

计算机科学基础

彭 沛

JISUANJI KEXUE
JICHU

东南大学出版社

计算机基础

彭沛

清华大学出版社

东南大学出版社

内 容 提 要

本书以程序设计方法、软件开发方法和使用计算机解决问题为主线,将计算机系统、算法设计、PASCAL 语言、数据结构、数值与非数值算法、软件工程等基本知识贯穿起来。本书结构严谨、说理清晰,附有大量例题,每章末均有小结和习题,便于自学。学完本课程并完成适当的实践性环节后,软件设计能力可达到较高水平。

读者对象:高校非计算机类的电类专业学生,一般工程技术人员和计算机业余爱好者。

责任编辑:朱经邦

计算机科学基础

彭 沛

*

东南大学出版社出版发行

(南京四牌楼2号 邮编 210018)

江苏省新华书店经销 南京雄州彩色印刷厂印刷

*

开本 787×1092 毫米 1/16 印张 26 字数 714 千

1995 年 6 月第 1 版 1995 年 6 月第 1 次印刷

印数:1—3000 册

ISBN 7—81050—020—1/TP·1

定价:17.80 元

(凡因印装质量问题,可直接向承印厂调换)

前言

计算机在各种领域中的应用正在不断深入与扩大。计算机不仅是解决电类工程的设计计算问题、信息处理问题的强有力的工具,而且在许多情况下,计算机(或单片机)就是电气、电子系统的有机组成部分。如果要利用计算机来设计电气、电子系统或用来处理信息,那么需要大量应用软件的支持,而这些应用软件与专业知识密切相关,开发这类应用软件的主导者应当是电类专业人才;如果电气、电子系统中电脑就是系统的有机组成部分,那么这些系统中的电脑硬件与软件的设计与电气、电子工程专业知识密切相关,它们的设计工作不能主要依靠计算机系的毕业生,而应依靠非计算机系电类专业的毕业生。这就是说,时代要求当代电类专业的学生对计算机科学领域有更广泛、深入的了解,时代对学生利用计算机解决问题的能力,包括硬件与软件能力提出了更高的要求。

我们认为,对于电类非计算机专业的学生而言,软件能力的基本要求是具有较好的程序设计能力;较高要求是具有开发与专业有关的应用软件的能力。

众所周知,程序是由算法和数据结构两大部分组成,而程序设计语言是表达算法和数据结构的工具。为了培养学生的程序设计能力,就应当让学生在算法设计方法、数据结构的选取与设计方法、程序设计语言的语法与语义这三个方面都受到教育与训练,缺一不可。仅掌握程序设计语言不一定能设计出好的程序,肯定设计不出好的大程序,正如熟悉汉语语法并掌握了大量词汇,哪怕是大量美丽词汇的人不一定能写出好的文章一样。

为了使具备开发与本专业有关的应用软件的能力,我们还应当让学生掌握一些常用的非数值算法、数值算法以及必要的软件工程知识。还应当让学生在开发小型软件方面受到一定的训练。

上述的各种基本方法、基本知识和基本技能,在内容上涉及到计算机专业中的〈算法的设计、分析与证明〉、〈数据结构〉、〈程序设计语言〉、〈数值分析〉、〈软件工程〉和〈软件实践〉等课程。由于学时有限,我们不可能把这些课程内容照搬到非计算机专业中。为了探索完成以上教学任务的模式,东南大学无线电工程系从1991年开始,开设了〈计算机科学基础〉课程。在这门课程中以程序设计方法、软件开发方法和使用计算机解决问题为主线把计算机系统、算法设计、PASCAL语言、数据结构、数值与非数值算法和软件工程等基本知识贯穿起来。

本书是在1991年所编写的〈计算机科学基础〉讲义基础上修改而成。下面介绍本书内容安排的考虑。

算法设计活动是一种构造性的思维过程,而大多数学生却习惯于数理化课程中使用的演绎性思维方法,不容易进入算法设计的思维方法。教学实践证明,要让学生掌握正确的算法设计方法并使坚持使用自顶向下的算法设计方法是一个很艰巨的任务。算法设计既是教学的重点又是教学难点。为了提高教学效果,本课程将算法设计的地位加以突出,具体表现在两个方面:其一,在内容安排的顺序上是先讲算法设计后讲程序设计语言;

