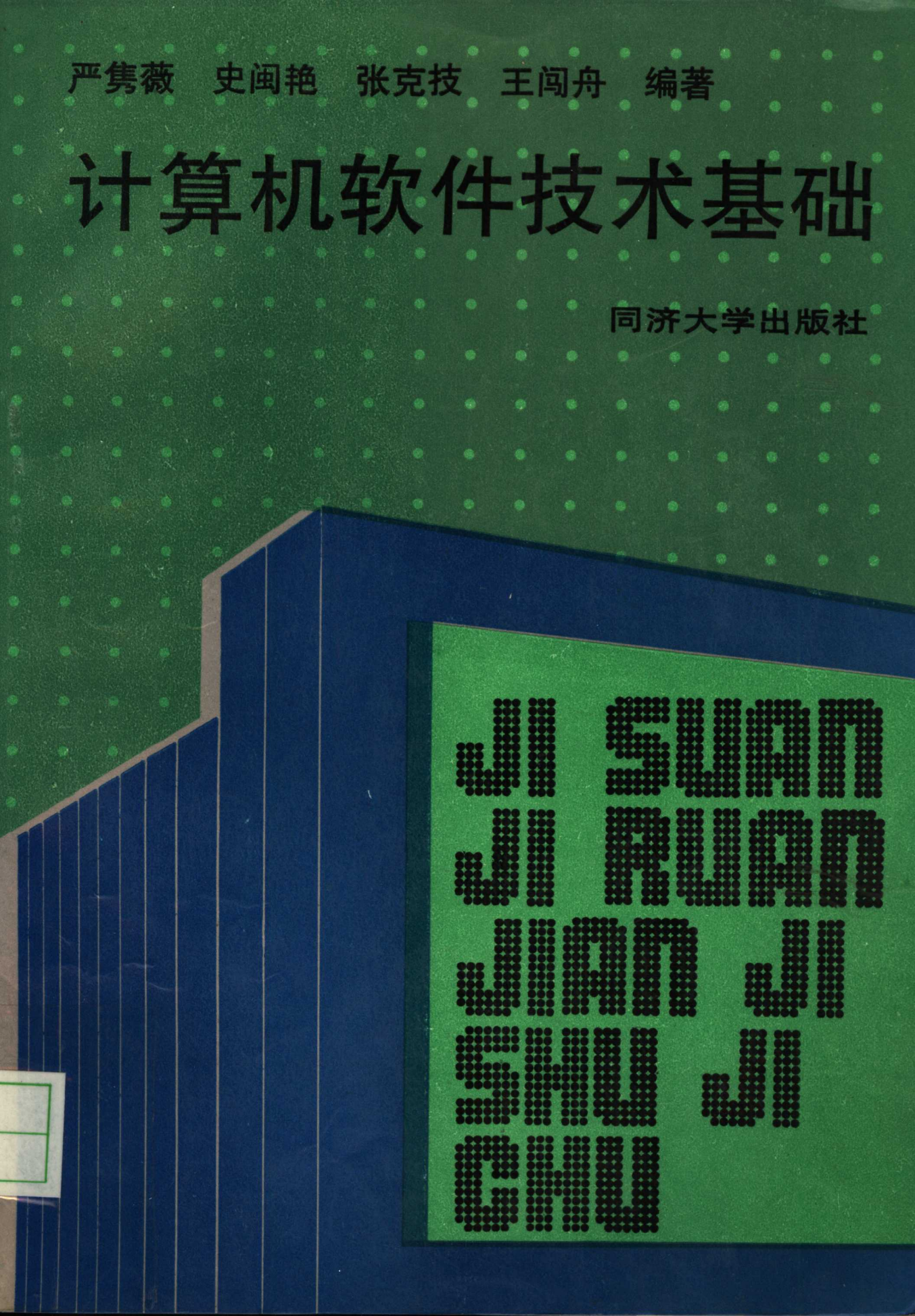


严隽薇 史闽艳 张克技 王闯舟 编著

计算机软件技术基础

同济大学出版社



↓ SWAP
↓ ROLL
↓ AN ↓
SW ↓
CPU

计算机软件技术基础

严隽薇 史闽艳 张克技 王闯舟 编著

同济大学出版社

(沪)新登字 204 号

内 容 提 要

本书从高等理工院校非计算机专业学习软件技术基础的需要出发,以计算机软件技术中的基础内容为线索,向读者全面、系统地介绍了软件的基本内容及主要应用技术。全书分概论、数据结构、非数值程序设计技术、操作系统、数据库系统、应用程序开发技术和计算机网络等七章。

本书可作为高等院校非计算机专业本科生、研究生的软件技术基础课程的教材,也可作为从事计算机应用工作的科技人员的自学参考书。

责任编辑 胡兆民

封面设计 李志云

计 算 机 软 件 技 术 基 础

严隽薇 史闽艳 张克技 王闯舟 编著

同济大学出版社出版

(上海四平路 1239 号)

