

李芳芸 沈被娜
王选民 编著

(第二版)

计算机软件技术基础

清华大学出版社



《计算机基础教育丛书》出版说明

近年来,我国的计算机应用事业迅速发展,大批科技人员、大中学生、管理人员,以及各行各业的在职人员都迫切要求学习计算机知识,他们已经认识到,计算机知识是当代知识分子的知识结构中不可缺少的重要部分。

计算机应用人才的队伍由两部分人组成:一部分是从计算机专业毕业的计算机专门人才,他们是计算机应用人才队伍中的骨干力量;另一部分是各行各业中从事计算机应用的人才,他们既熟悉本专业的业务,又掌握计算机应用的技术,人数众多,是计算机应用人才队伍的基本力量。他们掌握计算机知识的情况和应用计算机的能力在相当程度上决定了我国计算机应用的水平。因此,在搞好计算机专业教育的同时,在广大非计算机专业中开展计算机基础教育是十分必要的。

非计算机专业中的计算机教学,无论就目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同,它以应用为目的,以应用为出发点。如果不注意这个特点,将会事倍功半。广大非计算机专业的师生、在职干部迫切希望有一套适合他们的教材,以便循序渐进地迈入计算机应用领域,并且不断地提高自己的水平。我们在前几年陆续编写了一些适合初学者使用的教材,受到广大群众的欢迎。许多读者勉励我们在此基础上进一步摸索和总结规律,为我国的广大非计算机专业人员编写一整套合适的教材。

近年来,全国许多专家、学者在这个领域作了有益的探索,写出了一批受到群众欢迎的计算机基础教育的教材。特别是全国高等学校计算机基础教育研究会作了大量的工作,在集思广益的基础上,提出了在高等学校的非计算机专业中进行计算机教育的四个层次的设想,受到广泛的注意和支持。我们认为:计算机的应用是分层次的,同样,计算机人才的培养也是分层次的;非计算机专业中各个领域的情况不同,也不能一律要求,在进行计算机教育时也应当有不同的层次。对于每一个学习计算机知识的人,还有一个由浅入深、逐步提高的过程。

我们认为,编辑出版一套全面而有层次的计算机基础教育的教材,目前不仅是十分必要的,而且是完全有条件的。在全国高等学校计算机基础教育研究会和许多同志的积极推动和清华大学出版社的大力支持下,我们决定编辑《计算机基础教育丛书》。它的对象是:高等学校非计算机专业的学生、计算机继续教育或培训班的学员、广大在职自学人员。

本丛书包括计算机科学技术的一些最基本的内容,例如计算机各种常用的高级语言、计算机软件技术基础、计算机硬件技术基础、微型计算机的原理与应用、算法与数据结构、数据库基础、计算机辅助设计基础、微机网络与应用、系统分析与设计等,形成多层次的结构,读者可以根据需要与可能选学。

本丛书的宗旨是针对广大非计算机专业的需要和特点来组织教材。敢于破除框框,从实际出发,用读者容易理解的体系和叙述方法,深入浅出、循序渐进地帮助读者更好地掌握课程的基本内容。希望我们的丛书能在这方面闯出自己的风格,在实践中接受检验。

本丛书的作者大多数是高等学校中有较丰富教学经验的教师。但是,由于计算机科学技术飞速发展以及我们的水平有限,丛书肯定会存在许多不足,丛书的书目和内容也应当不断发展和更新。我们热情地希望得到社会各界和广大读者的批评指正。

主编 谭浩强 林定基 刘瑞挺

1988.10

前 言

《计算机软件技术基础》于1987年10月由清华大学出版社出版第一版。通过几年的教学实践,广泛听取了广大读者提出的宝贵意见,特别是得到了清华大学教务处及相关各系同行们的支持,与我们共同讨论确定了第二版的内容大纲。

为使书的内容精练,第二版中删除了原书中部分章节,保留了概论、数据结构、操作系统、数据库系统这四部分软件技术的基本内容。并在内容上进行了修改与补充,加强了系统性和应用性。为便于教学和自学,每章后面都附有习题。

本书作为《计算机基础教育丛书》之一,适用对象为非计算机专业的本科生、研究生及广大从事计算机应用工作的科技人员。在计算机应用日益广泛,特别是计算机软件飞速发展的今天,广大计算机应用软件的开发者和使用者迫切需要掌握计算机软件的有关知识。本书包括了**计算机软件的基础知识——数据结构,计算机系统的核心软件——操作系统,以及广为使用的数据处理软件——数据库系统**。通过这些内容的学习能使读者在掌握这些知识的基础上,进一步提高使用计算机的能力和研制应用软件的水平。

