程序的设计	80	参考文献	181
4.6 结构化程序设计实例	86	第七章 程序的效率	182
习 题	92	7.1 正确的效率观	182
参考文献	97	7.2 程序优化	184
第五章 算法设计	98	7.2.1 不同层次的优化	184
5.1 算法的概念	98	7.2.2 局部优化和全局优化	184
5.2* 算法的性质	99	7.3 FORTRAN程序优化的措施	185
5.3 基础算法	100	7.3.1 时间优化	185
5.3.1 枚举法	100	7.3.2 空间优化	196
5.3.2 归纳法	107	习 题	199
5.3.3 回溯法	117	参考文献	202
5.3.4 模拟	120	第八章 程序的风格	203
5.4* 算法分析	126	8.1 什么是程序的风格	203
5.4.1 算法的时间复杂性	127	8.2 设计的风格	204
5.4.2 算法的最优性	130	8.2.1 结构应清晰	204
习 题	133	8.2.2 程序应简明	204
参考文献	140	8.2.3 Hrair极限	204
第六章 数据结构	141	8.3 语言运用的风格	205
6.1 数据结构的概念	141	8.3.1 关于程序结构	207
6.2 数据结构分类	142	8.3.2 关于过程	208
6.3 表	143	8.4 程序正文表示的风格	208
6.3.1 向量和数组	143	8.4.1 充分利用注释行	209
6.3.2 链表	147	8.4.2 一致性与助忆	210
6.3.3 串	148	8.4.3 正文编排格式	212
6.3.4 队列	148	8.5 关于程序的输出	216
6.3.5 栈	149	8.6 怎样阅读易读性不好的程序	217
6.4 图	152	习 题	217
6.4.1 图的机器表示	153	参考文献	224
6.4.2 图的运算	154		
6.5 树	164	第二篇 程序设计实现	
6.5.1 树结构及其表示	164	第九章 程序设计基本环境	225
6.5.2 树的运算	166	9.1 程序实现的软件支持	225
6.6 文 件	168	9.1.1 程序实现的各步骤	225
6.6.1 文件的概念及其逻辑结构	168	9.1.2 基本的软件环境	230
6.6.2 文件运算	169	9.2 操作系统简述	231
6.7* 递归程序的非递归变换	169	9.2.1 多道程序运行	231
6.7.1 尾递归	169	9.2.2 分时系统	232
6.7.2 设中间信息堆栈	171	9.2.3 操作系统的组成及功用	233

9.3 人机接口	235	12.1.1 完全测试不可能	288
9.4 典型的操作系统	236	12.1.2 测试的经济性	289
9.4.1 UNIX	237	12.2 测试的基本原则	289
9.4.2 CP/M	240	12.3 测试用例设计	290
习 题	242	12.3.1 逻辑覆盖测试	290
参考文献	242	12.3.2 黑箱测试技术	293
第十章 程序覆盖与虚存概念	243	12.3.3 测试用例的设计策略	299
10.1 存储管理	243	12.4 不同层次的测试	299
10.2 程序覆盖	244	12.5 程序测试的策略	301
10.2.1 覆盖原理	244	12.5.1 由底向上测试	301
10.2.2 程序分节和覆盖树	245	12.5.2 渐增测试	302
10.2.3 程序覆盖实例	246	12.5.3 自顶向下测试	303
10.3 虚拟存储技术	249	12.6 程序排错	304
10.3.1 虚存原理	249	12.6.1 错误分类	304
10.3.2 多级存储体系	250	12.6.2 排错技术	308
习 题	251	12.6.3 排错原则	315
参考文献	251	12.6.4 测试与排错工具	316
第十一章 数据管理与数据库	252	习 题	316
11.1 文件系统	252	参考文献	318
11.1.1 逻辑文件与物理文件	252		
11.1.2 文件的属性和分类	253	第三篇 系 统 开 发	
11.1.3 文件管理系统的功能	254	第十三章 数值程序简论	319
11.2 对文件的操作与使用	254	13.1 数值程序特点	319
11.2.1 对文件的操作	254	13.2 数值程序常用算法	324
11.2.2 应用实例	255	13.3 数值程序设计要点	327
11.3 数据库	261	13.4 数值程序开发实例	328
11.3.1 数据库系统	262	习 题	333
11.3.2 数据的相关性和独立性	263	参考文献	334
11.3.3 数据库的优缺点	265	第十四章 系统开发实例	336
11.3.4 数据库语言	266	14.1 系统描述	336
11.3.5 关系数据库原理	267	14.2 软件计划	336
11.3.6 关系数据库应用	271	14.3 需求规格说明书	338
11.4* 程序再入	278	14.4 设计文档	340
习 题	286	14.5 用户界面描述	369
参考文献	286	14.6 确认测试规格说明书	374
第十二章 程序的测试与排错	287	参考文献	378
12.1 测试与排错的基本概念	287		