上海中行印刷厂常熟分厂印刷

新华书店上海发行所发行

开本 787×1092 1/16 印张:14.25 字数:360千字

1995年2月第1版

1995年2月第1次印刷

印数:1—5000

定价:8.20元

ISBN7-5668-1502-2/TP·456

前 言

当前计算机技术已经大量渗入到各个学科领域。对于大学各类专业而言, 计算机软件应用与开发技术显得越来越重要和必不可少。为此, 教委已经开始在高等工业院校中推行非计算机专业分层次的计算机教学的设想, 《计算机软件技术基础》一书正是为满足这种教学需求而编写的。原铅字油印本已试用三年, 这门课程将系统地培养非计算机专业类大学生的基本软件能力, 内容主要包括: 软件基础知识; 软件应用技术和开发方法。

本书共分七章:

第一章 概论 对软件的发展过程、基本内容作一概述, 一些在本书中不详细展开的内容也在这一章中作了扼要介绍。

第二章 基本数据结构 内容包括线性表、栈、队列、数组、二叉树等概念及有关算法。

第三章 非数值程序设计方法 内容包括查找和排序算法。

第四章 操作系统 介绍操作系统的四个功能, 即处理机管理、存储器管理、设备管理及文件管理。

第五章 数据库系统 论述数据库系统的基本概念, 介绍关系型数据库管理系统 FoxBASE 的原理及应用。

第六章 应用软件开发技术, 内容包括结构化分析、结构化设计, 编程方法和测试方法, 同时还介绍了数据库的设计方法以及软件开发的三个实例。

第七章 计算机网络 论述了网络的基本概念, 同时介绍了 Ethernet 网的基本内容。

每一章后面都附有一定数量的思考习题。

本书油印本的第一、四、六章由严隽薇编写, 第二、三章由史闽艳编写, 第五章由王闯舟编写, 第七章由张克技编写。王闯舟、张克技、程振宏、左文生参加了第四、六章中实例的编写工作。这次正式出版前, 由严隽薇、张克技对全书各章作了精简、调整, 并附上了习题。全书由严隽薇进行审阅。

本书的编写和出版得到了同济大学教务处、教材科、电气工程系及自动控制教研室有关领导和教授的帮助和支持, 也得到了同济大学出版社的大力协助, 在此表示衷心的感谢。

由于水平有限, 书中有不当之处, 恳请读者批评指正。

编者

目 录

前 言

第一章 概论	1
1.1 计算机软件的发展	1
1.1.1 汇编语言和手工操作	1
1.1.2 高级语言与系统管理程序	2
1.1.3 操作系统的形成与发展	3
1.1.4 计算机网络软件、数据库软件的出现	4
1.2 计算机软件技术基础中的主线	5
1.3 计算机软件	6
1.3.1 汇编程序	7
1.3.2 各种高级语言	8
1.3.3 编译程序和解释程序	9
1.3.4 各种标准程序库	10
1.4 程序设计	11
1.5 软件工程概述	12
1.5.1 软件危机	12
1.5.2 软件工程化	13
1.6 小结	15
习题一	16
第二章 数据结构	17
2.1 什么是数据结构	17
2.2 线性表	18
2.2.1 线性表的逻辑结构及其运算	18
2.2.2 线性表的顺序存储结构	18
2.2.3 线性表的链式存储结构	20
2.3 栈与队列	24
2.3.1 栈	24
2.3.2 队列	26
2.3.3 栈和队列的应用	30
2.4 数组	31
2.4.1 数组定义和运算	31

2.4.2 数组的顺序存贮结构	32
2.4.3 矩阵的压缩存贮结构	33
2.5 树和二叉树	36
2.5.1 树的定义和基本运算	36
2.5.2 二叉树的定义及性质	37
2.5.3 二叉树的存贮结构	38
2.5.4 遍历二叉树	39
2.5.5 二叉树的应用	40
2.6 小结	41
习题二	42
第三章 非数值程序设计	44
3.1 查找	44
3.1.1 顺序查找	44
3.1.2 折半查找	45
3.2 Hash 表及其查找	47
3.2.1 Hash 表的概念	47
3.2.2 Hash 函数的构造方法	47
3.2.3 处理冲突的方法	48
3.2.4 常用的 Hash 表及其查找	49
3.3 排序	49
3.3.1 插入排序	50
3.3.2 简单选择排序	51
3.3.3 冒泡排序	52
3.3.4 快速排序	53
3.4 小结	54
习题三	55
第四章 操作系统	56
4.1 引言	56
4.1.1 操作系统的任务	56
4.1.2 操作系统的分类	57
4.2 处理机管理	59
4.2.1 进程	60
4.2.2 进程调度	63
4.2.3 交通控制	64
4.2.4 作业管理	68
4.3 存贮器管理	69
4.3.1 存贮器管理的功能	69
4.3.2 存贮管理方式	71

