

第一篇 编程命令与技巧

第一章 概 述

计算机的应用领域已经从早期以数值应用为主转变为现在非数值应用为主。目前非数值数据的处理几乎占用了90%以上的计算机时间。当前一个重要的课题是:如何有效地对数据进行组织和管理。数据库的出现为大量的形式繁多的非数值数据的管理提供了科学的办法。本章将扼要地阐述什么是数据库、数据库的种类、建立数据库有哪些要求、数据库文件有哪些性质等,使读者对数据库的基本概念有所了解,然后再介绍微型机数据库管理系统dBASE III及其组成。

第一节 数据库的定义及其用途

在日常工作和生活中,人们经常在接触各式各样的信息,并且频繁地在传递和利用这些信息。例如当我们进入商店时,总是根据自己对商品的需要去了解价格信息,然后根据这个信息和自己的情况决定是否购买。又如,人们每天早晨收听天气预报,然后根据这个信息决定是否要带上雨具,这也是在了解信息和利用信息。例如,制鞋厂的产品管理系统,必须及时了解社会上需要的鞋子式样、数量及种类,并且利用这些商品信息确定鞋厂的生产品种和产量,及时满足市场需要。计算机在处理这些信息时,需要将外界的信息转换成计算机能识别的符号,这就是通常所说的转换为数据。

数据可以这样定义,它是用来描述客观事物,并且能输入到计算机中被计算机所识别的数字(如0,1,2,...,9)、文字(如大、小写英文字母和汉字)、图形和声音等的总称。

在利用计算机管理数据时,需要对大量输入的数据进行分类、存贮、查找、修改和增删等,这些工作统称为数据管理。数据库管理系统就是全面地实现这种管理的一种软件系统。确切地说,数据库是一个通用化的综合性的数据集合,它是按数据的自然联系来构成的,它包含各个体数据的描述以及各个体数据间的联系,在不同的应用场合,可以对数据进行相应的各种组合,以满足给定的应用要求。

为了使读者能更好地了解数据库的概念及如何利用数据库,下面举一个简单地例子(见表1-1)。

表1-1 为一个工资表,共包含三个成份:

表的名称:表的名称用来区分各种不同的表。表1-1的名称为工资表。

表的格式:表的格式用来指出表中共有多少项,每一栏的标题和宽度。表1-1中共有14项。即14个标题,各栏的标题是工号、姓名、页号、基础工资、浮动、岗位工资、年功工资、副补、食补、粮差、扣病假、扣事假、洗理、实发。每个栏的宽度是按照本栏中的内容而定。

表的内容:表的内容就是表中所填入的数据,也叫每一栏的内容,它可以是文字,也可以是数字。表 1-1 的内容是 10 个人的工资表。

表 1-1 工资表

记录号	工号	姓名	页号	基础工资	浮动	岗位工资	年功工资	副补	食补	粮差	扣病假	扣事假	洗理	实发
1	13	冯春	20902	125.00	62.00	50.00	32.00	7.50	20.00	12.00	0.00	0.00	8.00	316.50
2	65	郑蔚	10101	111.00	50.00	28.00	19.00	7.50	20.00	12.00	0.00	8.88	9.00	247.62
3	303	王敏	20405	90.00	43.00	22.50	2.00	7.50	20.00	12.00	5.76	3.60	9.00	196.64
4	537	张萍	20102	132.00	45.00	25.00	30.00	7.50	20.00	12.00	0.00	0.00	9.00	280.50
5	20	吴城	10102	193.00	135.00	65.00	34.00	7.50	20.00	12.00	0.00	0.00	8.00	474.50
6	22	刘生	10102	161.00	105.00	65.00	33.00	7.50	20.00	12.00	0.00	0.00	9.00	412.50
7	27	张凤	10102	90.00	42.00	28.00	7.00	7.50	20.00	12.00	0.00	0.00	9.00	215.50
8	28	苏娜	10102	104.00	42.00	25.00	19.00	9.00	22.00	12.00	0.00	0.00	9.00	242.00
9	32	孟敏	10103	90.00	42.00	22.50	4.00	7.50	20.00	12.00	25.92	28.80	9.00	152.28
10	33	蒋兰	10103	90.00	42.00	22.50	5.00	7.50	20.00	12.00	0.00	0.00	8.00	208.00

将整张表放入计算机中,若需要某一个人的情况时,就可以用数据库管理系统中所描述的基本语句来查找。如果需要这张表按名字排序,那么可以利用“排序命令”将这张表重新排列后,放入一个新文件中。若需要察看一下某一页号(如 10102)的工资情况,则可以用“表达式”来查找满足页号为 10102 条件的记录。若想用工资进行排序,则可以用“索引”或“分类”命令对工资进行排序。同样对姓名或工号也可以用“表达式”来查找各种需要的表达式条件。从这里可以看出,数据库可以使大量数据在统一的控制下,有效地实现存储、查询、检索和排序等各种管理。详细的命令叙述在后面。不仅如此,数据库还能大大地减少数据的重复,特别是使程序和数据各自独立,互不依赖,从而使各个程序能共享这些数据。

表 1-1 是工资表,用户可以利用所建立起来的表进行添加、修改、编辑、报表等各种功能,这些功能我们都将一一做说明。

第二节 数据结构及数据库的种类

数据结构是数据的组织形式,也可以说是数据之间的联系。它可以用一个数学表达式来定义,即 $DS = (D, R)$,其中 DS 称为数据结构,D 表示数据,R 是数据对象之间存在的关系的集合。

