

第一章 汉字 FoxBASE+ 概述

§ 1.1 数据库的基本概念

汉字 FoxBASE+ 是一种关系型数据库管理软件,凡装有该软件的计算机,都可以接受用户输入的汉字 FoxBASE+ 的命令或由这些命令编写的程序,以便完成数据处理过程。在学习汉字 FoxBASE+ 之前,应该先搞清楚与汉字 FoxBASE+ 有关的以下概念即:数据库、数据工作区和数据库别名、记录指针、数据类型和种类、运算符和表达式、函数、命令和程序、文件、命令执行状态和程序运行方式、系统配置等。

为了叙述简便,本书把汉字 FoxBASE+ 简称为 FoxBASE+, 即两种称呼均为同一概念。

一、数据库

数据库是结构化的数据集合,是根据数据的性质、内在联系,对所有相关数据实行统一的、集中的、独立管理的要求来设计和组织的。

数据库有三种类型模型,即关系型模型、层次型模型、网络型模型,一般将以上三种类型模型简称为关系型、层次型、网络型。在微机上使用最多的是关系型数据库,如 dBASE II、Informix、UNify 和本书介绍的 FoxBASE+。

二、关系型数据库

关系型数据库又称关系数据库,它的模型概念及操作运算方法简述如后。

关系型数据模型中,所有的数据都用二维表的形式来组织,如表 1.1 职工登记表就是一个二维表。该表就是一个关系,每一个关系均由同类记录组成。如表 1.1 中就有 5 个同类记录。不同类型记录不能在同一关系中,但不同关系中,可以有相同的属性。表 1.1 中就有 6 个属性,如职工编号、职工姓名、工作部门等。

表 1.1 职工登记表

职工编号	职工姓名	工作部门	职 务	职 称	工 资
126	王大力	计财处	科 员	经济师	110.00
076	李一云	行政处	办事员	经济员	98.00
128	杨 元	计财处	科 长	经济师	160.50
054	朱大江	办公室	主 任	工程师	160.50
100	鹿 龙	计财处	科 员	会计师	160.50

在关系内,把一个记录行表示成元组,每一元组集中反映了某特性,把列表示成属性,反映元组的某种性质。在表 1.1 中,有 5 个元组,反映了 5 名职工的情况,每一元组又有 6 列,反映每一职工的 6 个属性。

关系数据模型的主要特点表现在数据描述的统一性,描述的数据与数据之间的联系只能用关系来表示。关系必须是规范化的关系,它的每一列是一个不可分割的数据项,也就是说不

允许表中再有表。

关系数据库的操作方法有其严格的数据基础。作为一个关系数据库管理系统,一般可以给用户提供对数据库数据的查询、插入、删除和更新等四种基本操作,其中,查询不会引起数据库中数据的变化,其余三种操作都会使其数据库中数据发生改变。

关系数据库管理系统对数据库的操作按关系的运算可分为两大类:(1)从行的角度进行传统的集合运算,包括并运算、交运算和差运算;(2)从列的角度进行专门的关系运算,包括选择运算、投影运算、联结运算和除运算。下面分别介绍两类常用的操作运算。

1. 传统的集合运算

传统的集合运算用在两个同类的关系中,即两个关系的度数(属性的个数)相同。设关系 A 和关系 B 具有相等的属性个数 n, 关系相容可并, 则可定义如下三种运算操作。

(1) 并运算。关系 A 和关系 B 的并记为:

$$A \cup B = [t | t \in A \vee t \in B]$$

其中: t 为元组变量, $\in$ 表示属于, $\vee$ 表示逻辑和。 $A \cup B$ 的结果仍为 n 个属性的关系。

(2) 交运算。关系 A 和关系 B 的交记为:

$$A \cap B = [t | t \in A \wedge t \in B]$$

其中: $\wedge$ 表示逻辑与。 $A \cap B$ 的结果仍为 n 个属性的关系, 由既属于 A 又属于 B 的元组组成。

(3) 差运算。关系 A 和关系 B 的差记为:

$$A - B = [t | t \in A \wedge t \notin B]$$

其中 $\notin$ 表示不属于。 $A - B$ 的结果仍为 n 个属性的关系, 并由属于 A 而不属于 B 的元组组成。

综上所述, 并、交、差的运算可由图 1.1 来说明, (a)、(b) 是 A 和 B 两个关系, (c) 表示并运算, (d) 表示交运算, (e) 表示差运算。

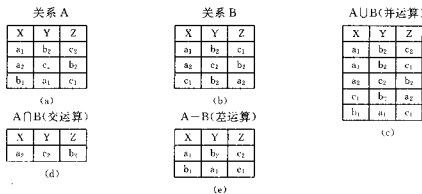


图 1.1

2. 专门的关系运算

专门的关系运算包括有: 选择运算、投影运算、联结运算。

(1) 选择运算。选择运算是按照某种条件在给定的关系中选取若干元组, 得到一个属性保持不变的另一个关系。选择是通过条件来判断的。条件的关系运算符有 $>$ (大于)、 $<$ (小于)、 $=$ (等于) 等 6 种, 选择的逻辑运算符有 NOT (非)、AND (与)、OR (或) 3 种。将单个的条件和关系运算符组成一个复合选择条件, 即可完成各种选择运算。例如表 1.1 中, 选出计财处中工资在 150 元以上的职工, 则可通过条件复合选择运算来实现:

工作部门="计财处".AND. 工资>150
就可产生出一个新的关系如表 1.2 所示。

表 1.2 选择运算后产生的职工登记表

职工编号	职工姓名	工作部门	职 务	职 称	工 资
128	杨 元	计财处	科长	经济师	160.50
100	鹿 彪	计财处	科员	会计师	160.50

(2)投影运算。 投影运算是对一个关系中某些属性的选择,也就是对表示关系的二维表的某些列进行操作。投影运算的结果将会产生一个新的关系,即一个新的二维表。该关系只会包含投影运算中指定的那些属性,并与原来关系中的元组数目相同,也就是原关系中的行数不变,列数变化。例如表 1.1 中,对职工姓名、职务及工资三个属性进行投影运算,则得到一个新的关系如表 1.3 所示。

表 1.3 投影运算产生的职工登记表

职工姓名	职 务	工 资
王大力	科 员	110.00
李一云	办事员	98.00
杨 元	科 长	160.50
朱大江	主 任	160.50
鹿 彪	科 员	160.50

(3)联接运算。 联接运算是将两个关系联结成一个关系的操作。被联接的两个关系必须具有一个以上的公共属性。否则不能进行联接运算。在通常情况下,联接按照两个关系中公共属性值相等的条件进行操作,产生一个新的关系,这个新关系包含了两个被联接的关系的全部属性。例如,将关系 R1 和 R2 进行联接运算后产生新关系 S。关系 R1,关系 R2 及新关系如图 1.2 所示。

关系 R1			关系 R2			关系 S				
A	B	C	B	D	E	A	B	C	D	E
a	c	d	e	a	r	a	c	d	a	r
d	b	f	b	f	h	d	b	f	f	h
e	e	a	a	d	b					

图 1.2 联接运算产生关系 S 的示意

在图 1.2 中是按 B 属性(用 * 号表示)进行联接的。关系 R1 和 R2 中属性相同的值是 c 和 b,因而联接的结果产生的新关系 S 同样在 B 属性中包含了 c 和 b,且只能包含 c 和 b,不能含有其它不相同的属性的值。

以上讨论的关系数据库的常用算法和操作的过程是系统内部进行的,用户不须对它们进行安排,用户只须向数据库管理系统提出数据组织管理要求,数据库管理系统便会自动对数据库进行运算操作,以实现用户的要求。

关系数据库的存储是以一个关系即一个文件为单位存储在磁介质上的。一个数据库文件中,记录数即元组数,属性的多少即字段数。字段是以它的名称、类型、宽度来定义的,因此,关系数据库的结构由字段的多少来确定。一个数据库可称为一个数据库文件或数据库,又可称库文件。

关系数据库与另外两类数据库比较,有数据表示能力强、使用方便、易于实现数据之间的

管理等优点。

三、数据工作区和数据别名

要使用数据库应该将它事先打开,在 FoxBASE+ 中可以同时打开和使用 10 个数据库文件和 21 个索引文件。为了在同时使用多个库文件时,互不干扰,识别方便,在 FoxBASE+ 中采用了不同的数据工作区来分隔它们。由于允许同时打开 10 个库文件所以设有 10 个数据工作区,每个工作区允许打开 1 个库文件。数据工作区即工作区用编号 1~10 或名称 A~J 的单个字母表示。例如,1 工作区的名称是 A,2 工作区的名称是 B,等等。

FoxBASE+ 虽然可以同时使用多个数据工作区,但在任一时刻只有一个工作区是主要的,这个主要工作区即称为当前工作区。在 FoxBASE+ 中,大多数数据库的操作都是针对当前数据库进行的。需要同时使用其它工作区中的库文件的数据时,必须在该库名称前冠以所在工作区的编号或名称或数据别名。由于别名是用户任意指定的,且与所在工作区没有必然的联系,可以用工作区名来标识非当前工作区的数据库,很容易实现数据与操作功能的独立。

例如,有一数据库文件别名是 BM,则无论它在那个工作区被打开,使用 SELECT BM 就可以选择该文件所在的工作区为当前工作区。

四、记录指针

一个数据库文件中一般有多个记录,在操作数据库时,总是逐个记录进行的。为了表明操作的当前记录,FoxBASE+ 为每个数据工作区分别设置一个记录指针,来标识各工作区中的当前记录。记录指针可以前后移动,如工作区中不打开数据库就不存在当前记录,记录指针的值为 0。如果记录指针指向文件尾时,记录指针值应比该文件的最大记录号大 1,如果记录指针指向文件头时,指针值为 1。

五、数据类型

数据库中信息的表达形式是数据,数据按其构造、处理方式、用途等属性,分为若干不同的类型。FoxBASE+ 的数据类型有 6 种,即数值型(N 型)、字符型(C 型)、日期型(D 型)、逻辑型(L 型)、备注型(M 型)、屏幕型(S 型)。此外,FoxBASE+ 中还有数组,数组并非一个独立的数据类型,其中的各单元可有自己的数据类型。

在数据库的操作中,只有相同类型的数据才能进行运算或操作,不同类型的数据通过类型转换成一致后,也可以进行运算或操作。

六、数据种类

FoxBASE+ 的数据有三种,即常量、变量、函数。常量就是固定不变的数据,它有数值型、字符型、日期型、逻辑型之分。变量是内容可以变化的数据,它有字段变量、内存变量之分。字段变量即数据库字段,内存变量又有标量变量和数组变量。字段变量及内存变量也有多种类型,只有相同类型的变量才能互相传递,参与数据的操作。

七、运算符与表达式

运算符是表示数据之间运算方式的符号。在 FoxBASE+ 中,运算方式有:算术运算、字符串运算、关系运算、逻辑运算,由此而对应的运算符也有 4 种,即算术运算符、字符串运算符、关