4.4 设备管理	81
4.4.1 概述	81
4.4.2 设备管理程序	81
4.5 文件管理	83
4.5.1 概述	83
4.5.2 文件组织	85
4.5.3 文件目录	89
4.5.4 文件的保护	92
4.5.5 文件存储空间的管理	92
4.5.6 文件的使用	93
4.5.7 文件系统的一般模型	94
4.6 实际操作系统介绍	97
4.6.1 PC-DOS 操作系统	97
4.6.2 UNIX 操作系统	106
4.7 小结	113
习题四	114
第五章 数据库系统	115
5.1 概述	115
5.1.1 什么是数据库系统	115
5.1.2 数据描述	121
5.1.3 关系模型的数学概念与关系代数	126
5.2 数据相关和关系规范化理论	129
5.2.1 数据相关	129
5.2.2 规范化	130
5.3 微机关系数据库管理系统 FoxBASE 简介	134
5.3.1 FoxBASE 的历史	134
5.3.2 FoxBASE 的运行环境	134
5.3.3 FoxBASE 的主要指标	134
5.3.4 FoxBASE 的文件类型	135
5.3.5 FoxBASE 的数据类型	135
5.3.6 启动和退出 FoxBASE 系统	136
5.3.7 FoxBASE 的表达式	136
5.3.8 FoxBASE 的函数	137
5.3.9 FoxBASE 的命令	137
5.3.10 FoxBASE 程序设计	138
5.4 小结	151
习题五	151
第六章 应用软件开发技术	152

6.1 结构化分析方法	152
6.1.1 分析阶段的任务	152
6.1.2 基本思想和步骤	152
6.1.3 数据流图	154
6.1.4 数据词典	156
6.1.5 加工小说明	157
6.2 结构化设计方法	158
6.2.1 设计阶段的任务	158
6.2.2 基本思想和原则	159
6.2.3 总体设计的主要方法	161
6.2.4 详细设计的基本方法	165
6.3 编程方法	170
6.3.1 编程阶段的任务	170
6.3.2 程序设计的风格	170
6.4 测试方法	172
6.4.1 测试阶段的任务	172
6.4.2 测试过程	173
6.4.3 白盒法	174
6.4.4 黑盒法	175
6.5 数据库的设计方法	178
6.5.1 设计过程	178
6.5.2 概念结构设计	179
6.5.3 逻辑结构设计	183
6.5.4 物理结构设计	184
6.5.5 数据词典、数据安全性、保密性及完整性问题	185
6.6 软件设计实例	187
6.6.1 小型数据库设计实例	187
6.6.2 一个分级计算机控制系统软件的设计	192
6.6.3 逐步求精算法设计实例	196
6.7 小结	198
习题六	199
第七章 计算机网络	200
7.1 概述	200
7.1.1 计算机网络的形成过程	200
7.1.2 计算机网络的组成	202
7.2 网络的拓扑结构和访问控制方式	203
7.2.1 网络拓扑结构	203
7.2.2 网络中信道访问控制方法	204
7.3 网络协议与基准模型	205

7.3.1 网络协议	205
7.3.2 开放系统互连网络基准模型(OSI)	206
7.4 网络环境下的操作系统	207
7.4.1 概述	207
7.4.2 通信方式与中断处理程序	207
7.4.3 进程管理	208
7.4.4 通信软件和网络软件	208
7.5 Ethernet 局部网络介绍	209
7.5.1 概述	209
7.5.2 Ethernet 结构	209
7.5.3 Ethernet 网络控制策略	211
7.5.4 Ethernet 网络通信协议	212
7.5.5 Ethernet 中帧的格式	214
7.5.6 Ethernet 信息交换过程	214
7.5.7 Ethernet 的改进	215
7.6 小结	216
习题七	216
参考文献	217

第一章 概 论

1.1 计算机软件的发展

一个完整的计算机系统由硬件层和软件层两个基本部分构成。硬件层在内部表示机器的可见结构,即裸机。其主要组成部分是:可执行一组指令的中央处理器(简称 CPU),一定规模的存储器(包括内存和外存),还包含大批完成信息输入与输出的装置及其接口。硬件是计算机系统工作的基础。随着计算机硬件技术的不断发展和广泛使用,在外部软件层也逐步丰富与完善,而软件的发展又大大促进了硬件的进展。