数据结构分为数据的逻辑结构和数据的物理结构。数据的逻辑结构是从逻辑的观点来观察数据,与数据的存储位置无关。数据的物理结构是指元素在计算机中存放的结构,也称为存储结构。数据的存储结构是逻辑结构在计算机存储结构里的实现,数据的物理结构是依赖于具体计算机的。数据的逻辑结构分为三种,即分层数据结构,网状数据结构及关系数据结构。

下面是一张工资表的数据结构。它是关系式数据结构,是用 dBASE II 建立起来的,它可以给用户一个直观的概念。

记录号:	字段名	类型	宽度	小数
1	工号	N	4	0
2	姓名	C	6	0
3	页号	C	5	0
4	基本工资	N	8	2
5	浮动	N	6	2

6	岗位工资	N	6	2
7	年功工资	N	6	2
8	副补	N	6	2
9	食补	N	6	2
10	粮差	N	5	2
11	扣病假	N	5	2
12	扣事假	N	6	2
13	洗理	N	5	2
14	实发	N	8	2

在这个结构表中共有 14 项,其中有每个字段(域)的名称、类型、宽度和小数点位置。

本节就这三种数据的逻辑结构及其相应的数据库作简要说明。

一、分层数据结构及其层次式数据库

分层数据结构是一种有限的定向有序树,例如在图 1-1 中所表示的我国东北地区行政区划分图。这个行政区划分图象一棵树,树根只有一个,即东北。省和直辖市为枝叉点,树根与枝叉点之间的联系称为键(连接键),它是一个有向的键,树根与连接键的比例关系为 1 : N,即树根有一个,树枝为 N 个。这种数据结构叫分层数据结构。由分层数据结构组成的数据库就叫作层次式数据库。

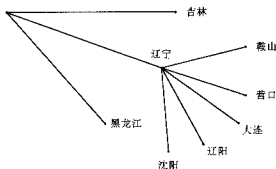


图 1-1 分层数据结构

二、网状数据结构及网络式数据库

在图 1-2 中所表示的是医院病房的数据库管理系统,该图中医生、病房和病人之间的联系是:每个病人一定住在一个病房中,也一定有一个医生负责。图中有四个病房,四名医生,柳医生负责三个病人,它们分别住在第一病房,第二病房和第三病房。汪医生负责的三个病人分别住在第一病房、第三病房和第四病房,章医生负责的病人分别住在第一病房、第二病房和第四病房。侯医生负责的病人住在第二病房、第三病房、第四病房。医生与病人(连接键)的比例关系是 M : N,也就是说有 M 个医生给 N 个病人看病治疗,而 N 个病人分别住在 P 个病房中。这种数据结构就叫作网状数据结构。用网状数据结构组成的数据库称为网络式数据库。通过数学关系式可以把网状数据结构转化为分层数据结构,网络式数据库也可以转变为层次式数据库。因为 dBASE III 不是网络式数据库,所以这种网状形式的管理不是十分适合。它适合关系式数据结构。

三、关系式数据结构及关系式数据库

关系式数据结构把数据的逻辑结构归结为满足一定条件的二维表的形式,并采用了逻辑运算和关系运算对数据进行操纵。所谓二维表就是指具有 m 行 n 列的一张表,每一行是一个 n 元组,相

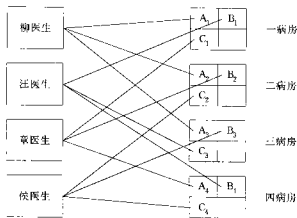


图 1-2 网状数据结构

当一个记录(所谓一个记录就是用以描述一个个体的若干个个体数据的总和),每一列被称为一个字段,字段名相当于标题栏中的标题。如表 1-2,表的名称为旅馆火车票预订表。表的格式有 6 行 12 列,表的内容已有 6 个记录,这 12 列也叫 12 个字段,字段名分别为编号、房间号、终点站、车次、发车时间、车票类别、数量、预付款、票价、退款及备注。

表 1-2 旅馆火车票预订表

记录号	编号	房间号	终点站	车次	发车日期	发车时间	车票类别	数量	预付款	票价	退款	备注
1	101	203	天津	247	06/30/93	08.00	软坐	2	4.80	2.40	0.00	Memo
2	102	208	上海	21	06/29/93	11.00	软卧	2	460.00	220.00	20.00	Memo
3	103	205	烟台	54	07/02/93	09.00	硬卧	4	420.00	100.00	20.00	Memo
4	302	123	大连	129	07/03/93	10.00	硬卧	2	680.00	140.00	400.00	Memo
5	401	308	厦门	145	07/01/93	12.00	软卧	1	780.00	760.00	20.00	Memo
6	204	440	杭州	33	07/03/93	14.00	硬卧	2	440.00	430.00	10.00	Memo

第三节 建立数据库文件的要求及数据库文件的特性

一、建立数据库文件的要求

无论使用哪种数据库管理系统,在建立数据文件时都必须注意以下几点:

- (1)尽量减少数据的重复,使数据具有最小的冗余度。
- (2)避免同一数据描述方法的不一致性。如同一个人的学历,在人事档案中是大学毕业,而在科技档案中是大学程度,这样容易造成混乱。
- (3)保持数据的完整性。如地质、水文、天气及人口资料应完整无缺地保存,以备用过期的数据来预测未来的情况时使用。
- (4)尽量使为了不同目的的许多用户共享数据资料,以提高数据的利用率。
- (5)由于某些数据需要保密,故必须增设保密措施。
- (6)根据需要数据应能及时维修。
- (7)数据的查询率高。