本书第一、第四章由李芳芸编写,第二章由王选民、沈被娜、李芳芸编写,第三章由沈被娜编写。

我们对在第二版的修订出版过程中所得到的各方面的帮助和支持表示感谢。限于我们的水平,书中缺点和问题难免,敬请计算机软件教学工作者和广大读者提出宝贵意见。

编 者

目 录

第一章 概论	1	2.5.1 树的定义及其存储结构	52
§ 1.1 计算机软件的发展	1	2.5.2 二叉树及其性质	53
1.1.1 汇编语言的出现	1	2.5.3 二叉树的遍历	55
1.1.2 高级语言的出现	3	2.5.4 二叉树的应用	57
1.1.3 操作系统的形成	5	§ 2.6 图	65
1.1.4 计算机网络软件、数据库 软件的出现	7	2.6.1 图的定义和基本概念	65
§ 1.2 计算机软件	8	2.6.2 图的存储结构	68
1.2.1 计算机系统资源	8	2.6.3 图的遍历	71
1.2.2 计算机软件定义	9	2.6.4 最短路径	73
§ 1.3 软件的研制	10	2.6.5 关键路径法	78
1.3.1 软件工程学的基本概念	10	§ 2.7 检索	81
1.3.2 系统分析与设计方法介绍	12	2.7.1 线性检索法	81
1.3.3 结构化系统分析与设计	15	2.7.2 对半检索法	82
习题一	17	2.7.3 分块检索法	85
参考文献	17	2.7.4 散列检索法	85
第二章 常用数据结构及其运算	18	§ 2.8 排序	89
§ 2.1 概述	18	2.8.1 选择排序	90
2.1.1 引言	18	2.8.2 冒泡排序	90
2.1.2 什么是数据结构	19	2.8.3 线性插入排序	91
2.1.3 与数据结构相关联的运算	22	2.8.4 对半插入排序	92
§ 2.2 线性表	23	2.8.5 快速排序	93
2.2.1 线性表的逻辑结构	23	2.8.6 归并排序	94
2.2.2 线性表的存储结构	23	习题二	95
2.2.3 线性表的基本运算	25	参考文献	99
2.2.4 线性表的应用	28	第三章 操作系统	101
§ 2.3 数组	31	§ 3.1 引论	101
2.3.1 数组的定义	31	3.1.1 什么是操作系统	101
2.3.2 数组的顺序存储结构	31	3.1.2 操作系统的分类	102
2.3.3 稀疏矩阵	34	3.1.3 操作系统的功能和特性	105
§ 2.4 栈与队	40	§ 3.2 存储管理	108
2.4.1 栈的结构与运算	40	3.2.1 存储管理的功能及有关 的概念	108
2.4.2 栈的应用	41	3.2.2 实存储管理	110
2.4.3 队的结构与运算	45	3.2.3 虚拟存储管理	116
2.4.4 队的应用	46	§ 3.3 处理机管理	124
§ 2.5 树	51	3.3.1 概述	124

3.3.2	作业调度和进程调度	126	4.1.2	数据描述	187
3.3.3	并行程序设计中的 几个问题	130	4.1.3	三类数据模型	189
§ 3.4	设备管理	140	4.1.4	数据库管理系统 (DBMS)	194
3.4.1	概述	140	4.1.5	用户访问数据的过程	195
3.4.2	设备分配程序	143	§ 4.2	关系数据库理论基础	196
3.4.3	设备处理程序	144	4.2.1	关系代数	196
3.4.4	虚拟设备——假脱机 系统	145	4.2.2	数据依赖概念	203
§ 3.5	文件管理	146	4.2.3	规范化理论	205
3.5.1	引言	146	4.2.4	结构化查询语言 SQL	209
3.5.2	文件结构及存取方式	148	§ 4.3	数据库设计	213
3.5.3	文件目录管理	150	4.3.1	数据库设计的目的及设计 阶段	213
3.5.4	文件存储空间管理	153	4.3.2	概念结构设计工具 ——E-R 方法	216
3.5.5	文件的共享与文件系统的 安全性	155	4.3.3	逻辑结构设计	220
3.5.6	文件的使用及文件系统的 一般模型	156	4.3.4	物理结构设计	224
§ 3.6	用户接口	160	4.3.5	数据字典	230
3.6.1	概述	160	§ 4.4	数据完整性及安全性	230
3.6.2	系统调用命令	160	4.4.1	数据完整性	230
3.6.3	脱机作业控制方式	162	4.4.2	数据安全性	232
3.6.4	联机作业控制方式	162	4.4.3	数据库恢复	233
§ 3.7	网络环境下的操作系统	167	§ 4.5	分布式数据库系统	234
3.7.1	计算机网络	167	4.5.1	概述	234
3.7.2	网络操作系统	172	4.5.2	分布式数据库的系统类型及 数据分布方式	235
习题三		182	4.5.3	分布式数据库系统设计中 的问题	237
参考文献		183	习题四		241
第四章 数据库系统		184	参考文献		243
§ 4.1	绪论	184			
4.1.1	什么是数据库系统	184			