“软件”这个名词在 60 年代初才从国外传来。当时许多人并说不清它的确切含意,只是根据 Software 一词译成“软制品”或“软件”,现在我们统一称为“软件”。计算机发展的早期,人们就有了程序设计的概念,并认为写出的程序只要能在计算机上得出正确的结果,程序的写法可以不受任何约束。当时用于写程序的语言也仅是低级的机器码或汇编语言。程序在计算机上的装入、汇编、执行均是靠面板上的命令开关逐步地手工操作。这种小规模的低级语言编制、靠手工方法在计算机上操作并使其运行的程序,可以看作是软件的前身。发展到今天,人们对软件早已是一种新的认识,这种认识有两层意思。首先,谈论所谓软件应是一个程序的集合,这种程序不只是用户为解决某一个具体问题而编制的程序,它具有支持计算机工作和扩大计算机功能的作用。其次,随着程序规模及复杂程度的增大,软件的内容不仅仅是其程序实体,还包括开发程序、使用程序、维护程序所需要的一切文档。第一层意思强调的是计算机软件的丰富内容;第二层意思强调的是软件的实质,充分表明软件与程序的区别。

我们按计算机从第一代至第四代的发展过程来看一下计算机软件发展的四个阶段。

1.1.1 汇编语言和手工操作

在第一代计算机阶段,机器速度较低,外围设备较少,计算机又是无任何软件的裸机。这时程序设计人员只能用机器指令或称机器语言来编写程序,这样就要求程序设计人员熟记计算机的指令,工作量大,易于出错且不易于修改。又由于每种计算机的机器指令各不相同,所编制的程序只适用于某一特定的机器,局限性很大。因此在早期的计算机中,采用了建立标准子程序或标准程序库的办法,这虽然在一定程度上缓和了工作量大的矛盾,但不能从根本上解决问题。为了摆脱用机器指令编码的困难,出现了用指令符号来编制程序的办法。用符号语言编制的程序称为符号程序,在编制程序时,只要记住指令的助记符就可以了,指令助记符是指令英文名称的缩写,较指令编码容易记忆。例如:取数用 LDA、加法用 ADD 等。这种符号语言的扩大就是汇编语言。

用汇编语言编制程序要比用机器的指令代码方便得多,不仅易于检查程序错误和修改错误,而且指令、原始数据和结果数据的存放单元可由机器根据定位伪指令自动分配。例

如用 Z-80 机器语言和汇编语言编制的计算 $Z = X + Y$ 程序，如表 1-1 所示。

表 1-1

机器语言程序		汇 编 语 言 程 序		
地址码	指令代码	地址标号	指 令 符 号	注解
			ORG 200H	; 定位伪指令，程序从 2000H 号地址开始存放
2000H	3A 0C 20	START	LD A, (CNO1)	; $x \rightarrow A$
2003H	47		LD B, A	; $x \rightarrow B$
2004H	3A 0D 20		LD A, (CNO2)	; $y \rightarrow A$
2007H	88		ADD A, B	; $x + y \rightarrow A$
2008H	32 0E 20		LD (CNO3), A	; $z = (x + y) \rightarrow (CNO3)$
200BH	76		HALT	; 停机
200CH		CNO1:	x	; 原始数据 x
200DH		CNO2:	y	; 原始数据 y
200EH		CNO3:	z	; 计算结果 z
			END	; 程序汇编结束

不过，计算机的内部结构是根据指令代码设计的，它只能“识别”和“理解”用二进制代码表示的机器指令，不能“识别”和“理解”指令助记符。因此，人们用汇编语言编出计算题后，必须将此程序“翻译”成机器语言程序，机器才能执行，并算出结果。这个翻译工作是由预先装入计算机中的“汇编语言”完成的，汇编程序是一种“编译”程序，计算机有了它，才能允许用户在该计算机上使用汇编语言程序，因此汇编程序是计算机必不可少的软件。机器语言程序是它的工作目的，称为目标程序（或目的程序），而用汇编语言编制的程序称为汇编语言程序（或汇编源程序、源程序）。

在计算机上使用汇编程序过程有三部：

- (1) 通过引导程序把汇编解释程序装入到计算机；
- (2) 通过汇编解释程序读入源程序，并执行汇编过程；
- (3) 产生一个目标程序，并输出到卡片或纸带上。

当时计算机需要手工用开关命令对程序进行汇编，并把该目标程序和原始数据装入计算机执行，才能将运行结果输出到卡片或打印纸上。这种手工操作方式的主要特点是：

- 独占计算机
- 串行工作
- 人工干预

这在机器速度低、程序简单的情况下还可以适应。

1.1.2 高级语言与系统管理程序

采用汇编语言编制程序，仍要记住机器指令的助记符，且所编的程序只针对某一种计算机。为解决这些问题，在 50 年代末进入第二代计算机时期，出现了高级语言程序。它使得程序编制工作从程序设计员手中解放出来。高级语言的出现大大提高了程序的设计工

