计算机软件基础

孟彩霞　主编
孟彩霞　王曙燕　陈莉君　编著

机械工业出版社

计算机软件基础

计算机软件基础

孟彩霞　主编
孟彩霞　王曙燕　陈莉君　编著

机械工业出版社

本书较全面地介绍了计算机软件基础的几个主要部分。全书分11章，介绍了数据结构及部分典型算法，操作系统的基本原理，关系数据库的基本理论，并详细讨论了如何应用FoxBASE+进行数据库设计，以及软件工程的基本原理、方法。全书内容紧凑、深入浅出、注重实用。

本书可作为高等院校作计算机专业计算机软件基础课程的教材，也可供工程技术人员作为提高软件水平的参考书，成人教育和职业培训亦可用它作教材。

图书在版编目（CIP）数据

计算机软件基础/孟彩霞主编，王曙燕，陈莉君编著.--北京：机械工业出版社，1997.11

ISBN 7-111-06031-8

Ⅰ.计… Ⅱ.①孟…②王…③陈… Ⅲ.软件-基础理论 Ⅳ.TP31

中国版本图书馆CIP数据核字（97）第23229号

出 版 人：马九荣（北京市百万庄南街1号 邮政编码100037）
责任编辑：何文军 版式设计：霍永明 责任校对：李秋荣
封面设计：姚 毅 责任印制：路 琳
机械工业出版社印刷厂印刷·新华书店北京发行所发行
1997年12月第1版第1次印刷
787mm×1092mm 1/16·15.75印张·384千字
0 001—3 000册
定价：25.00元

前　言

在普通高等院校中对学生的计算机基础知识与应用能力的培养已成为各学科各专业教学计划的重要组成部分，高等院校本、专科毕业生的计算机基础知识与应用能力的水平也已成为绝大多数用人单位选择录用人员的重要依据之一。对于大学各类专业来说，计算机软件应用与开发技术显得越来越重要和必不可少了。为此，教委已经开始在高等院校中推行非计算机专业分层次计算机教学。《计算机软件基础》一书正是为满足这种教学需求而编写的。本书的目的是培养非计算机专业学生的软件开发基本能力。考虑到学时和读者的特点，不追求计算机学科体系完整，尽量做到深入浅出，讲求实用。

全书共分 4 大部分 11 章：

第 1 章　概述　对计算机软件的发展过程、基本内容作一概述。

第 1 部分　数据结构（2～4 章）　内容包括线性表、栈、队列、数组、二叉树、图等概念及有关算法，以及查找和排序的方法及相应算法。

第 2 部分　操作系统（5～6 章）　内容包括操作系统的发展，操作系统的基本原理以及常用的几种操作系统。

第 3 部分　数据库系统（7～10 章）　内容包括关系数据库的原理，数据库设计理论，数据库应用系统的设计以及 FoxBASE⁺ 系统。

第 4 部分　软件工程（11 章）　内容包括软件工程的形成以及软件的生存周期。

每一章后面都附有一定数量的习题。

本书主要按照对该课程的教学基本要求而编写，它是从事相关课程的一些教师长期教学经验的总结。本书由孟彩霞主编。其中第 2、3、4 章由王曙燕编写；第 5、6 章以及第 1 章的部分内容由陈莉君编写；第 7、8、9、10、11 章以及第 1 章的部分内容由孟彩霞编写。在编写本书的过程中得到了南京邮电学院计算机系唐家益教授的大力支持，机械工业出版社为本书出版做了大量工作，使本书能尽早与读者见面，在此一并表示感谢。

由于作者水平有限，书中有不当之处，恳请读者批评指正。

编　者

1996. 12

目　录

第1章 概 述

一个完整的计算机系统由硬件和软件两个部分组成。硬件是机器的可见部分，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，软件也逐步丰富与完善，而软件的发展又大大促进了硬件的进展。

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。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。计算机软件技术发展到今天，已经形成了一个完整的技术基础学科体系。

1.1 计算机软件

1.1.1 计算机软件定义

在飞速发展的计算机产业中，计算机软件所承担的角色越来越重要，“软件”这一词汇在不同的场合其含义可能不尽相同。习惯上，人们认为软件就是程序或程序就是软件。随着计算机的发展及软件规模愈来愈大，人们发现程序和软件是两个不同的概念，于是有人提出这样一种观点：软件是由程序和开发它、使用它、维护它所需要的一切文档组成。这一观点强调了文档在软件研制中的重要性。1983年，IEEE组织明确地给软件作了定义：软件是计算机程序、方法、规则相关的文档以及在计算机上运行它时所必须的数据。

