

计算机软件基础 与微型计算机 体系结构

梁德群 编

高等教育出版社

内 容 提 要

本书是计算机应用的基础性教材。它用通俗的叙述将软件领域中的数据结构、程序设计语言、程序设计方法、操作系统和数据库等多方面的知识,以及微型计算机体系结构的知识有机地组织在一起,注意了基础和实用相结合。

本书可作为生物医学工程与仪器专业、自动控制专业的本科生教材或研究生教材,也可作为其他非计算机类专业研究生的选修课教材,还可作为计算机应用工作者和业余程序员自学用书。

本书经国家教委高等工业学校生物医学工程与仪器专业教材委员会于1987年4月召开的全体会议审稿,同意作为高等学校试用教材出版。

计算机软件基础 与微型计算机体系结构

梁德群 编

清华大学出版社

新华书店北京发行所发行
北京顺义县印刷厂印装

*

开本 850×1168 1/32 印张 16.875 字数 410 000

1989年2月第1版 1989年3月第1次印刷

印数 0 001— 1 150

ISBN7-04-001159-X/TN·102

定价 4.00 元

前 言

对于生物医学工程与仪器、自动控制及电类专业的科技工作者来说,计算机已不仅仅是他们进行科学计算的工具,而且已成为系统(如一个医疗器械,一个自动控制系统,一台通信装置等)中的一个重要环节,这个环节正起着越来越大的作用。

今天的计算机已不再是以硬件为主的机器,而是在操作系统统一管理下硬件和软件紧密结合的完整系统。它提供了各种软、硬件的工具,使用户可以方便地调用系统的各种资源,为提高计算机的使用效率提供了良好的环境。然而,这个环境需要人去熟悉它,而且对环境了解得越好,则使用计算机的效率也越高。计算机的种类多种多样,每种计算机都有自己的环境。本书的目的在于,通过对硬件和软件系统的一些基础知识的讲述,使学生掌握计算机(主要是微型计算机)系统环境的共同特点,为在实际应用中掌握各种具体的计算机系统打下基础。

本书包括如下内容:

第一、二章是基础部分,包括数理逻辑初步、集合中的关系和本书所使用的描述工具——算法描述语言以及数据结构。

第三章在总结过程式程序设计语言共同特点的基础上,集中地介绍了PASCAL语言的各种概念和具体结构。PASCAL语言将是以后几章的描述工具之一。

第四章介绍了现代程序设计方法的基本理论和技术,以求使学生自觉地培养起良好的程序设计习惯,提高开发软件的效率。

第五章从体系结构的角度出发,对几种典型的微处理器的结构、微型计算机的存储器组织、寻址方式和指令系统的组成做了总

结性的介绍。还介绍了硬件对操作系统的支持技术、非微型计算机中的体系结构知识和并行处理技术。

第六章重点介绍了CP/M微型机磁盘操作系统的内部结构和IBM-PC机中的MS-DOS操作系统的内部功能调用方法,另外还涉及到一般操作系统的原理。

第七章在简单讲述了数据库一般原理后,着重介绍了dBASE III的使用方法。

这本教材是按70~80学时的讲授内容编写的,讲授时可根据实际学时数以及学生的基础,对内容做适当的删减。

如果学生已学过集合论,则在讲授第一章时应把重点放在关系的形式化表示上。关于算法描述语言,可在以后各章中具体应用时加强解释,这样可以减少学时。

在讲授第二章的数据结构时,应把重点放在各种结点的表示和指针的灵活应用上,不一定详细介绍所有算法。

如果另有专门的PASCAL语言课,则第三章可不讲,或只总结一下过程式程序设计语言的共同特点。

若学时较少,可跳过第五章中的大部分内容,只讲那些与操作系统有关的部分。关于8088的汇编语言,尽管在讲述MS-DOS内部功能调用时要用到,却仍可不讲,只需在用到时对指令进行逐条解释即可,因为我们的读者均应具备某种汇编语言知识。

