

最新FoxBASE+2.10

教程

甘登岱 王 强 编著
友 元 审

航空工业出版社

(京)新登字161号

内 容 提 要

本书全面系统地介绍了运用 FoxBASE+2.10 进行程序设计的基础知识。全书共分九章, 主要内容包括: FoxBASE+ 概述, 数据库基本操作, FoxBASE+ 函数, FoxBASE+ 结构程序设计, FoxBASE+ 输入输出设计, FoxBASE+ 与其他语言的接口, FoxBASE+ 高级编程技巧。本书在附录部分汇集了所有 FoxBASE+2.10 命令与函数, 并给出了 FoxBASE+2.10 错误信息一览表。此外, 为了便于读者学习, 作者还精心设计了大量的编程实例。

本书不同于一般的使用手册, 注重编程与实例, 集知识性与技术性于一体, 是一本不可多得的参考书。本书适用于大专院校师生、培训班师生及数据库管理应用软件开发人员。

最新 FoxBASE+2.10 教程

甘登岱 王强 编著

友 元 审

航空工业出版社出版发行

(北京市安定门外小关东里14号)

—邮政编码: 100029—

全国各地新华书店经售

通县向阳印刷厂印刷

1993年10月第1版

1993年10月第1次印刷

开本: 787×1092 1/16

印张: 26

印数: 1—4500

字数: 639 千字

ISBN 7-80046-553-5/TP·038

定 价: 19.50元

序 言

随着微机管理信息系统的推广与普及,关系型数据库系统越来越受到广大用户的欢迎。FoxBASE+是1986年6月由美国Fox软件公司推出的关系型数据库管理系统,它与目前国内流行的dBASE III Plus数据库完全兼容,但在性能上却大大优于dBASE III Plus。这首先表现在,FoxBASE+的速度平均比dBASE III和dBASE III Plus快6倍,而比Clipper要快2.3倍。其次,FoxBASE+不仅和dBASE III和dBASE III Plus完全兼容,而且增加了数十个命令和函数。

此外,由于FoxBASE+是用C语言编写的,因此,它的可移植性很好。它适用于多种硬件环境,如IBM=PC/XT、AT、286、386等。它还可以在多种操作系统下运行,如DOS、XENIX等。而且FoxBASE+的多用户版本和单用户版本百分之百兼容。本书全面系统地介绍了运用FoxBASE+2.10进行程序设计的基础知识。全书共分九章,主要内容包括:FoxBASE+概述,数据库基本操作,FoxBASE+函数,FoxBASE+结构程序设计,FoxBASE+输入输出设计,FoxBASE+与其他语言的接口,FoxBASE+高级编程技巧。本书在附录部分汇集了所有FoxBASE+2.10命令与函数,并给出了FoxBASE+2.10错误信息一览表。此外,为了便于读者学习,作者还精心设计了大量的编程实例。

本书不同于一般的使用手册,注重编程与实例,集知识性与技术性于一体,是一本不可多得的参考书。本书适用于大专院校师生、培训班师生及数据库管理应用软件开发人员。

本书由北京航空航天大学自动控制系304教研室甘登岱和管理系王强同志主编,参加本书编写工作的还有:许建平、刘东、潘磊、林海、张开伦、朱立志、丁建民、刘连生等。本书的录入排版工作由张东凌、张万芹和李燕负责,他们为本书的出版付出了辛勤的劳动,在此对他们表示由衷的感谢。

全书由友元工作室审校,在此对友元工作室的全体同仁表示谢意。友元工作室虽成立时间尚短,但成绩已很显赫,它们开发的通用数据库生成器、经济型字幕机、彩色人像打印系统、工业控制软件等已相继推出。有关这方面的问题,读者如有兴趣的话,可直接与他们联系。他们的通信地址是:100083 北京航空航天大学304信箱 友元收。最后,我们预祝友元工作室今后能推出更多更好的为社会所需的产品。