作。有人统计，在同等条件下，一天生产的高级语言程序行数大致等于汇编语言程序的行数，但对于同一个问题，用高级语言书写的程序可比用汇编语言写出的程序缩短 3~7 倍，所以两者的生产率也要相差数倍。高级语言的出现也促进了计算机的应用，使得计算机的使用范围由计算专业人员扩大到各行各业。

高级语言是由表达各种意义的“词”和“数学公式”，按照一定的“语法规则”组成的。通常把用程序设计语言编制的程序称为源程序，而计算机进行运算仍要根据目的程序来进行。因此在计算机中要先放好“编译程序”。这种翻译过程有两大类，一类是解释程序，如 BASIC 语言解释程序。它是先将源程序“扫视”一遍，然后一句句翻译成目的程序，每译完一句，就执行一句，当源程序翻译完了，目的程序也就执行完了。另一类是编译程序，如 FORTRAN 语言编译程序，是将源程序完全翻译成目的程序后，再交计算机执行。

这种程序设计的方法比较接近于人们的习惯，编出的源程序与具体机器的指令无关，可适用于任何机器，因此人们把程序设计语言称为高级语言。

目前，世界上有着近千种程序设计语言，较通用的也有几十种，例如用于科学与工程计算的 FORTRAN 语言，风格严谨的 ALGOL 60 语言，数据处理语言 COBOL，系统程序设计语言 PASCAL，人工智能语言 LISP，大型通用语言 PL/1 和 ALGOL 68，会话式语言 BASIC，此外还有各种模拟语言、数据库语言等。

高级语言解决的是程序的生产效率，但计算机软件手工操作的低效仍严重影响发挥第二代计算机在速度和存贮容量上的优势。例如，在一台第一代计算机上花一个小时计算的一个问题，在计算过程中，人工操作需花三分钟，仅占全部时间 5%；但是，当计算机的速度提高了十倍以后，这个问题只要计算 6 分钟，而人工操作如仍需三分钟，就占了全部时间的 50%。另外，用户“独占”丰富的计算机资源也是不合理的。这时出现了对计算机硬件和软件进行管理与调度的软件——管理程序。它的主要功能是：向用户提供多个共享资源来运行他的程序；帮助操作员控制用户程序的执行和管理计算机的部分资源。管理程序为用户提供一套控制命令，并以一定格式穿在卡片上，称为控制卡。用户用控制卡来表达对程序进行控制的要求。

有了管理程序以后，用户不必亲自上机操作，而是由专业化的操作员代劳。操作员只要通过控制台打字机打入控制命令就可以操纵计算机。操作员打入的命令由管理程序来识别和执行。这样不仅操作速度快，而且操作员可以方便地进行一些较为复杂的控制。当计算机运行中发生错误或意外时，管理程序通过计算机从控制台打字机上输出信息向操作员报告。这种输出信息不仅比“亮灯显示”所表达的更为丰富，而且操作员也易于理解。总之，用这种半自动方式来控制计算机不仅提高了效率，而且方便了使用。

管理程序不仅协助操作员操纵计算机，而且还管理计算机的部分资源。例如，用户程序总是通过管理程序去启动外围设备；用户可以按文件名字而不是物理地址来存取信息；管理程序不允许几道程序同时被接受进入计算机并同时执行等等。

1.1.3 操作系统的形成与发展

到了第二代计算机后期，特别是进入第三代（60 年代中）以后，硬件的大发展，特别是主存贮器容量的增加和大容量辅助存贮器——磁盘的出现，给发展更先进的管理程序准备好了物质条件。最后，计算机应用的日益广泛和深入，也要求进一步发展和扩大功能简单的管理程序。这样，管理程序就迅速地发展成为一个重要的软件分支——操作系统。

操作系统实现了计算机操作的自动化。在操作系统控制下，系统对计算机资源的管理水平又提高了一步。这些资源包括中央处理机、存贮器、各种外部设备和各种软件。操作系统的统一调度和管理使这些资源发挥了极大的效率，各种设备、多道作业协调一致，有条不紊地工作。第四章我们将全面地学习操作系统的各种管理功能。它们是：CPU 管理、存贮器管理、设备管理、文件管理和作业管理。

最先投入使用的操作系统是批处理多道系统。它允许多个程序成批地输入到计算机中并实现多道程序同时运行。为了给多个用户提供使用计算机的方便，给中央处理机设置多个终端，出现了分时操作系统。由上可知，操作系统的发展经历下面四个阶段：

- 手工操作阶段
- 管理程序阶段
- 批处理多道系统阶段
- 分时系统阶段

批处理多道系统和分时系统的出现，标志着操作系统的正式形成。随着硬件、软件技术的发展，以后又出现实时系统及网络 and 分布式系统。进一步还将要求研究和开发支持数据库和知识库应用的新型操作系统。