(8)尽量减少所占用的数据存储空间。

二、数据库文件的特性

数据库文件是具有相同性质的记录的集合,其特性如下:

- (1)一般说来,一个数据文件所含有记录是同格式的、等长的。
 - (2)不同的行是不同的记录。
 - (3)不同的列用不同的字段名。
 - (4)同一列中数据的性质是相同的。
 - (5)每一行的内容都是不可分割的,但行的顺序和列的顺序是无关紧要的。
- 从表 1-1 和表 1-2 中不难看出,每一行就是一个记录。

第四节 dBASE II 的组成

dBASE II 是目前国内外相当流行的一种关系数据库管理系统。它的前身是用 FORTRAN 语言专为大型机编写的,后来被移到微型机上,并用 Intel 8080 汇编语言进行了改写,然后又不断完善,才成为 dBASE II 早期产品 dBASE II,后来美国 Ashton-Tate 公司不断改进成为目前最流行的微机数据库语言 dBASE III。1982 年引入我国以后,由于成功的汉化,使它很快的在我国微机应用市场上占了主导地位。后来发展了 dBASE IV。

dBASE II 具有一套功能极强的人机会话式数据库的管理系统的命令和语言,简便易学,用户可以用这些命令和语言作为编制程序的工具,进行数据的自动处理。它的编程效率大大超过高级语言。

dBASE II 对硬件的要求是 16 位的 CPU,最小内存为 512K 字节,外存至少要一个软盘驱动器,键盘、显示器各一个。为了打印,需要有并行打印机一台。在软件方面能适合多种操作系统(如 CP/M, PC-DOS, MS-DOS 操作系统),但要求操作系统版本号在 2.0 以上,即 PC-DOS 2.0, MS-DOS 2.0 等以上的版本。

dBASE II 可以在多种微机上使用,它能为各种类型的终端提供全屏编辑操作,并且具有独特而丰富的查错功能和提示信息,还为用户提供了方便的帮助信息(Help),使用时具有很大的灵活性,虽然处于单用户状态,但能在网络上运行的 dBASE II Plus 为开发大型的应用系统创造了良好的环境。由于它的使用相当方便,所以有大众数据库之称。

dBASE II 有许多修改版本,这些版本的内容基本一致,本书是以 3.3 版本汉化 dBASE II 为例。它是由四个文件组成的,其中包括:

- (1)DBASE.EXE dBASE II 的总控程序及常驻内存模块。负责调用磁盘上的各种辅助程序。
- (2)DBASE.OVL dBASE II 的可覆盖模块,平时常驻磁盘,使用时由总控程序部分地调入内存。负责解释和执行 dBASE II 的各种命令。
- (3)HELP.DBS 帮助命令所用的文件,帮助用户自学 dBASE II 的各种命令和函数,提供语法、功能和使用说明。
- (4)ASSIST.HLP 菜单辅助命令。用于辅助用户正确选用 dBASE II 命令。

其中前两个文件是 dBASE II 运行时的必备文件,后两个文件可根据需要再调入的文件。

第五节 dBASE III 的基本约定

考虑到微机的特点,dBASE III 作了如下的规定:

- (1)每个文件最多允许 10 亿个逻辑记录。
- (2)每个记录最多有 128 个字段。
- (3)每个字段的字符数最多为 254 个。
- (4)每个记录的字符数最多为 4000 个。
- (5)每个命令行的长度最大不超过 254 个字符。
- (6)每个文件名长度不超过 8 个字符。
- (7)每个字段名长度不超过 10 个字符。
- (8)字段类型共有五种,字符型字段、数值型字段、逻辑型字段、日期型字段及备注型字段。
- (9)每个字段的宽度如下:

字符型字段长度前面已讲了,数值型字段长度为小于或等于 19 位数字(包括小数点),逻辑型字段长度等于 1,日期型字段长度等于 8,备注型字段长度为 512~4096。

- (10)报表标题长度不大于 254 个字符。
- (11)索引关键字长度最多为 100 个字符。
- (12)字符串长度最长不超过 254 个字符。
- (13)最大数不大于 1×10^{308} 。
- (14)最小数不小于 1×10^{-307} 。
- (15)精确度为数值位数 15 位,小数点为 9 位。
- (16)求和表达式为最多 5 个。
- (17)数据库文件类型共 9 种:

① *.DBF 数据库文件,而且是数据库的主要文件,每个数据库文件包含若干个记录,每个记录有若干个字段。

② *.TXT 文本输出文件(或称正文文件),可以用 Type 命令打印出来,它全部由可打印的 ASCII 字符组成,并作为与其它高级语言之间交换数据的接口。

③ *.NDX 索引文件,由索引命令 INDEX 所生成,该文件只有两个字段,一个是索引关键字字段,还有一个是记录指针。

④ *.PRG(或 *.CMD) 应用程序文件,也叫命令文件,它是由一系列命令语句组成,是用户自己编写的一种运行程序,它可以用 dBASE III 的内部和外部命令来建立和修改,执行该文件时用 DO 命令来完成,即 DO <命令文件名>。

⑤ *.FRM 报表格式文件,它是 dBASE III 用来产生报表的文件,它是用 REPORT 命令生成的。可用 MODIFY 命令以及 MS-DOS 或 PC-DOS 下的编辑文件来修改。建立时用 CREATE REPORT 命令。

⑥ *.MEM 内存变量文件或者叫存储变量文件,它是通过 SAVE 命令或 STORE 命令来建立的,该文件最多为 256 个内存变量,可以使用 RELEASE 命令释放内存变量,下次使用时,可用 RESTORE 命令恢复。