系运算符及逻辑运算符。

在以上 4 种运算方式中,算术运算的对象是数值型数据,运算的结果也仍然是数值型数据。算术运算符在算式中也有运算的优先级别之分,按级别高低从高到低进行运算。字符串运算是把两个字符串型数据连接起来,形成一个新的字符串。关系运算是用来比较两个数的大小或前后,它只能在数值型数据、字符串型数据、日期型数据之间进行。逻辑运算只能对逻辑型数据进行,它的运算结果仍然是逻辑型数据。逻辑运算符也有运算的优先级别,并从高到低运算。

表达式是数据之间运算关系的表达形式,可以由常量、字段变量、内存变量、函数等和运算符组成。单个数据也可视为表达式的一个特例。表达式在 FoxBASE+ 的各类数据操作和运算中十分重要,正确合理地应用表达式,可收到事半功倍的效果。

八、函数

函数是一种不同于常量和变量的特殊数据,它的值是由其它数据通过某种特定的操作后得到的,每使用一次函数就进行一次数据操作。FoxBASE+ 中有近百个内部函数及用户自行定义的自定义函数。

FoxBASE+ 的内部函数按其功能或用途可分为 7 类,即字符处理函数、数值运算函数、日期函数、转换函数、标识函数、测试函数及输入函数。

FoxBASE+ 函数的表现形式除宏替换函数外,一般是在函数名后跟一对圆括号,圆括号内给出函数的自变量,自变量就是函数的操作数,其个数、类型、值域均有规定。

九、命令及程序

在 FoxBASE+ 中,用以完成数据库操作、程序控制、系统状态、参数设置等功能,都是由 FoxBASE+ 的各类命令或通过命令调用函数来实现的。因此,在 FoxBASE+ 中用了较大的篇幅来描述它们。

FoxBASE+ 命令一般由英语动词和限定该命令动词的若干短语或子句组成。命令动词完要要进行的操作,短语或子句用以说明该操作的对象、范围、条件等。有些命令由于操作的对象、范围、条件是固定的,也可以不带短语或子句。短语或子句一般由功能短语、表达式等构成。

FoxBASE+ 命令可以由用户在计算机键盘上输入执行,也可以将命令编成程序即编成命令文件在计算机中执行。

FoxBASE+ 程序是用户把要计算机解决及处理的问题的逻辑思维过程,用一系列有关的 FoxBASE+ 命令来表达的过程。FoxBASE+ 程序一般又称为命令文件。计算机在对 FoxBASE+ 程序进行执行时,一般是解释一句执行一句,并把命令编译成可执行代码,然后执行。代码执行后并不消除,仍然保留在内存的程序缓冲区里,如重复执行该程序时即可直接执行已有的代码,加快了程序的运行速度。

FoxBASE+ 的程序也可以对它进行准编译,生成的目标程序不能单独运行,必须由 FoxBASE+ 执行,但具有程序装载快、内容保密、长度减小等优点。

FoxBASE+ 的程序逻辑结构一般有顺序结构、分支结构、循环结构。FoxBASE+ 的程序按其用途和构造分为一般程序、过程、屏幕格式及用户定义函数。FoxBASE+ 还可以使用汇编语言子程序或其它语言编制的可以独立执行的程序。

十、文件

FoxBASE+的各类数据和程序都是以文件形式存储在磁盘上的。FoxBASE+共设置了9种类型文件,即数据库文件、备注文件、索引文件、内存文件、格式文件、标签文件、报表格式文件、命令文件、文本文件。其中应用最多的是数据库文件和命令文件。

十一、命令执行状态及程序的运行

FoxBASE+的命令需执行时,有两种执行状态,即一种是由用户通过计算机的键盘输入一条命令,FoxBASE+就执行一条命令并将执行结果反馈给用户。另一种方法是将被命令编成程序来运行,只有当该程序全部运行完毕后,用户才能得到执行结果。第一种执行是圆点提示符状态,即点状态。第二种是程序运行状态。

以上两种FoxBASE+的命令执行状态,使FoxBASE+的运行有多种方式。但较常用的有三种:在点状态下使用FoxBASE+的命令或函数;在操作系统下直接执行程序;对编译过的FoxBASE+程序在运行版本下执行。以上三种方式的详细操作,将在后面有关章节中说明。

十二、系统配置文件

为了使FoxBASE+更好地适应用户的环境和需要,优化FoxBASE+的性能,可以由用户对FoxBASE+的系统状态、特性、参数进行设置。设置方法一般有:一种是在点状态下或程序运行中用有关命令如SET、SET ON/OFF等命令来设置;另一种是在FoxBASE+启动时通过系统配置文件CONFIG.FX来设置,前一种设置方法比较灵活,后一种设置若要改变,应退出FoxBASE+以便修改CONFIG.FX后再重新启动系统。但对内存变量最多个数、文件打开个数等,只能通过后一种方法来设置。许多读者在运行FoxBASE+的程序时,往往会出现许多系统设置上的问题就是因为CONFIG.FX设置不当而产生的。

§ 1.2 汉字FoxBASE+的特点及技术指标

一、主要特点

FoxBASE+是美国Fox公司开发的先进关系数据库系统,它保留了新颖关系数据库dBASE III的全部优点,除有较强的数据管理功能和灵活的程序设计能力外,还在dBASE III的性能和功能上有了许多扩充和重要发展。例如具有更快的运行速度,FoxBASE+比dBASE III快6~7倍;FoxBASE+可以同时打开48个文件,而dBASE III最多只能打开16个文件;FoxBASE+的内存变量可达3600个,而dBASE III的内存变量最多为256个;在FoxBASE+中可以使用一维或二维内存变量数组,可以使复杂的程序编制简单化,并大大提高计算速度,但在dBASE III中却不能使用一维或二维内存变量数组;FoxBASE+除改进了dBASE III的许多命令功能外,还增加了数十个新的命令和函数。此外,FoxBASE+还具有过程文件组合、命令文件(即用户编制的应用程序)编译等一系列显著的优点。

