

高等学校试用教材

数据库设计 和管理基础

崔 巍 于志钧 编著
俞盘祥 审校



高等教育出版社

高等学校试用教材

数据库设计和管理基础

崔巍 于志钧 编著

俞盘祥 审校

高等教育出版社

内 容 提 要

本书介绍数据库的一般知识, 主要包括: 文件系统、数据库概念的形成、数据库系统的构成和原理、关系数据库及关系 DBMS Oracle、CODASYL 和 IMS 数据库概述、数据库设计及数据库环境管理、数据库远程通信及分布式数据库等内容。

本书以关系数据库为主, 特别较详细介绍了目前占主流位置的关系 DBMS Oracle, 注重理论和实践的结合, 使读者能够对关系数据库有个直观、全面的了解, 突破了近些年以 dBASE 作为教学样本的一些局限性。

全书以实例贯穿始终, 通俗易懂, 实用性强。

本书可作为高等院校信息管理、计算机应用和有关的非计算机专业的“数据库”课程的教材或教学参考书, 也可供从事计算机和信息管理的科研人员、工程技术人员、管理人员及其他有关人员阅读参考。

(京)112号

高等学校试用教材 数据库设计和管理基础

崔巍 于志钧 编著
俞盘祥 审校

高等教育出版社出版
新华书店总店北京科技发行所发行
河北省香河县印刷厂印装

开本 787×1092 1/16 印张 19.5 字数 440 000

1990年9月第1版 1991年9月第2次印刷

印数 3 521—5 031

ISBN 7-04-002907-3/ TP·67

定价 5.45 元

前 言

本书是一本如何使用数据库管理系统这样一个相当新型的工具来实现大型企业数据管理的书。也就是说,本书所涉及的不仅是计算机科学的一个分支,而且也是管理科学中相当专门的一个分支,同时也是对企业和社会机构的全面管理的十分重要的一个分支。

在实践中,管理方法千变万化,然而给管理下定义却不难。可以说,管理就是指建立和维持一个实用、有效的秩序。为方便起见,我们可以把建立一个实用有效秩序的管理过程同这一秩序的维持过程分成两部分,虽然这种划分也许是过分简单化了。据此,建立一个新工厂是管理的一个任务,而维持这个工厂的实用和高产则是管理的另一个任务。

如同工厂管理以及其它类型的工作一样,企事业数据的管理也需要建立各种实用而有效的结构。在电子计算机发明之前,人工保管帐目一直是企事业数据管理的主要手段。随着计算机的出现,产生了将成千上万的数据存放于计算机文件中的数据管理方法。这确实是个实用而有效的方法,至少在刚开始时是这样。在大企业中,这种有序数据结构的管理是由数据处理部门来完成的。为了保证数据的有效性,计算机文件中的内容必须定时更新。在一个大机构中,各类人员都需要获取信息,从而使得所要保管的计算机文件在数量和种类上都不断增加。开始时,这种用计算机文件来管理数据的方法的确很有效,但随着计算机文件数量的增加,人们便逐渐意识到,这种计算机文件的集合对于企事业的数据管理并不是十分实用和有效。事实上可以不过分地说,在许多企业中,那些曾经是有秩序的文件集合逐渐变得灾难性地混乱不堪。这一现象的原因将在第四章中加以讨论,而这里只陈述这样一个事实,即到60年代末期,某种新的数据结构产生了,并且势在必行。

这种新的数据结构就是数据库。可以说,现代数据库属于人类曾经设想过的高度有序的数据集合。数据库的管理要用到称为数据库管理系统这样一种复杂的软件产品。数据库这样一种高度有序的数据集合有一个独特的优点,就是其内在的灵活性。具体地说,尽管企业的需要是在不断地变化着,但是用先进的数据库管理系统可以相当方便地重新组织一个数据库以满足企业的需要。然而我们必须记住,尽管数据库本身十分奇妙,却不是我们的最终目的,它的存在只是为了帮助企业达到实用和高效率。

常规数据处理所用文件一般认为是相互独立的。而实际上越来越多的数据往往重复出现在不同文件中,而且它们之间常常是互相矛盾的,这些文件的灵活性极差。而数据库则是一组高度集中的、同时又是十分灵活的计算机文件,这些文件相互引用却极少数据重复。

在数据库管理系统刚刚出现时,人们曾把建立单个大型中心数据库以取代企业中所有死板的计算机文件作为数据库管理的最终目标。然而不久,无情的事实便表明,这种美好的想法实际是行不通的。为了获取利益而用于先进技术的投资不一定不是浪费。事实上,为取代所有传统文件而急于花费在建立单一、全面的中心数据库上的钱极有可能就是浪费。因此,至少在近期

内,大多数企业的管理将同时包括传统文件和数据库。

数据库的管理往往比建立数据库更为重要。数据库中的数据是由企业中许多部门以及许多不同管理层次的用户共享的,这样做有利于企业的经营者们提高其决策的正确性。对于当今高度自动化同时又是十分复杂的企业环境,这是十分重要的。理由是,正确的决策要比勤奋地工作更为有效。