计算机软件发展非常迅速，其内容又十分丰富，对它进行分类也比较困难，仅从用途来划分，大致分为3种：

1. 服务类软件

这类软件是面向用户的，为用户服务的，包括各种语言的集成化软件如Turbo C软件、Windows下的Borland C++软件；各种软件开发工具及常用的库函数等等。

2. 维护类软件

此类软件是面向计算机维护的。包括错误诊断和检测软件、测试软件、各种调试用软件如Debug等等。

3. 操作管理类软件

此类软件是面向计算机操作和管理的，包括各种操作系统、网络通信系统、计算机管理软件等等。

若从计算机系统角度看，软件又分为系统软件和应用软件。

系统软件是指为管理、控制和维护计算机及外设，以及提供计算机与用户界面等的软件。如操作系统、数据库管理系统、各种语言编译系统及编辑软件等。

系统软件以外的其它软件称为应用软件。目前应用软件的种类很多，按其主要用途可分

为5类：科学计算、数据处理、实时控制、辅助设计和人工智能软件。应用软件的组合可称为软件包或软件库。数据库及数据库管理系统过去一般认为是应用软件，随着计算机的发展，现在已被认为是系统软件。随着计算机技术的不断发展，应用领域不断拓宽，应用软件种类将日益增多，在软件中所占比重越来越大，如今已是市场上的主要软件。

1.1.2 计算机软件开发环境

学习软件基础知识（包括数据结构、操作系统、数据库及软件工程），其主要目的是为了进行应用软件的开发。搞好软件开发，除了要掌握先进的开发技术外，还要求有良好的软件开发环境。

在软件开发环境中，用户界面占有重要的地位。近十多年来开发的应用软件，多数开发者都十分注意用户界面的设计。其中“多窗口”、“菜单”与“联机帮助”被称为用户界面的三大友好技术。

随着计算机的普及与性能的提高，人们越来越重视用户界面的改善。在80年代，图形用户界面（graphical user interface；简称GUI）取得了重要的进展。Microsoft公司的Windows，麻省理工学院DEC公司开发的X-windows，精彩纷呈。非键盘输入工具鼠标器也随之得到广泛的使用。与此同时，包括文字、图形、声音、图象等多媒体（multi-media）用户界面也应运而生，受到广泛的注意。

操作系统是开发环境的重要基础。它不仅通过对其它系统软件和一切服务软件的支持给开发环境提供各种有用的开发工具，还以数以百计的键盘命令与系统调用，向用户直接提供功能强大的服务。比较著名的操作系统如DOS、UNIX及Windows 95已经向我们展示了现代操作系统丰富多彩的用户界面。

在软件开发中，无论技术活动还是管理活动，都离不开环境的支持。近十几年来，各技术先进国家大力发展软件环境的研究，一批实用的环境应运而生。Case（Computer Aided Software Engineering）环境和工具，几乎成为一切现代化软件开发环境总称的趋势。

1.1.3 面向对象的软件开发方法

面向对象（Object-oriented简称OO）方法是当代计算机科学领域，特别是软件领域的发展主流。面向对象方法起源于70年代，在80年代出现了一大批面向对象的编程语言，标志着OO方法在编程领域走向成熟和实用。但是OO方法的作用和意义决不只局限于编程技术，它是一种新的程序设计范型，是一种具有深刻哲学内涵的认识方法学和系统构造理论。面向对象方法的主要特点和优势表现在以下几点：

1）强调从现实世界中客观事物（对象）出发来认识问题领域和构造系统，大大减少了系统开发者对问题域的理解难度，使系统能准确地反映问题域。

2）运用人类日常的思维方法和原则（体现于OO方法的抽象、分类、继承、封装、消息通信等基本原则）进行系统开发，有益于发挥人类的思维能力，并有效地控制了系统的复杂性。

3）对象的概念贯穿于开发过程的始终，使各个开发阶段成分具有良好的对应，从而显著地提高了系统的开发效率与质量，并大大降低了系统维护的难度。

4）对象的相对稳定性和对易变因素的隔离，增强了系统的应变能力。

5）对象类之间的继承性关系和对象的独立性，对软件复用提供了强有力的支持。

正是由于上述特点，使面向对象方法在计算机领域产生了巨大影响。近十年来，它的影

响渗透到计算机科学技术的几乎每一个分支领域，如编程语言、系统分析与设计、数据库、人机界面、知识工程、操作系统、计算机体系结构等等。所以80年代后期人们预言，面向对象方法将成为90年代计算机领域，特别是软件领域的主流技术，现在这一预言已成为现实。