第0章 绪 论

0.1 信息社会与计算机文化

人类制造工具增进了人类征服自然的能力。生产工具的革新,促使社会生产力达到新的水平。医用工具的改进,挽救、延长了更多人的生命。现代运输工具,使人类得以驰骋乾坤。艺术工具(书、报、画笔、乐器)为我们创造了日益增进的精神文明。人类总是为了某种特定的需要去创造工具,工具使我们更有效地去完成前所不能做到或做好的事情。

计算机出现以前的工具,不断地延展人类的体力和感观能力,使人们能在更大的范围内驾驭力、能、声、光、电,创造出现代工业化的社会。计算机延长了人脑,使人们得以高速采集、贮藏、加工信息,使人类进入信息社会。

正确的信息可以使人们有效地利用所据资源创造出更加合理的前景。后工业化社会告诉我们,尽管生产力大大地提高了,但若没有正确的信息,人们辛辛苦苦创造的机器没有开封就要回炉,生产的农作物会在仓库里烂掉。人们在生产斗争、社会斗争中一刻也离不开正确的信息。然而获取正确信息并非易事。科学家花毕生精力也只能在他的专业上取得为数不多的正确信息。在现实社会中,由于信息不足或不确,打败仗、走弯路、赔本、事倍功半、发展缓慢乃至一无所获、全军覆没的事例举不胜举。也许我们今日以为在当今环境下做得不错的事,五十年后我们的子孙会说:“他们本来还可以做得更好,由于缺乏某些知识……”因而,一旦有了高速处理信息的工具,人们就尽最大的可能收集、积累、利用信息,使我们做的每件事花费的代价最少,取得的成果最大。

信息的概念范畴是什么呢?信息是对客观规律表示出来的映射。它在逻辑上是知识、经验、形态、情报、信号。在物理上是书报、图片、符号代码、数字、语言、声音、光线、电波……,就计算机而言,信息的物理存在就是数据。严格地说:

信息是现实事物间根据某种自然规律或人为的约定建立联系的一种形式,是被表示出来的事物的增添了的确定性,或者被消除了的不确定性。

信息社会以大量采集、加工、贮藏、生产信息为其主要特征。各生产生活部门利用电子计算机和电子通信设备有目的地收集、生产、传送、存储、利用信息。其结果,工农业生产率成倍增长,新的工作领域大量出现,人们的物质文化生活水平高度、快速地发展,生产、生活各领域全面信息化。

社会信息化导致大量知识涌现。新知识领域日益增多,知识的老化和更新加速,这就是社会学家讨论的“知识爆炸”问题。事实上当今世界正面临知识爆炸的情景,试看1985年的统计数字:

全世界发表的科学论文,每年500万篇,其中包含新知识的论文每天13000~14000篇,即每6秒钟即有一篇科学论文问世。

全世界每年出版图书50万种(不是册!)每秒钟就有一种新书出版。

据联合国“世界科学技术情报系统”统计,60年代科学知识每年的增长率从9.5%提

到10.6%，而80年代头几年是12.5%，即5年翻一番。

知识老化速度由20~30年的更新周期变为5~7年，而对生物工程、有机化学合成、计算机科学等新兴科学，其更新周期只有3~5年。

如果说一位化学家每周拿出40小时（一周的工作日！）浏览化学方面的论文和著作，一年内发表的论文和著作他就要读48年。

在知识爆炸面前，我们的学习生活也要有相应的、深刻的变化。如果没有学会高速处理信息的技能，很难跟上时代的步伐。

一定的社会生产方式就有与之相适应的文化，没有这种文化，社会的成员就难以运用当代的生产和生活工具。工业化社会要求它的劳动者具有初等中学的文化水平。信息社会要求它的劳动者具有一定的处理信息的能力，也就是要会使用信息处理工具——计算机。这就是所谓的“人类第二文化——计算机文化”。