⑦ *.FMT 格式文件。由 @ 和 * 二种命令组成。

⑧ *.LBL 标签文件,可以用 LABEL 命令建立和修改该标签文件,建立时用 CREATE LABEL 命令,修改时用 MODIFY LABEL 命令。

⑨ *.DBT 备注文件,它是数据库文件的辅助文件,用来存放数据库中备注型字段的具体内容,如果在建立数据库文件时有备注字段,则系统将自动生成一个同名的备注文件。

*.DBT 文件在 *.DBF 文件中占了 10 个字节,与 *.DBF 文件一起建立和一起使用。*.DBT 文件最多有 4096 个字符,可以按字处理方式使用,非常灵活和方便。但是当字段宽度少于 254 个字符时,不要使用备注型字段,以免降低使用数据库的效率。

dBASE Ⅲ 上述 9 种文件提供了几种管理功能,例如显示文件目录,更改文件名及删除文件等。

这几种文件都是标准的 CP/M、PC-DOS、或 MS-DOS 文件,文件名最多用 8 个字符,扩展名 (PRG,FRM,MEM,FMT,LBL 等均叫扩展名)都是 3 个字符。

(18)dBASE Ⅲ 对命令的书写作了如下规定,读者学习时应注意。

① 用户键入的命令可用大写或小写字母,二者是等价的。

② 屏幕上,dBASE Ⅲ 的提问、回答以及所提示的信息一般用大写字母表示。

③ 在命令行的尖括号 < > 中所表示的项是必选的,用户键入命令时必须注意。

④ 在命令行的方括号 [] 中所表示的项是可选的。

⑤ 单引号 ' ' 与双引号 " " 的作用相同。

⑥ <CR>,↵,<enter>均表示回车符号。

⑦ dBASE Ⅲ 常用的几个成份说明如下:

$$\langle \text{范围} \rangle = \begin{cases} \langle \text{ALL} \rangle \\ \langle \text{NEXT } n \rangle \\ \langle \text{RECORD } n \rangle \end{cases}$$

用以指定该语句的有效作用范围。⟨ALL⟩表示对文件中所有记录都要进行按规定的操作;⟨NEXT n⟩表示对文件中从当前记录开始的几个记录都进行按规定的操作;⟨RECORD n⟩表示只对第 n 个记录进行操作。

FROM ⟨文件名⟩表示该语句操作时所使用的数据是来自⟨文件名⟩所指出的文件。

TO ⟨文件名⟩表示该语句操作过程中所产生的数据送到⟨文件名⟩所指定的文件中。

对于键入的命令,dBASE Ⅲ 只要求用前四个字符即为有效,但这四个字符必须书写正确。

(19)dBASE Ⅲ 的文件操作命令

dBASE Ⅲ 比 dBASE Ⅰ 增强了好多功能,特别是在数据库的圆点提示符下,可以完成操作系统下文件操作命令,如 TYPE 命令,ERASE 命令等等。

第六节 dBASE Ⅲ 中使用的 DOS 命令

——文件操作命令

一、显示文件目录命令 DIR

命令格式 DIR [驱动器名:]

DIR [文件名[.扩展名]][/W]

命令功能 显示当前驱动器或指定驱动器上的文件目录,其中包括文件名、记录个数、文件更新日期、文件长度及文件总数。如果不写驱动器名,则默认当前驱动器名。如果文件名和扩展名都

不选,则将盘上全部文件目录显示出来。可选项/W表示只显示文件名与扩展名和下一级子目录名,显示屏幕每次容纳5个文件名或者子目录名。

```
例如: • DIR *.COM
      dBASE.COM
      COMMAND.COM
      IBMBIO.COM
      IBMDOS.COM
      .....
```

在本例中使用的统配符“*”是和操作系统状态下使用的*号含意一样,表示不管什么文件名,只要是扩展名是.COM的均可以。“?”也是和在操作系统状态下含意一样。如:DIR ??P.DBF,则显示有三个字符的文件名,其中前两个是任意的,而第三个是字母P,扩展名是.DBF文件。这比dBASE II前进了 一步。

二、更改文件名命令 RENAME

命令格式 1 RENAME <源文件名.扩展名> TO <新文件名.扩展名>

命令格式 2 RENAME [驱动器名:] <源文件.扩展名> TO <新文件名.扩展名>

命令功能 将默认驱动器名下的源文件更改为新文件。若指定驱动器上文件则用第二种命令格式把驱动器名加上。更改后的新文件不能与磁盘上的文件名相同。

例如:RENAME LI.DBF TO LIM.DBF

这是把 LI.DBF 的名更换为 LIM.DBF。

值得注意的是不能对已打开的数据库更换文件名,另外有备注字段的数据库文件更改文件名时,则必须更改相应地*.DBT 文件名。

三、拷贝命令 COPY

命令格式 COPY FILE <源文件名> [扩展名] TO <新的文件名> [扩展名]

命令功能 拷贝指定驱动器(A,B,C,均可)上的文件为新的文件名,即用此方法可以生成一个新文件。如果不选驱动器则指定系统默认的驱动器。

例如:• COPY FILE LU.DBF TO A:LIN.DBF

这是将 C 盘上(目前默认盘为 C 盘)的文件 LU.DBF 拷贝到 A 盘上 LIN.DBF。和 RENAME 命令的要求一样,不能对正在打开的数据库文件进行复制。