1.2 数据结构概述

1.2.1 数据和数据结构

现代数字计算机原是作为能快速地进行复杂、耗时计算的工具而发明的。随着计算机的发展，在计算机的绝大多数应用中，能够存取、处理大量信息的能力却被认为是计算机的首要特征，而它的计算能力在许多情况下已经几乎被人们忽略了。有位美国学者曾批评说："计算机"这个词只告诉我们以前能做的事，却未道出它的潜能。有鉴于此，人们常常把计算机称作数据处理机。

什么是数据？数据就是计算机可以保存和处理的信息。可以看出，数据这个概念本身是随着计算机的发展而不断扩展的概念。在计算机发展的初期由于计算机主要用于数值计算，数据指的就是数值。计算机硬件和软件技术的不断发展，扩大了计算机的应用领域，诸如文字、表格、图形、图象、声音等也属于数据的范畴。目前非数值问题的处理占用90%以上的计算机时间。

组成数据的基本单位是数据元素。例如：全部学生的学籍登记卡组成学生的学籍数据，每个学生的学籍登记卡就是学籍数据的一个数据元素。数据元素可以是一个数或字符串，也可以是由若干数据项组成。在这种情况下，通常把数据元素称为记录。如表1-2-1所示的学生学籍登记卡，在这个表中每一个学生的学籍登记卡作为一个数据元素，每一个元素由学号、姓名、性别、民族、籍贯、专业六个数据项组成。

表1-2-1 学生学籍登记卡

学　号	姓　名	性　别	民　族	籍　贯	专　业
1	王　安	男	汉	陕西	计算机通信
2	李　华	男	回	宁夏	软件
3	张　莉	女	汉	山西	计算机应用
4	赵　平	女	汉	广东	软件
…	…	…	…	…	…

什么是数据结构？在任何问题中，构成数据的数据元素并不是孤立存在的，它们之间存在着一定的关系以表达不同的事物及事物之间的联系。所以简单地说数据结构就是研究数据及数据元素之间关系的一门学科。它不仅是一般程序设计的基础，而且是设计和实现编译程序、操作系统、数据库系统及其它系统程序和大型应用程序的重要基础。它包括三个方面的内容：

- 数据的逻辑结构
- 数据的存储结构
- 数据的运算

1. 数据的逻辑结构

数据的逻辑结构就是数据元素之间的逻辑关系。可以用一个二元组，给出其形式定义为

$$\text{Data-Structure} = (D, R)$$

其中：D 是组成数据的数据元素的有限集合，R 是数据元素之间的关系集合。

根据数据元素之间关系的不同特性，数据结构又可分为两大类：线性数据结构和非线性数据结构。按照这种划分原则，本书要介绍的所有数据结构大致归结如下：

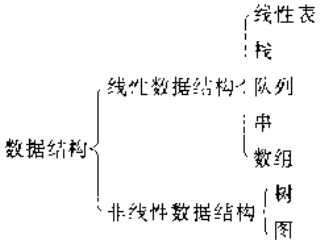

图 1-2-1　数据结构分类

2. 数据的存储结构

数据的逻辑结构是从逻辑上来描述数据元素之间的关系的，是独立于计算机的。然而讨论数据结构的目的是为了在计算机中实现对它的处理。因此还需要研究数据元素和数据元素之间的关系如何在计算机中表示，这就是数据的存储结构。

大家知道，计算机的存储器是由很多存储单元组成的，每个存储单元有唯一的地址。数据的存贮结构要讨论的就是数据结构在计算机存贮器上的存贮映象方法。

一般来说，数据在存储器中的存储有四种基本的映象方法。

(1) 顺序存储结构　就是把数据元素按某种顺序放在一块连续的存储单元中，其特点是借助数据元素在存储器中的相对位置来表示数据元素之间的关系。

(2) 链式存储结构　有时往往存在这样的情况：存储器中没有足够大的连续可用空间，只有不相邻的零碎小块存储单元。另一种情况是在事先申请一段连续空间时，因无法预计所需存储空间的大小，需要临时增加空间。所有这些情况，使得得到一块合适的连续存储单元并非易事，即这种情况下顺序存储结构无法实现。

链式存储结构的特点就是将存放每个数据元素的结点分为两部分：一部分存放数据元素(称为数据域)；另一部分存放指示存储地址的指针（称为指针域），借助指针表示数据元素之间的关系。