汉字FoxBASE+是由FoxBASE+的多用户版本汉化开发而成,保留了FoxBASE+的全部性能和功能,既可在单用户环境中运行,也可在网络软件支持下多用户工作。FoxBASE+在CCDOS、XENIX等多种操作系统环境下适用于IBM PC XT、AT、286、386、486及其兼容机如

长城、东海、浪潮、联想等多种机型。

用汉字 dBASE III 编写的程序,用户无需改动即可在汉字 FoxBASE+ 环境下运行。

由于 FoxBASE+ 的优秀性能,向读者介绍和推广汉字 FoxBASE+ 必将在办公自动化、企事业单位管理、财政金融等经济领域的数据处理中起到非常积极的作用。

二、主要技术指标

1. 数据库文件

记录个数 ≤ 10 亿个

字节个数 ≤ 20 亿个

记录大小 ≤ 4000 个字节

逻辑字段 ≤ 1 个字节

备注字段 $\leq 64K$ 字节

2. 字段宽度

数值字段 ≤ 19 个字节

字符字段 ≤ 254 个字节

日期字段 ≤ 8 个字节

逻辑字段 ≤ 1 个字节

备注字段 $\leq 64K$ 字节

3. 内存变量

内存变量个数:默认 256 个,最多 3600 个

字符串变量可用字节数:默认 6K 字节,最多 64K 字节

其它变量可用字节数:受可用内存大小限制。

每个数组的单元数 ≤ 3600 个

4. 数值精度

16 位有效数字

5. 可使用的最大最小正数

可使用的最大正数 1×10^{38} , 可使用的最小正数 1×10^{-307}

6. 文件操作

可同时打开的文件个数 默认 16 个,最多 48 个

可同时打开的库文件 ≤ 10 个

可同时打开的索引文件每个库文件最多可以打开 7 个索引文件,一共可以打开 21 个索引文件。

7. 命令行

命令行长度 ≤ 254 个字节。

索引(INDEX)的关键表达式的长度 ≤ 100 个字节。

索引关键表达式的长度 ≤ 180 个字节

关联(RELATION)的关键表达式的长度 ≤ 60 个字节。

过滤器(FILTER)中条件表达式长度 ≤ 160 个字节。

每个命令行中的 &(宏替换)次数 ≤ 256 次。

8. 程序结构

DO 命令嵌套层数 ≤ 24 层。

程序结构嵌套层数 IF、WHILE、CASE 等 ≤ 64 层。

一个命令文件中变量个数 ≤ 2048 个。

过程文件中子过程个数 ≤ 128 个。

屏幕格式文件中屏幕分页数 ≤ 128 页。

§ 1.3 系统配置与运行环境

一、系统配置

FoxBASE+有多种版本,如 1.10、2.00、2.10 等。以 FoxBASE+2.00 版为例有两张 5.25" 软盘。

其中 1 号盘上有:	2 号盘上有:
MFoxPLUS.EXE	MFoxPLUS.OVL
FoxPCOMP.EXE	FoxPHELP.HLP
FoxBIND.EXE	EUROPEAN.MEN

FoxBASE+的主要部分是执行程序 MFoxPLUS.EXE 和复盖程序 MFoxPLUS.OVL。执行程序启动 FoxBASE+后驻留内存,复盖程序在 FoxBASE+启动时只有一部分装入内存,其余在需要时以复盖方式调入内存,使用后又被新的内容所复盖。

FoxBASE+的辅助部分有帮助文件 FoxPHELP.HLP,过程组合程序 FoxBIND.EXE,汇编编译程序 FoxPCOMP.EXE 和欧洲文字排序基准文件 EUROPEAN.MEN 等。

FoxBASE+还有一个只能运行编译目标模块文件的特殊版本 FoxBASE RUNtime。它由三个文件组成,一个是系统运行文件 FoxPRUN.EXE(单用户),或 MFoxPRUN.EXE(多用户),另两个是复盖文件 FoxPRUN.OVL 和 FoxPRUN.OV2。要注意在使用 FoxBASE RUNtime 时,有部分 FoxBASE+的功能不能得到支持。如 CREATE/MODIFY REPORT、CREATE/MODIFY LABEL、MENU、SET STEP ON、BROWSE 等。

二、运行环境

1. 单用户 DOS 系统下

一台 IBM PC/XT 或其兼容机,内存容量应大于 360K 字节;

一个或二个软盘驱动器,一个 10MB 以上的硬盘驱动器;

一台 24 针的行式打印机或类似打印机,一台显示器;

操作系统为 DOS 2.0 以上各种版本,如 CCDOS 2.13H 等。

2. 在多用户 XENIX 系统下

一台 IBM PC XT/AT 286、386、486 或兼容机,内存容量应大于 1MB,每增加一个用户应多加 0.4MB 的内存空间;

二个软盘驱动器,一个 20MB 以上的硬盘驱动器;

一台 24 针行式打印机或类似打印机,一台彩色显示器;

操作系统为汉化 XENIX V 系统;