我国是否为信息社会？何时信息化？这当然要由社会学家来回答。但是全世界各国都在迅速地信息化，这是不以人们意志为转移的客观规律。因此，作为现代的科技人员，除了本专业业务而外，还要接受计算机文化，继而去推动整个社会信息化，这是理所当然的了。

0.2 电子数字计算机系统概述

电子数字计算机（以下简称计算机）是当今处理信息的主要工具。有关计算机的介绍，读者已经或多或少有所了解。这里只是系统地作一个复习。

0.2.1 计算机系统

我们使用的计算机，实际上是一个系统。运算器加上控制器我们称之为中央处理器（CPU），连同主存储器则构成计算机的主体，我们叫它主机。除了主机而外，所有用金属、塑料、玻璃制成的设备，我们叫它外部设备（相对于主机而言内外），简称外设。如终端、读卡机、磁带机、打印机、输出/输入装置等。习惯上把所有的硬设备称为硬件。

当今计算机使用的最基本的运算仍是二进制运算。其工作原理是，将数据和命令机器对这些数据进行运算动作的指令都存放在存储器里，由控制器逐条取出指令，解释（译码）执行。动作的结果是将存储器单元里的内容（值）不断地改变。这就是Von Neumann原理。Von Neumann机的特点是程序执行的顺序性以及改变数据值的强制性。

应用计算机就是把计算的意图通过指令表示出来。计算机执行了装入存储器中的指令就实施了计算。这些指令序列就叫做程序。

计算机不同于计算器主要是它能自动计算。人们为了更方便有效地使用计算机，编制了许多自动完成计算、协调主机和外设的各种程序。在管理程序执行期间输入用户程序并计算。管理程序自动调入各种辅助程序，使应用程序不需顾及机器内部的操作。从而大大方便了人们的应用。这些系统内部的程序叫系统程序。用户为着某种实用目的编制的程序叫应用程序，或用户程序。

一般说来，系统程序是经常要用到的。一开机就离不开它，所以常把它们存储在磁带或磁盘中，随用随调。此外，卓有成效的应用程序，也把它们存放在系统的设备内。它们都是为用户使用机器方便而存储在软介质（纸带、磁带、卡片、软盘）上的程序，传统上称为软件。

• • •

显然,按用途它们又分为**系统软件**和**应用软件**。

当今成熟的软件有用电路固化下来的趋向,因而不能仅从介质形态去理解软件.软件仅是程序(或信息)的集合,其功用是使得用户使用机器方便。

软件和硬件是计算机中既相联系又相互独立的部分.相联系指计算机没有软件仅仅是裸机.裸机只能在很不方便的情况下做点简单的题目.正如只有光板床,冬天不能睡觉一样.没有硬件的执行去实施软件的功能,软件毫无意义.然而,当今的软件开发已脱离了某种具体型号的机器,软件自成行业.软件工程是研究大量社会化生产软件的学科.它标志着软件已完全独立于硬件而自成体系。

按照软/硬件商品产值统计,随着计算机的发展,1970年购置一个计算机系统,软、硬件的成本各占50%(如图0.1).而50年代软件费用不足20%,90年代相反,软件从硬件的附属物逆转为硬件按软件要求设计。