第一章 概 论

§ 1.1 计算机软件的发展

自 1946 年世界上出现第一台电子数字计算机以来,仅仅几十年时间,计算机系统得到了飞速发展。人们习惯按组成计算机的元件的演变来划分计算机的发展阶段。计算机元件的演变分为四个阶段,计算机的发展也相应分为四个时代。

第一代**为电子管时代**(1945—1959)。在这个阶段计算机的主要逻辑元件是电子管,存储器为水银延迟线或静电屏。

第二代**为晶体管时代**(1959—1965)。在这个阶段计算机的主要逻辑元件是晶体管,存储器由磁芯构造。

第三代**为集成电路时代**(1965—1971)。在这个阶段计算机主要逻辑元件是集成电路,存储器除磁芯外还增加了镀线。

第四代**为大规模集成电路时代**(自 1971 年开始)。大规模集成电路的集成度超过 100 个门电路,每平方英寸包含 50000 个元件以上,存储器用 MOS 和双极型元件。

现在正在研制**第五代计算机系统**,简称 FGCS(Five Generation Computer System),其关键技术是超大规模集成电路(VLSI)结构、并行处理、基于关系数据库的知识库,以及人工智能和模式处理的应用。其主要目标是具有问题求解和推理能力,能进行知识库管理,具有智能接口。因此 FGCS 是具有人工智能知识信息处理能力和非数值运算能力的高性能计算机。

FGCS 硬件与软件设计和制造的目标是为用途广泛的知识工程服务,因此应该包括专家系统、知识库,能迅速进行相联检索,能进行逻辑推理操作。在程序结构和软件中利用并行方式达到高速运行,而且机器和使用者之间可以使用自然语言和图像进行通信。

FGCS 为实现以上目标和功能按照三个子系统组织工作,第一是管理知识库子系统,第二是问题求解和推理子系统,第三是人与机器之间的交互子系统。

图 1.1 示出第五代计算机系统的硬件、软件及与人的关系。硬件与软件的关联随计算机的发展越来越紧密,在 FGCS 中就充分显示出硬件、软件及人的统一。

在整个计算机的发展过程,随着计算机硬件技术的不断发展及广泛使用,软件也逐步丰富与完善,而软件的发展又大大地促进了硬件的发展,最终成为一个完整统一的整体。

计算机软件也有标志其发展的四个阶段,下面作一个详细的介绍。

1.1.1 汇编语言的出现

一台没有装入任何软件的计算机,我们称它为**裸机**。裸机只认识“0”、“1”两种代码,程序人员只能用**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全部指令,工作量很大且容易出错又不易于修改。由于各种计算机的机器指令不一定相

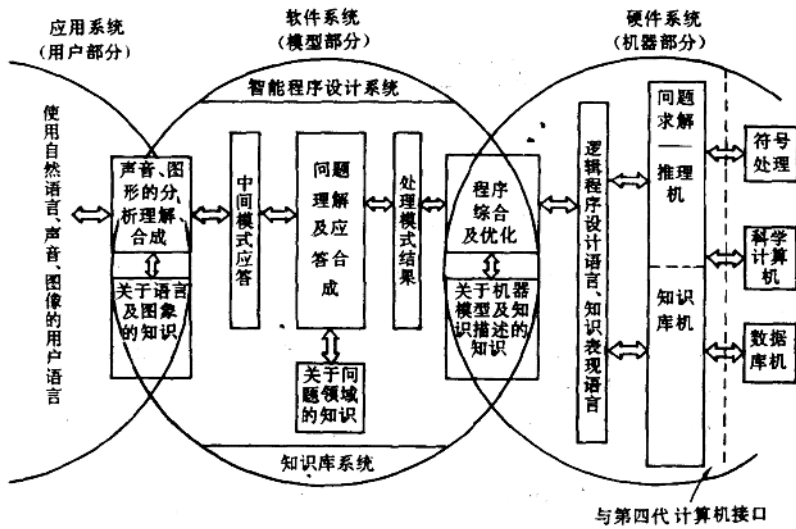


图 1.1

同,所编制的程序只适用于某一特定的机器,局限性很大。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。用指令符号编制的程序称为**符号程序**,在编制程序时,只要记住用英文名称缩写的指令的助记符就可以了。例如取数用LDA、加法用ADD等。在符号语言的基础上进一步发展就是**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,便于检查和修改错误,而且指令、原始数据和结果数据的存放单元可由机器自动分配。

然而,计算机的内部结构是根据指令代码设计的,也就是说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程序后,必须将此程序翻译为机器语言程序,称为**目的程序(目标程序)**,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“**汇编程序**”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编制程序,这个程序称为**汇编语言程序**或称为**汇编源程序**。汇编程序是计算机必不可少的软件。

用汇编语言写的程序和用机器语言写的程序有相同之处,又有不同之处。相同之处在于程序主体部分几乎是一一对应的,不同的是0、1数码换成了符号,地址换成了可读的名字,另外还增加了关于工作单元和常数单元的成分。这些不同之处也正是汇编语言的优点,使得用汇编语言编写的程序好写、好读、好改。