最好有一个 80287 或 80387 数字协处理器,以提高用户程序运行速度。

§ 1.4 汉字 FoxBASE + 命令文件的编译、过程文件及运行

一、汉字 FoxBASE + 命令文件的编译

汉字 FoxBASE + 命令文件即软件开发人员或用户用汉字 FoxBASE + 编写的程序,一般称为命令文件或源程序(文件扩展名是 .PRG),命令文件可以在汉字 FoxBASE + 系统下直接运行,但如果在 FoxBASE + 系统下先对命令文件进行编译,然后再运行则会更好些。

编译过的文件从磁盘装入内存时,其速度就如同读入他们时一样快。而源程序读入内存的同时就进行编译,一旦装入结束,编译也就结束了。一个给定的程序不管它是从一个源程序装入还是从一个被编译的程序装入,其执行的速度总是相同的。但是,编译过的程序比源程序装入的时间要短得多。因此,编译的程序可以达到最大的装入速度,同时又可检测出任何存在的语法错误。

FoxBASE + 的源程序在 DOS 或 XENIX 下通过以下命令进行编译:

```
foxpcomp <源程序名 1>[<源程序名 2>...<源程序名 N>]
```

编译命令的功能是将扩展名 .PRG 的源程序编译为扩展名为 .FOX 的目标程序,但目标文件名与源文件名相同。源程序文件扩展名 .PRG 不被默认,因此,在指定文件名的同时必须给出其扩展名。源文件名及扩展名均可以使用通配符“*”和“?”,这样可以简便地操作编译多个程序,而只需使用一条命令。例如所有名为“RS”开头的人事档案管理程序的编译可用下面一条命令来实现:

```
foxpcomp RS *.PRG
```

从编译结果来看,FoxPCOMP 是一个伪编译程序(又称准编译程序),没有生成在操作系统下执行的代码程序(扩展名为 .EXE)文件,而是产生了一个执行速度很快且在解释程序下运行的特殊代码文件。

用 FoxPCOMP 对源程序进行编译时,不管编译的源程序是否有错误,总会将目标程序(扩展名 .FOX)文件输出,并给出运行错误信息,以便用户修改源程序后重新进行编译。

foxpcomp 编译命令也可以用于将 .FMT 文件编译成目标 .FMX 格式文件。

例 1.1

```
foxpcomp mulh.prg
foxpcomp    *.prg
foxpcomp    RS *.prg  GZ *.prg
foxpcomp    *.fmt
```

应当注意的是:如果存在有编译好的 .fox 文件,FoxBASE + 总是执行它。如果某一程序不是使用 MODIFY COMMAND 命令来修改,则修改后的程序不会被运行,除非该程序再被编译,或者 .fox 文件被删除了。

当一个程序由 MODIFY COMMAND 命令进行修改后,需调用编译命令重新编译,这样,原来相应的 .fox 文件将被新的 .fox 文件覆盖,从而保证 DOS 命令能够运行最新的 .fox 文件。

除了在操作系统下完成上述的源程序编译外,还可以在操作系统下将源程序文件加密后放到指定的驱动器上去。操作命令及命令的使用如下:

命令格式如下:

FoxPCOMP -e -o<驱动器名或文件路径><文件名 1>[<文件名 2>]...

命令格式中的 -e 项用于指定按加密方式进行编译; -o<驱动器名或文件路径>项用于指定编译后的目标文件路径; <文件名 1>... 是被编译文件的路径和名称, 不默认文件扩展名, 可以含有通配符“?”和“*”。

例 1.2 在根目录下将 A 盘上的全部命令文件, 加密编译后放到 C 盘中去。

```
C>FoxPCOMP -e -oc: A: *.PRG
```

例 1.3 将 C 驱动器硬盘中子目录 GZ 下的 GZ123.PRG 加密编译后放到 A 驱动器软盘的子目录 GZ 中去。

```
c>FoxPCOMP -e -OA:GZ C:\GZ\GZ123.PRG
```

二、汉字 FoxBASE+ 过程文件的建立

FoxBASE+ 允许在一个过程文件中调用多达 128 个不同的程序。通过将多个同类的应用程序组合成一个过程文件, 然后用 foxpcomp 进行编译, 并且使用 SET PROCEDURE TO 语句来选择过程文件, 这样可以减少和磁盘打交道的次数, 提高程序的运行速度。

开发调试和维护一个应用程序的过程中, 总是要维护一系列各自独立的 .prg 文件, 于是 FoxBASE+ 提供了一个实用程序——foxbind, 使过程文件的维护非常方便。

该实用程序可以迅速地将多个 .prg 文件组合成过程文件, 其组合的方法是将若干 .prg 程序一个接一个地拷贝到一个输出文件中, 每一个程序前都加上一个注释说明过程, 如下例:

```
*****
*      progname      *
*****
```

PROCEDURE progname

foxbind 的使用方法非常简便, 在 DOS 或 XENIX 下, 直接打入命令:

foxbind<过程文件名><源程序名 1>[<源程序名 2>]...<源程序名 N>]

<过程文件名>是命令执行后生成的过程名称, 扩展名可以指定也可以不指定, 缺省扩展名为 .prg。

如果给出的过程文件名已经存在, 则将这些程序追加到已存在的过程文件之中。

在源程序名中, .prg 扩展名不被默认, 而必须明显地给出来, 源程序名可以包含通配符“*”和“?”。

例如: 假定将人事档案系统的所有输入程序组成一个过程文件, 并且各个程序都以“RSIN”开头, 这时可以用下面一条命令生成过程文件, 其过程名为: “RSPROCE.PRG”。

```
foxbind RSPROCE RSIN *.PRG
```