数据库的应用对企业管理方法会产生明显的影响。而企业管理者们只有在对数据库及其发展过程有了基本了解之后,才能全面理解数据库环境所提供的各种机会和其固有问题。据此,我们将首先并主要讨论数据库的基础,而在本书的最后概述一下数据库环境和数据库管理的问题。

数据库管理是由传统计算机文件构成的数据处理技术发展而来的。应用的范围不同,计算机文件的组织方法也不同。当各种计算机文件集中受控于一个称为数据库管理系统的软件时,就形成了数据库。所以,欲全面地理解数据库,有必要对计算机文件有所了解。如前所述,虽然数据库是今后的发展方向,非数据库类型的计算机文件仍将长期存在。所以本书在前两章专门回顾计算机方面的知识。第一章主要讨论用于读、写和存储的计算机设施,以及虚拟操作系统、批处理和联机处理的概念。第二章则主要介绍几种传统数据文件的使用及其有关问题。

本书共分十二章,从第三章开始逐步引入数据库的概念。在第三章,介绍倒排文件和多链表这样的辅助数据结构及相应的辅助关键字的概念,接着介绍数据库的雏形——初级数据库。第四章从数据独立性开始,讨论存储模式、概念模式、外部模式及三级模式数据库管理系统;介绍包括数据库设计、数据库操作和数据库管理等诸方面的数据库管理系统的应用问题。第五章首先讨论信息结构及实体-联系方法;然后通过实例讨论概念数据库结构及描述概念数据库的扩展 Bachman图;讨论在设计数据库过程中可能会出现的诸如连接陷井等方面的问题及解决方法;最后将讨论常用的三种主要数据模型及相应的数据库系统方法。第六章介绍数据库系统的关系方法,讨论关系代数和 QBE。第七章介绍一个关系数据库管理系统实例——能够运行于多种机型上的、功能完善的关系数据库管理系统 Oracle,并主要讨论已成为关系数据语言标准的结构化查询语言 SQL;也将介绍 Oracle 在安全性和并发控制等方面采取的措施,介绍 Oracle 的转储、恢复功能及 Oracle 的数据库结构。第八章介绍概念数据库设计的一些问题,包括函数依赖,模式的分解,关系的设计与规范化,最后讨论递归关系。第九章简单介绍数据库系统的 CODASYL 方法。第十章简单介绍数据库系统的层次方法 (IMS)。第十一章介绍数据库远程信息处理,主要包括用于数据管理的计算机通信系统和分布式数据库。第十二章概述数据库应用系统和数据库环境管理的一些问题,将特别强调数据库管理员的重要性和数据字典的作用。

本书主要讨论关系数据库,特别是九、十两章给出的概念文件不仅满足关系的定义,而且也象关系那样可通过连接字段产生联系。并简单介绍关系数据语言应用于 CODASYL 和 IMS 系统的趋势。

由于 Oracle 在我国的应用日趋广泛,特别是有在 IBM-PC/XT 档次的微型机上运行的版本,一般院校很容易具备上机条件,所以我们在附录 A 中介绍了 Oracle 的上机过程及其常用的

命令。同时由于历史的原因,很多院校都以并不完备的数据库系统 dBASE III 作为教学样本,为了满足不同的需要,我们在附录 B 中对 dBASE III 也做了简单介绍。但是,如果条件具备,高等院校的数据库教学还是应以象 Oracle 这样的完备的数据库管理系统作为教学样本,使学生对数据库管理系统有个真正的全面了解。

本书为教学需要,每章附有大量习题,可供主讲教师选用,同时也可作为自学者自检、练习之用。

本书可作为信息管理专业、计算机应用专业和有关的非计算机专业本科生的教材或教学参考书,也可供有关技术人员和管理干部阅读参考。

索梅编写了第七章的部分章节和附录 A, B。另外参加过部分初稿工作的有任雪松、林小茶、郁红英和陈甦。

清华大学俞盘祥副教授校阅了本书的全部书稿,提出了很多宝贵的意见和建议,在此表示衷心的感谢。

由于编者水平有限,难免有疏漏之处,恳切希望广大读者提出批评和建议。

编 者

1989年8月

目 录

第一章 基础知识

1.1 基本概念	1
1.1.1 计算机的基本结构	1
1.1.2 多道程序操作系统	2
1.1.3 I/O 处理器的作用	2
1.1.4 记录块	4
1.1.5 逻辑文件与物理文件	4
1.2 磁带与磁盘存储介质	5
1.2.1 磁带和磁带机	5
1.2.2 磁盘和磁盘驱动器	6
1.3 虚拟操作系统	8
1.4 数据处理的方式	8
1.4.1 批处理	8
1.4.2 联机处理	9
1.4.3 分布处理	9
1.4.4 适合于批处理和联机处理的操作系统	9
习题一	10

第二章 传统数据文件

2.1 顺序文件	12
2.1.1 顺序存储文件的建立	12
2.1.2 顺序文件的更新	15
2.1.3 顺序文件的应用	18
2.2 索引顺序文件	19
2.2.1 基本概念	19
2.2.2 索引顺序文件的操作	22
2.3 散列文件	27
2.3.1 散列原理	27
2.3.2 建立散列文件	31
2.3.3 散列文件的应用	34
习题二	36