结点的结构如下：

数据域	指针域

链式存储结构可用一组任意的存储单元来存储数据元素，这些存储单元可以是连续的，也可以是不连续的。

(3) 索引存储结构　在线性表中，数据元素可以排成一个序列：R_1、R_2、R_3，…，R_n，每个数据元素 R_i 在序列里都有对应的位置数据 i，这就是元素的索引。索引存储结构就是通过数据元素的索引号 i 来确定数据元素 R_i 的存储地址。一般索引存储结构的实现方法：建立附加的索引表，索引表里第 i 项的值就是第 i 个元素的存储地址。

(4) 散列存储结构　这种存储方法就是在数据元素与其在存储器上的存储位置之间建立一个映象关系 F。根据这个映象关系 F，已知某数据元素就可以得到它的存储地址。即 D=F(E)，这里 E 是要存放的数据元素，D 是该数据元素的存储位置。可见，这种存储结构的关键是设计这个函数 F，但函数 F 不可能解决数据存储中所有问题，还应有一套意外事件的处理方

法、它们共同实现数据的散列存储结构。

3. 数据的运算

数据的运算是定义在数据逻辑结构上的操作，如检索、插入、删除等。每种数据结构都有一个运算的集合。

1.2.2 算法的描述及评价

算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。有时，把运算的实现称之为算法。我们知道，运算是定义在逻辑结构上的操作，是独立于计算机的，而运算的具体实现则是在计算机上进行的，因此算法要依赖于数据的存储结构。

作为算法应具有下述的五个重要特性：

(1) 有穷性　一个算法必须是在执行有穷步后结束，且每一步都能在有限的时间内完成。

(2) 确定性　算法中每一条指令必须有确切的含义，读者理解时不会产生二义性。并且，在任何条件下，算法只有唯一的一条执行路径，即对于相同的输入只能得出相同的输出。

(3) 可行性　一个算法是能行的，即算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现。

(4) 输入　一个算法应该有零个或多个输入。

(5) 输出　一个算法应该有一个或多个输出。

1. 算法的描述

算法需要用一种语言来描述，同时，算法可有各种描述方法以满足不同的需求，常用的有：自然语言描述、流程图描述、类 PASCAL 语言描述等。本书中使用类 PASCAL 语言作为描述算法的工具。类 PASCAL 语言绝大多数语句都是 PASCAL 语言中的相应语句，修改并增加了个别的语句。表 1-2-2 中给出了类 PASCAL 语言语句一览表。

2. 算法的评价

在算法是“正确”前提下，评价一个算法主要有两个指标：

时间复杂度：依据算法编制成程序后，在计算机上运行时所消耗的时间。

空间复杂度：依据算法编制成程序后，在计算机执行过程中所需要的最大存储空间。

要确定实现算法在运行时所花的时间和所占用的存储空间，最直接的方法就是测试，即将依据算法编制的程序在计算机上运行，所得到的结果就是算法运行时所花的时间。这种方法有时也称为事后统计的方法。不过，可以想到同一算法在不同档次的计算机上运行所花的时间肯定不同，这取决于计算机系统的速度。

另外一种方法就是先验分析估算的方法，这是人们常常采用的一种方法，下面将详细讨论之。

(1) 时间复杂度　假定知道算法中每一条语句执行一次所花的平均时间，则有：

算法运行所花的时间＝语句执行一次所花的平均时间×语句执行次数

其中语句执行一次所花的平均时间取决于计算机系统中硬件、软件等环境因素，而一个算法中语句的执行次数一般来说是确定的。因此，对于先验分析方法我们讨论的目标集中在确定语句的执行频度上，即把算法的语句执行频度作为衡量一个算法时间复杂度的依据。

实际分析中，关注的是频度的数量级，即按重复执行次数最多的语句确定算法的时间复杂度。引入符号“○”来表示这种数量级。算法的时间复杂度记作 $T(n) = ○(f(n))$