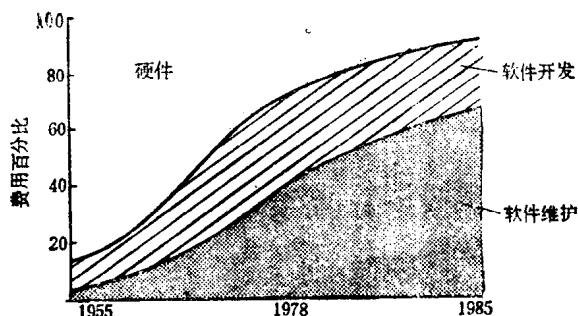


图 0.1

0.2.2 计算机系统的特点

计算机由于它有一系列特点,所以发展特别迅速.因而社会信息化的势头不可低估.它的特点是

● 普适性

计算机是处理信息的工具.如前所述,各行各业都有信息,都需要处理信息.这就决定了社会对计算机的需要量非常大.这也是计算机工业迅速发展的前题。

● 技术密集

计算机以前的工具无论如何发展总是要保持一定的物理属性,而计算机是信息和处理信息的载体,只要能表示信息并处理这些信息就可满足使用要求.50米长的超级轰炸机缩小100倍,只能充当玩具,缩小1000倍只能是装饰品,缩小十万倍什么也不是.然而,50毫米长的真空管是信息的一个表示单位(bit),即表示1或0,缩小十万倍仍可以在超大规模集成电路中充当一个表示单位,表示1或0.因此,技术越高,体积越小,耗用物质材料愈少.硬件成本10年以两位数字递减.这就决定了计算机工业发展的高速性.正是由于消耗的社会物质财富越来越少,占用的精神财富愈来愈多.它是典型技术密集型产品.产品的更新换代特别容易.所以世界上还没有任何一种工业,其产品更新速度能与计算机媲美.在短短的40年里,硬件更迭了几代.软件更新随时都在进行。

● 运算的高速性

计算元件集成度愈高,电子走的路程越短,运算速度愈高.世界上第一台数字计算机运算速度为每秒做5000次加法.到目前每秒作4亿次加法的计算机已投入使用.运算的高速性是处理复杂问题的前题,因而运算速度一直是衡量计算机先进性的指标之一。

- **信息容量大**

计算机容量大小,说明计算机贮藏和处理信息的能力.由于元件集成化和运算速度的提高,各型计算机内存存储器日益增大.海量存储器几乎可以不受限制地加大.信息量大也是处理复杂问题的前提,所以计算机容量成为衡量计算机先进性的指标之一.一般以千或兆(千千)字节(KB或MB)度量内存.外存以兆字节度量.

- **精确度高**

表示一个数据的二进制位位数愈多(即字长越长)计算精度愈高.由于软件技术的发展,字长可以成倍地增加,因而可满足任意精确度要求.主存及运算器中设计的物理字长,也是衡量计算机的指标之一.目前4, 8, 16, 32, 64, 128位机均有型号.4, 8位是单板控制机,8, 16位是微型计算机.超级微型机和中型机是16, 32位.大型机和巨型机有64, 128位的.一般通用机32位作科技计算和数据处理精度已令人满意,因而最普遍.

- **使用方便**

计算机是电子设备,很容易和电子通信设备联接.由于计算机软件技术的发展,通过远程终端、计算机网络和通信卫星,使用计算机几乎不受任何地域限制.我们可以在北京利用美国或欧洲的信息资源.

现代计算机一般为用户提供良好的使用环境.好的软、硬件配置能方便用户设计、测试、查询程序.用户通过软件(操作系统)可以方便地使用主机与外设.因而,就用户观点,计算机配备什么样的操作系统,有哪些软件支持也是衡量计算机先进性的参考因素之一.

0.2.3 通用机与专用机

市售计算机一般是通用电子数字计算机.

通用机提供常用语言软件,配备常规外设和操作系统.可作一般科技计算和数据处理、教学与科研.

根据不同用途通用机分为巨、大、中、小、微型.其大小按内存存储器容量区分.由于硬件集成迅速提高,这种区分界线很难固定.过去一般认为64KB以下为微机,256KB以下是小型机,1MB以下是中型机,4MB以上是巨型机.而如今超级微型机一般可达2MB,超级小型机可达4MB.微机IBM PC/AT内存1MB.微机长城0520C内存512KB.

通用机有时也有侧重,例如UNIVAC系列就侧重于科技计算,可提供128位字长的运算.再如,32位APPLE就是图形功能很强的超级微型机,并配备绘图机.

如果单为某种用途开发的计算机系统叫专用机,它多半是用通用的单板和集成元件的部件组建的,并配备有专用软件.其特点是专业性强,结构紧凑,功能简明.例如,飞机和导弹用机载、弹载机、实时控制机.在使用过程中它们只需程序正确,很少或不需要终端显示,也无需高级语言.因而传统上开发这类应用软件用本机的机器语言,这就导致开发周期长.近代软件技术可以在通用机上开发专用机的软件.开发其他机器所需软件的机器叫宿主机(host),使用该软件系统的叫目标机(target),开发出的计算机系统叫嵌入式系统(Embedded System).