其二,在整个课程中贯穿算法设计方法的应用。在进行程序设计语言的教学之前进行算法设计方法教学的好处有二。一是教、学双方都可集中精力搞好算法设计的教学,有利于培养学生的过程抽象能力,否则,若算法设计教学穿插在程序设计语言的教学过程中进行,则容易被程序设计语言的诸多语法规则而干扰;若是在程序设计语言的教学结束后再讲算法设计方法,那么学生在学习程序设计语言过程中形成的不良算法设计习惯就要付出额外的努力去纠正。教学实践证明,为纠正不良的算法设计习惯所付出的代价是很大的。好处之二是,先学了算法设计方法后,在本课程后面的程序设计语言、数据结构、非数值与数值算法的教学中,所有的程序设计举例都可应用算法设计方法,使学生反复接触正确的算法设计方法。

在该课程中使用伪码表示算法,其好处也有两个:一是伪码中只允许顺序、分支和循环这三种基本的算法结构,因而用伪码有利于写出结构良好的算法;二是伪码表示算法比较简单易行,减少了算法设计的工作量。根据以往的教学经验,我们认为第二点极为重要。学生之所以不易坚持使用自顶向下的算法设计方法,一个重要原因是嫌麻烦,在设计算法时总希望能一眼看穿,一口气就把算法的细节直接写出来,结果是欲速而不达,适得其反。虽然,表示算法的N—S图(盒图)也有利于设计出结构良好的算法,但是画图的工作量大,尤其是在算法设计的过程免不了要反复修改(学生是初学者,反复修改的情况是普遍的),每修改一次一般就要重画一次图,花费较多时间,容易导致学生对算法设计的厌烦,也不利于学生坚持使用正确的算法设计方法。

设计好的算法和数据结构用某种程序设计语言表示则成了程序。在本课程中详细地介绍PASCAL语言。之所以选用这种语言是基于以下几个原因:首先,该语言很严谨并支持结构化程序设计方法,将这种语言作为学生接触的第一语言有利于培养学生具有良好的程序设计和程序的书写风格。当学习了PASCAL语言后,有利于自学其它语言。第二,在进行数据结构内容的教学时,需要有描述数据结构的语言。PASCAL语言不但对几种常用数据结构(类型)提供了描述方法而且提供了构造各种新数据结构(类型)的方法。因此,在PASCAL语言的教学过程中,可以很方便地进行数组、记录等数据结构的教

学,利用指针类型可以讲解链表、堆栈、队列和树这几种数据结构,利用PASCAL提供的构造类型的能力可讲解自顶向下设计数据结构的方法。第三,PASCAL语言允许递归,便于向学生介绍递归概念和递归算法。

本课程把数据结构的基本知识分两部分:一部分是关于数组、记录、串、集合、文件、链表、堆栈、队列和二叉树这几种数据的构造形式及允许进行的操作;另一部分是查找、排序和归并算法。前一部分安排在PASCAL部分,与PASCAL语言的数据类型的教学结合进行。这样安排既可节省学时,又可帮助学生加深对数据类型的理解。后一部分安排在第九章中,这一章安排的内容并不多(约6个学时的内容),但对拓宽学生的视野和增加解决问题的手段很有好处。实践证明,学生学习了非数值算法的基本知识后,程序设计实践的课题大大地扩展了,其中不仅可以有本专业的非数值算法课题,也可以有各种信息管理方面的课题,提高了学生的学习兴趣。

本课程在讨论了算法设计的问题之后,设立了‘计算机系统与语言’一章,在这一章中简要地介绍了计算机的硬件系统,机器语言、汇编语言和高级语言的特征,数据在计算机

系统中的表示方法,编译程序和操作系统的概念。设立这一章的目的是为了提高PASCAL语言与数据结构的教学效果。例如,有了浮点数和定点数的概念就容易理解PASCAL中为何要区分整型数和实型数;有了编译的概念就容易理解PASCAL中的一些语法规则,也便于用堆栈概念讲清过程(或函数)调用和递归程序;有了内存贮器、外存贮器的概念就便于讲清文件用法比较复杂的原因。