第三章 初级数据库

3.1 辅助数据结构与辅助关键字	37
3.1.1 倒排文件	37

3.1.2 多链表的使用	40
--------------	----

3.2 初级数据库

3.2.1 相关的存储文件	43
3.2.2 数据库文件的连接	44
3.2.3 数据库管理之需要	46

习题三

第四章 三级数据库

4.1 数据独立性	48
4.1.1 存储文件、概念文件与外部文件	48
4.1.2 存储数据独立性	50
4.1.3 概念数据独立性	51
4.1.4 数据独立性的数据管理含义	53
4.2 三级数据库管理系统	54
4.2.1 数据库管理系统的各级抽象	54
4.2.2 一个典型的模式的例子	55
4.2.3 数据库系统	60
4.2.4 数据字典	62
4.3 数据库的设计、应用与管理	63
4.3.1 数据库设计过程概述	63
4.3.2 数据库操作语言	63
4.3.3 数据库的操作	64
4.3.4 联机数据库系统的并发使用	65
4.3.5 数据库管理	66
习题四	67

第五章 概念数据库结构

5.1 信息结构	68
5.1.1 三种世界	68
5.1.2 实体-联系方法	69
5.2 概念数据库结构——实例研究	71
5.2.1 概念数据库中的基本联系	71
5.2.2 具有大量概念文件的数据库	77
5.2.3 连接陷阱	80
5.2.4 实践中的订货文件	83
5.3 数据模型及其数据库系统的主要方法	84

5.3.1 层次模型与层次方法	85
5.3.2 网络模型与 CODASYL网络方法	87
5.3.3 关系模型与关系方法	91
习题五	92

第六章 数据库系统的关系方法

6.1 关系模型的基本概念	93
6.1.1 笛卡尔积与关系定义	93
6.1.2 其它一些基本概念	94
6.2 关系模式	95
6.2.1 关系概念模式	95
6.2.2 关系存储模式	96
6.2.3 关系外部模式	97
6.3 关系数据语言概述	97
6.4 关系代数	98
6.4.1 传统的集合运算	98
6.4.2 专门的关系运算	99
6.5 QBE	105
6.5.1 检索操作	106
6.5.2 更新操作	118
习题六	119

第七章 Oracle 数据库系统

7.1 Oracle 概述	121
7.2 Oracle 的系统结构及其实用程序	122
7.2.1 系统结构	122
7.2.2 Oracle系统的实用程序	123
7.3 交互式SQL	124
7.3.1 查询	125
7.3.2 SQL 的定义功能	136
7.3.3 SQL 的操作功能	142
7.3.4 SQL 的控制功能	144
7.4 宿主型SQL	145
7.4.1 主变量	145
7.4.2 错误处理	146
7.4.3 与Oracle连接	146
7.4.4 位置指针	147
7.4.5 应用程序体	147
7.4.6 程序实例	148
7.4.7 预编译	152

7.5 安全性、一致性与并发控制	152
7.5.1 安全性	152
7.5.2 一致性与并发控制	155
7.6 转储与恢复	159
7.6.1 装载和卸载	159
7.6.2 后映象日志	160
7.7 Oracle 数据库结构	160
7.7.1 分区和物理文件	161
7.7.2 表和索引的空间要求及空间定义	162
7.7.3 使用多个分区	163
7.7.4 存储空间满的两种情况	164
习题七	165

第八章 概念数据库设计

8.1 函数依赖	167
8.2 关系模式的分解	170
8.2.1 模式分解的必要性	170
8.2.2 模式分解的无损连接	171
8.2.3 保持函数依赖的分解	172
8.3 关系的设计(规范化)	174
8.3.1 单关键字关系中的非函数依赖与消除	174
8.3.2 复合关键字关系中的非函数依赖与消除	177
8.4 多值依赖	179
8.5 关系的分类	180
8.6 递归关系	182
8.6.1 一对多递归联系	182
8.6.2 多对多递归联系	184
习题八	185

第九章 数据库系统的 CODASYL 方法

9.1 CODASYL数据库系统的历史	187
9.2 CODASYL概念模式	188
9.2.1 定义 CODASYL 概念文件	189
9.2.2 定义 CODASYL 系	191
9.3 CODASYL数据库的操作	196
9.3.1 CODASYL 外部模式	196
9.3.2 CODASYL 数据库的载入	198
9.3.3 CODASYL 的当前值概念	199

9.3.4 CODASYL 数据操作语言命令	200
9.4 基于自然量词的 SQL/N	207
习题九	213

第十章 数据库系统的层次方法

10.1 IMS 简介	215
10.2 IMS 概念数据库	216
10.2.1 一个实例	216
10.2.2 层次型与层次值	216
10.2.3 层次数据库中的非需函数依赖	219
10.2.4 层次序列和相关的概念	219
10.2.5 IMS数据库描述的概念部分说明	222
10.3 IMS 外部数据库	223
10.4 用 DL/1 操作 IMS 数据库	225
10.4.1 与 DL/1 通信	225
10.4.2 DL/1 检索命令	228
10.4.3 DL/1 更新命令	230
10.5 用 SQL/DS 操作 IMS 数据库	232
10.6 用 IMS 管理网络结构数据库	232
习题十	237