由于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,如累加器的个数、每条指令的执行速度、内存容量等等。但也正由于它依赖于

机器,就可以与机器语言程序一样可结合机器特点编出短小、高质量、执行速度快的程序。所以,时至今日,汇编语言仍起着重要作用。在一些计算机公司中仍用汇编语言编写系统软件,以保证高质量软件的功效。

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。这由三个过程来完成,如图 1.2 所示。

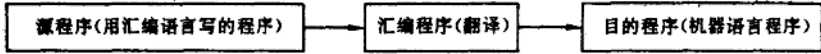


图 1.2

第一个过程,用汇编语言编制出源程序;

第二个过程,将源程序输入到计算机内,由汇编程序把它加工成计算机能够执行的目标程序;

第三个过程,执行目标程序,得到计算结果。

源程序由一串符号化的指令组成,一般采用标准的 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。从头到尾扫描一次源程序为扫描一遍。一般汇编程序均扫描两遍,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,把每条符号化指令转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:

- 查错功能:当用户写的汇编语言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位置和出了什么性质的错误。

- 修改功能: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,由汇编程序实现对源程序的自动修改。

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

1.1.2 高级语言的出现

采用汇编语言编制程序,仍要记住机器指令的助记符,且所编的程序只能针对某一类机器。为了解决这些问题,出现了高级语言程序。它使得程序编制工作可以从专职程序设计员手中解放出来。高级语言的出现大大方便了程序的设计工作,也促进了计算机的应用,使得计算机的使用由计算机专业人员扩大到各行各业。

高级语言是按照一定的“语法规则”,由表达各种意义的“词”和“数学公式”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。这中间也如同汇编程序一样,要有一个“编译器”,把源程序“翻译”成目标程序。这种程序设计的方法比较接近人们的习惯,编出的源程序与具体机器的指令系统无关,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,世界上有几百种程序设计语言,较通用的也有几十种,各有其特点。我们以常用

的几种语言为例,介绍其特点。

· **FORTRAN 语言**:最常用的科学与工程计算的语言。其程序结构是分块结构,一个 FORTRAN 程序包含一个主程序块和若干个子程序块。程序的执行由主程序块开始,主程序块可以调用子程序块,子程序块间也可以调用,但子程序块不可以调用主程序块。另外,程序中的变量地址是静态分配的,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、分块编译,故使用起来灵活、方便。

· **ALGOL 60 语言**:主要用于科学与工程计算的语言,其程序结构是分程序嵌套结构。整个 ALGOL 60 是一个大分程序,分程序中又可以嵌套进一层的分程序,其嵌套的深度可以任意,内层分程序可以使用外层分程序定义的量,反之则不可以。其变量地址是动态分配的,在进入分程序时分配,在退出分程序时释放,因此节省内存。而且允许递归子程序和动态数组。其程序结构比较严谨、清晰。

· **COBOL 语言**:FORTRAN 和 ALGOL 60 语言都是适于数据处理的计算机语言,COBOL 语言中引入了按层次结构来描述的数据部分,这种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,程序的书写采用接近英语的形式。由于这些原因,在商业、工业和行政管理部门被广泛地使用。

· **PASCAL 语言**:这是一种典型的系统结构化程序设计的语言,PASCAL 语言的出现和结构程序设计技术的发展推动了编译程序工程技术的发展。PASCAL 语言强调概念清晰,实现简化,方便用户;强调程序的可靠性,易于验证;有较好的结构,减少了错误的发生并易于阅读。因此虽然问世较晚,但却得到人们的极大重视。

· **C 语言**:C 语言是 1972 年设计出并首先在 UNIX 操作系统、PDP-11 机器上实现的。UNIX 操作系统是用 C 语言书写的,C 语言也与 UNIX 操作系统紧密地联系在一起。但目前,C 语言却独立于 UNIX 系统,独立于 PDP-11 系列而蓬勃发展了起来。用 C 语言编写的程序很容易移植,所以具有“可移植的汇编语言”的称号。此外,由于它具有丰富的运算符和表达式,以及先进的控制结构、数据结构,因此还具有表达能力强、编译出的目标程序质量高、语言简单灵活、易于实现等特点。C 语言不仅可用来写操作系统、编译程序、数据库管理系统等系统软件,也可以用来写各种应用软件,例如企事业管理、办公自动化、工业自动控制、图形处理等。C 语言已成为当今最流行的程序设计语言。

· **PL/I 语言**:这是一种大型通用语言,它综合了 FORTRAN、ALGOL 60、COBOL 等语言的特点。其表达式和语法来自 FORTRAN 语言,分程序结构和类型说明来自 ALGOL 60 语言,数据描述取自 COBOL 语言,控制结构和总体结构则兼有 FORTRAN 和 ALGOL 60 语言的特点。它包含了各种功能的语言成分,既能进行数值计算又能进行数据处理;可以作表处理,又可以作符号串处理;还具有格式丰富的输入/输出设施和中断并行等能力。用户可以根据需要选用它的一部分或全部。