还有一类机器系统叫开发系统,它的软件功能一般比通用机强,可对被开发的目标系统作模拟测试,亦可向只读存储器(ROM)写入信息等.

0.2.4 计算机系统发展概览

传统的断代原则是硬件元件的更新换代。计算机发展至今日已有五代，但是第三代之后硬件元件只有集成度大的量变，没有质变，因而划分界线不十分明晰。但第三、四、五代机在软件方面差别明显。以下表格是五代计算机发展简表。

特 征 项 目		第 一 代 1946~1958	第 二 代 1958~1964	第 三 代 1964~1971	第 四 代 1971~	第 五 代 1982~
硬 件	代 表 元 件	电 子 管	晶 体 管	集 成 电 路 SSL, MSL	集 成 电 路 LSL, VLSI	集 成 电 路 SLSI
	存 储 器	磁鼓 水银延时电路 磁芯 磁带	磁芯 磁带	磁芯 磁盘	半导体存储器 磁盘	半导体存储器 磁泡存储器 磁盘 光盘
	代 表 机 型	IBM-704 (1953.4) IBM-650 (1954.11)	IBM-7090 (1959.11) IBM-7094 (1962.9)	IBM-370 (1971) IBM-360 (1964) PDP-11系列	巨型 ILLIAC-IV, 1.5~2亿次/秒 (1973) CRAY-1(1976) 大型 IBM3033.3081 M-280.200 超级小型 VAX-11/780 超级微型 MC68000 Intel iAPX432 微型机 MC6800 Intel 8086 Z 8000 IBM PC	
软 件	代 表 性 软 件	汇编语言 FORTRAN	批处理管理系统 FORTRAN W (1960) COBOL(1960) PL/1(1964)	分时操作系统 IBM360交互式 语言 BASIC(1964) 结构程序语言 Pascal(1970) 软件形式产业	软件工程工具 数据库 大型程序系统 并发嵌入式语言 Ada, Modula 软件开发环境 网络软件	知识库 人工智能语言 知识工程 智能数据库 机器人系统 专家咨询系统

注：关于集成电路集成度作以下说明：

SSI——小规模集成 (Small Scale Intergration) 电路。指在单个硅片上集成10个以下门电路或100个以下晶体管的集成电路。

MSI——中规模集成 (Medium Scale Intergration) 电路。指在单个硅片上集成10~100个门电路或100~1000个晶体管的集成电路。

LSI——大规模集成 (Large Scale Intergration) 电路。指在单个硅片上集成100~1000个门电路或1000~10000个晶体管的集成电路。

VLSI——甚大规模集成 (Very Large Scale Intergration) 电路。指在单个硅片上集成1000~10000个门电路或10000~

100000个晶体管的集成电路。
SLSI——超大规模集成(Super Large Scale Intergration)电路。指在单个硅片上集成10000个门电路以上或100000个晶体管以上的集成电路。

0.3 计算机应用的发展

人们为了某种应用去发展一种工具，工具发明后又大大地扩充了应用范围。在科技发明史上这种事例多不胜举，中国爆竹就是最明显的例子。谁会想到驱邪避瘟，渲染气氛的吉祥物的后代竟成了战争的“瘟神”？从爆竹—火炮—原子弹—氢弹的演变大约经历了1200年。所以在一代人的时间里还不致于把它们的应用弄混。计算机由于演变太快，以致于还有相当多的人认为计算机只是数值计算工具，它不过比计算尺和机械计算机更快、更准罢了。殊不知它已经能在多方面代替人们作简单脑力劳动，并成为我们学习、生产、科研、生活有力的辅助工具。因而，在非数值计算方面应用的开发，很可能成为使用计算机的主导方面，科技领域也不例外。

本节综述计算机应用的各方面。

0.3.1 早期的应用

● 科学与工程计算

电子数字计算机一开始就是为了科学与工程计算而设计的。第一台数字计算机ENIAC就是“电子数值积分和计算机”(Electronic Numerical Integrator And Calculator)的缩写。它用于美国阿贝丁火炮试验场，计算导弹弹道。