第十一章 数据库远程信息处理

11.1 基本(单线)联机系统	239
11.2 包交换通信系统	240
11.2.1 联机系统中的通信概述	240
11.2.2 包交换原理	241
11.3 多线联机系统	243
11.4 分布式数据库	244
11.5 Oracle 的分布处理能力	246
习题十一	247

第十二章 数据库应用系统与数据库环境管理

12.1 数据库应用系统	248
12.2 数据库管理员	249
12.3 数据字典及 Oracle 的数据字典功能	246

12.4 数据库应用的高级管理	253
习题十二	254

附录 A Oracle 上机与命令要览

A.1 Oracle 的目录结构	255
A.2 Oracle 的安装与初始化	255
A.3 Oracle 的运行	256
A.4 UFI 命令	257
A.5 SQL 命令	261
A.6 SQL 运算符	265
A.7 SQL 函数	265
A.8 Oracle 保留字	267

附录 B dBASE III 简介

B.1 dBASE III 的特征和使用环境	269
B.2 dBASE III 的文件类型	270
B.3 功能与典型操作	270
B.4 全屏幕操作控制键	273
B.5 dBASE III 函数表	275
B.6 dBASE III 命令表	276
B.7 dBASE III 系统参数设置	279

附录 C 数据库系统开发示例

C.1 基于 E-R 方法的数据库设计	281
C.2 关系数据库模式的设计	282
C.3 建立数据库及一些特性的考虑	284
C.3.1 字段类型、数据量与存储空间	285
C.3.2 基本表和索引的建立	286
C.3.3 数据的载入	288
C.4 数据库应用程序的开发	289
C.4.1 数据录入程序	289
C.4.2 用于查询的程序	292
C.4.3 定期报表程序	296
C.4.4 更新程序	298
C.5 其它几个问题	299
主要参考书目	301

第一章 基础知识

1.1 基本概念

1.1.1 计算机的基本结构

计算机是存储信息、处理信息的机器。为了处理信息,计算机需要有输入、存储、运算、输出以及控制这些过程的功能。这五种功能叫做“信息处理的五大要素”,计算机在执行这五种功能的过程中实现信息处理。

图 1.1 说明了计算机系统的基本构成,包括输入设备、输出设备、存储器(主存储器和辅助存储器)、控制器和运算器。

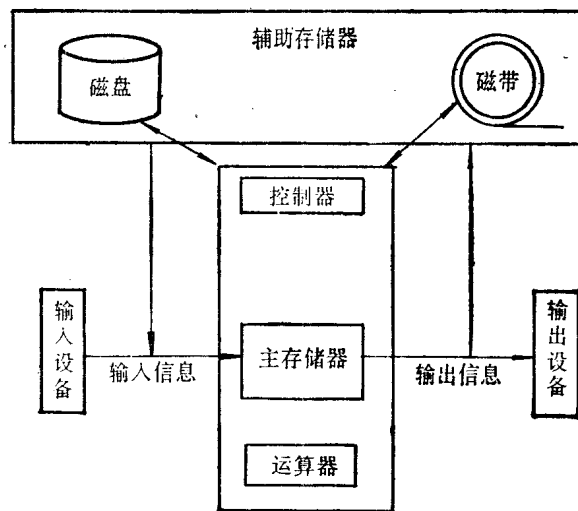


图 1.1 计算机系统的基本组成

输入设备完成计算机从外部取进需要处理的信息的功能。

输出设备完成将计算机处理信息的结果送往计算机外部显示的功能。

存储器负责信息的记忆和存储。

运算器负责对主存储器中的信息进行运算、比较、判断、组合等信息处理。运算器处理的结果仍存储在主存储器中。

计算机完全是按照人编制的处理步骤进行工作的,即由程序控制其工作。每一条指令由控制器进行译码,产生所需要的命令去控制信息处理。在执行处理之前,程序预先被存储在主存储器中,将构成存储程序的这些指令一条一条地送入控制器,控制器便将所需要的命令发送到相应

的装置上.

一般我们把运算器和控制器统称为中央处理器(CPU).

主存储器通常又分为程序区和数据区两大部分.

1.1.2 多道程序操作系统

计算机的输入、输出措施是数据处理的基础. 需做处理的数据必须从某种外部存储介质输入计算机;而经计算机处理过的数据又必须从计算机中输出或存放在外部存储介质上. 与 CPU 的速度相比, 这种输入、输出过程是非常慢的. 一般 CPU 访问磁盘 10 次所用的时间足够访问主存储器 100 万次. 换句话说, 在用来从外存设备中获取附加数据的时间内, CPU 可以处理主存储器数据区中极大量的数据(原因是外部设备为机电设备, 而主存储器则是电子器件). 这样, 输入、输出就构成了计算机系统的瓶颈部位.

为了使计算机大量的 CPU 资源在输入、输出时不被浪费, 最好的办法就是允许多个程序同时进驻内存, 并且交替执行. 这样的系统就是多道程序操作系统.

在多道程序操作系统中, 可以认为主存储器是(动态地)划分成区域的, 每个分区内有一个执行某项数据处理功能的程序. 在任一确定时刻, CPU 只能执行一个分区中的程序. 当正在执行的程序需要访问外部设备时, 操作系统将调用 I/O 处理程序(通道程序)去完成输入、输出任务. 而 CPU 则转去执行另一个分区中的另一个程序. 在其后的某个时刻再转回来执行原来的程序. 这样 CPU 总是在工作之中, 充分利用了计算机的宝贵资源.