过程文件建成以后, 最好使用 foxpcomp 命令编译所有的过程文件, 以便提高过程文件的运行速度。

如果一个应用系统的整个程序全部被包进一个过程文件中, 则可以除了过程文件名外不需要任何 .prg 或 .fox 文件, 而执行 SET PROCEDURE TO 语句。

在将一个应用程序组合进进一步过程文件时, 应按照以下次序编写。

(1) SET PROCEDURE TO... 语句位于第一个过程说明语句, 其后跟初始化程序 DO 语句。

(2)用过程文件名运行该程序或从圆点“·”状态提示符下执行这一过程。

下面举例说明过程文件的编制和调用：

<在过程文件“RSPROG”中>

SET PROCEDURE TO RSPROG

DO RSINP

PROCEDURE RSINP

：

PROCEDURE RSOUT

：

<在 DOS 或 XENIX 系统下启动程序>

MFoxPLUS RSPROG

或

FoxPLUS RSPROG

在上述两条命令后各应有相应的回车命令以命令输入完毕，请求计算机进行处理。

三、汉字 FoxBASE+ 命令文件的运行

一个 FoxBASE+ 程序的执行有多种方法，下面介绍常用的三种方法。

1. 在 FoxBASE+ 系统下执行

如在 DOS 操作系统下则键入 MFoxPLUS；如在 XENIX 操作系统下则键入 FoxPLUS。一但进入 FoxBASE+ 的环境，即 FoxBASE+ 系统装入完成，在显示屏幕上就会出现圆点“·”提示符，便可在 FoxBASE+ 下执行 FoxBASE+ 的各类命令和函数。

2. 在操作系统下直接执行

在 DOS 或 XENIX 系统下键入：

MFoxPLUS <命令文件名>或 FoxPLUS <命令文件名>即可。

其中命令文件的扩展名不必给出，若命令文件的编译程序(.fox)存在则 FoxBASE+ 将其装入，否则就装入源文件(.prg)。

3. 在运行版本下执行

FoxBASE+ 在运行编译目标模块的版本即 FoxBASE+ Runtime 时仅能运行编译过的程序(扩展名为 .fox)，Runtime 的运行方法是：

MFoxPRUN <命令文件名>

或

FoxPRUN <命令文件名>

也可在 CONFIG.FX/DB 中设置 COMMAND DO<命令文件名>，然后键入 MFox-PRUN 或 FoxPRUN 来启动。

§ 1.5 汉字 FoxBASE+ 命令格式、规则、符号约定及全屏幕操作

一、命令格式

使用 FoxBASE+ 各种命令，或用各种命令编写程序时，必须严格按照规定的命令格式表

达, FoxBASE+ 的命令一般由两部分构成。第一部分是命令动词, 位于命令行之首, 第二部分是动词短语(或称子句), 位于命令动词之后, 在一个命令行中可以有若干个动词短语, 动词短语是为了对所要执行的命令作某些限制性说明。下面以 DISPLAY 命令格式为例进行说明。

DISPLAY [<范围>][FIELDS<字段名表>][FOR/WHILE<表达式>][OFF][TO PRINT]

其中 DISPLAY 是命令中的命令动词, 用来表示要完成的任务, 而其余部分则是动词短语, 用来提供命令执行时的各种参数或命令执行范围。DISPLAY 命令用以完成显示任务。如命令仅用 DISPLAY 表示时, 即不指明显示范围时, 显示记录的当前一行内容; 如果用 DISPLAY ALL 表示时, 即指明显示范围为 ALL 时, 将显示当前内容的全部记录。

命令格式中出现的中文部分如范围、字段名表、表达式等是由用户选择自定的参数。格式中出现的 DISPLAY、FIELDS、FOR/WHILE、OFF、TO PRINT 等是命令中格式决定的, 用户不能随意改动, 一般称为命令关键字。

命令格式中出现的符号 < > 表示该部分内容如选用该命令时, 必须出现在命令中, 称为命令的必选部分。命令格式中出现的符号 [] 表示该部分内容在选用该命令时可以出现在命令中, 也可以不出现在命令中, 称为命令的可选部分。如 DISPLAY 命令中, 除命令动词 DISPLAY 外, 其余均为可选部分, 即都是可选动词短语。

在一条命令中如有必选部分而又缺少必选部分时, 即缺少必选短语时, FoxBASE+ 会判定该命令出现语法错误。

二、命令规则

命令也称语句, 讨论命令规则就是讨论语法规则。因此, 在书写每条命令时, 都必须严格按照规定的命令格式来书写。

(1) FoxBASE+ 的命令名都是英文动词, 它必须是命令行的第一个非空白字符。

(2) 命令行中的短语的先后位置顺序可以是任意的。例如 DISPLAY OFF FOR<表达式> 与 DISPLAY FOR<表达式>OFF 两条命令的作用完全相同。

(3) 一条命令只能写一行, 当命令行很长时, 超过显示屏上一行时, 可用分号(;)加上回车键操作在行尾分行, 然后在下一行继续输该命令的剩余部分。命令行的总长度不能超过 254 个字符。

(4) 在命令中, 命令名与短语, 短语与短语之间要用一个以上空格分隔。

(5) 命令中必须包括命令名, 命令格式中方括号中的内容是任选项, 可由用户根据具体情况指定。

(6) 命令行中的命令名或关键字, 可以缩写为四个字符或更多。例如 DISPLAY STRUCTURE 和 DISP STRU 或 DISPL. STRUCT 的作用完全一样。