第六章的重点应放在CP/M的文件管理、系统调用的原理和MS-DOS中内部功能的调用方法上。

如果有条件,建议把第七章中关于dBASE III各种命令的具体使用方法作为实验课的内容。

这本教材涉及的面是很广泛的。如何保证在有限的篇幅内把这些内容讲透,避免泛泛而谈,是编写本教材中着力克服的难点。按照这一目标,作者在1982年编写了《计算机控制软件基础》讲

义,并为西安交通大学自控专业高年级开出了选修课;1985年继续以《计算机控制软件基础》为基本教材,扩充了一些内容,为信息与控制工程系的研究生开设了《计算机软件基础和体系结构》课;1986年编出了《计算机软件基础和体系结构》的讲义,并继续为信息与控制工程系研究生开课。现在这本教材是在上述两本讲义的基础上,根据教材委员会和诸位审稿者的意见改写而成的。特别是担任主审的南京工学院吕玉琦副教授,在取材、结构和具体内容上都提出了有价值的意见。另外,陕西机械学院林魁明老师审查了部分章节并提出宝贵意见,还有作者所在单位西安交通大学图象处理研究室的蔡元龙教授和傅世崇副教授及其他同志对本教材的编写提供了支持。在此一并表示衷心的感谢!

本书第七章中 dBASE III 内容的原稿是由西安交通大学刘甘娜老师编写的;第六章中汇编语言程序的例子和 MS-DOS 内部功能调用的素材是由西安交通大学王瑛老师提供的。

作者在多年的教学实践和编写本教材的过程中,深深体会到,要把这样广泛的内容有机地组织成一本篇幅不大的教材实非易事,因此书中的错误在所难免,恳请使用本书的各位老师和学生以及其他读者给予指正。

梁德群

1988年1月

目 录

第一章 预备知识	1
§1.1 集合与关系.....	1
1.1.1 集合的定义.....	1
1.1.2 数理逻辑中的一些概念和符号.....	2
1.1.3 集合的表示.....	7
1.1.4 集合的包含和相等.....	7
1.1.5 集合的某些基本运算.....	8
1.1.6 集合中的关系.....	11
§1.2 算法及算法描述语言.....	21
1.2.1 算法.....	21
1.2.2 算法描述语言.....	22
习题.....	30
第二章 数据结构	34
§2.1 数据结构的定义、表示及分类.....	34
2.1.1 定义.....	34
2.1.2 表示.....	36
2.1.3 分类.....	41
§2.2 有序表.....	41
2.2.1 有序表的定义.....	41
2.2.2 有序表的操作类型.....	42
2.2.3 有序表的不同形式.....	43
2.2.4 有序表的几种主要操作算法.....	44
2.2.5 有序表的应用举例.....	63
§2.3 树.....	73
2.3.1 定义.....	73
2.3.2 树的表示法.....	74
2.3.3 二叉树.....	75

2.3.4	树的二叉树表示	84
2.3.5	树的搜索算法	85
2.3.6	树的应用举例	89
§2.4	图	92
2.4.1	定义和术语	92
2.4.2	图的表示法	94
2.4.3	图的遍历算法	97
2.4.4	图的应用举例	99
	习题	101
第三章	PASCAL 语言及过程式程序设计语言	106
§3.1	数据类型及其在 PASCAL 中的表现	107
3.1.1	变量名	107
3.1.2	变量类型的定义及相应的操作	108
3.1.3	表达式	132
§3.2	程序的结构	136
3.2.1	组织结构	136
3.2.2	程序的控制结构	166
	习题	167
第四章	程序设计方法	179
§4.1	概述	179
§4.2	问题定义及问题的分解	184
4.2.1	问题定义	184
4.2.2	问题的分解	191
§4.3	结构化程序和程序的正确性证明	196
4.3.1	结构化程序	196
4.3.2	结构化程序设计及程序正确性证明	200
§4.4	算法设计及分析	208
4.4.1	算法设计	209
4.4.2	算法分析	231
§4.5	程序的优化	234