· **LISP 语言**:在计算机的应用中有一个非常有意义的领域叫做非数值处理,例如公式的推导、用机器证明定理、下棋、印刷排版、编辑修改、版面设计、机器翻译、语言研究等人工智能问题。这些非数值处理问题是 FORTRAN、ALGOL 60、COBOL 等语言不能解决的,例如 $(X+Y)^2$ 的计算,在 FORTRAN 语言中只能先给出变量 X 和 Y 的数值,然后才能计算,绝不能推导出 $(X+Y)^2$ 等于 $X^2+2XY+Y^2$ 来,然而 LISP 语言可以解决这类

问题。LISP 语言有两个基本的数据类型,叫做原子和表。一般原子定义为以大写字母打头,由大写字母和数字串组成,其长度不超过 30 个字符的串。表是递归定义的,也就是在表的定义中又可以用到表,每个表可以构成一个复合表。还定义了一组简单而有效的操作,例如由两个表构造一个复合表的运算,从一个表中选出其第一个分量的运算和从一个表中去掉其第一个分量而取余下部分的运算。通过原子、表、表的运算和定义的 LISP 函数,就可以进行符号演算、公式推导、博弈等种种非数值处理的工作。LISP 语言的理论基础是数理逻辑,它简单但又有很强的表达能力。在运行过程中动态分配大量的存储单元,可以随时把无用的存储单元收集回来再用,这也是非数值处理语言的一个共同特点。

除以上通用的程序设计语言外,还有专用于某种特定用途的语言,这种语言种类很多,例如飞机制造业中的机翼设计语言、造船设计中的船体放样语言、专用于计算机线路设计的语言、专用于系统仿真的语言以及用于编辑、排版的语言等。

近年来为了适应大型软件开发的需要,并由于软件工程学概念的广泛普及,还出现了超高级语言、蓝图语言等。

超高级语言功能强且简洁,但运行时需要大型支撑系统的支持,这增加了存储容量,也降低了用这种语言写的程序的执行速度,因此它不适于用来开发实际的大系统,而是用来开发模型系统。模型系统是对系统分析阶段的检验,是进一步确定系统需求的工具。模型系统通常具有实际系统的主要功能而不是全部性能,而且可以降低对接口、可靠性和程序质量的要求;此外还可以省掉许多文档资料方面的工作,所以可以大大降低模型系统的开发成本。在 UNIX 操作系统上的命令解释语言 Shell 就是超高级语言的代表。

蓝图语言是为设计软件和描述软件的一种方法论——软件蓝图方法论使用的语言。软件蓝图方法论是由美国马里兰大学朱耀汉教授提出来的。它指导软件工程师采用自顶向下、模块化和结构化的设计原理;它只规定了必须实现的设计结构,并没有硬性规定必须采用的具体设计方法。它把软件设计和软件实现作为相对独立的两个不同阶段,把软件设计分为三级。A 级用概念性的设计提供软件的整体概貌;B 级用一个结构化设计描述控制流(数据操作序列)和数据流(数据元素的序列);C 级用详细的设计来描述软件的实现。所用蓝图语言是自然语言和形式化的软件设计语言的混杂语言,它有统一的语法,严格规定了三级设计中什么地方用自然语言、什么地方用软件设计语言。使用自然语言描述简明、易读易理解;使用软件设计语言描述精确,可保证设计的完整性。蓝图语言吸收了两者的优点。

超高级语言和蓝图语言都是将软件分析和设计两个阶段结合起来,为实现软件工程提供工具,是具有发展前景的语言。

1.1.3 操作系统的形成

为了充分发挥计算机的高工作效率,需要解决以下问题。首先要处理好人对机器过多的干预或手动操作过多的问题。这就要求用户组织好交给计算机来完成的每个作业。这就需要有一个管理软件,这个管理软件具有能接受和处理用户提交作业的功能。

其次,要处理好中央处理机与外围设备在速度上快慢不匹配的问题。要求管理软件具有处理中断的功能,应提供一种简便的、统一的调用外围设备的手段,统一管理外围设备

的输入和输出。

第三,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。因此要求管理软件具有处理来自不同终端的**多道作业**的功能。

总之,要使计算机所有资源(包括中央处理机、存储器、各种外部设备和各种软件)协调一致、有条不紊地工作,就要有一个功能强大的管理软件进行统一调度与管理。这个软件称为**操作系统**,具有操作系统的计算机系统常称为**虚拟计算机**。

计算机操作经过了手工阶段、早期的批处理阶段、执行系统阶段、多道程序系统和分时系统阶段。多道程序系统和分时系统的出现才标志着操作系统的正式形成。

1. 手工操作阶段(50年代初至50年代中期)

用户使用计算机是将机器语言或高级语言编制的程序凿在纸带或卡片上,用户扳动扳键、启动按钮、光电输入机或卡片读入机读入程序,程序运行的结果由打印机印出。出现故障时,用计算机上的扳键进行修改。