(7) 命令可以用大写或小写字母书写。

三、命令格式中的符号约定

主要的符号的约定规则如下:

(1) < >: 必选参数表示符, 表示用户根据具体使用要求填入的内容。

(2) []: 可选参数表示符, 用户可根据具体情况确定选用或不选用。

(3) <范围>: 表示对库文件进行筛选的记录范围, FoxBASE+ 用下面的格式来定义:

ALL 表示对库文件全部记录进行操作。
 NEXT n 表示只对当前记录以下的 n 个记录进行操作。
 RECORD n 表示仅对第 n 个记录进行操作。
 REST 表示从当前记录开始直到文件结束进行操作。

(4) <表达式>：是指任何一种表达式，如逻辑的、数字的、字符的、条件的都可以。

<逻辑表达式>：只能是逻辑型表达式，逻辑表达式用于程序分支判断时又称条件。

<数值表达式>：只能是数值型表达式，数值表达式也称为数字表达式。

(5) <文件名>：指任何一种 FoxBASE+ 的文件。

(6) <变量>：表示内存变量或字段名。

<内存变量>：专指内存变量。

(7) <表达式表>：表示用逗号“,”分开的若干表达式。

<内存变量表>：表示用逗号“,”分开的若干内存变量。

<字段名表>：表示用逗号“,”分开的若干字段名。

(8) <命令组>：表示一组有序列的命令。

(9) <字符串>：表示字符型常量。

(10) FOR <逻辑表达式>：表示只对满足 <逻辑表达式> 条件的记录进行操作。

(11) WHILE <逻辑表达式>：表示 <逻辑表达式> 条件为真时，才进行当前循环体操作。

(12) 小写字母：指由用户提供的部分。

(13) 符号 < > , [] 无须在命令中写出或键入计算机中。

四、全屏幕操作

全屏幕操作又称全屏幕编辑，就是允许在计算机显示器的全屏幕范围内将光标移动到任一位置，并可以在光标指向处直接用控制键或一般的 ASCII 码键进行增加、删除、修改等操作，大大方便了用户。

在 FoxBASE+ 中的许多命令都要涉及全屏幕操作。例如，CREATE、APPEND、EDIT、BROWSE、CHANGE、INSERT、MODIFY、READ 等命令。

在全屏幕操作方式下，数据以一定的格式出现，供你填入或修改。数据项通常是反象显示，箭头键用于把光标设置在任意的位置上，数据项是在标识的数据区中键入完成的，达到规定长度时，机器响铃并被接受，十分方便。

下面列出了与全屏幕命令一起使用的键：

键	功能
→/^ D/^ L	光标移到右面一个字符
←/^ S	光标移到左面一个字符
END/^ F	光标移到右面一个词组
Home/^ A	光标移到左面一个词组
↵/^ B	光标移到行尾
↶/^ Z	光标移到行头
↓/^ J/^ X	光标移到下一行
↑/^ E/^ K	光标移到上一行
Return/^ M	如果 INSERT 是 ON 则插入一行，否则移到下一行的开始

PgDn/^ C	光标移到下一页
PgUp/^ R	光标移到上一页
^ KE	光标移到文件尾
^ KT	光标移到文件头
Ins/^ V	插入/覆盖状态的转换开关
^ N	插入一个新行
Del/^ G	删去光标处一个字符
Backspace/^ H	删去光标左面一个字符
^ T	删去光标处一个词组
^ Y	删去光标所在行
^ KB	重新格式一个段落
^ KF	搜索指定的字符串,并将光标移到第一个发现的串上
^ KL	继续按指定字符串向下搜索,将光标移到下个发现的串上
^ KR	将其它文件读入当前文件,并插在当前光标所在位置
^ KW	将当前文件写到另外的文件
F1	光标控制键的说明菜单的显示开关
ESC/^ Q `	退出编辑,并且不保存本次所做的修改
^ End/^ W	退出编辑,并且存储本次所有修改结果

其中符号/是等价键的分隔符,^ 即表 Ctrl 键,^ → 即表 ^ 键与 → 键两键同时键入(又可表示为 Ctrl→)。

与全屏幕操作有关的单一命令操作的具体说明将在后面有关章节中介绍。

第二章 汉字 FoxBASE + 基础

汉字 FoxBASE+ 基础主要介绍 FoxBASE+ 的各种文件类型和功能、文件的基本操作,文件中的数据及分类以及主要的函数的定义及使用等。

§ 2.1 文件类型与功能

FoxBASE+ 系统将有用的数据和处理问题编制的程序以文件的形式存储到磁盘中,并对文件进行各种操作和管理。

一、文件名

为了对文件进行操作和管理,每个文件都有一个名字。FoxBASE+ 文件名的格式为:

文件名. 扩展名

文件名长度不能超过 8 个字符(如字母、数字、其它字符),在汉字 FoxBASE+ 中允许直接用汉字作文件名。

扩展名由一个圆点和三个字符组成。文件名及扩展名中均不能包括空格。

文件名是在建立文件时由用户指定的,同时指定文件的扩展名。

二、文件类型

FoxBASE+ 将数据以九种不同的格式存储在磁盘文件中,每一种格式对应一种文件类型,它服务于特定的 FoxBASE+ 处理的需要。为了区分各类文件, FoxBASE+ 为各种文件规定了不同的扩展名。下面给出九类文件及其扩展名:

文件类型	扩展名	文件类型	扩展名
数据库文件	.dbf	标签文件	.lbl
数据库备注文件	.dbt	报表格式文件	.frm
索引文件	.idx	命令文件(未编译的)	.prg
内存文件	.men	命令文件(已编译的)	.fox
格式文件(未经编译的)	.fmt	文本文件	.txt
格式文件(已经编译的)	.fmx		