习题	238
第五章 微型计算机体系结构	244
§5.1 冯·诺伊曼(Von Neumann)型计算机结构特点	244
§5.2 微处理器的结构	246
5.2.1 累加器型处理器	246
5.2.2 通用寄存器型处理器	247
5.2.3 堆栈型处理器	248
5.2.4 其它类型的处理器	249
§5.3 存储器、寻址方式及存储管理	250
5.3.1 存储器的组织结构	250
5.3.2 寻址方式	252
5.3.3 存储管理	255
§5.4 指令	258
5.4.1 指令格式	259
5.4.2 指令的种类	261
5.4.3 指令的实现	262
§5.5 输入/输出	265
5.5.1 编址	265
5.5.2 接口	268
5.5.3 数据缓冲队列	269
5.5.4 通道	269
§5.6 中断	271
5.6.1 中断的概念	271
5.6.2 中断的处理	272
5.6.3 中断的级别与优先权	273
5.6.4 直接存储器存取(DMA)	274
§5.7 IBM-PC的结构及汇编语言	274
5.7.1 8088 处理器	275
5.7.2 IBM-PC系统的组成	282
5.7.3 8088用的汇编语言	291
§5.8 并行处理计算机系统	305

5.8.1	概述	305
5.8.2	分布式系统	307
5.8.3	数据流计算机	311
	习题	311
第六章	操作系统	314
§6.1	概述	314
§6.2	CP/M 操作系统的内部结构	319
6.2.1	CP/M 的基本结构及命令	319
6.2.2	磁盘及文件结构	328
6.2.3	BDOS 的结构及执行过程	339
6.2.4	BIOS 的结构	350
6.2.5	CCP 的结构及执行过程	358
§6.3	操作系统中一些其它的概念	363
6.3.1	进程	364
6.3.2	存储管理	377
§6.4	IBM-PC MS-DOS 的功能及使用	382
6.4.1	MS-DOS 的内部结构	382
6.4.2	MS-DOS 的功能	389
6.4.3	MS-DOS 的使用	400
	习题	408
第七章	数据库的使用	410
§7.1	数据库管理系统的原理	410
§7.2	dBASEIII 的基本性能和进入	417
7.2.1	dBASEIII 的运行环境	418
7.2.2	dBASEIII 系统的启动和退出	419
7.2.3	dBASEIII 的求助措施	419
7.2.4	dBASEIII 的文件参数和技术参数	420
§7.3	dBASEIII 的语法基础和非文件性操作	422
7.3.1	dBASEIII 的语法基础	422
7.3.2	dBASEIII 命令系统	425
7.3.3	dBASEIII 的系统环境状态的改变及利用	426

7.3.4	dBASEIII 的两种命令方式	430
§7.4	数据库文件的操作	430
7.4.1	有关文件名的操作	430
7.4.2	数据库文件的建立	431
7.4.3	数据库文件的间接建立	433
7.4.4	数据库文件的修改	435
§7.5	数据库文件的使用	440
7.5.1	数据库文件的检索	440
7.5.2	数据库文件的索引和排序	444
7.5.3	数据库文件的统计和计算	450
7.5.4	报表生成及输出	453
§7.6	dBASEIII 的应用程序	457
7.6.1	应用程序的设计及运行	457
7.6.2	程序控制语句	458
7.6.3	应用程序设计实例	465
§7.7	dBASEIII 与高级语言的接口	470
7.7.1	用输出文件(.TXT)实现数据共享	470
7.7.2	使用 RUN 命令连接两种程序	471
	习题	473
附录 1	PASCAL 语法图	475
附录 2	CP/M 操作系统的系统调用命令	480
附录 3	8088 的指令及时序	485
附录 4	MS-DOS 的内部功能	496
附录 5	MS-DOS(2.0)常用操作命令	502
附录 6	IBM-PC 的全屏幕编辑及功能键	514
附录 7	dBASEIII 的内部函数	516
附录 8	dBASEIII 命令	518
	参考文献	527

第一章 预备知识

§1.1 集合与关系

本节的重点是介绍关系的数学表示，在介绍关系之前先介绍集合的定义、简单运算和数理逻辑中的一些概念与符号，以此作为本书的数学基础。

1.1.1 集合的定义

集合是一种元概念，即无法用“通常的办法”定义的概念。这种“通常的办法”是，用已经定义了的概念去定义新的概念。这种办法势必产生一个“定义源”的问题，到了“定义源”就再也找不到已经定义过的概念了。这个定义源就是元概念。