1.1.3 I/O 处理器的作用

图 1.2 详细地画出了主存储器中存有应用程序 A 的一个典型分区. 图中还画出了一个 I/O 处理器(通道), 以及与之相连的磁盘控制器. 控制器又与一个或多个磁盘相连. 为了搞清楚内

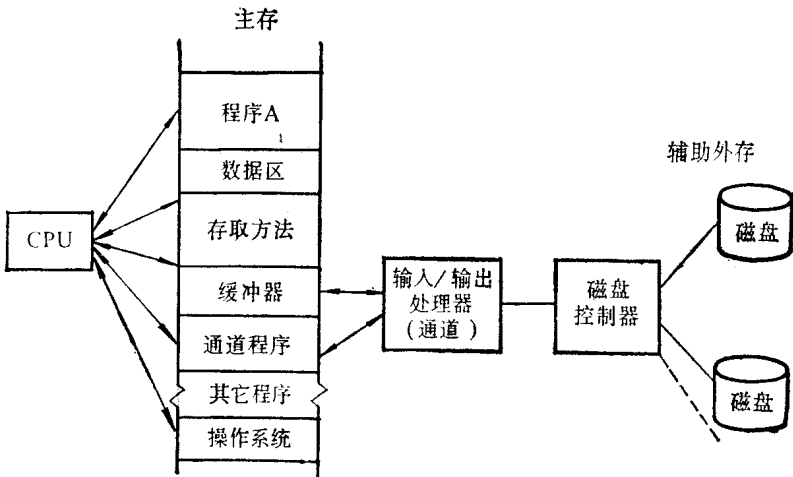


图 1.2 主存储器中存有应用程序 A 的一个典型分区

存分区, I/O 处理器及磁盘控制器的作用, 我们假定程序 A 正在运行之中, 并且这一程序含有要求从磁盘获取数据的命令 READ. 当 CPU 执行 READ 命令时, 将执行操作系统的存取方法程序, 它存在于程序 A 所在分区内. 存取方法首先检查一个称为缓冲器的区域内是否含有所需数据. 若有, 则将其移入程序 A 的数据区, 程序 A 继续被执行. 如果缓冲区内并无所需数据, 则会有一个相当复杂的处理过程. 下面大致描述一下这一过程.

首先, 存取方法将产生一个消息, 并将其置于操作系统所在分区中一个称为消息缓冲区的尾部, 再将控制转交给操作系统. 操作系统首先检查程序 A 中存取方法送来的消息 (其它分区送来的消息亦在操作系统的消息缓冲区内排队, 等待处理. 在此, 我们不妨忽略这一点). 这条消息要求从某个特定磁盘中获取数据并将其存放于程序 A 的缓冲区中. 操作系统开始调用某个处于空闲状态的 I/O 处理器, 由它去执行一个通道程序. 这个程序也在程序 A 的分区内. I/O 处理器通过与存放通道程序的存储区直接相连来执行这一程序的每条指令. 同样, I/O 处理器也可和磁盘控制器直接相连. 在执行了通道程序之后, I/O 处理器便调动磁盘控制器将特定磁盘的读/写臂移至所需要的磁道处, 然后将数据传入缓冲区. 缓冲区的作用相当于外存与主存间传递数据的一个装卸过渡区.

一旦开始执行通道程序, 操作系统便将控制转给其它分区内的某个应用程序, 使那个程序又开始运行. 就是说, 当 I/O 处理器在执行程序 A 分区中的通道程序时, CPU 则在执行其它分区中的某个程序. 前面说过, 从外存设备中获取数据是相当慢的, 因而在通道装满缓冲区的过程中, CPU 可以执行大量的应用程序指令. 实际上, 往往有若干个通道在同时工作, 各自为某个应用程序完成输入、输出工作. 当程序 A 的缓冲区被填满后, I/O 处理器向操作系统发送一个相当于消息的信号, 以便操作系统再次将执行控制转给程序 A 的分区. 然而, 首先恢复运行的是存取方法, 由它将一部分数据从缓冲区移入程序 A 的数据区, 再把控制交给程序 A. 当 CPU 执行 WRITE 命令时, 需要将程序数据区中的数据移入外存设备. 这是个与上述过程类似但却是反向的过程.

多道程序操作系统和 I/O 处理器的作用可以在相当程度上同糕点厂的工序管理相比较. 假设某车间的唯一制作师 (CPU) 正在同时烤制不同类型的糕点, 每种糕点都有自己的制作配方 (程序) 和配料 (数据). 我们进而假设每种糕点所用的配料大部分都存放在各自的仓库中 (假设仓库和车间有一定的距离). 各车间都有专门从事将配料从仓库搬入靠近车间的一间小贮藏室中去的搬运工. 现在我们假设制作师正在配制糕点 A (按配方 A). 这时, 某种配料用完了. 根据某项管理规程 (存取方法), 制作师先去查看相应的贮藏室 (缓冲区). 如果那里有所需要的配料, 制作师便拿一部分去车间放入配料堆 (糕点 A 的程序数据区); 若贮藏室中没有该种配料, 则按另一管理惯例 (操作系统), 制作师将请一位搬运工到相应的仓库中去取. 那位搬运工 (I/O 处理器) 将按照提取该种配料的特定手续 (通道程序) 从仓库 (外存储器) 中搬运部分配料放入车间旁的贮藏室中 (缓冲区). 由于仓库距离很远, 搬运工的搬运工作很费时间. 而在同时, 制作师则正按配方忙于制做另一种糕点. 其后某一时刻, 制作师又会从贮藏室中获得糕点 A 的配料, 将其与其它配料放在一起, 继续按配方 A 制做糕点 A.