表 1-2-2 类 Pascal 语言语句一览表

种类	名称	表示形式	备注
算法形式	过程	PROC 算法名（〈参数表〉） BEGIN 〈语句组〉 ENDP；{过程名}	a. 除参量表外，所有过程中出现的局部变量都不加类型说明 b. 假设类 Pascal 语言允许一切可能出现的变量类型
	函数	FUNC 函数名（〈参数表〉）：〈类型〉； BEGIN 〈语句组〉 ENDF；{函数名}	
简单语句	赋值语句	〈变量〉：＝〈表达式〉	复合语句，类 Pascal 语言中用“〔”和“〕”替代 BEGIN 和 END
	复合语句	[<语句组>]	
	输入语句	READ（〈变量表〉）	
	输出语句	WRITE（〈表达式表〉）	
分支语句	条件语句	IF 〈条件〉 THEN 〈语句 1〉 ELSE 〈语句 2〉	
	分情形语句	CASE〈变量名〉OF 〈常量表 1〉：〈语句 1〉； 〈常量表 2〉：〈语句 2〉； … ENDCASE	
重复语句	FOR 语句	FOR I：＝〈初值〉 TO 〈终值〉 DO 〈语句〉； FOR I：＝〈初值〉 DOWNTO〈终值〉 DO 〈语名〉；	
	WHILE 语句	WHILE〈条件〉DO 〈语句〉；	
	REPEAT 语句	REPEAT 〈语句〉 UNTIL〈条件〉；	
其它	过程调用	CALL 过程名（参量表）	
	函数调用	变量名：＝函数名（参量表）	
	函数值返回	RETURN（表达式）	
	出错处理	ERROR（‘文字’）	
	基本函数	可以使用 Pascal 语言中的标准函数及一些数学符号来描述一些简单操作；	

它表示随问题规模 n 的增大算法执行时间的增长率和函数 f（n）的增长率相同，称作算法的渐近时间复杂度，简称时间复杂度。

按数量级递增次序排列，常见的几种时间复杂度有：

$O(1)$，$O(\log n)$，$O(n)$，$O(n\log n)$，$O(n^2)$，$O(n^3)$，$O(2^n)$，这里，n 表示问题的规律。

例如：在下列三个程序段中，

(a) x：=x+1；

(b) FOR i：=1 To n DO x：=x+1；

(c) FOR j：=1 To n DO

FOR k：=1 To n DO x：=x+1；

含基本操作“x 增 1”的语句 x：=x+1 的频度分别为 1，n 和 n^2，则这三个程序段的时间复杂度分别为○ (1)，○ (n) 和○ (n^2)，分别称为常量阶、线性阶和平方阶。

需要指出的是：有些算法的基本操作的频度不仅仅依赖于问题的规模，还取决于它所处理的输入数据集的状态。对于这一类算法，一般按每种情况发生的概率求出其数学期望值作为算法的平均时间复杂度，或者按最坏情况下基本操作的执行频度得出算法最坏情况下的时间复杂度，以此作为该算法的时间复杂度。

(2) 空间复杂度　一个算法的实现所占用的存储空间大致有这样三个方面：其一是指令、常数、变量所占用的存储空间；其二是输入数据所占用的存储空间；其三是算法执行时必需的辅助空间。前两个空间是计算机运行时所必须的。因此，把算法在执行时所需的辅助空间的大小作为分析算法空间复杂度的依据。

与算法时间复杂度的表示一致，也用辅助空间大小的数量级来表示算法的空间复杂度，仍然记为：○(x)。常见的几种空间复杂度有：○(1)，○(n)，○(n^2)，○(n^3) 等。

事实上，一个问题的算法实现，时间复杂度和空间复杂度往往是相互矛盾的，要降低算法的执行时间就要以使用更多的空间为代价，要节省空间就可能要以增加算法的执行时间作代价，很难两者兼顾。因此，只能根据具体情况有所侧重。

1.3　数据库系统概述

1.3.1　信息、数据和数据处理

人类的社会活动和生产活动，离不开对信息的收集、保存、利用和处理，特别是当今生产力突飞猛进，新技术层出不穷，信息量迅速剧增，人类社会进入了信息化的阶段。那么什么是信息呢？信息是人们用以对客观世界直接进行描述的、可以在人们之间进行传递的一些知识。信息需要被加工和处理、需要被交流和使用。随着计算机技术的迅速发展，计算机具有的高速处理能力和存储容量巨大的特点，使得人们有可能对大量的信息进行保存和加工处理。为了记载信息，人们使用各种各样的物理符号和它们的组合来表示信息，这些符号及其组合就是数据。数据是信息的具体表示形式，信息是数据的有意义的表现。由此可见，信息和数据有一定的区别，信息是观念性的，数据是物理性的。在有些场合信息和数据难以区分，信息本身就是数据化了的，数据本身是一种信息。因而在很多场合不对它们进行区分，信息处理与数据处理往往指同一个概念，计算机之间交换数据也可以说成是交换信息等等。

有了数据就产生了数据处理的问题，人们收集到的各种数据需要经过加工处理。所谓数据处理包括对数据的收集、记载、分类、排序、存储和计算等工作。其目的是使有效的信息资源得到合理和充分的利用，从而促进社会生产力的发展。

数据处理经过了手工处理、机械处理、电子数据处理三个阶段。今天，用电子计算机进行数据处理方法的研究已成为电子计算机科学技术中的主要课题之一，数据库技术已成为社会信息化时代不可缺少的方法和工具。