数值计算仍然是电类专业应用计算机的一个重要领域,因此,在这门课程中给予了数值算法相当的地位,内容包括线性方程组、非线性代数方程和常微分方程的数值解法,插值与拟合,以及数值积分等。

一个能投入实用的软件不仅仅是程序,还应当包括开发、运行和维护程序的全部文档;另一方面,程序设计也仅仅是软件开发过程中的一个环节。作为一个将来可能肩负开发应用软件重担的电类专业的本科生来说不仅要掌握程序设计技术,还应当掌握基本的软件工程方法。为此,在本课程最后安排的是‘软件工程’一章。在这一章中,较详细地介绍了软件需求结构化分析技术、软件的结构化设计技术、软件的测试技术和建立文档的方法。

〈计算机科学基础〉课程是东南大学无线电系课程体系改革的产物之一。学生在学习了这门课程之后继续必修一门实践性课程〈综合程序设计〉,目的是让学生在开发小型软件方面接受一次较系统的训练。在〈综合程序设计〉课程中向学生提供并讲解一个按软件工程方法开发一个小型软件的详细例子,学生在做开发小型软件的实践课题时以这个例子作为样板。然后,给每个学生一个开发小型软件题目,给每个学生30小时的上机机时。要求每个学生交出一份文档齐备的作业。

经过3年的教学实践证明,学生在大学一年级期间学完这两门课程后,软件设计能力可达到较高的水平。

东南大学计算机科学与工程系秦振松副教授认真审阅了本书稿,提出了许多宝贵意见,在此表示衷心感谢。

作 者

于东南大学无线电工程系

1994.4

目 录

1 计算机与计算机科学	(1)
1.1 计算机系统的组成	(1)
1.2 电子计算机的发展简史	(3)
1.3 计算机科学	(4)
2 算法的设计与分析	(6)
2.1 算法的概念	(6)
2.2 算法的表示	(7)
2.2.1 赋值指令	(7)
2.2.2 循环指令, 输出指令	(9)
2.2.3 跟踪	(11)
2.2.4 条件指令	(14)
2.2.5 算法的三种基本结构	(19)
2.2.6 算法的两种图形表示	(20)
2.3 几种典型的算法	(24)
2.3.1 直接计算	(24)
2.3.2 枚举	(25)
2.3.3 递推	(29)
2.3.4 迭代	(31)
2.3.5 递归	(33)
2.4 结构化设计方法	(33)
2.4.1 自顶向下设计	(33)
2.4.2 算法的表达风格	(43)
2.5 算法的分析	(44)
2.5.1 执行时间函数	(45)
2.5.2 执行时间分析	(47)
2.6 小 结	(50)
2.7 习 题	(50)
3 计算机系统与语言	(53)
3.1 数据在计算机中的表示方法	(53)
3.1.1 位权	(53)
3.1.2 数系的转换	(54)
3.1.3 比特、字节和字	(57)
3.1.4 数在计算机中的定点表示法	(57)
3.1.5 数在计算机中的浮点表示法	(58)
3.1.6 字符和字符串的表示	(59)

3.2	数字计算机的硬件组成	(59)
3.2.1	中央处理器	(61)
3.2.2	主存贮器	(62)
3.2.3	辅助存贮器	(63)
3.2.4	输入输出子系统	(64)
3.3	程序设计语言的概念	(65)
3.3.1	机器语言	(65)
3.3.2	汇编语言	(67)
3.3.3	高级语言的特征	(69)
3.4	系统软件	(70)
3.4.1	翻译程序	(70)
3.4.2	操作系统	(71)
3.5	小 结	(75)
3.6	习 题	(75)
4	PASCAL 的简单数据类型与程序结构	(76)
4.1	PASCAL 语言的基本成分	(76)
4.1.1	字 符	(76)
4.1.2	词	(77)
4.2	数据类型	(79)
4.2.1	整型	(80)
4.2.2	实型 REAL	(84)
4.2.3	字符型 CHAR	(86)
4.2.4	布尔型	(87)
4.2.5	枚举型	(88)
4.2.6	子界型	(89)
4.3	变量说明与常量定义	(90)
4.3.1	变量说明	(90)
4.3.2	常量和常量定义	(91)
4.4	PASCAL 程序的基本结构	(92)
4.5	赋值语句与表达式	(95)
4.6	输入输出语句	(99)
4.6.1	输入语句	(100)
4.6.2	输出语句	(102)
4.7	简单的 PASCAL 程序	(105)
4.8	小 结	(108)
4.9	习 题	(108)
5	PASCAL 的控制结构	(112)
5.1	复合语句	(112)
5.2	重复性语句	(113)
5.2.1	当语句	(113)