1.1.4 记录块

到目前为止,我们还没提及外存与程序数据区间是怎样传递数据的,而只说到传输过程分为两步:在通道程序控制下大批量地从外存向内存缓冲区的传输和小批量地从缓冲区到程序数据区的传输。这里每个“小批量”是缓冲区和程序数据区间数据传输的基本单位,我们把它称为记录;而每个“大批量”是内存和外存间数据传输的基本单位,它包含若干个记录,我们把它称为记录块,或传输数据块,同时它也被作为外存的一个基本存储单位。

图 1.3 示意了上述传输过程。外存(这里指磁盘)含有一连串的物理存储块(记录块),每块中既含有输入磁盘的原始数据块,又有一些附加的系统数据表明该块的位置和标志。我们知道,通道向主存传送的数据是首先放在缓冲区中的,因而缓冲区应有足够的空间来容纳一个数据块或记录块。实际上,操作系统给与程序有关的每个文件分配至少一个缓冲区,而每个缓冲区的大小与传输数据块相同。

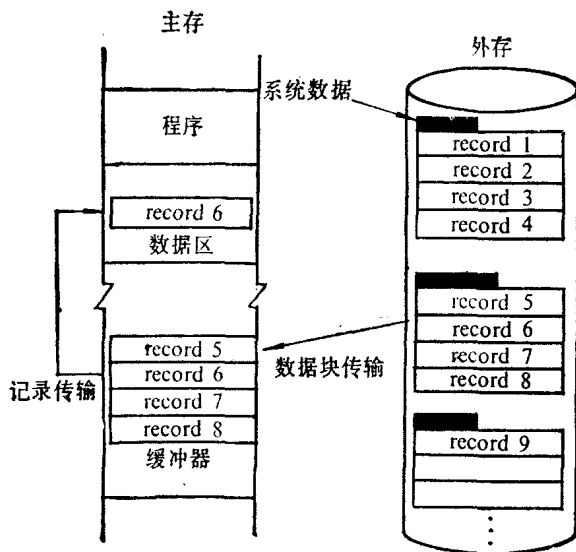


图 1.3 数据传输过程示意

每个记录块所包含的记录个数称为块因子。每个记录又由若干数据项或字段构成。在向缓冲区传入一个记录块之后,存取方法又可根据关键字字段从缓冲区中选取某个记录并将其送入程序数据区以供处理。

1.1.5 逻辑文件与物理文件

只要知道每个字段中的字节数,就可以用简单加法算出每个记录的字节数,再用块因子与之相乘,便可得出每个记录块的字节数(假定所有记录等长,通常这种假设也是合理的)。存储在存储介质上的物理记录或记录块构成了物理存储文件。为方便起见,我们设想还存在一个逻辑存储文件,它由相应的逻辑记录或逻辑存储块构成。需要注意的是,物理存储文件在物理上真实存

在(在磁盘上),传输数据块也存在(在缓冲区内),而逻辑存储文件则只是程序所处理各记录假想中的组合,并不以某种组合形式实际存在。

当程序员用 COBOL(或其它高级语言)写程序时,他们所关心的只是逻辑存储文件,因而不必考虑复杂的物理存储文件。但若需要计算一个文件所占的存储空间,就不得不仔细考虑物理存储文件了。

1.2 磁带与磁盘存储介质

大量的计算机数据通常是存放在磁带和磁盘上的,但磁带上的记录只能被顺序读取,而磁盘由于允许直接访问数据,因而一般被用作数据库的存储介质。然而我们也不能忽视磁带,它们在数据库中也是有用的,尽管其用处受到限制。

1.2.1 磁带和磁带机

图 1.4 画出了一台磁带机的示意图。与普通磁带录音机不同,当磁带文件接受处理时,磁带并不作连续运动,而是时走时停。磁带上各数据块之间有空隙相隔,称为块间隙,或起-止间隙。我们知道数据是分块被读取的。当数据被处理时,磁带是停止不动的,此时读/写磁头处于块间隙处。为了能让磁带迅速地加速和减速,在读/写磁头和两个磁带盘之间各留有一段松弛的磁带作为缓冲(图中 A, B 处)。缓冲带是靠光电控制来维持的。