人们发明了各种不同的数学方法去描述客观事物，正确地映射各事物数量间的关系，数学描述一般是对该事物定性的描述。而真正解决问题还要靠精确的计算，即定量。不幸的是，定量问题一开始就遇到障碍。早期的天文航海测量就无法精确地计算 π ， e ， $\sin\theta$ 等超越量和超越函数。为求得精确计算只好求助于“数值分析”：用一个代数运算序列去表示非代数数量，转换为代数数量就可用手工准确算出。然而，精确度越高，计算量越大。虽经多方努力，但多数精确计算的计算量仍然很大，简直到了手算一辈子也算不完的地步。

人们渴求快速准确的计算工具由来已久。但直到计算机出现之前，要快只好用计算尺，但精度只有两、三位，要精确则要用对数表插值。Gauss八位对数表曾在工程计算中作出了杰出的贡献，但科学每一步的进展都要求更加精确的计算，八位精确度则远远不够了。例如，射电天文望远镜已可探测145亿光天的天体(145亿光年换成公里是 1.3776×10^{23} 公里)。如果第八位数有误差就可差出 1×10^{17} 公里的距离，它是地球到太阳的距离(1.45×10^8 公里)的六亿八千九百万倍！因此，精确计算不解决，进一步发展是不可能的。

我们的数学工具，描述问题的能力远远超出我们计算它的精确解的能力。例如，我们可以列出120阶的联立方程式，说明120个变量间的关系，但手工无法解出。在科学的前沿问题上，写得出方程给不出解的情况比比皆是。然而，解不出来就很难去用实验验证，只好不了了之。

电子计算机的出现可以说是科学发展史上的新的里程碑。原来需要数学家花毕生精力去从事的计算，现在一个普通学生就可以轻易地做到了。例如，1878年英国数学家 William Shanks声称他花了15年时间把 π 值精确地计算到707位，现在用普通计算机10分钟可以算到1000

位(1984年日本有人用24小时将 π 值计算到1000万位).许多列得出方程解不出来的问题,可用数值方法近似解出来.传统的数值问题(如实验数据的处理)得到长足的发展,并且推动了以计算机为主要后盾的一批新兴数学学科的发展.例如,数学规划、最优化数学等.但是人们并未就此止步.精确度和计算量进一步增长,到目前为止,依然是研制巨型、阵列、分布式计算机的主要动力.计算机科技应用始终是计算机应用的一个重要方面.不过,它在整个计算机应用中所占的比重日益缩小.例如,1979年美国国防部系统统计表明,该部用于科技计算的软件费用仅是整个软件费用的5%.

● 数据处理

在计算机使用的初期,人们就试图把一些简单单调的脑力劳动交给机器做.比如,各种日常流水帐,登记表格,统计资料等,效果十分良好.它不仅记帐准确、方便,而且可以随心所欲地处理数据.例如,重新排表、选项打印等等.于是,从简单的统计、登记数据到生产管理决策的应用,很快就发展起来了.到第二代计算机初期(1960年前后),数据处理的应用按耗用机时统计,已超过科学计算的应用.数据处理一举成为计算机应用的主要领域.直到今天仍然如此,约占应用领域的70%.

数据处理是人类生产、生活中的古老课题.出土的巴比伦陶瓷就放有借贷、贸易等帐目.它们的分类存储仍然是今日数据处理的中心问题之一.数据处理就是数据的收集、分类、加工、统计.其特点是数据量大,处理数据的算法相对定型.数据间的关系(即数据结构)大体上就决定了程序如何展开.

现代社会非常复杂,各种因素纵横交错、相互制约.例如,为了解决职工上班迟到问题,就要增加公共车辆;车辆增加道路拥挤又引起轧死人的问题;扩建道路又增加改善居民住房问题的负担;民用投资增大,生产投资只得减缩又降低社会生产.那么,为解决提高生产率要解决的迟到问题应该如何决策呢?事实上,我们日常生活中的不协调、不平衡、矛盾、混乱情况,很大一部分是管理不科学.即缺乏从整个的宏观角度去处理各种数据,并作出正确决策.数据处理是现代化管理的基础.