对于元概念可用如下三种方法定义：

a. 解释性定义 许多元概念为大众常识所承认，是不必给予严格定义的，只需做出一定解释即可。比如，我们可以把集合说成是“具有某些特殊性质的客体的全体”。特殊性质的客体是指组成集合的各个部分，这些部分之间由于性质的特异而相互区别。客体是理解集合这一概念的关键，只要有客体存在就能形成集合。根据这个定义，我们举出一些具体集合的例子，如：“整数的集合是由整数的全体组成的”，“人类是全体个人的集合”。

b. 枚举法 有时我们可以通过列举一事物的所有性质或包含的客体来定义这一事物。比如，我们可以把“十以内整数的集合”

· · ·

定义为 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$; 又比如, 可以把英文字母表定义为二十六个英文字母的集合, 即 $\{A, B, C, \dots, X, Y, Z\}$ 。

c. 递归定义法 这种方法是, 经过一系列的定义后又回到最初要定义的元概念本身。比如, 我们可以这样定义集合: “集合是元素组成的全体。元素是集合中的成员。”我们用元素定义集合, 又用集合定义元素。初看起来, 似乎根本没给出定义! 实际上这是一种很好的定义法。它清楚地指明了如下的事实: 只要有能分清的元素存在, 并把它放在一起, 就是一个集合。至于元素是什么, 定义不做任何规定。这就使集合具有高度的抽象性。在科学的基础上, 抽象性越强, 越能深入事物的本质, 并且具有更广泛的应用性。比如, 可以把人类说成是“每一个人”的集合; 也可以说成是“黄种人”、“白种人”、“黑种人”和“棕种人”的集合; 还可以说成是“中国人”、“美国人”、“苏联人”、……的集合。上述三种说法中, 用了不同的元素来定义“人类”这个集合。

1.1.2 数理逻辑中的一些概念和符号

1. 命题

一个命题就是一个语句, 但必是一个能判断其真假的句子。比如, 感叹句、疑问句、祈使句等就不能判断其真假, 而陈述句就能判断其真假。

例如: “ $1 < 2$ ”和“狗是动物”都是真命题, 而“ $1 > 2$ ”和“狗是植物”则是假命题; 象“您好吗?”和“您请坐!”这样的句子就不能称为命题。

命题又被称为断言。

如果一个命题不能进一步被分解为更简单的命题, 则该命题叫原子命题。由原子命题可组成复合命题, 复合命题还可组成新的复合命题。

一个命题可取两种真值之一：真(T)和假(F)。

常用大写英文字母 A, B, C, \dots 表示命题，并且以如下形式表示一个具体的命题：

字母：句子

比如， A ：月亮是行星。

命题有变量和常量之分。若一个字母表示一个特定的命题，则说该字母是一个命题常量，特定命题的真值就是此命题常量的真值。若一字母表示任一命题时，该字母是一个命题变量，它是没有值的，只有用一个具体的命题取代时才能有值，这一取代叫命题变量的指派。上面的例子中 A 是命题常量，其值为“月亮是行星”。若规定 P 是命题变量，当用 A 取代 P 时， P 就有值，即被指派了一个值。

2. 联结词

把命题联结起来组成复合命题的符号称为联结词。表 1-1-1 给出了五个常用联结词的定义。表 1-1-1 是命题的真值表。

表 1-1-1 联结词的定义表

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

其中各符号的含义是：

“ \neg ”叫否定。 $\neg P$ 是一个复合命题，其值总是与 P 的值相反（读作“非 P ”）。

“ \wedge ”叫合取。 $P \wedge Q$ 是一个复合命题，当 P 与 Q 同时为真时它才是真（读作“ P 与 Q ”，或“ P 同时 Q ”）。

“ \vee ”叫析取。 $P \vee Q$ 是一个复合命题，当 P 或 Q 之一或者两者均为真时它是真(读作“ P 或 Q ”)。

“ \rightarrow ”叫条件。 $P \rightarrow Q$ 是一个复合命题，只有 P 为真而 Q 为假时 $P \rightarrow Q$ 才为假(读作“如果 P 则 Q ”)。我们可以把 P 理解为 Q 的充分条件，因为在 $P \rightarrow Q$ 为真的前提下，只要 P 为真， Q 一定为真。