手工操作的特点是计算机软、硬件资源只能由一个用户的一个程序独占,而且操作过程必须人工干预。

2. 早期批处理阶段(50年代中期至50年代末期)

手工操作占用的时间与计算机运行时间的不匹配,随着计算机运行速度的提高,问题越来越大,因此出现了早期的批处理方式。计算机的系统操作员将各个用户提出的作业组合成一批作业,利用常驻在内存中的监督程序,将各个用户的作业逐个输入到磁带上,由磁带再逐个将作业调入内存运行,运行结果也先输入至另一磁带上,再逐个打印出来。

这种批处理方式为**联机批处理**,外部设备如输入机、打印机和主机的工作仍串联运行。为解决这个问题采用**脱机批处理**方式。脱机批处理方式是增加一台卫星机(小型计算机),通过卫星机将卡片输入到磁带上,同样在磁带上的输出结果也通过卫星机后再打印出来。这样可以达到主机和输入、输出设备并行工作,从而解决了外部设备与主机运行速度不匹配的矛盾,提高了主机使用效率。

3. 执行系统阶段(50年代末至60年代初)

由于早期批处理方式中的监督程序常常被用户程序破坏,需要解决对监督程序的保护问题,因此产生了**执行系统**。在执行系统中使用了通道设备并提出了**中断处理**的概念。

通道设备解决了多台外设与主机并行工作的问题;中断处理是指外围设备或通道发出信号要求暂时停止主机当前的工作,而转去处理外设提出的要求——**中断处理程序**。

中断处理程序以及负责控制输入、输出的服务程序称为**执行系统**,它对其它的系统程序和应用程序起控制指挥作用。而原来的监督程序只起相互调用的作用。执行系统常驻内存,其它程序在其指挥下工作,从而增加了对系统的保护能力。

执行系统的出现还促进了软件技术的发展,使系统程序设计实现模块化结构,使系统结构简单,易于设计、修改和扩充。

4. 多道程序系统(60年代初至60年代中期)

为提高计算机中央处理器(CPU)的运行效率,在内存中应保持有**多道程序**,而且程序可以交替占用CPU和外部设备。因此在多道程序系统中解决了以下几个主要问题:

- 对CPU来说,要解决在何时、以何种策略把CPU分配给哪一道程序使用的问题。

通常把解决这一问题的过程称为**调度算法**。

- 对存储器来说,要解决多道程序在内存中定位、分配及避免互相冲突和破坏的问题。把解决这一问题称为**存储管理问题**。

- 对外部设备来说,在多道程序下提高设备利用率,避免发生矛盾,负责外部设备的分配、释放、控制及信息交换的程序称为**设备管理程序**。

- 在辅助存储器的管理方面,出现了文件系统,使用户可以把程序和数据以文件形式存放于辅助存储器中。管理及使用文件的程序称为**文件管理程序或文件系统**。

5. 分时系统(60年代中期以后)

多道程序系统保持并发展了执行系统的特点,使操作更加自动化,但对一般的使用者来说,更希望能直接与机器接触,以便随时发挥人的主观能动作用,这样出现了**分时系统**。分时系统的特征是在一台计算机上挂有若干个终端,每一个终端提供给一个用户使用,因此用户可以通过自己的终端,用对答方式(交互式)直接控制自己的程序,随时对程序作必要的修改和补充。在分时系统中,系统将CPU的时间轮流分给每个联机终端,这样使用户感到好像只有他在使用机器。因此称这种分时系统为一种**高级联机操作方式**。

多道程序和分时系统是操作系统形成的标志,随着计算机应用的扩大和计算机硬件的不断发展,操作系统更加完善,功能更为扩大,出现了**实时操作系统、网络操作系统**等。

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(既包括硬件也包括软件)最大限度地发挥着作用,因此操作系统是软件系统的核心。

目前,操作系统已进入总结提高阶段,形成了几类较成熟的系统,如UNIX分时操作系统、RSX-11M实时操作系统、DOS微机操作系统等等。现在进行多方面的理论研究工作,如操作系统的结构设计、工具语言、进程通讯、关于死锁和各种调度算法、可靠性问题、性能测试和分析的研究等。

1.1.4 计算机网络软件、数据库软件的出现

计算机的应用不断向纵深发展。由辅助人们进行科学计算、实时控制、数据处理,发展到辅助设计、信息管理,辅助人们进行思维、决策;由单机系统发展到联机系统、复合计算机系统、计算机网络系统。

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,它是更高意义下的操作系统。它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起来,使网上用户可以实现数据传送、实现共享网络上的所有硬件、软件和数据等资源。并且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,同时便于系统的扩展。

目前,按照计算机网络的跨越距离可分成远程计算机网、局部计算机网两种。远程网如ARPA网、DEC网等,局部网如Ethernet、PCnet、PLAN 4000等等。