5.2.2 重复语句	(117)
5.2.3 循环语句	(120)
5.3 条件语句	(124)
5.3.1 如果语句	(124)
5.3.2 情况语句	(130)
5.4 臭名昭著的转语句	(133)
5.5 综合应用举例	(136)
5.6 小 结	(142)
5.7 习 题	(143)
6 PASCAL 的结构数据类型	(147)
6.1 一维数组	(147)
6.1.1 一维数组的类型定义与变量说明	(147)
6.1.2 一维数组的操作	(149)
6.1.3 一维数组应用举例	(151)
6.2 二维数组	(157)
6.2.1 二维数组的类型定义与变量说明	(158)
6.2.2 二维数组的操作	(159)
6.2.3 二维数组应用举例	(160)
6.3 字符串类型	(163)
6.3.1 串类型的定义与串变量的说明	(163)
6.3.2 字符串的操作	(164)
6.3.3 字符串应用举例	(165)
6.4 记录	(168)
6.4.1 记录的类型定义与记录变量的说明	(168)
6.4.2 记录变量的操作	(170)
6.4.3 数据的构造	(171)
6.4.4 开域语句	(181)
6.4.5 记录的变体部分	(183)
6.5 集合	(186)
6.5.1 集合的类型定义与集合变量的说明	(187)
6.5.2 集合的构造与运算	(188)
6.6 数组、记录和集合的顺序存贮结构	(192)
6.6.1 数组存贮结构	(193)
6.6.2 记录存贮结构	(194)
6.7 小 结	(194)
6.8 习 题	(194)
7 PASCAL 的函数与过程	(198)
7.1 函数	(198)
7.1.1 函数说明	(199)
7.1.2 函数的引用	(201)

7.2	过程	(204)
7.2.1	过程说明	(204)
7.2.2	过程调用	(205)
7.3	值参数与变量参数	(206)
7.4	存贮分配	(208)
7.5	程序的层次结构	(212)
7.5.1	层次结构的产生	(212)
7.5.2	标识符的作用域	(213)
7.5.3	调用关系	(216)
7.6	函数的副作用	(217)
7.7	递归	(218)
7.8	程序设计举例	(221)
7.9	小 结	(225)
7.10	习题	(225)
8	动态数据结构	(228)
8.1	指针类型与动态变量	(228)
8.1.1	指针类型与指针变量	(228)
8.1.2	动态变量	(229)
8.1.3	利用指针反映动态变量之间的关系	(230)
8.2	链表	(232)
8.2.1	链表的建立	(232)
8.2.2	访问链表中的元素	(234)
8.2.3	链表元素的插入与删除	(237)
8.3	堆栈与队列	(243)
8.3.1	堆栈的操作	(244)
8.3.2	队列的操作	(245)
8.4	二叉树	(147)
8.4.1	二叉树的遍历	(248)
8.4.2	二叉树的插入与删除	(254)
8.5	小 结	(258)
8.6	习 题	(258)
9	排序及文件	(262)
9.1	查找	(262)
9.2	排序技术	(265)
9.2.1	选择排序	(265)
9.2.2	交换排序	(267)