1.3.2 数据管理技术的发展

随着计算机数据处理的发展，数据管理技术先后经历了三个发展阶段，即人工管理阶段文件系统管理阶段和数据库系统管理阶段。

1. 人工管理阶段

这是计算机用于数据管理的初级阶段，计算机本身只相当于一个计算工具。对数据的管理是由程序员个人考虑和安排的，一个程序对应于一组数据，进行程序设计时，往往也要对数据的结构、存储方式、输入输出方式等进行设计。严格地说，这种管理只是一种技巧，这是数据自由管理的方式，因此，这一阶段又称为自由管理阶段。其特点是：数据不能长期保存，数据与程序不独立，一组数据对应于一个程序，没有软件系统对数据进行管理，基本上没有文件的概念。程序与数据的存放形式如图 1-3-1 所示。

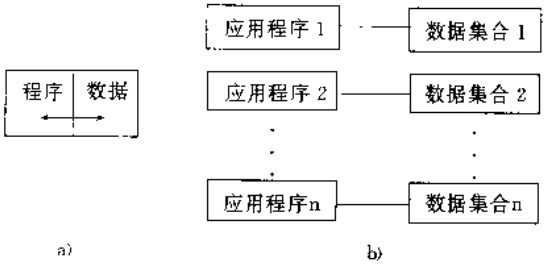

图 1-3-1　程序与数据的存放

2. 文件系统管理阶段

当计算机的操作系统包含有文件系统后，把数据组织成文件的形式可以随机进行查询、增删改等处理，就使得计算机数据管理方法得到了极大的改善。数据以文件形式长期保存于计算机外存储器里，可以离开处理它的程序而独立存在，从而实现了以文件为单位的数据共享。

所有文件由一个称做文件管理系统的专用软件对其进行管理和维护。文件管理系统是应用程序和数据文件之间的一个接口。应用程序必须通过文件管理系统才能建立和存储文件，反之，应用程序也只有在文件管理系统的支持下才能检索数据文件中的数据。文件系统工作方式如图 1-3-2 所示。

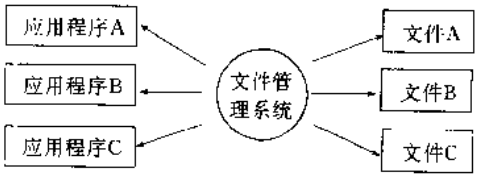

图 1-3-2　文件管理系统示意图

3. 数据库系统管理阶段

上述数据管理方式存在着一系列缺点：

(1) 数据依赖于程序，增加了程序的维护工作；

(2) 数据文件只对应一个或几个应用程序，数据和文件之间相互依赖；

(3) 数据冗余度大，用户各自建立文件，相互不能共享，数据重复，工作重复；

(4) 缺乏统一的管理机构，安全性、完整性差。

自 70 年代开始，数据库技术得到了迅速的发展和广泛的应用，尤其是使用了大容量的外存储器以后，开始出现了数据库管理系统 DBMS（Data Base Management System），以实现大量数据的集中存储和数据资源的高度共享。

数据库管理系统对数据的处理方式与文件管理系统不同。数据库管理系统把所有应用程序中所使用的数据汇集在一起，并以记录为单位将其存储起来，以便于应用程序的查询。在数据库系统中，应用程序与数据之间的关系可用图 1-3-3 来表示。

数据库系统管理方式的基本特征是采用了逻辑的数据模型，具有完整的数据结构，数据

之间存在着内在联系，可以最大限度地避免数据的重复；它把文件、数据的定义，从应用程序中独立出来，使得程序独立于数据而存在，所有的应用程序都可以随意取用数据库中的任何数据，从而实现多个用户多种语言对数据的共享。数据库系统的特点概括起来有：数据的结构化、数据的独立性、可控冗余度和数据共享。

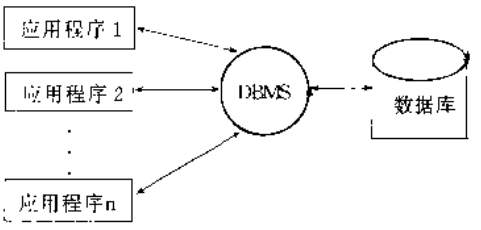

图 1-3-3　数据库管理系统示意图

1.3.3　数据库系统的组成与体系结构

1. 数据库系统的组成

数据库是数据处理的最新技术，那么什么是数据库呢？数据库是存储在计算机内的有结构的数据的集合。形象地说，数据库是一个存放数据的基地。数据库系统是指计算机系统中引进了数据库后的系统。因此，数据库系统是由计算机系统、数据库及其描述机构、数据库管理系统和有关人员组成，是由这几个方面组成的具有高度组织性的总体。带有数据库的计算机系统如图 1-3-4。