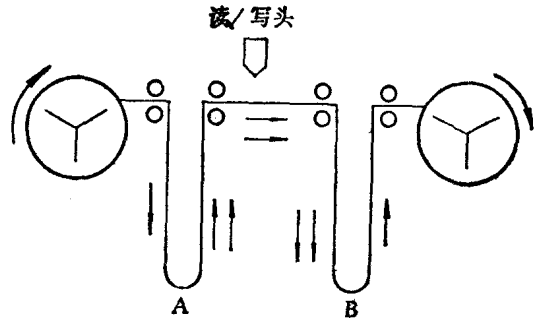


图 1.4 磁带驱动示意图

一盘磁带一般有 2400 英尺长(指新磁带,旧磁带有时会被剪短)。磁带的数据密度和块间隙大小有若干规格,详见下表:

密度(位/英寸)	隙宽(英寸)	隙宽(字节)
800	0.6	480
1600	0.6	960
6250	0.3	1875

其中 1600 位/英寸(bpi)密度的磁带是最常用的。注意:数据密度是以位/英寸为单位的。由于每个数据字节是在垂直于磁带走带方向上记录于磁带上的,因而在 1600-bpi 的磁带上每英寸实际上可记录 1600 个字节。相应地,0.6 英寸的块间隙占据了 0.6×1600 个字节,即 960 个字节。

在此我们注意到,磁带上的块间隙是相当大的。如果我们用块因子为 10 的数据块在 1600 bpi 的磁带上存储记录长度为 104 字节的数据,物理存储块的长度应为 1040 字节。当把块间隙亦考虑在内时,则必须分配出 $1040 + 960$ 字节的磁带长度,即 2000 字节。这样,整盘磁带所能存

放的块数为

$$\frac{2400 \times 12 \times 1600}{2000} \text{块}$$

即 23040 块,或 230400 条记录。例如没有用块间隙,则磁带在理论上可以存放

$$\frac{2400 \times 12 \times 1600}{1040} \text{块}$$

即 44307 块,或 443070 条记录,几乎比前者翻了一番。然而,取消块间隙是不行的,唯一可采取的措施是尽量加长物理存储块的长度,可是这意味着在主存储区中要有相应的大缓冲区,而这往往是不现实的。一般,块的大小在 2000 到 3000 字节之间。

在磁带上读取一块数据所需时间取决于两个因素:

1. 启动磁带所需时间;
2. 传送数据所需时间。

磁带加速时,磁头尚处于块间隙处,而当它移至数据处时,磁带已达到其正常速度,这时数据以标准传输速率传输。常用的 IBM 磁带机使用 1600-bpi 磁带时的读写速度为每秒 100 英寸(相当于每小时大约 5 英里),与之相应的标准传输速率为 1600×100 ,即 160000 字节/秒。

在上述情况下,传送一个 2000 字节的数据块需时 $2000/160000$ 秒,即 12.5 毫秒。磁带启动时间约为 4 毫秒,因而总共需要时间为 $12.5 + 4$,即 16.5 毫秒。

按计算机的标准,这确实是个很长的时间。与之相比,CPU 可以在小于 1 微秒的时间内访问主存储器内的数据。后面我们将会看到,从磁盘上传出 2000 字节的数据块所需时间也是很长的。

1.2.2 磁盘和磁盘驱动器

磁盘分为硬磁盘和软磁盘两种。我们主要讨论硬磁盘,它是由装在一根转轴上的若干金属盘片构成的磁盘组。当磁盘驱动器开启后,转轴便不停地旋转(与之不同,磁带机仅在传送某个数

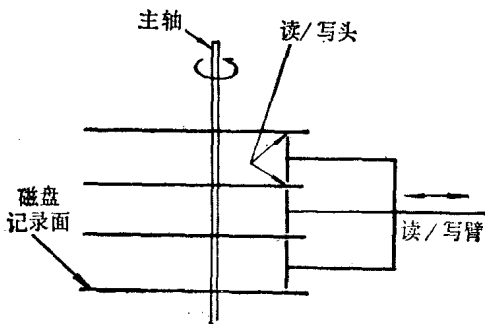


图 1.5 磁盘组与磁盘驱动器的读/写臂

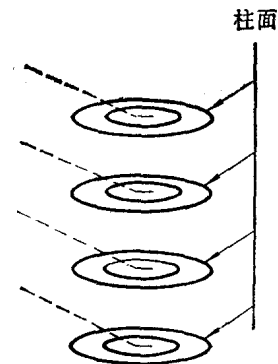


图1.6 同一半径的磁道形成一个柱面

据块时才动作)。除了最上面的盘面是用于控制目的外,所有盘面上都有许多同心圆状的存储面(就象土星光环)。如图 1.5 所示,梳齿状排列的读/写磁头可以同时移动,因此在读/写臂端的读

/写磁头可以被移至具有同一直径的磁道的上方(或下方)。同一直径的所有磁道的集合称为磁道柱面。当读/写磁头所组成的“梳子”在任一特定位置上时,磁头将占据相应磁道柱面上的所有磁道,使这些磁道都可以被读/写而不用进一步移动读/写臂。读/写臂向某个磁道柱面的移动称为查找,这一过程较慢。从存储角度看,磁盘组是由磁道柱面构成的,每个柱面又由许多磁道构成(见图 1.6)。

在磁盘上存储数据的常用方法是将磁道划分成等长区段。每个区段一般较小,典型值为512字节。由于常用的数据块一般都大于一个区段,每个数据块往往要占据若干区段。在磁道上的数据块可以用柱面号,磁道号和起始区段号来唯一确定。每个磁道所能存储的数据数量都与其它磁道相同,而与其直径无关。显然,在内磁道上的数据要比外磁道上的数据紧凑。