计算机网络当前研究的课题有高速数据通信和使用效率问题,不同类型的计算机联网问题,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,数据存储、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。

数据库系统软件是数据经过人工管理和文件管理两个阶段发展起来的。计算机信息管理的大量数据需要较长时间的存储、检索,并且需要对数据进行删除、更新等处理,这样提出了需要有一个系统软件来统一管理这些数据的要求,因此出现了数据库管理系统。

数据库管理软件具有数据结构化、最低冗余度、较高的程序与数据间的独立性,并且易于扩充、易于编制应用程序等优点。此外,还具有安全性(防止不合法的使用)、完整性(保证数据的正确性、有效性、相容性)及并发控制(避免并发程序之间的干扰)等功能,较大规模的信息系统都是建立在数据库系统的基础上。

数据库的出现以 60 年代后期的 3 个事件作为标志:

(1) 1968 年美国 IBM 公司首先研制了第一个数据库管理系统 IMS (Information Management System), IMS 是**层次结构模型**。

(2) 1969 年美国 CODASYL (Conference on Data Systems Languages) 组织的 DBTG (Data Base Task Group) 小组发表了 DBTG 报告。提出了 DBTG 系统,它是**网状结构模型**。

(3) 1970 年美国 IBM 公司的研究员 E. F. Codd 开始研究**关系数据库系统**。

当前数据库系统软件的发展动向是数据库标准化——为各种数据库提供统一的基础,建立公用数据库结构;数据库规范理论的研究,用这种规范化的理论构成规范的数据模型;面向对象数据库的研究,用面向对象的技术设计面向对象的数据库管理系统;数据库机器的研究也是当前数据库系统发展的动向,用部分硬件来完成数据库软件的工作,以改进数据库现行体系结构;分布式数据库系统的研究是当前数据库技术与计算机网络相结合的产物;随着人工智能技术的发展,高级数据库和知识库的结合,使得具有更强的信息处理能力,以及灵活和方便的接口,这促进了数据库向工程数据库发展。

可以预见,随着计算机硬件的进一步发展和计算机广泛的应用,以及科学技术的进步,计算机软件还会有更大的飞跃。

§ 1.2 计算机软件

1.2.1 计算机系统资源

计算机系统资源是指计算机硬件和计算机软件资源。

计算机硬件资源包括中央处理机(简称 CPU)、主存储器和显示终端、键盘、打印机、磁带机、磁盘机等外部设备。

计算机软件资源一般认为包括以下几部分:

- 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
- 数据库系统软件
- 计算机网络软件
- 各种应用程序

计算机硬件和软件组成了计算机使用的统一整体。因此,我们称之为**计算机系统**。对于使用计算机的人来讲,面对的是以**中央处理机**和**外围设备**为物质基础的扩充了的计算机,称为**虚拟计算机**。计算机系统使用的好不好,不仅仅指物质基础使用是否正确、运行是否可靠,而且指软件掌握得如何、发挥了多少效用。而后者往往是更为关键的一个方面。

例如以下一些问题没有合适的计算机软件是无法解决的:

(1) 多道程序的处理问题。如果内存中保持有多道程序,如何实现交替运行,以便有效地利用 CPU 和外设。软件应解决什么时候把 CPU 分配给某道程序占用,每个程序和其执行的中间结果、最后结果如何存放才不致造成混乱,又如何将各道程序执行结果打印输出。

(2) 多个用户使用一台计算机问题。一台计算机挂上若干个终端,每一个终端被一个用户使用,多个用户有效地使用 CPU 和外设的问题,也要依靠计算机软件才能实现。

(3) 当某种高级语言编制的应用程序需要占用内存的容量大于内存可用空间,那么这个应用程序是否可以正常运行,如何在系统软件的支撑下,利用计算机的外存来保证应用程序的执行呢?

(4) 多台计算机联成计算机网络,建立通信,扩大功能,提高可靠性,这也是靠计算机硬件和软件共同实现的。

诸如此类的问题很多,说明了虚拟计算机的功能大大超过了裸机。因此当讲到计算机资源时应该既考虑硬件资源又考虑软件资源。

1.2.2 计算机软件定义

从功能上可以认为软件是利用计算机本身提供的逻辑功能,合理地组织计算机的工作,简化或代替人们在使用计算机过程中的各个环节,提供给用户的一个便于掌握操作的工作环境。因此,不论是支撑计算机工作还是支持用户应用的程序都是软件。

在 50、60 年代就曾认为计算机软件就是程序,软件就是指如汇编程序、编译和解释程序、操作系统和支撑操作系统的各种管理程序、服务程序以及用户用各种高级语言编制的程序等。