操作系统的出现是软件发展的一个重大转折，也是计算机系统的一个重大转折。操作系统本身是一个庞大的程序，在它的控制下，计算机系统的每个部件（不仅硬件部件，也包括软件部件）最大限度地发挥作用，因此，操作系统是软件系统的核心。

1.1.4 计算机网络软件、数据库软件的出现

从第三代计算机的中后期（1970 年后）开始，计算机的应用正向纵深发展：由辅助人们进行科学计算、实时控制、数据处理等发展到辅助设计、信息管理，甚至辅助人们进行思维和决策；由单机系统发展到简单的联机系统、复合计算机系统、分时系统直到计算机网络系统。由少数高等院校、科研单位走向整个社会，日益广泛地应用于国民经济的各个领域。

计算机网络软件是计算机技术和通信技术两者高度发展和密切结合的结果。从某种意义上讲，它是更高水平上的操作系统。它是利用通信线路把分布在不同地点上的多个独立的计算机系统连接成为一种网络，使网上用户能够实现数据传送，共享网络中的所有硬件、软件和数据等资源。这不仅提高了计算机的可靠性，均衡了网络中各种计算机的加载情况，并且便于系统的扩展。

目前，按照计算机网的跨越距离，又可分成远程计算机网和局部计算机网。远程网如美国 ARPA 网、DEC 网，局部网有 Ethernet, PC-net, PLAN 4000 等等。

当前计算机网络研究的课题有：高速数据通信和使用效率，不同类型的计算机结合，不同类型计算机网络彼此互连，网络操作语言，网络操作系统，以及完善各种服务级协议，建立网络服务系统，扩大网络服务范围等等。

随着计算机广泛应用于工业生产，商业管理，财政贸易，交通运输等部门，数据存贮、数据处理的要求越来越高，因而在操作系统的支持下建立和发展了各种类型的数据库系统软件。

数据管理经过人工管理、文件管理两个阶段逐步发展成为数据库系统软件。

人工管理阶段是计算机用于数据处理的初期阶段，对数据的管理是由程序员个人考虑

和安排的。这种管理只是一种技巧，是数据自由管理的方式。文件系统管理方式是由操作系统提供的。它把数据按一定的规则组织起来，成为一个有效的数据组合体。数据库的目标是要对所有的数据实行统一的、集中的、独立的管理，数据独立于程序而存在，并可以提供给各类不同的用户共享使用。

当前数据库系统软件的发展动向有：数据库标准化——为各种数据库提供统一的基础，建立共用数据库结构；数据库规范理论——研究构成规范的数据模型；数据库机器——改进数据库现行体系结构，部分地用硬件去完成数据库软件的工作；分布式数据库系统——是数据库技术与计算机网络相结合的产物；高级数据库和知识库——具有更强的处理能力，灵活性和方便接口的工程数据库、知识库等。

可以预见，随着计算机硬件发展和计算机广泛的应用以及科学技术的不断进步，计算机软件还会有更大的飞跃。

1.2 计算机软件技术基础中的主线

计算机软件技术发展到现在，已经形成了一个完整的技术基础学科体系。图 1-1 构画出了软件技术基础中的一条主线和拓宽部分。这条主线包括三个基础部分：

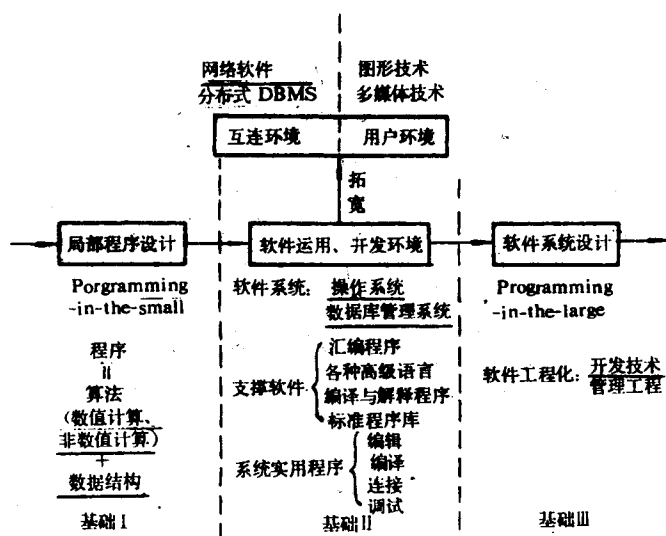


图 1-1 软件技术中的主线

基础 I：局部程序设计；主要内容是算法设计和数据结构知识。

基础 II：重要的软件系统；主要内容将在 1.3 节中专门论述。

基础 III：软件系统设计；主要内容是软件工程化。

基础 II 是软件的主体，它保证了软件的运行和开发，其拓宽部份涉及到一系列更新的软件技术。这部分将涉及到多机互连成网技术和人机界面的问题。例如图形技术、多媒体技术、网络软件和分布式 DBMS 等等。