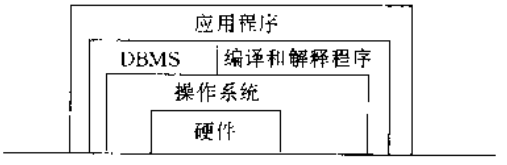

图 1-3-4　数据库系统基本构成

（1）计算机系统　计算机系统指的是用于数据库管理的计算机硬件和软件资源。硬件资源包括中央处理器、内存储器、外存储器及其它外部设备。软件资源包括操作系统、数据库管理系统及应用程序。

（2）数据库及其描述　数据库有存放实际数据的物理数据库和存放数据的逻辑结构的描述数据库。数据库的数据是通过模型来描述的，即通过外模式、概念模式（简称模式）和内模式三级进行描述。

（3）数据库管理系统　数据库管理系统 DBMS（Data Base Management System）是一个帮助用户建立、使用和管理数据库的软件系统，是数据库与用户之间的接口。主要功能是维持数据库系统的正常运行，接收并回答用户提出 的访问数据库的各种应用要求，如检索、存储数据等。

（4）数据库系统的有关人员

①　数据库管理员 DBA（Data Base Administrator）

负责对整个数据库系统进行总体控制和维护，以保证数据库系统的正常运行。

②　应用程序员　负责编写和维护应用程序，这些应用程序能对数据库进行通常的操作，如检索、建立、删除或改变信息等操作。

（3）终端用户　他们通过联机终端，使用查询语言或者运行应用程序对数据库进行检索、插入、删除或更新等操作。

2. 数据库系统的体系结构

数据库系统的体系结构分为三级：外模式、概念模式和内模式，如图 1-3-5 所示。

（1）外模式　外模式是用户与数据库的接口，是应用程序可见到的数据描述，不同用户看到并获准使用的数据是不同的，对每个用户所涉及到的那部分数据结构，用子模式 DDL（Data Definition Language）描述，称之为子模式。

(2) 概念模式　概念模式是指数据库管理员(DBA)所看到的数据库，其结构用模式DDL描述。它对应着数据库的数据模型，是对数据库整体逻辑的描述。

(3) 内模式　内模式又称存储模式，它描述了数据的物理结构（包括数据项、记录和数据集在内的数据库的全部存储数据），描述了数据在存储器上的实际安排与存取方式。

三级模式之间有两个映射，以实现两者的对应的转换，它们由软件完成，这些软件是DBMS的主要组成部分之一。

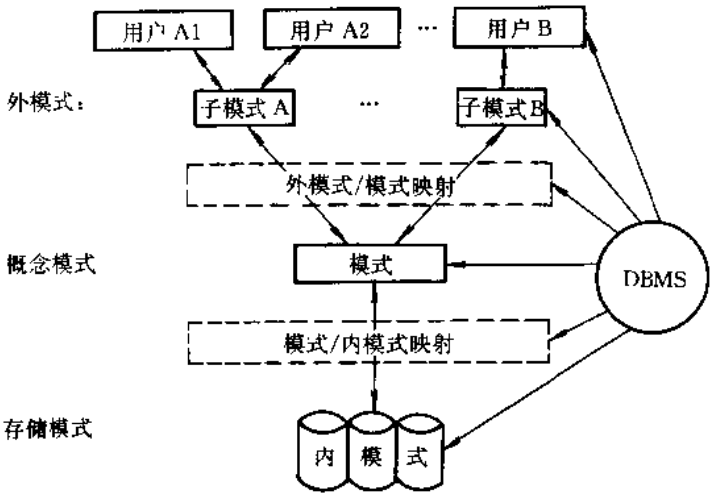

图 1-3-5　数据库系统的分级结构

1.3.4　数据库管理系统

现在，将结合DBMS的组成进一步介绍它的功能。

DBMS通常有三个组成部分，它们是：

(1) 数据描述语言及其翻译程序；

(2) 数据操作语言及其翻译程序；

(3) 管理和控制例行程序。

以下分别对它们作一简单介绍。

1. 数据描述语言 (Data Description Language)