9.2.3	插入排序	(274)
9.3	合并	(275)
9.4	文件	(277)
9.4.1	PASCAL 文件的定义与说明	(278)
9.4.2	PASCAL 文件的使用	(279)
9.5	小结	(292)
9.6	习题	(292)
10	数值算法——方程求解	(295)
10.1	数值算法的基本概念	(295)
10.1.1	误差的概念	(295)
10.1.2	数据误差对计算结果的影响	(297)
10.1.3	算法的数值稳定性	(299)
10.2	非线性代数方程的数解法	(301)
10.2.1	简单迭代法	(301)
10.2.2	牛顿(Newton)迭代法	(305)
10.3	线性方程组的数值解法	(308)
10.3.1	高斯消去法	(309)
10.3.2	选主元的高斯消去法	(313)
10.3.3	三角分解法	(316)
10.3.4	迭代法	(318)
10.4	常微分方程的数值解法	(322)
10.4.1	台劳方法	(322)
10.4.2	线性多步法	(324)
10.4.3	稳定性	(329)
10.4.4	变步长方法	(333)
10.4.5	算法	(334)
10.5	小结	(334)
10.6	习题	(335)
11	数值算法——插值、拟合与求积	(337)
11.1	插值	(337)
11.1.1	拉格朗日插值	(337)
11.1.2	牛顿插值	(342)
11.1.3	高次插值的缺点及分段低次插值	(348)
11.1.4	曲线拟合的最小二乘法	(350)
11.2	数值积分	(352)
11.2.1	基本概念	(353)
11.2.2	牛顿-柯特斯(Newton-Cotes)公式	(354)

11.2.3 复合求积	(358)
11.2.4 步长的自动选择	(360)
11.2.5 龙贝格(Romberg)方法	(362)
11.3 小 结	(366)
11.4 习 题	(366)
12 软件工程	(368)
12.1 软件工程的概述	(368)
12.1.1 软件生命期	(368)
12.1.2 软件的质量标准	(370)
12.2 需求分析	(372)
12.2.1 数据流图	(372)
12.2.2 数据词典	(375)
12.2.3 小说明	(377)
12.3 软件设计	(379)
12.3.1 程序的模块结构	(379)
12.3.2 设计阶段的任务	(384)
12.3.3 概要设计	(385)
12.3.4 详细设计	(390)
12.4 编程	(390)
12.5 测试	(391)
12.5.1 测试的目的与特性	(392)
12.5.2 测试方法	(395)
12.6 维护	(398)
12.7 小 结	(399)
12.8 习 题	(399)
参考文献	(400)

1 计算机与计算机科学

电子计算机(简称为计算机)是 20 世纪科学技术的卓越成就之一。由于它具有极高的运算速度、极高的运算精度、极强的记忆功能和卓越的逻辑判断能力,目前已广泛应用于科学研究、生产、生活和军事等各种领域。研究和使用计算机的人越来越多,又促使了计算机与计算机科学的迅速发展。计算机与信息工程、电子工程关系尤为密切,这不仅因为信息工程与电子工程中的许多问题需要使用计算机这样一个强有力的工具来解决,而且在许多情况下计算机中的核心部件就是其它电子产品和电子系统中的有机组成部分。

本章我们将说明什么是电子计算机系统和什么是计算机科学这两个问题。为了说明第一个问题,将介绍计算机的基本组成和基本原理以及计算机的发展简史。为了说明第二个问题,我们将简要地介绍计算机科学的研究领域。

1.1 计算机系统的组成

电子计算机是用于信息处理的现代化的电子设备。根据所处理的信息是数字量或是模拟量(连续变化的量),电子计算机又可分为数字计算机和模拟计算机。由于当前广泛应用的是数字计算机,因此,通常把电子数字计算机简称为电子计算机。

为了说明电子计算机的组成和基本工作原理,让我们先考察人用纸和笔来解题的过程。首先,根据给定的题目,确定解题的方法和解题步骤,为了备忘,应当把原始数据和解题步骤写在纸上,对于简单的题目,这一步似乎可以省去,但在一般情况下,这一步是必须的。然后,按解题步骤逐步计算,把中间结果写在纸上,直到算出最后结果。最后,把结果写在纸上,计算完毕。

例如,对于计算问题 $112 + 72 \times 4 = ?$, 其解题过程为

第一步,根据先乘除后加减的运算次序,可写出原始数据与解题步骤如下。

原始数据: $a=112$, $b=72$, $c=4$

解题步骤: 1. 计算 $b \times c$ 的值;

2. 计算 $a + (b \times c)$ 的值)

第二步,按解题步骤逐步计算,把中间结果写在纸上。

1. 计算 $b \times c$ 的值,值为 288, 写在纸上;

2. 计算 $a + 288$, 最后结果为 400

第三步,把最后结果 400 写在纸上。

从以上解题过程来看,如果要设计一个机器自动地执行以上计算步骤,这个机器必须具备以下物质基础:

1) 记忆装置。用来存放原始数据、解题步骤、中间结果和最后结果。记忆装置相当于人工方式解题时用的纸。

2) 控制与计算装置。它用来控制整个计算过程并进行计算。这相当于人工方式解题

时人的大脑。

3) 输入输出装置。输入装置用来将原始数据、解题步骤输入机器中。这相当于人工方法解题时用的眼、手和笔。输出装置用于把计算结果送出机器。

电子计算机就是模拟上述解题过程的现代化电子设备。因此,它也具备相应的三大组成部分:

1) 主存储器。它用来存贮为求解问题所需的原始数据和解题的详细步骤,还要存贮中间结果和最后结果。在计算机科学中把解决问题的一个步骤称为一条指令。能解决问题的指令序列称为算法。

主存储器好比办公室中的文件立柜。文件柜中整齐地排列了许多小抽斗,每个小抽斗内可存放文件,对于所有小抽斗按顺序编上号码。当要存或取文件时,先根据该文件存放的抽斗号码到相应的抽斗里去存(或取)文件。

计算机的主存储器是由许多排列整齐的存贮单元组成。

一个存贮单元相当于文件柜中的一个小抽斗。对所有存贮单元也按顺序编上号码(实际上,这号码并没有写在存贮单元上

面,而是一种约定)。在计算机科学中,把给存贮单元编的号码称为地址,如图 1.1 所示。当要从某个存贮单元取出所存贮的

内容(信息)或将一信息存放到某个存贮单元中去时,应当先给出该存贮单元的地址,然后按这个地址对相应的存贮单元

进行内容的取或存的操作。

计算机的存储器有两个重要的特性:

其一,设在某单元中已存有一数据(或其它信息),现将一新数据存进去,则老数据自动消失。因此,存贮单元总是保留最新存进去的数据或指令。其二,若从某存贮单元取数据(或其它信息),取走的是副本(相当于复印件),原件仍在该存贮单元中。存贮单元这两个特性可用‘取之不尽,一挤就走’这两句话来概括。

2) 处理器。它的主要作用是从主存储器中按指令执行的次序依次取出并执行各条指令。当处理器从主存储器取来一条指令后,接着就分析这条指令(即弄清楚这条指令的含义),然后依照这条指令的要求向计算机各部件发出各种控制信号(或称为命令),使主存储器和其它部件与处理器协调地完成这条指令规定的操作。当各个部件执行完处理器发出的命令后,会向处理器汇报命令执行情况的反馈信息。当处理器得知这条指令执行完成后,自动地去取下一条要执行的指令,重复上面的过程。当所有指令执行完毕,问题的解就得到了。

3) 输入和输出设备。它们负责将信息送入和送出计算机。输入设备用于输入指令和数据,输出设备把结果送出计算机。输入输出设备又称为外部设备。在计算机科学中常用符号‘I/O’表示输入输出,其中 I 是 INPUT 的第一个字母, O 是 OUTPUT 的第一个字母。

以上三个计算机的组成部分构成了计算机的硬件系统。

从以上介绍可见,光有计算机的硬件设备和原始数据,计算机是什么事也不会做的。我们还必须向它提供解决问题的详细步骤以便让计算机能按照人们的意图进行计算。如

N-1	
N-2	
⋮	⋮
2	
1	
0	
地址	内容

图 1.1 主存储器的表示

果你想要计算机为你解题,首先,你应当用计算机能够理解的语言来编制解题的步骤,这就是编写程序。接着把编好的程序连同原始数据,通过输入设备存入计算机的主存储器中。然后启动计算机,计算机便在程序的控制下,按人的意图自动地进行操作,直到全部计算完毕,通过输出设备送出结果。电子计算机能存储程序并在程序的控制下工作是它区别于其它机器的一个重要特征。普通袖珍计算器虽也有所谓存储器,但一般很小,只能存放几个参加运算的数据,根本不存放指令,计算器的运算指令是由按键输入的,并且是随按随算,不按则不算。这是普通计算器不能称为计算机的原因。迄今为止,电子数字计算机所共同遵守的程序存储与程序控制的原理是在 1945 年由冯·诺依曼提出的,故这原理又称为冯·诺依曼型计算机原理。