假定我们在磁盘上存有一个逻辑块长为 2000 字节的文件。从理论上讲,对这个文件的处理既可以是顺序的,亦可以是直接的。采用顺序处理时,先顺序读第一柱面上的各块,然后是下个柱面上的各块,依此类推。这样,在同一个柱面内读/写臂不需作任何移动便可读下一块的内容,即不必查找。由于柱面上各块是逐个被读取的,且每两块的读取过程之间有个不确定的时间间隔,所以任一块的读取都必须让读/写磁头等待磁道旋转到一定程度才行。这就形成了数据传输之前的一个旋转延时。旋转延时的平均值为磁盘旋转一周所需时间的一半。在顺序文件中读取下一块数据所需的时间应为:

1. 旋转延时。
2. 直接传送数据所需时间。

以 IBM 3350 磁盘为例,它旋转一周需要 16.7 毫秒,相应的旋转延时为 8.3 毫秒,传输速率为 120 万字节/秒。这样传送一个 2000 字节的数据块仅需 $2000/1200$ 毫秒,即约 2 毫秒。而读或写一个这样的数据块实际上需要的时间为 $8.3+2$,即约 10 毫秒。我们还记得,用 1600-bpi 磁带时,读/写时间大约为 16 毫秒。

对磁盘上的文件常常可以采用直接方式进行处理。就是说可以按任意顺序读取数据块。这时,在两次读块之间往往会有一个查找过程。IBM 3350 的平均查找时间为 25 毫秒。我们说平均时间,因为当读/写臂要从最外柱面移至最内柱面时,需要的查找时间最长,而移向相邻柱面时,需时最短。当读/写臂找到所需柱面后,又必须经历一个旋转延时,接着是直接传输数据块的时间。这样,读/写一个块所需时间为:

1. 查找时间。
2. 旋转延时。
3. 直接传输数据所需时间。

如此,一个 2000 字节块的读/写时间为 $25+8.3+2$,即平均 35 毫秒(其中旋转延时和传输时间与顺序传输时相同)。按照计算机的标准,这个时间是太长了,它往往是数据库和文件系统性能不佳的主要原因,当然,采用特别昂贵的磁盘机是可以消除查找时间的。在这种磁盘机内,每条(或几乎每条)磁道上都有一个读/写磁头。这种磁盘只用于对访问时间要求很高的系统,如虚拟操作系统(见 1.3 节)。

1.3 虚拟操作系统

出于讲解的需要,我们在前文中有意回避了多道程序操作系统的一个重要方面,即多道程序操作系统有两个基本类型:传统型(我们在前文中介绍的)和所谓虚拟操作系统。后者自 70 年代初开始取代前者,现在已成为最常用的操作系统类型。其实虚拟操作系统的工作方式与传统操作系统没有多大差别,只不过采用了请求式页面调动系统而已。这一改进使得那些大得无法容纳于主存储器内的程序也可以在系统上运行了。

在任一时刻,运行在虚拟操作系统上的用户程序都只有一部分存储于主存储器内。而用户程序的主体、所有通常意义的缓冲区存取方法和通道程序(即传统操作系统的一部分)都存放在一个特殊的辅助存储器内。这一辅助存储器称为外部分页存储器,通常是一个高速磁盘系统。应用程序和所有相关子程序以及数据区都是划分成页的(每页约 4000 字节长)。换言之,虚拟操作系统外部分页存储器中的内容与传统操作系统的主存储器中的内容没有什么两样,只不过外部分页存储器是分页的而已。我们仍可把外部分页存储器看作是划分为分区的,每个分区含有一个用户程序。

在任一时刻,每个分区内的内容只有几页是在主存储器内的,而且正在运行的则只是某一页上的某个指令。当一页上的所有指令都已执行完毕,而下一页尚未进入主存时,所需页将被读入主存储器。执行此任务的是操作系统中的一个单元,称为分页管理程序。与此同时,CPU 则去执行其它某应用程序的某一页,只是在其后某个时刻才又开始执行刚刚调入的那一页。

可以认为,虚拟操作系统与传统系统的基本不同点在于前者不断地将各区域中的内容分页调入或调出主存储器。这种请求式页面调度功能使大于主存或大于主存中空余分区的程序得以在系统上运行,从而使计算机系统的效率提高。然而请求式页面调度的应用,使人们对计算机系统工作过程的了解变得更为复杂,而且也使我们几乎无法画出明了的示意图。这使问题从根本上复杂化了。为了简化进一步的讨论并且能够画出示意图来,我们不妨接受下述观点,即一个应用程序加上所有相应子程序和数据区可以占有主存储器中任意大的一个分区。但我们必须明白,在实际系统中这样的分区是存在于外部分页存储器中,不断地被分页调入、调出主存的。

1.4 数据处理的方式

数据的处理方式有三种,它们是批处理、联机处理和分布处理。

1.4.1 批处理

在数据处理技术发展初期,几乎都采用批处理方式。这种方式是把原始数据先“收集”在一起,然后统一送入计算机进行处理。成批处理的数据,一般按文件组织分为两类:即主文件和细