数据处理应用非常广泛,几乎深入到我们生产、生活的各个领域.大体说来它可以分为以下四种类型:

管理型 各种经济计划、工程项目管理计划的制订、执行和检查,各职能部门业务及日常管理.高级的有军事决策、投资决策、中长期发展规划等.低级的就是商店、银行的流水帐.管理数学、概率统计、博弈论、运筹学、对策论、经济学是它的理论基础.

服务型 计算机系统存有大量有关数据.输入用户的请求或询问,由系统作出应答.如图书资料检索、航空订票系统、医院、旅馆、车站问讯服务处等.它的特点是有大量应用软件.在用户界面上,使用极其方便.

设计型 将有关设计规范、标准、资料存入计算机.待设计人员作出少量选择后,自动计算、绘图、制表.每当设计人员修改某处,则自动修改所有应改之处.不仅自动设计,而且少量模拟试验也可以在机器上进行(试算).

各种CAD(即计算机辅助设计“Computer Aided Design”)可以在各专业领域代替设计人员作大量简易设计劳动,其代替程度以CAD软件功能而定.

教育型 各种技术培训、图形文字解说系统,以及CAI(计算机辅助教学“Computer

Assisted Instruction")系统,可根据学员文化程度,安排讲授内容.自然,整本整本书的内容存放在计算机中是教育型的前题.

各类数据处理系统界线并不严格,随着进一步发展,各种技术交织的系统不断出现.

数据处理要求存储大量信息,因此它刺激着大容量存储器以及高速、高质量输入/输出设备的发展.一般说来,数据处理系统都要求有软件,用户在软件基础上编制自己的应用程序,使得程序简短运用方便.因而,数据处理的发展促成软件工业的形成.数据处理以交互式终端应用为其主要特色.

● 工程控制

计算机的工程控制应用是二次大战后发展起来的自动控制工程的继续.即由数字计算机控制各种自动装置、自动仪表、机床工具的工作进程.原则上,所有的生产、生活工具都可以自动控制.

工程控制也叫实时控制,其基本原理是通过各种传感器采集现场控制对象的数据,按控制模型进行计算,再给出相应的控制信息,驱动伺服机,实现装置工作过程的全盘自动化.早期工程控制用的是模拟计算机,模拟计算机运算模拟量,其特点是响应快、体积小、成本低,所以至今航空技术部门仍然广泛采用.但是模拟机专业性强,不宜于大量生产,技术要求高,开发周期长,更重要的是难于实现整个系统的通信协调.全系统全面自动化必然导致数字计算机控制,目前正研究全数字控制的机载系统和弹载系统.

控制用数字计算机,一般情况下任务专一,要求可靠性高,体积小,成本低,有特殊的输入输出设备.所以,直到60年代以后,集成电路和超大规模集成电路出现才大量发展起来.此外,由于数字计算机系统本身就是一个自动控制系统,第一、二代机系统软件开发的经验这时也为工程控制技术提供了发展的基础.而外围设备的完善,如自动检测设备、数模、模数(D/A, A/D)转换设备等,都促进了工程控制的应用.

工程控制按自动化程度可分四级:

第一级,直接控制与生产过程联系最紧的参数.给出初步加工信息,通知操作者作某种干预.

第二级,使用最广的直接数字控制(Direct Digit Control 简称DDC).即采样—计算—控制命令三个步骤完全自动.

第三级,最佳控制或叫监督控制、自适应控制.自动选择与本参数最有影响的量作控制量.这是真正的自动过程控制.

第四级,管理信息系统.即把生产管理与生产的过程控制二者结合起来,它不仅最优选择控制生产过程中的参数,还考虑成本、完工周期等等管理因素.

工程控制机一般是一个执行系统.它的控制软件以机器码形式写入控制机(或者写入到只读存储器ROM中).传统上这个层次的软件用手工编码或汇编语言,研制开发周期较长.目前已经开始在宿主机上用高级语言开发目标机上的软件,即嵌入式应用.在军事部门嵌入式应用是大量的.例如,1973年美国国防部统计数表明,嵌入式计算机系统所耗软件费用占整个软件费用的56%.