拷贝命令 COPY 可以对 dBASE II 9 种文件的任意一种文件进行复制。除了数据库文件以外其它 8 种文件必须写扩展名,否则出错。

四、删除命令 ERASE

命令格式 ERASE [驱动器] <文件名.扩展名>

命令功能 从磁盘上删除用户指定的文件。如果不指出驱动器则指默认驱动器上的文件。

例如:• ERASE A:LI.DBF

该命令执行后,则把 A 盘中 LI.DBF 文件删除了,但不能对正在打开的数据库进行删除操作。

五、删除命令 DELETE

命令格式 DELETE [驱动器:] 〈文件名, 扩展名〉

命令功能 从磁盘上删除指定的文件。两个删除的命令对删除文件是等效的, 不过 DELETE 命令用法很多, 在第三章中详细叙述。

例如: • DELETE C:GZB1.DBF

这是表示删除 C 盘中 GZB1.DBF 文件。

值得注意的是同样不能删除正在打开的数据库文件。使用这两个删除命令时要十分慎重, 不能随便使用, 因为一旦删除文件将永远不能恢复。

六、显示文本文件命令 TYPE

命令格式 TYPE 〈文件名, 扩展名〉 [TO PRINT]

命令功能 显示文本文件的内容。如选用 TO PRINT 则将同时把文件内容传送到打印机上打印。

例如: • TYPE A.PRG

这时屏幕上显示了当前默认驱动器上的文件为 A.PRG 的内容。

值得注意的是不能用 TYPE 命令显示数据库文件、索引文件和已打开的数据库文件。若想显示上述文件的内容, 则需把数据库文件或索引文件转换为文本文件, 即扩展名为 *.TXT 的文件。

第七节 dBASE III 自学命令及使用

前面已说明的 HELP 文件就是帮助用户自学 dBASE III 的命令, 常用的 HELP 命令有二种格式。

一、帮助命令 HELP

命令格式 1 HELP

命令功能 用来学习整个 dBASE III 语法规则、定义、命令功能和操作方法, 包含有全部命令语法和用法。

例如: • HELP

屏幕上显示 HELP 系统的主菜单, 用户可以按菜单上的提示逐步进行选择。这样可以通过人机对话了解 dBASE III 全部命令。

命令格式 2 HELP 〈命令关键字〉

命令功能 用于帮助用户学习需要的命令和函数的使用方法。

例如: • HELP STORE

屏幕上出现了有关 STORE 命令的语法及其功能的描述。

例如: • HELP LIST

屏幕上出现了有关 LIST 命令的语法和功能。

二、自动检错功能

dBASE III 提供很强的自动检错功能, 当用户输入的命令不符合语法规则时, dBASE III 向用户

提示命令语法出错,给出错误信息,并显示出错的语句,同时用“?”符号标识出出错的位置。然后,系统询问用户是否需要帮助。如果用户对命令语法不清楚,需要帮助,则选择“Y”,否则回答“N”。键入“Y”以后,系统进入到 HELP 模块,并且给出了该命令的详细信息。键入“N”以后,系统又回到了圆点提示符,此时用户需要再重新键入命令。

第八节 CONFIG.DB 文件的使用

在 dBASE III 中,可以使用一个与 DOS 操作系统中相同的 CONFIG.SYS 文件和一个批处理文件 CONFIG.DB 文件,它与 AUTOEXEC.BAT 文件相类似。CONFIG.DB 文件中的内容具有一些特殊格式的命令行。当 dBASE III 启动时,系统将自动查找 CONFIG.DB 文件,若能找到,就执行 CONFIG.DB 中的命令,执行完了以后,回到了命令状态;若找不到 CONFIG.DB 文件,则系统就按照既定的值进行初始化 dBASE III,然后再进入命令状态。

CONFIG.DB 是标准的文本文件,可用任意字处理或编辑程序建立。CONFIG.DB 文件的内容主要包括六个方面。

一、设置功能键

命令格式 <功能键名称>=<命令>[,<命令>...]

命令功能 设置功能键。

例如:F2= CLEAR

当按 F2 键时,就相当于键入 CLEAR 命令进行清屏。

二、设置 dBASE III 系统的最大内存

命令格式 MAXMEM=<数值>

命令功能 设置 dBASE III 系统的最大内存。

其中<数值>必须是整数,其值的范围为 200K~720K。

例如:MAXMEM=640K

该命令使系统给 dBASE 留下了 640K 的内存空间供用户使用。

三、设置 SET 命令的 ON/OFF 状态

例如:TALK=OFF

该命令使 TALK 为关状态,进入 dBASE III 就自动地执行 SET TALK OFF。

四、规定 dBASE III 使用的编辑方式

命令格式 TEDIT=<程序名>

WP=<程序名>

命令功能 规定 dBASE III 使用哪种编辑方式。

TEDIT 规定了 MODIFY COMMAND 命令所用的编辑程序;WP 规定了建立或修改记忆性字段所使用的编辑程序。

五、规定内存变量可占用的最大内存空间容量

命令格式 MVARISZ=<数值>

命令功能 扩大内存变量所允许使用的内存空间数目。

dBASE Ⅱ 允许使用的内存变量为 256 个,允许占用的空间为 6000 字节。在实际编程中有些内存变量要求占用字节数较多,变量个数虽然未达到 256 个,但总的占用内存量超过了 6000 字节,这时用户需要在使用 dBASE Ⅱ 之前用这个命令定义扩大的内存空间数目。

六、希望进入 dBASE Ⅱ 后,立即执行的命令

命令格式 COMMAND=〈命令〉

命令功能 执行列出的命令

例如:COMMAND= CLEAR

系统进入 dBASE Ⅱ 时,就自动执行 CLEAR。

下面列举一下 CONFIG. DB 的内容:

DEFAULT=C

DECIMAL=2

COLOR=3

F10=DISP

MAXMEM=512K

MVARsiz=10K

TEDIT=EDLIN

WP=WPS

COMMAND=DO LIN. PRG

当用户进入 dBASE Ⅱ 以后,执行了 CONFIG. DB 文件,使系统配置为:指定当前工作盘为 C 盘;小数点后 2 位;指定屏幕颜色为蓝色;F10 功能键定义为 DISPLAY;最大可用内存为 512K;内存变量所使用的容量为 10K;编辑或修改命令文件用 EDLIN 行编辑;建立和修改记忆型字段用 WPS 编辑程序。自启动文件为 LIN. PRG 程序。

第九节 dBASE Ⅱ 调试程序时常见错误

整个程序的错误分两大类:逻辑错误和语法错误。

一、逻辑错误

逻辑错误很复杂,概括起来有下面几种。

1. 建立数学模型错误。由于采用的算法不合适或者错误地理解了管理模式中的数学模型,或者有特殊需要处理的地方被忽略了,因此调程序时整个结果不正确。
2. 数据引用错误。例如引用了没有赋值的内存变量,或者引用了错误的数据。
3. 统计结果中数值型数据出现上溢或者下溢,这是由于取值的数据过大或者过小而导致的后果。
4. 相同类型的数据项比较出错。一般出现在数值型数据的等值比较上,这是由于二进制运算,十进制显示的误差而引起。所以选用数值型数据比较时要注意。
5. 控制流程时错误。这种情况是程序编写时缺了 ENDIF, ENDCASE 或 ENDDO。特别在多重循环时容易产生。

6. 模块连接错误,有时一个模块单独运行时正确,但连接起来容易出错。一般在全程变量的引用时要注意,局部变量不能作为全程变量。全程变量必须在各个层次模块中都有定义。还有时错误地调用了与当前入口无关的参数。

二、语法错误

一般用户应首先将程序清单调出,打印,逐项查找。

dBASE III 有自动查错功能,会在屏幕上把错误显示出来,所以很容易排除。

例如:

.....

```
STORE  x="ABCD"
```

```
STORE  y=123
```

```
?  x+y
```

.....

当运行第三句会自动显示出错信息:

数据类型不匹配

这就是由于 x 为字符型常量,y 为数值型常量,这两种不同类型的数据不能相加。

第二章 数据库文件的建立

本章主要介绍数据库文件的建立、数据库文件中字段的定义、数据的添加、文件的打开、数据库文件记录的显示、字段的显示、数据库文件的关闭等等。

第一节 如何进入 dBASE III 状态

一、安装操作系统(PC-DOS,MS-DOS)

DOS 操作系统是微型机的磁盘操作系统,它由四个部分组成,第一部分是引导部分,它负责每次机器启动时自动装入到内存,并由引导程序负责将 DOS 的其它部分装入内存中。第二部分是 IBMBIO.COM 文件,它负责提供内存与外部设备之间的数据传送。第三部分是 IBMDOS.COM,它包含了一个文件管理程序和一系列的实用程序。第四部分是 COMMAND.COM,它是命令处理程序,负责接收键盘发来的命令并运行相应的实用程序。

首先必须准备好一张无病毒的 DOS 系统软盘,即包含上述四个文件的软盘。在格式化硬盘和软盘时加上 /S,表示将上述系统中隐含文件即引导程序装入硬盘或软盘中。

命令格式 C>FORMAT A:/S 或者

C>FORMAT C:/S

命令功能 在格式化时将操作系统中的隐含文件装入盘中。

命令格式 COPY [原盘符:]*. * [目的盘:]

命令功能 首先用 COPY 命令将 COMMAND.COM 文件装入格式化后的盘上,然后再将系统盘上的文件全部拷贝到用户所使用的系统盘中。

最后由用户自己拷贝的盘进行启动,如果正常,则几分钟后出现了盘符提示符。如 C>,或 A>,一般在出现盘符前首先出现输入日期和时间,按二次回车即承认系统中设置的日期和时间,紧接着出现系统提示符 C>,或 A>,表示操作系统按装完毕。

如果日期和时间不对,用户可自己重新输入新日期和新时间。

有些操作系统在屏幕上提示信息:

THE IBM PERSONAL COMPUTER

DOS VERSION 3.0

COPRING IBM CORP 1982,1983

input new date:

input new time:

A>

二、安装 dBASE III

计算机装上操作系统后,就可以将带有 dBASE III 的软盘装到机器中,但是要注意必须先安装

中文操作系统,才能使用汉化后的 dBASE Ⅱ。

命令格式 DISKCOPY A: *. * C:

命令功能 将 A 盘中装入的 dBASE Ⅱ 的文件全部装入到用户的硬盘 C 上。有时用户的文件很多,如有图形软件、查病毒软件、工具软件等等,这时必须先建立一个 dBASE 子目录,然后再拷贝到目录上。

操作步骤如下:

第一步:建立 dBASE 子目录。

命令格式 C>MKDIR DBASE

命令功能 表示用 MKDIR 命令建立一个子目录,即为 DBASE 目录

第二步:拷贝到 dBASE 目录中。

命令格式 CD DBASE

DISKCOPY A: *. * C:

命令功能 先用 CD(改变目录)命令进入 dBASE 目录,再将带有 dBASE Ⅱ 文件的盘插入驱动器 A 中,然后键入拷贝命令。如果 dBASE Ⅱ 是二张盘则装完第一张后,再装入第二张盘。当二张盘的内容全部装好后,用 C>DIR 命令列目录,查看所需要的文件是否全部装好,如果全部装好则表示 dBASE Ⅱ 全部装完。然后用任意种编辑命令编辑 CONFIG.SYS 文件,使 FILES=20,BUFFERS=24。

下面介绍用 EDLIN 命令来编辑 CONFIG.SYS 文件。

三、文本编辑命令 EDLIN

命令格式 EDLIN [驱动器名:] [路程] (文件名, 扩展名)

命令功能 编辑一个文本文件,它可用于对源文件(如用 dBASE Ⅱ 语言编的程序)的建立、修改、删除、插入、查找与文字替换等。它用来对指定的文件进行编辑。如果你是首次对该文件编辑,就必须在磁盘上建立新文件,否则就是对磁盘上已经存在的文件进行修改。

前面提到装入 dBASE Ⅱ 后,需对 CONFIG.SYS 文件进行修改。此时用户可用 EDLIN 命令来编辑该文件以满足用户的需要。

例如:C>EDLIN CONFIG.SYS

```
1 * FILES=20
2 * BUFFERS=24
3 * ^ C
```

* e

这个例子是编辑 CONFIG.SYS 文件,让文件数等于 20,缓冲区为 24。到了第三行按 CTRL + C 之后出现提示符 *,再键入一个 e 表示把该文件存盘,再接一个回车表示结束 EDLIN 编辑命令。

因为 DOS 系统下只允许同时打开 8 个文件,而 DOS 本身已经占用了 5 个文件,那么留给用户只有三个文件,这给用户运行 dBASE Ⅱ 带来很多不便。这时用户必须用 EDLIN 命令来修改 CONFIG.SYS 文件,改善运行环境,使系统允许同时打开的文件个数最大为 20 个,系统同时开辟的文件缓冲区个数最多为 24 个。如果内存容量扩大,则设置的数字可以再增加。

四、运行 dBASE Ⅱ

用户把装有 dBASE Ⅱ 内容的盘片全部装到硬盘以后,再用 EDLIN 命令设置好 CONFIG.SYS

文件,然后再装入汉字的操作系统。

汉字的操作系统种类很多,像 CCDOS,UCDOS,XSDOS 等等,不论用户使用那种汉字操作系统,都必须在硬盘上装入汉字操作系统后才能运行中文的 dBASE Ⅱ。

在所有这些准备工作完成以后,用户才能运行 dBASE Ⅱ,此时要查看一下拷贝的文件。

在用户的硬盘中查看了有 dBASE. EXE 及 dBASE. OVR 文件后,键入下述命令。

命令格式 C>dBASE

紧接着在屏幕上有二行信息:

COPYRIGHT(C) 1983

dBASE Ⅱ 3.0 版本

命令功能 dBASE Ⅱ 全部装入后,启动 dBASE Ⅱ,即把 dBASE Ⅱ 从盘上装入到内存。显示的二行信息是签到信息,第一行是表示 1983 年拷贝来的,第二行是表示 dBASE Ⅱ 的版本是 3.0。

显示完信息后,在屏幕左上侧出现一个圆点·,这是 dBASE Ⅱ 的提示符。它表示该计算机已经在 dBASE 的状态下,可以使用 dBASE 的命令来操作数据。也可以用 dBASE Ⅱ 提供的命令语句进行各种操作,或者执行 dBASE Ⅱ 语句所编制的数据库文件的应用程序。

在向 dBASE Ⅱ 输入命令时,一般只要求前四个字符即可,但是必须是正确的,而且前四个字符必须次序正确,如果不熟练,按照书中命令全部输入也可以。初学者要特别注意 dBASE Ⅱ 的圆点提示符·。如果命令正确,则执行;如果命令有错误,dBASE Ⅱ 会立即指出,并询问用户是否要改正。若回答 Y,则表示用户要改正,这时屏幕上先提示错误的命令,紧接着问用户怎么改,如果仍没改对,则会立刻提示是否要帮助,如果要帮助,会进入到帮助信息,并显示出正确命令格式,用户可参考正确命令键入。一旦命令改正好后,该语句会立即被执行。如果你不要帮助则回答 N,或者你的错误命令不要求改正也回答 N,那么错误的语句就自动作废。你可以重新输入命令。

例 2-1 修改错误命令的过程

· LLST

在屏幕上会有如下显示:

不认识的命令。

校正和重试吗?(Y/N)Y

从哪改变:LL

改变到哪:LI

LIST

还要进一步修改吗?(Y/N)N

本例中输入的是 LIST 命令(显示记录内容),但是 LIST 误写为 LLST 了,那么 dBASE Ⅱ 提示了错误信息,并等待用户改正,接着询问要不要改正,如要修改则等用户改正后,屏幕上出现了正确的命令,接着又问要不要继续改正?回答了 N,表示不继续修改,于是该命令立即执行。命令执行结果如下:

记录号	学号	姓名	性别	高考总分
1	7865321	李静	女	492
2	7637627	余清	男	512
3	7886563	刘红	男	557
4	7536786	胡建	男	562
5	7436765	张莉	女	620
6	7712121	陈和	男	606

这里是假定打开了一个高考成绩单数据库(命令以后讲),然后用修改后的 LIST 命令再来查看结果。该文件假定只有 6 个记录,4 个字段,它们分别为学号、姓名、性别、高考总分。这就是命令修改后的结果。有关打开数据库命令和显示数据库命令在后面介绍。