一个完整的计算机系统,除了硬件系统外还包括软件系统。而软件系统又是由系统软件和应用软件这两类软件组成。

计算机制造厂家为了让用户方便地使用计算机和为了充分利用计算机的硬件设备,向用户提供一系列程序和这些程序的使用手册,这些程序和使用手册总称为系统软件。

为解决用户的某一类问题而设计的程序及其使用手册称为应用软件或用户软件。

计算机区别于其它机器的第二个重要特征是它具有通用性。即它能解决不同类型的信息处理问题。实际上,只要你给计算机的程序不同,它就可以执行不同的任务。例如,一个程序可以让计算机来平衡帐目,另一个程序可以让计算机下国际象棋,第三个程序可以让计算机求解联立方程组,等等。

什么是计算机?我们可以这样回答:电子计算机是一种能按预先存储的程序,对以数字形式表示的信息进行处理的通用电子设备。

电子计算机的发展历史

1.2 电子计算机的发展简史

自从第一台电子计算机问世以来,已经历了四代。

1. 第一代电子计算机

第一代电子计算机即电子管计算机。第一台电子计算机于 1946 年在美国制成,取名为 ENIAC。它是一个由 18000 多个电子管制成的庞然大物,占地面积达 170m^2 ,重量有 30 多吨,耗电功率 140kW ,其运算速度为每秒 5000 次加法。此后,从 1947 年至 1959 年的大约 10 年期间,电子计算机一直成为广泛研制生产和使用的机种。

第一代电子计算机的主要特征是采用电子管组成基本逻辑电路,用磁鼓或延迟线作为主存储器。第一代电子计算机使用机器语言或汇编语言编制程序。它主要用于科学计算(数值计算)。

由于第一代电子计算机使用了大量电子管,使得它不但造价高、体积大、耗能高,而且故障率也高,平均稳定运行时间只能达到几小时。正是它的这些缺点,使它很快被第二代电子计算机所淘汰。

2. 第二代电子计算机

20 世纪 50 年代末期,随着半导体材料的发展,人们制造出体积比电子管小得多,耗能也比电子管低得多,而功能完全可以代替电子管的半导体晶体管。于是,全部采用晶体

管组装的晶体管计算机出现了,这就是第二代电子计算机。第二代电子计算机的生存时间大约是从1957年至1964年。

第二代电子计算机的主要特征是采用晶体管作基本逻辑电路,以磁芯存储器作为主存储器,在外部设备中开始使用磁盘;在软件方面,出现了高级的程序设计语言,如ALGOL语言,FORTRAN语言和COBOL语言等。同时在这一代计算机上出现了管理计算机资源和方便用户使用计算机的操作系统。第二代电子计算机的运算速度提高到每秒钟可做几十万次以上的加法。它的使用范围扩展到数据处理、自动控制、企业管理各方面。

3. 第三代电子计算机

从1965年开始研制的用中、小规模集成电路组成的计算机就是第三代计算机。所谓集成电路就是将多个晶体管和电阻元件集中做到一小块硅片上,制成门电路、触发器等具有一定逻辑功能的电路器件。与晶体管电路相比,集成电路大大地缩小了体积,降低了功耗,因此也就提高了可靠性。

第三代电子计算机的主要特征是采用中、小规模集成电路作为基本逻辑电路。主存储器仍以磁芯存储器为主。外围设备的种类不断增加,终端设备迅速发展。操作系统得到发展和普及。计算速度可达到每秒钟做几百万次以上的加法。

4. 第四代电子计算机

一般把在一块硅芯片上集成100个门电路以上或上千个晶体管元件以上的集成电路叫做大规模集成电路。由大规模集成电路构成的电子计算机就是第四代电子计算机。

第四代电子计算机的研制开始于70年代初,至今还是兴旺时期。由于大规模集成电路的应用使得这一代计算机比前几代计算机在性能上有很大提高,而且随着集成电路规模的迅速扩大使得第四代计算机发展速度也越来越快。其发展趋向向两端发展,即出现了运算速度超过亿次的巨型计算机和以微处理器为核心组装的微型计算机。在这一代计算机的硬件结构中,磁芯存储器基本被淘汰,普遍使用半导体存储器。软件系统的飞速发展更是这一代计算机的明显特征,近20年来对高级语言、操作系统、数据库、应用软件等的研究和应用越来越深入、完善。软件行业已成为一个现代工业部门。