“ \Leftrightarrow ”叫双条件。 $P \Leftrightarrow Q$ 是一个复合命题，只有 P 和 Q 取同一值时 $P \Leftrightarrow Q$ 才为真(读作“ P 当且仅当 Q ”)。我们可以把 Q 理解为 P 的充分必要条件。

3. 合式公式

用联结词和圆括号把命题变量或常量联结起来而形成的式子称为合式公式。圆括号和联结词在式中是有优先级的，按高到低的次序是： $()$ ， \neg ， \wedge ， \vee ， \rightarrow ， \Leftrightarrow 。

合式公式不是命题，只有当变量成为确定的命题时它才变为命题。变量的一组取值叫做公式的一个真值指派。可以用真值表来确定公式的一个真值指派所对应的真值。若对一个公式的任一真值指派，它的真值均为真(T)，则该公式称为永真式(或重言式)；反之称为永假式(或不可满足式)。

4. 蕴含和等价

类似于布尔代数的基本定律，数理逻辑中有命题定律，这些定律被分成两种类型：蕴含式和等价式。

蕴含 设 A 与 B 是合式公式，当且仅当 $A \rightarrow B$ 是永真式时，称 A 永真蕴含 B ，简称 A 蕴含 B ，记作 $A \Rightarrow B$ 。此时“ $A \rightarrow B$ 为真”这一前提是肯定的，所以只要 A 为真，则 B 一定为真，即 A 真意味着 B 真。

等价 设 A 与 B 是合式公式，若对两者的所有真值指派均获得相同值，则称 A 与 B 等价，记作 $A \Leftrightarrow B$ 。当 $A \Leftrightarrow B$ 为永真式时，必有 $A \Rightarrow B$ ，也只有 $A \Leftrightarrow B$ 为永真时，才有 $A \Leftrightarrow B$ 。

5. 谓词

已经指出,命题是陈述句。陈述句中总有主语和谓语之分。主语可看成是谓语描述的对象,这种对象均可被看成为单个的客体,称为个体。个体可以是任何物体或事物。谓词描述了个体的性质和个体之间的关系。直观地看,谓词就是谓语。比如:在“1是整数”这个命题中,“1”是个体,“是整数”是谓词;在“ x 比 y 小2”这个命题中, x 和 y 是两个个体,“…比…小2”是谓词。其中后一个命题中有两个个体,则它的谓词称为二元谓词。个体数为谓词的元数。元数在两个以上的谓词称为多元谓词。

类似于命题变量,在谓词逻辑中有个体变量的概念。个体变量不表示具体的个体,仅表示个体的变化范围,这个变化范围被称为个体域(或论域),它可以是有限的,也可以是无限的。

通常用大写英文字母表示谓词,用小写英文字母表示个体。如用 $P(x)$ 表示“ x 是人”,其中 $P(\quad)$ 是谓词, x 是个体。又如用 $L(x, y)$ 表示“ x 比 y 小”,其中 $L(\quad)$ 是二元谓词, x, y 为两个个体。如果一个谓词中的个体用个体变量代替,这个谓词也无具体含义,则称之为谓词变量命名式。显然,谓词变量命名式还不是命题。只有将其中的谓词赋予确定的含义,用具体的个体代替个体变量时,该谓词才变成一个确定的命题。如果,仅给谓词赋予确定含义,仍保留个体变量,则该谓词仍不是命题,而称为谓词常量命名式。如“ x 在 y 和 z 之间”,其中 x, y, z 表示任何实数,这个谓词就是谓词常量命名式。

对于用谓词刻划的命题,前述的命题联结词全部适用。

6. 量词

设有两个谓词

$$P(x):x^2-1=(x+1)(x-1)$$

$$Q(x):x-2=1.5$$

其中，谓词 $P(x)$ 中的个体 x 对于整个有理数均使 $P(x)$ 为真，而 $Q(x)$ 中的 x 仅对于 $x=3.5$ 才使 $Q(x)$ 为真。这就是说，可以在同样的个体域上对不同谓词所描述的个体的取值范围加上限制。为此引入量词的概念：全称量词和存在量词。