这个定义是基于手工业方式进行软件开发而提出来的,从设计、编程到调试均由个人独立完成。但要开发一个大型软件,特别是实用化、商品化、通用化的软件就碰到不少问题。不仅效率低、开发周期长,而且各个模块间的联系和接口很难协调,人的思维也很难胜任以几万条或百万行计数的程序,因而出错率高,维护工作量大,导致成本高。在 60 年代末出现了软件危机。当时 IBM 公司 OS/360 系统负责人 Brooks 形象地形容这种手工业方式开发软件说:“……像巨兽在泥潭中作垂死挣扎,挣扎得越猛,泥浆就沾得越多;最后没有一个野兽能逃脱淹没在泥潭中的命运,……,程序设计就像是这样一个泥潭……,一批批程序员在泥潭中挣扎……没人料到问题会这样棘手。”

软件设计者逐渐感到要有一定规范的文档以保证程序从设计、调试到运行的成功。这样从70年代开始,认为软件不仅是程序还包括开发、使用、维护这些程序所需要的一切文档。

到了80年代,为了加强工程化、规范化,从软件工程的概念上更为全面地给软件下了定义,认为:计算机程序、实现此程序功能所采用的方法、规则以及与其相关联的文档和在机器上运行它所需要的数据都是计算机软件。计算机软件不断发展出新产品,使得应用的灵活性和人机交互能力不断提高,充分体现了软件支撑计算机工作,扩大计算机系统功能的作用。

§ 1.3 软件的研制

1.3.1 软件工程学的基本概念

在50、60年代开发大型系统软件用手工方式进行,其生产效率低、出错率高。例如,IBM公司的OS/360操作系统用了5000人年开发,软件产品每个版本均有1000个大大小小的错误。这种状态不能满足日益增长的软件生产的需要。产生了以下几个方面的问题:

(1) 软件复杂性飞速增长。

个体思维不能胜任开发一个需上千人年甚至几千人年的大型软件,这种软件程序量是以几万条、百万行计数的,而且相互关联,造成错误率大大提高。

(2) 软件成本极高。

由于硬件迅速发展,集成度大大提高,硬件成本下降,靠落后开发方式来编制的软件却随着工作量的增长,人工费用也增长,造成软件、硬件投资比率变化,美国做了一个统计如表1.1所示,可见软件投资增长率之快。

表 1.1

年 份	软件投资占总投资的百分比
1955 年	20%以下
1970 年	60%左右
1980 年	85%以上

(3) 开发周期长。

这种手工方式开发的程序,所需人年是随程序代码行的上升按指数曲线增长的,造成大型软件开发周期长,如表1.2所示。

(4) 维护工作量大。

通过对软件错误的分析和统计,看到由于设计产生的错误占70%,由于编码产生的错误占30%,因此保证设计过程软件的正确性且易于检验,是一个亟需解决的问题。此外,当用户需求有变化时、当硬件设备更新后或操作系统有新的版本出现时,都需要修改

程序,以适应新的环境,这使得软件的维护工作耗费相当的人力和资源。50年代衡量一个软件的质量是看其是否节省存储空间、运行速度是否足够快,而到80年代衡量的标准是软件正确、易读、易修改、易测试、有完整的文档资料。

表 1.2

代码行	所需人年
100K	8.3
200K	83
300K	170
400K	250
500K	380
600K	500

由于以上四个方面的原因,出现了“软件危机”,于是在1968年北大西洋公约组织的计算机软件学术会议上第一次提出“软件工程”这个词。考虑到研制一个软件系统同研制一台机器或建造一座楼房有许多共同之处,因此参考机械工程、建筑工程中的一些技术,来指导软件的研制,像处理其他工程一样来处理软件研制的全过程,形成了软件学科中的一门新学科——**软件工程**。

软件工程强调使用生命周期方法学和各种结构化分析与设计技术。用“系统”的观点来分解问题然后再分别解决各个子问题。

生命周期方法学是从时间角度对软件的开发和维护的复杂问题进行分解,分成若干个阶段,每个阶段有相对独立的任务,前一个阶段任务的完成是开始进行后一阶段工作的前提和基础,而后一阶段任务的完成是使前一阶段提出的问题给予解决的具体化。每一阶段的开始和结束都有严格标准,因而,前一阶段的结束标准就是后一阶段的开始标准。而且每一阶段结束之前都必须进行正式严格的技术审查和管理审查。审查的一条主要标准就是每个阶段都应该交出高质量的文档资料。文档资料是通信的工具,清楚准确地说明了任务是如何完成的、完成的情况以及下一步工作是什么等;文档资料也起备忘录的作用。

软件生命周期一般分为问题定义和可行性论证时期、系统分析时期、系统设计时期、编码和单元测试时期、系统测试时期与软件维护时期等六个阶段。也有人将其分为四个阶段即软件定义时期,包括问题定义、可行性论证和系统分析;软件开发时期,包括系统初步设计、系统详细设计、编码、单元测试和综合测试;软件维护时期包括修改、更新等活动。

我们将六个阶段的任务及结果文档用表 1.3 给出:

表 1.3

阶 段	任 务	结果及文档
问题定义和 可行性论证	明确问题是什么? 有可行的解否?	调研报告 可行性论证报告