1.3 计算机科学

要回答‘什么是计算机科学’是困难的。但人们仍然力求对这个问题作出回答,典型的回答有这样两种:

一种是从面向工程的角度,将计算机科学定义为研究计算机。这个定义强调了计算机是实在的工具也强调了计算机的性能,但忽视了这些工具的使用和使用它们的技术。

另一种定义走向另一个极端,按程序设计的观点把计算机科学定义为研究计算机能解决什么样的问题,如何表达问题以及解决问题的技术。这个定义强调了如何使用计算机及使用的技术,而对计算机本身漠不关心。

这两个定义都过于狭窄,就是将两个定义加以合并形成一个新的定义也不合适。原因之一是,计算机科学是一门非常年轻的科学,‘计算机科学’这个名词才出现30年,而且,这门学科仍然处于迅速发展之中,至今我们无法知道这门学科的全貌。第二个原因是,计

计算机科学涉及到其它多种学科,如数学、机械工程、电气工程、语言学等等。有许多问题是属于计算机科学家还是属于数学家、电气工程师、语言学家的专业范围尚不清楚。例如,计算机的设计问题照理应当由计算机科学家来解决,但是计算机设计的问题包含了大量的电子工程问题与数学问题,因此,计算机设计的理论问题似乎应当由电气工程师和数学家来解决。而关于编程语言的许多问题由语言学家或心理学家来回答可能更好。

由于以上原因,目前想用很简单的一种概念来概括全部计算机科学的内容是困难的。但是,不能定义计算机科学并不妨碍计算机科学的发展,也不妨碍我们对这个学科的学习和研究。目前许多大学设有计算机科学系,许多工厂和研究单位需要计算机科学系的毕业生,对非计算机专业的毕业生也要求具备计算机科学的基础知识。这些足以表明人们对计算机科学领域有大体一致的观念,一般认为计算机科学包括如下的四个子领域:

1. 计算机科学理论

计算机科学理论是以计算机的算法作为研究对象的科学,在这门科学中把所研究的问题抽象成数学形式然后利用数学方法进行研究。计算机科学理论研究的内容包括:形式语言理论,程序设计方法学,算法的分析等。

2. 计算机系统

毫无疑问,计算机系统本身应当是计算机科学的研究对象。计算机系统这个领域由硬件技术和软件技术这两个分支构成。硬件技术关心的是如何利用最新和最先进的物理部件设计具有某种性能的计算机硬件系统。例如,设计由卫星线路连接的计算机网是一项大的计算机硬件设计工程。软件技术包括操作系统的研究和开发,程序设计语言的翻译技术的研究,软件开发工具的研究与开发等。

3. 计算机应用

这是一个广泛的领域,也是设计制造计算机的最终目的。用计算机来进行各种科学和工程中的数值计算是早期出现的(50年代到60年代)计算机的主要用途。当前,虽然数值计算仍然是计算机的一个重要的应用领域,但非数值方面的应用发展很快。目前计算机已被广泛地用于政府机构、学校、金融业、商业、企业等的管理工作中。其中,计算机可胜任的工作有库存控制,排序处理,人员管理,会计,报表产生,办公室自动化等。在这些应用中往往要使用数据库管理系统。

人工智能和机器人是一个正在兴起和应用广泛的领域。它包括问题求解技术,定理的自动化证明,自然语言的理解与产生,话音和视觉信息处理,专家系统开发和工业机器人。

计算机还广泛地用于生产过程、武器系统和航空系统的控制等。用于控制的计算机与正在运行的系统密切相关,我们要求计算机对输入的数据信号必须在指定的时间内作出响应,即在指定的时间内要产生计算结果并发出相应的控制信号。

4. 社会问题

随着计算机的广泛应用,各种工作的自动化程度越来越高。自动化对社会的影响错综复杂,如,自动化程度提高可以增加劳动生产率,降低产品成本,提高产品质量;另一方面,存在工作岗位减少,工人再培训,人们空闲时间增多带来的问题。另外,当人们的档案保存在大型数据库中时,也构成了对人们隐私的威胁。还有,由于大量使用计算机,也给罪犯提供了利用计算机犯罪的机会。以上问题都引起了计算机科学家与社会学家们的注意。