第二节 数据库文件的建立

数据库管理系统 dBASE Ⅲ 的主要处理对象是以表格形式组织起来的数据,它们以文件的形式存放在磁盘中,称作数据库文件。因此一个数据库文件就相当于一张表格,如表 1-1 中所表示的工资表(GZB)。一个数据库文件至少包含三个部分:表的名称、表的格式及表的内容。dBASE Ⅲ 在处理这些表格时,要定义文件名,这个文件名是和表的名称相对应,可以用 dBASE Ⅲ 的命令来定义;同时还要定义文件结构,文件结构就是表的格式(也叫数据结构或记录结构),但是表的格式在计算机中是不能以直观形式给出的,它只能以一种描述格式的数据给出。在描述时,对表格中每一栏均定义为字段(也称区段、场或域等);同时还要定义字段名、字段宽度及字段类型,文件内容就是表格中所添入的数据。表格的每一行叫文件的一个记录,每个记录在文件中都有一个编号,该编号在文件中叫记录号,它表示该记录在文件中的位置。如表 2-1 所示,该表的名称叫做 GZB,表的格式为 4 行 14 列,第一列是记录号,是系统添加的,文件内容是 3 个记录(除表头外的三行内容)。在该书中大部分例子都是以 GZB 为例。

表 2-1 GZB(工资表)

记录号	工号	姓名	页号	基本工资	浮动	岗位工资	年功工资	副补	食补	粮差	扣病假	扣事假	洗理	实发
1	1	刘宏	10501	56.00	105.00	40.00	20.00	0.00	20.00	12.00	10.00	0.00	9.00	259.50
2	2	李莉	10501	56.00	52.00	28.00	15.00	7.50	20.00	12.00	0.00	0.00	9.00	199.50
3	3	曹西	10502	56.00	51.00	40.00	18.00	7.50	20.00	12.00	15.00	0.00	8.00	197.50

表 2-1 中,实发=基本工资+浮动+岗位工资+年功工资+副补+食补+粮差-扣病假-扣事假+洗理。不难看出,文件内容是由每个记录的数据所组成。只有在表的格式确定以后,才能输入文件内容。而格式是建立数据库时首先需要定义和说明的。所以定义一个文件的名称和定义文件的格式是建立数据库文件的关键。

在 dBASE Ⅲ 中建立数据库文件有两种方式。一种是直接建立数据库文件,另一种是间接建立数据库文件。

一、直接建立数据库文件

直接建立数据库文件,就是直接向 dBASE Ⅲ 定义该文件的名称和文件结构。

命令格式 CREATE [[盘符]:]<数据库文件名>

命令功能 建立数据库文件,如果定义时该格式命令的可选项[盘符:]没定义,则认为是当前默认盘的值,数据库文件名若没定义,则屏幕上提示信息为:

输入文件名:

即要求用户在冒号后面添上文件名称,然后按回车键。这样数据库文件就建立了。注意:文件名不能超过 8 个字符(即四个汉字),必须是字母开头,不能有冒号和空格。文件名建立以后,就可以用 DIR 命令查一下刚刚建立的数据库文件是否存在。如果有该文件名,则其扩展名一定为 .DBF。

文件名建立以后,紧接着需要建立数据,这时用户可按照屏幕上显示的拟建数据库结构信息—

项一项地做。

屏幕上显示的信息:

剩下的字节数:3909

定义的字段数:14

字段名	类型	宽度	小数
1 工号	Numeric	4	0
2 姓名	Char/text	6	
3 页号	Char/text	5	
4 基本工资	Numeric	8	2
5 浮动	Numeric	6	2
6 岗位工资	Numeric	6	2
7 年功工资	Numeric	6	2
8 副补	Numeric	6	2
9 食补	Numeric	6	2
10 粮差	Numeric	5	2
11 扣病假	Numeric	5	2
12 扣事假	Numeric	6	2
13 洗理	Numeric	5	2
14 实发	Numeric	8	2

字段名以汉字或字母开头,后面可以是汉字、字母、数字或下划线

该信息说明了让用户按给定的次序输入记录结构,也就是数据结构,其中包括字段名、字段类型、字段宽度和十进制小数点位置。下面说明这几项数据填写的要求。

1. 字段名

字段名相当于表格中每一栏的标题,最多只允许 10 个字符(即 5 个汉字),开头必须是字母,中间允许有冒号,但不允许有空格。

如果规定的这些要求,用户没有作到,则会要求用户重新键入字段名。

例:字段名输入了空格,当你输完后,回车时,系统不承认,并提示重新输入文件名。在表中屏幕左边的数字 1 是 dBASE II 系统自动给出的字段序号,长方框表示屏幕上的反相高亮度显示部分。此时光标停在欲输入字段名的第一个字符的位置上,等待用户键入数据库结构中的字段名。如果你不知道咋错的,系统还提供询问是否要帮助?如下面一个数据结构中第 7 项键入的文件名为空格,则屏幕提示修改数据结构,重新输入第 7 个字段的字段名。英文提示是(MODIFY STRUC DB06, Field7)

字段名	类型	宽度	小数
1 工号	Character	4	0
2 姓名	Character	8	0
3 晋级分	Numeric	8	2
4 年结转	Numeric	8	2
5 合计分	Numeric	8	2
6 累计分	Numeric	8	2
7	Character	0	0

修改后的文件结构的字段名应该是输入奖励分,见下面修改。

```
修改数据结构(C:\) \DB06          |          | 字段:7/7          |          |
                                     |          | 输入字段名          |          |
```