图中有下划线的项目是本教材涉及的内容。1.4 节简单介绍一下程序设计。1.5 节是软件工程化的概要内容。本书从第二章开始就有下划线的项目展开论述。第二章和第三章分别是基础 I 部分中的数据结构和非数值程序的设计技术；第四章和第五章讲授基础 II 部分中的操作系统和数据库系统；第六章讲授基础 III 部分中有关软件开发技术方面的内容；第七章是计算机网络的初步知识。这七章基本上是一种块结构，第一章至第五章为基本内容，第六章、第七章可作为提高和拓宽的内容。

1.3 计算机软件

如 1.1 节所述，软件是程序的集合，其功能是利用计算机本身提供的逻辑功能来合理地组织计算机的工作，以便简化或代替人们使用计算机过程中的各环节，并为用户提供一个便于掌握、操作简便的工作环境。图 1-2 给出了以层次的观点划分的计算机系统软件工作环境的示意图。整个计算机系统可以划分为四个层次：硬件、操作系统、实用程序和应用程序。后三层属于软件范围。这些软件层被设置在用户与硬件之间，形成了一种虚拟计算机。虚拟机的各层，从裸机开始，层层相继，复杂程度逐步提高，同时每一层都为计算机扩充一组功能并带有一个接口。接口的含义是指各层都为使用者提供实现本层功能所需的一组可见的约定。我们把接口的这些特性称为系统的一个视图。图 1-3 表示了一个

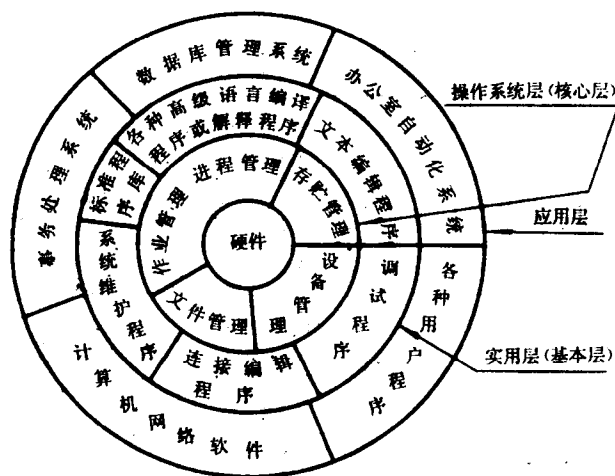


图 1-2 计算机系统软件工作环境示意图

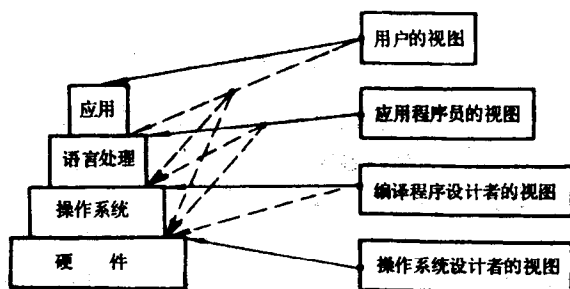


图 1-3 一个系统的分层和视图

四层结构，图中的实线箭头表示了各种层次的计算机使用人员将面临的不同的视图，而图中的虚线箭头则表示未能被外层隐藏的特性。计算机系统使用得好不好不主要是指物质基础即硬件的使用是否正确、运行是否可靠，而且包括对各个层次的软件掌握得如何，发挥了多少效用。这往往是更为关键的一个方面。

由图 1-2 可见，计算机各层次的软件主要有以下 8 个部分：

- 汇编程序
- 各种高级语言
- 各种语言的编译或解释程序
- 各种标准程序库
- 操作系统
- 数据库或信息管理系统
- 计算机网络软件
- 各种应用程序

这些软件按其功能通常划分为三类，即系统软件、支撑软件和应用软件。

系统软件能使计算机系统的各个部分，相关的软件和数据协调地、高效地工作。例如，操作系统、数据库管理系统以及计算机网络软件等。系统软件的工作通常要频繁地与硬件来往，大量地为用户服务，并经常伴随着资源的共享与复杂的进程管理，以及复杂数据结构的处理。系统软件是计算机系统不可少的一个组成部分。

支撑软件是协助用户开发软件的软件，包括帮助程序人员开发软件产品，也包括帮助管理人员控制开发的进程。例如文本编辑程序、关系数据库系统、各种语言的编译解释或汇编、连接、调试，也包括各种通用、专用的标准程序库和大量支持设计、测试方面的程序或软件包。