全称量词 当用谓词来刻划命题时，如果对于(个体域中的)所有的个体 x ，谓词 $P(x)$ 均为真，则称“所有的个体 x ”为**全称量词**，记为 $\forall x$ (\forall 为英文字 All 中的 A 的倒写)。而 $P(x)$ 对应的命题记为 $(\forall x)P(x)$ 。谓词 $P(x)$ 称为 $(\forall x)$ 的辖域或作用范围。

存在量词 当用谓词来刻划命题时，如果(在个体域中)存在某些个体 x 使谓词 $Q(x)$ 为真，则称“存在某些个体 x ”为**存在量词**，记为 $\exists x$ (\exists 为英文字 Exist 中的 E 的反写)。而 $Q(x)$ 对应的命题记为 $(\exists x)Q(x)$ 。谓词 $Q(x)$ 称为 $(\exists x)$ 的辖域或存在范围。

当某个量词作用于一个有确定个体域的一元谓词常量命名式，从而使其转化为一个命题时，这个“作用”称为**量化**。

注意：

(1) 量词本身不是一个独立的逻辑概念，可以用联结词 \wedge, \vee 代替之。比如：

$$(\forall x)A(x) \iff A(a_1) \wedge A(a_2) \wedge \dots \wedge A(a_n)$$

$$(\exists x)A(x) \iff A(a_1) \vee A(a_2) \vee \dots \vee A(a_n)$$

其中， x 的个体域为 $S = \{a_1, a_2, \dots, a_n\}$ 。

(2) 被量化的谓词的真值与个体域有关。如，对于命题 $Q(x)$ ： $\frac{1}{x} = \frac{x}{x^2}$ ， $(\exists x)Q(x)$ 在个体域为自然数时为真，在个体域为整数时则为假。

对于二元谓词 $P(x, y)$ ，可以联合使用 $\forall x, \exists y$ 和 $\forall y, \exists x$ ，但要注意次序。可能有的量化为：

$$(\forall x)(\forall y)P(x, y) \quad (\forall x)(\exists y)P(x, y)$$

$$\begin{aligned} (\exists x)(\forall y)P(x,y) & \quad (\exists x)(\exists y)P(x,y) \\ (\forall y)(\forall x)P(x,y) & \quad (\forall y)(\exists x)P(x,y) \\ (\exists y)(\forall x)P(x,y) & \quad (\exists y)(\exists x)P(x,y) \end{aligned}$$

其中, $(\exists x)(\forall y)P(x,y)$ 代表 $(\exists x)((\forall y)P(x,y))$, 而 $(\forall y)(\exists x)P(x,y)$ 代表 $(\forall y)((\exists x)P(x,y))$ 。

1.1.3 集合的表示

1. 列举成员法

(1) 列举所有成员,并用花括号括起来。如:

$\{a, b, c\}$ 表示由三个元素组成的集合。

(2) 列举部分成员,而其它成员可以一目了然地类推。如:

$\{1, 2, 3, \dots\}$ 表示自然数的集合。

2. 谓词法

对于任意集合 A 可有如下表示:

$$A = \{x | P(x)\} \text{ 或 } A = \{x : P(x)\}$$

其中, x 是元素的抽象表示, $P(x)$ 是元素的性质。如:

$$B = \{x : x = 2^n, n = 1, 2, 3, \dots\}$$

表示了以 2, 4, 8, ... 等数组成的集合。又比如:

$$C = \{\text{学生} | \text{年龄在 20 岁以下的大学生}\}$$

其中,谓词直接表示成命题形式。

3. 属于记号

集合是由元素组成的,反过来说,元素是属于集合的,用记号 \in 表示属于,用 \notin 表示不属于。比如: $8 \in B, \langle \text{王某} \rangle \in C, 3 \notin B$ 。

1.1.4 集合的包含和相等

1. 集合的包含

给定 A 和 B 两个集合。如果 A 集合的每一个元素都是 B 的元