这是在建立数据库时用来描述数据库结构的语言。也有些文献称之为数据定义语言(Data Definition Language)，简称都是DDL。

按照数据库的模式分级，DDL又可细分为：

(1) 模式DDL　用来定义数据库的整体（或全局）逻辑结构；

(2) 子模式DDL（或SDDL）　用来定义子模式（或局部）的逻辑结构；

(3) 物理模式DDL（或PDDL）　用来描述物理数据库的结构。

用三种DDL来描述三种不同的模式，有利于实现数据独立性。但实际的DBMS不一定将三种DDL分开，多数DBMS仅提供一种或者两种DDL，同时完成这三种语言的功能。

DDL是类似高级语言的形式化语言。用DDL描述的模式称为源模式，它构成数据库系统的“描述数据库”。通过DBMS配备的DDL翻译程序，能把源模式翻译为目标模式，构成系统的“目标数据库”。目标数据库通常由一组表格构成。

在第7章7.2节，还要对DDL作更多的讨论。

2. 数据操作语言 (Data Manipulation Language)

简称DML，它是DBMS提供给用户的对数据库进行检索和存储的工具，也是用户与数据库之间的接口。

检索就是查询，在DML的各类功能中，查询具有重要的地位，所以有些系统把DML直呼为查询语言。存储操作则包括数据的插入、删除和修改等操作。

在实际的DBMS中，DML又可分为两类。一类是可以独立使用的，它具有单独的编译或

解释程序，可以在终端上用输入命令的方式立即执行，或者用 DML 语言先编成应用程序，然后把程序送入计算机执行。通常称这类 DML 为自含型（Self-Containable），如后面将要介绍的 Fox Base 语言就属于自含型的 DML。另一类称为嵌入型，它不能单独使用，而是嵌入汇编语言或者 COBOL，FORTRNA 等高级语言之中，与被嵌入的语言一起使用。嵌入型 DML 又称数据子语言（Data Sublanguage，简称 DSL），被嵌入的语言称为宿主语言（Host Language）。有些系统的 DML 兼有自含型与嵌入型的功能，允许用户选择使用。SQL 语言就是这种语言的一例。

DML 的翻译程序可采用编译方式或解释方式。FoxBASE 语言既有解释型，也有编译型。

3. 管理和控制例行程序

这部分内容较杂，可包含安全性控制、完整性控制，并发控制、通信控制、数据存取、数据修改等例行程序以及工作日志、数据库转储、数据库初始装入、数据库重新组织等公用管理程序。对于不同的 DBMS，它们的 DDL 与 DML 功能也强弱不同，但差异最大的是这一部分。微机上使用的 DBMS 一般为简易型，许多管理与控制功能都被简化或省去了。

1.3.5 数据模型

世界上的事物是彼此联系的，因此，描述客观事物的数据之间也是有联系的。数据之间的联系有两种：一种是记录内部的数据联系；另一种是记录之间的数据联系。数据模型是对客观事物及其联系的数据描述，即实体模型的数据化，是指数据在数据库中排列、组织所遵循的规则，以及对数据所能进行操作的总和。简单地说，数据模型是表示实体及实体之间联系的模型。数据模型的好坏直接影响着数据库的性能，数据模型的设计方法决定了数据库的设计方法。当前较流行的数据模型有层次模型、网络模型和关系模型，下面分别作简单的介绍。

1. 层次模型

层次模型（Hierarchical Model）是以记录类型为结点的树，即树的每一个结点都表示一个记录类型，它有如下几个特点：

（1）有且仅有一个结点向上不与任何结点联系、这个结点即为树的根，称为根结点；

（2）其它结点向下可以与若干结点联系，但向上只与唯一的一个结点联系。

凡满足上面两个条件的数据结构称为层次模型。例如学院行政组织机构如图 1-3-6 所示，就是一个层次模型。

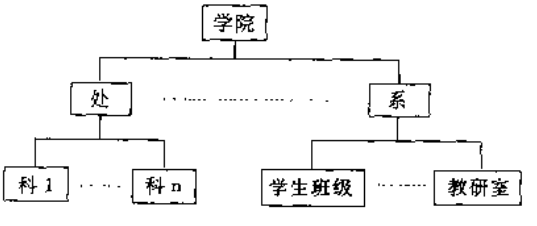

图 1-3-6 学院行政组织的层次模型

2. 网络模型

网络模型（Network Model）是以记录类型为结点的网络结构。在网络模型中，数据间紧密相连，呈现出一种网络状的关系形式。它的特征是：

（1）至少有一个以上的结点无父结点。

（2）至少有一个结点有多于一个的父结点。

（3）任何两个结点之间可以有两种以上的联系。

图 1-3-7 中的数据模型都是网络模型。网络模型与层次模型的主要区别在于：层次模型中