应用软件帮助人们完成特定领域的工作。除去商用事务信息处理软件、工程与科学计算软件以及办公室自动化软件等传统领域以外，近年来一些新的应用领域如异军突起，十分引人注目。比如，计算机辅助设计 (CAD)、系统模拟与仿真、计算机辅助教学 (CAI)、计算机辅助制造 (CAM)、计算机辅助工艺规划 (CAPP)、嵌入软件的智能产品以及人工智能软件等等。这些应用软件产品使得传统的产业部门面目一新，带来了惊人的生产效率和巨大的经济效益。

按功能分类的软件与前面按层次分类的软件既互相对应又有不同程度的交叉，这里的差别只是看问题的角度有所不同而已。

对于操作系统、数据库系统、计算机网络，本书将作全面论述。为了对软件有全面的了解，这一节我们简述软件中的前四个部分，即汇编程序、各种高级语言、各种语言的编译或解释程序以及各种标准程序库。

1.3.1 汇编程序

汇编语言的程序和对应的机器语言程序有相同之处，又有不同之处。相同之处是，程序主体部分几乎是一一对应的；不同部分是数码换成了符号，地址换成了名字，另外还增加了关于工作单元和常数单元的成分。这些不同之处正是汇编语言的优点，使得汇编语言的程序和机器语言程序比较起来，好写、好读、好改。

汇编语言也是依赖于机器的，因此称它为面向机器的语言。使用时必须了解机器的某

些细节，如累加器、每条指令的执行速度、内存容量等等，但正由于它依赖于机器，所以就可以与机器语言程序一样可结合机器特点编出高质量的程序，即程序又短，执行速度又快。所以直至今日，汇编语言仍起着相当重要的作用，在一些使用高级语言不能满足软件需求的场合，就要采用汇编语言以保证高质量的软件功效。

汇编程序的主要功能是把用汇编语言编写的源程序加工成机器语言写的目标程序。

源程序是由一串符号化的指令组成，国内采用标准的 ASCII 码，当汇编程序加工源程序时，总是从头到尾，一个符号接着一个符号地阅读，称为扫描源程序，把从头到尾扫描一次源程序称为扫描一遍，一般汇编程序是两遍扫描源程序的。第一遍把源程序中所有出现的名字进行造表，确定每个名字将占用的内存位置。第二遍扫描时，按所造出的表，把每条符号化指令代换成数码形式的机器指令。

此外汇编程序还具有一些附加的功能：

查错：当用户写的汇编语言源程序不符合汇编语言所要求的书写格式、不符合语法要求时，汇编程序指出源程序的某个位置出了什么性质的错误。

修改：汇编程序提供修改源程序的简便办法，用户把修改的要求提供给汇编程序，由汇编程序实现对程序的自动修改。

打印：在汇编过程中，当发现错误时，将错误信息打印出来，必要时还可打印出名字表及目标程序，还可打印出修改后源程序的文本等。

实用汇编语言中还可以包括一些用户编程序时很需要的指令，这些指令并无对应的机器指令，称为伪指令。

有关汇编程序的内容在微机原理课程中已详细讲解，在此不再赘述。

1.3.2 各种高级语言

高级语言种类繁多，总数已不下千种。我们以几种常用的语言为例介绍其特点。

FORTRAN 语言是用于科学和工程计算的语言，其程序结构是分块结构，每一块不包含进一层的结构。一个 FORTRAN 程序包含一个主程序块和若干个子程序块。程序的执行由主块开始，主块可以调用子块，子块间也可以调用，但子块不可以调用主块。另外在 FORTRAN 程序中变量的地址是静态分配的，因此不能包含递归子程序和动态数组。总的来讲，FORTRAN 程序的结构比较简单，而且可以分块书写和分块编译，故比较灵活方便。从 FORTRAN 77 起，在支持结构化控制结构与字符串处理等方面作了较大的改进，但数据类型仍欠丰富，对复杂的数据结构也缺乏支持。

COBOL 语言是通用商业语言，或称管理语言。对于数据处理，FORTRAN 和 ALGOL 60 语言都不是合适的计算机语言。为了解决大量数据的处理，发明了这种适用于商业数据处理的语言。在 COBOL 语言中引入了数据部分，按层次结构来描述，数据描述形式完全适合于现实生活中各种管理信息。另外，其程序设计采用了自然语言的程序设计风格。程序的书写是采用接近英语的形式。由于这些原因，COBOL 语言在商业、工业和行政管理部门被广泛地使用。

LISP 语言为表处理语言，是人工智能程序设计最常用的语言。LISP 有两个基本的数据类型，叫做原子和表。通过原子、表、表的运算以及定义一些 LISP 函数，就可以进行符号演算、公式推导、博弈及其他种种非数值处理的工作。

PASCAL 语言是第一个系统地体现结构化程序设计原则的描写算法的语言，是一种