三、文件功能

1. 数据库文件(.dbf)

数据库文件(又称库文件)是 FoxBASE+ 中最基础而又最重要的文件。它用来存储数据库基本数据。数据库文件以记录和字段(或称行和列)的形式存储数据,每个记录中存放一组信息。FoxBASE+ 数据库文件最多达 10 亿个记录,每个记录最多包括 4000 字符,分别存储于最多 128 个字段中。可由 CREATE 命令建立,并可由 MODIFY STRUCTURE 命令修改数据库结构。

2. 数据库备注文件(.dbt)

数据库备注文件(又名备注文件)是数据库文件的辅助文件,用于存储备注字段的内容。如果数据库文件中定义有备注型字段,那么在建立数据库文件的同时,也就建立了相应的数据库备注文件。但它们在磁盘上是单独分开存储的。

3. 索引文件(.idx)

这是一个快速数据操作的辅助文件。索引文件是对数据库文件的记录,是按索引关键字的值排序后建立的辅助文件。

索引文件把关键字与对应的数据库记录联系在一起。当数据库与一个索引文件一起使用时,数据库的排列看上去好像是按关键字项目的顺序排列的。该关键字可以由一个或多个字段组成,用以对特定记录提供直接存取。

索引文件由 INDEX 命令建立。

4. 内存文件(.mem)

内存文件又名内存变量文件,用于存储内存变量的内容以备将来使用,同时可提高内存空间的使用效率。由 SAVE 命令建立该文件,由 RESTORE 读回到内存中。

5. 格式文件(.fmt)

格式文件又名屏幕格式文件,用于建立屏幕显示格式以便数据的录入和输出,它可以提高数据输入输出或处理的质量和效率,是专供 READ 命令使用的文件。

格式文件也可以进行准编译,编译后的目标文件与源文件同名,但扩展名为 .fmx 执行时编译过的优先于未编译过的。

6. 标签文件(.lbl)

标签文件又名标签格式文件,用于规定标签或名片的输出项目和格式。FoxBASE+具有编辑和调用格式文件的专用命令,使用很方便。

7. 报表格式文件(.frm)

报表格式文件是 FoxBASE+ 的辅助文件之一,用于规定数据报表的输出格式。该文件一般含有报表的标题、数据内容、类小计、合计,以及打印格式等几部分。

8. 命令文件(.prg)

命令文件是用户编制的 FoxBASE+ 的应用程序,是 FoxBASE+ 的主要文件。它是数据库系统进行数据处理,实现各种管理任务的主文件。命令文件是一种 ASCII 文本文件,可以用 FoxBASE+ 的文本编辑命令进行编辑,也可以用其它任何一种字处理软件进行编辑。FoxBASE+ 的命令文件分为一般程序、过程、用户定义函数等几种。

命令文件可以进行准编译,编译后的目标文件与源文件同名,但扩展名改为 .Fox,目标文件不能单独执行,还得在 FoxBASE+ 的环境中执行。

9. 文本文件(.txt)

文本文件多用于和其它语言或应用软件的数据通讯。它只包含可打印的 ASCII 字符。

§ 2.2 文件的更名、删除、复制与文件目录的查询

一、文件更名

有时,我们需要更改磁盘上文件的名字。例如,磁盘上有一命令文件由于某种原因受到了

破坏,但是磁盘上还有一个扩展名为 .BAK 的备份文件,这时只须用更名命令就可以将备份文件更名为命令文件。

命令格式:

RENAME<原文件名> TO <新文件名>

命令功能:

把磁盘上名为原文件名的文件更名为新文件名的文件。

例 2.1 将文件 AA.DBF 更名为 BB.DBF 文件。

• RENAME AA.DBF TO BB.DBF

执行以上更名命令后,磁盘上的 AA.DBF 文件已改名为 BB.DBF 文件,AA.DBF 文件已不存在。

使用 RENAME 命令时的注意事项:(1)命令中的新文件名不得与同一磁盘上已有的文件名相同。(2)命令中的原文件名的文件应已经建立,但不能是打开的。

二、文件删除

不再需要的文件可以从磁盘上把它删除,以便腾出磁盘空间,供存放其它信息。

命令格式:

ERASE <文件名>或 DELETE FILE <文件名>

命令功能:

从磁盘上删除用户指定的任何一种文件,命令中的文件名应包括扩展名在内。

例 2.2 将磁盘上的文件 BB.DBF 删除。

• ERASE BB.DBF

三、文件复制

命令格式:

COPY FILE <源文件名> TO <目标文件名>

命令功能:

用于复制整个文件,文件的类型不限。<源文件名>和<目标文件名>必须标以全称,不能使用通配符“?”和“*”。复制后得到的目标文件与源文件完全相同。本命令不能把备注型字段的内容进行复制,必须再把相应的备注文件(.DBT 文件)也用本命令复制。

四、文件目录的查询

在要对某个文件操作之前,往往需要检查一下磁盘上是否存储着所需要的文件,如果有,就可以继续操作下去;如果没有,就要采取相应的措施:建立、拷贝或更换磁盘等等。查看磁盘目录的命令有三个:DIR、DISPLAY FILES 和 LIST FILES。

1. DIR 命令

命令格式:

DIR [驱动器:] [路径] [框架] [TO PRINT]

命令功能:

DIR 命令显示当前工作磁盘、当前文件目录上所有数据库文件的名字、每个文件的记录数、文件的最后修改日期以及文件的字节数。