尽管作者在编撰本书时已竭尽全力,但由于时间仓促,加之水平有限,不当之处在所难免,敬请读者批评指正。

编著者

1993年6月

目 录

第一章	FoxBase+概述	(1)
1.1	FoxBase+的环境要求与限制	(1)
1.2	FoxBase+的安装与启动	(3)
1.3	FoxBASE+系统环境定义	(5)
1.4	将 dBASE III PLUS 转换成 FoxBASE+	(10)
1.5	FoxBASE+数据类型	(12)
1.6	FoxBASE+交互功能	(17)
1.7	过程文件组合	(17)
1.8	命令文件编译	(18)
1.9	FoxBASE 工具集	(18)
1.10	光标控制键	(19)
1.11	名词解释	(25)
第二章	数据库的基本操作	(28)
2.1	建立数据库	(28)
2.2	数据库的输出	(35)
2.3	数据库的更新与维护	(37)
2.4	数据库的分类、索引和查找	(45)
2.5	数据库的运算操作	(55)
2.6	数据库间的操作	(59)
2.7	数据库常规辅助操作	(66)
第三章	表达式、常量、运算符、内存变量、函数操作	(76)
3.1	对表达式的操作	(76)
3.2	对常量的操作	(77)
3.3	对运算符的操作	(77)
3.4	对内存变量的操作	(80)
3.5	内存变量与数据库变量间的操作	(87)
3.6	函数调用的基本规则	(92)
第四章	FoxBASE+函数	(96)
4.1	FoxBASE+的数据库操作函数	(96)
4.2	FoxBASE+环境设置函数	(104)
4.3	数学运算函数	(105)
4.4	字符及字符串操作函数	(108)
4.5	日期运算函数	(116)
4.6	表达式运算函数	(120)
4.7	I/O 控制函数	(122)
4.8	运行错误信息捕获函数	(129)

4.9	SYS 系统信息函数	(130)
第五章	FoxBASE+ 结构程序设计	(136)
5.1	程序文件的建立、修改和运行	(136)
5.2	FoxBASE+ 的赋值语句	(138)
5.3	循环结构程序设计	(140)
5.4	分支结构程序设计	(144)
5.5	FoxBASE+ 的转移语句	(146)
5.6	过程及其调用	(150)
5.7	注释语句	(163)
5.8	错误捕获与键盘侦测	(163)
5.9	程序调试	(170)
5.10	程序文件的编译	(174)
第六章	输入输出设计	(176)
6.1	FoxBASE+ 的交互式命令	(176)
6.2	打印报表	(180)
6.3	打印标签	(183)
6.4	格式设计命令	(185)
6.5	使用格式命令设计屏幕输入格式	(193)
6.6	使用@命令设计报表	(199)
第七章	与其他语言的接口	(204)
7.1	与其他语言的数据交换	(204)
7.2	调用外部程序	(208)
7.3	调用二进制例程	(209)
第八章	FoxBASE+ 高级编程技术	(213)
8.1	系统的安全性设置	(213)
8.2	屏幕菜单制作与技巧	(218)
8.3	数组的建立与使用	(222)
8.4	数据库的查询与技巧	(223)
8.5	编制报表	(231)
8.6	简易绘图	(238)
第九章	多用户 FoxBASE+	(243)
9.1	系统运行环境与基本概念	(243)
9.2	数据库的加锁与解锁	(245)
9.3	锁的测试和处理	(250)
9.4	多用户命令与函数	(254)
9.5	死锁的处理	(261)
9.6	常见的多用户出错信息	(265)
附录 A	FoxBASE+ 2.10 命令基本列表	(267)
附录 B	FoxBASE+ 2.10 函数列表	(348)

第一章 FoxBase+概述

随着计算机及其应用软件走入人们日常事务处理和信息管理工作中,越来越多的事务管理工作离不开数据库管理系统。

在我国,由于 dBASE III 数据库管理系统具有的实用性强、易学易懂等特点,曾被誉为“大众数据库”而得到广泛的推广与应用,在各行各业的管理中得到最广泛的应用,成为我国流行最广,应用面最大的微机数据库管理系统,经过多年的发展与应用,开发了大量的应用程序。但是,由于 dBASE III 也有它固有的缺点,如计算能力弱,处理速度较慢,没有图形功能,没有数组的自定义函数,没有多用户功能等,这些弱点不但限制了 dBASE III 在更多应用领域的发展,而且导致了用 dBASE III 开发的应用项目大多只能停留在单项程序或比较低的水平上,难以满足高水平综合信息管理系统的要求。因此,近年来,不断有新的与 dBASE III 兼容的数据库管理软件问世,如 dBASE III Plus,clipper,FoxBASE 等,这些数据库中,当数美国 Fox 软件公司的 FoxBASE 关系型数据库管理系统最为成功,它不但完全与 dBASE III 兼容,而且又具有功能强,速度快等特点,是我国目前理想的 dBASE III 用户的换档产品。

FoxBASE+2.00,FoxBASE+2.10 三个版本,与其他数据库软件相比,FoxBASE+具有如下优点:

1. 兼容性

FoxBase+与 dBASE III,dBASE III PLUS 完全兼容,dBASE III 程序不作任何修改即可在 FoxBASE+环境中运行。

2. 可移植性

FoxBase 采用 C 语言开发,因此便于移植和推广,可运行于范围很广的操作系统(如 MS-DOS,XENIX,UNIX 等)和硬件环境(IBM-PC 系列兼容机,VAX/780 等),而且 FoxBase+的多用户版本与单用户版本完全兼容,用户的应用软件随着硬件和系统的发展也很容易向高档环境移植。

3. 运行速度快

利用“Data Based Advisor”年提供的 48 个基准检查程序来检验,FoxBase+在平均速度上比 dBASE III PLUS 快 5.9 倍,比 Clipper 快 3.2 倍,如果配置 80287/80387 协处理器,则运行速度可进一步得到提高。

4. 多种运行方式

FoxBase 既可在交互方式下运行,也可在程序方式下运行。在程序方式下既可解释执行,也可通过编译产生目标文件来执行,这使得用户在掌握 FoxBase+的应用、开发和调试 FoxBase 程序、提高运行速度和实现程序的保密性方面都得到了兼顾和提高。

1.1 FoxBase+的环境要求与限制

1.1.1 FoxBase+的环境要求

执行标准的 FoxBase+ 的最小硬件要求是：

(1) 一台支持 FoxBase+ 的计算机，如 IBM-PC/XT、IBM-PC/AT、长城或浪潮 286、386、486 等兼容机。

(2) 至少 360KB(千字节)的内存空间，如果是在汉字操作系统下运行，则至少需要 640KB 的内存空间，当然有更大的内存最好。

(3) 至少配备一个硬盘和一个软盘驱动器，将 FoxBase+ 的系统程序安装在硬盘中。用户开发的应用程序和数据文件则可通过软硬盘驱动器进行备份或读出复制到其他计算机上。

如果在微机上使用 8087、80287 或 80387 协处理器，则更有利于发挥 FoxBase+ 的最大性能。

如果希望在 386 的保护模式下运行 80386 FoxBase+ 的保护模式版本，则需要如下的配置：

- (1) 一台能使用 FoxBase+/386 的 80386 计算机；
- (2) 至少 2MB(兆字节，1MB=1024KB)的内存空间；
- (3) 一个硬盘，一个软盘驱动器；
- (4) 一个 80287/80387 协处理器。

1.1.2 FoxBASE+2.10 软件组成

FoxBASE+ 由下列文件组成：

FoxPLUS. EXE	执行程序
FoxPLUS. OVL	覆盖程序
FoxPHELP. HLP	帮助文件
FoxBIND	过程组合文件
FoxPCOMP	准编译文件
EUROPEAN	欧洲文字排序基准

1.1.3 FoxBase+2.10 的系统限制

1. 数据库文件

每个数据库文件的最大记录个数	1,000,000,000 条
每个数据库文件的字节数	最多 20 亿字节
每个记录的最大长度	4000 字节
每个数据库的最大字段个数	128 个

2. 字段宽度

数值型字段：	最多 19 个字节
字符型字段：	最多 254 个字节
日期型字段：	固定为 8 个字节
逻辑型字段：	固定为 1 个字节
备注型字段：	最多 64K 字节

3. 内存变量

内存变量的缺省个数:	256 项
最大内存变量数:	3600 个
字符串变量可用字节数:	缺省为 6KB, 最多 64KB
内存数组的最大元素数:	3600 个

4. 数值精度

每个数值型数据的有效数字:	最多 16 位
---------------	---------

5. 文件操作

同时打开的文件数:	缺省为 16, 最大 48 个
可同时打开的数据库个数:	最多 10 个
可同时打开的索引文件个数:	每个数据库可同时打开 7 个, 总共可打开 21 个。

6. 命令行

每个命令行的字符数:	最多 254 个
每个索引关键表达式字符个数:	最多 100 个
关键表达式字符数:	最多 180 个
关联的关键表达式的长度:	最多 60 个字节
过滤器中条件表达式字符数:	最多 160 个
每个命令行中 &(宏代换)次数:	最多 256 次

7. 程序结构

DO 命令的嵌套层数:	最多 24 层
IF、WHILE、CASE 嵌套层数:	最多 64 层
一个命令文件中变量个数:	最多 2048 个
过程文件中过程个数:	最多 128 个
屏幕格式文件中屏幕页数:	最多 128 页

8. 内存要求

FoxBase+ 标准版:	至少是用 360KB 内存
FoxBase+ 80386 保护模式版:	至少占用 2MB 内存

9. 操作系统

FoxBase+ 标准版:	MS-DOS 2.0 以上
FoxBase+ 网络版:	MS-DOS 3.1 以上
汉化 FoxBase+	汉化 DOS 操作系统

1.2 FoxBase+ 的安装与启动

1.2.1 FoxBase+ 的安装

在启动 FoxBase+ 运行之前, 必须先将它安装到硬盘中, 可按下列步骤进行 FoxBase+ 的安装:

1. 对 FoxBase 原盘制作一个备份, 并且对安装盘贴上写保护胶条;

2. 确定 FoxBase+ 的安装位置,在硬盘上建立一个子目录,并将该子目录设置为当前工作目录,如:

```
C:\>MD FOX
```

```
C:\>CD FOX
```

```
C:\FOX>
```

3. 将 FoxBase+ 系统盘的 #1 插入 A 驱动器中,然后启动 A 盘中的安装程序 INSTALL,如

```
C:\FOX> A:\INSTALL C:
```

4. 然后,将在屏幕上看到系统安装过程中所有的提示,以及进一步操作的说明,按下任意一个键后就可以继续执行;

5. 在显示一些信息后,屏幕将出现

```
Enter you Serial Number;
```

这时需要键入 FoxBase+ 软件的系列号码;

6. 在系列号码回答后,又出现

```
Enter your activation key;
```

这时需要键入“活动”钥匙,这是指 FoxBase+ 软件提供的字符串;

7. 在上述两步通过后,安装程序将依次提示插入 FoxBase+ 的以后各片盘;

8. 安装成功后屏幕上将显示:

```
FoxBase+Successfully installed.
```

9. 如果在安装中出现问题,安装程序将会提示用户,改正后再重复整个过程。

在结束基本的 FoxBase 系统后,如果还希望一些附加程序,可以按任意键继续。这时将出现一个

FoxBASE+ Optional Software Installation 菜单,用这个菜单就能自动安装全部的任选软件,或者安装其中的一部分,或者退出安装程序。

可供安装的附加程序是:

- FoxDoc ---- FoxBase+ 应用程序文献产生器;
- FoxCentral ---- FoxBase+ 控制中心接口;
- FoxView ---- FoxBase+ 屏幕设计工具和应用程序产生器;
- FoxCode ---- FoxBase+ 模块编译器;
- Templates ---- FoxView/FoxCode 已编译的模板;
- FoxView 演示和教授文件;
- Template source code ---- 源程序级的 FoxView/FoxCode 模板。

为了能自动安装全部任选软件,可用这个菜单进行选择,并按下 <Enter> 键,在自动安装过程中按下 <Esc> 键可随进中断程序的安装过程,并出现 Esc 选择菜单,此菜单内容为: Abort, Continue, Return 三项。

当完成 FoxBase+ 附加程序的安装后,屏幕上将显示:

```
Installation Complete,
```

按任意键后退出安装程序,用户就可准备启动 FoxBase 运行了。

1.2.2 FoxBase+ 的启动

在操作系统提示符下,可键入

FoxPlus 或 MFoxPlus.

如:

C:\Fox>FoxPLus

在执行启动命令后,即可进入 FoxBase+ 的交互环境。如果要执行多任务 FoxBASE+ 的 DOS 版本,则键入命令:

C:>\FOX>MFOXPLUS。

一旦装入 FoxBase+ 环境后,屏幕上出现 FoxBase+ 的圆点提示符,于是就可以开始工作了。

1.3 FoxBASE+ 系统环境定义

1.3.1 缓冲区

所谓磁盘缓冲区,是指内存的一块区域。DOS 使用这块区域来存储输入或从磁盘输出的数据。

当 DOS 被启动时,磁盘缓冲区即被设置。IMB-PC/XT 缺省磁盘缓冲区数目为 2 而 IBM-PC/AT 缺省磁盘缓冲区数目为 3,除非修改了配置文件(CONFIG.SYS)中的 BUFFER 个数,否则 DOS 的缺省值是不变的。

缓冲区是计算机用来暂时存储读出或写入磁盘数据的内存区。一般情况下,磁盘上每个扇区的容量为 512 字节,而每个记录的长度为 70 字节,因此,每一扇区将含有大约 7 个记录。在 FoxBASE+ 系统开始打开数据库文件时,FoxBASE+ 会立即要求操作系统从磁盘中读出一个或数个扇区的记录到内存,并且还将记录指针指向第一个记录,当要寻找另一个记录时,FoxBASE+ 便会先在内存中寻找符合的记录,若没有找到,则 FoxBASE+ 便立即从磁盘其他区域寻找。

如果磁盘缓冲区的数目比缺省值小,则可能会有不良影响。因为如果寻找数据记录次数较多,则磁盘驱动器的读写头移动次数频繁,无形中增加了存取时间;反过来,如果将磁盘缓冲区的数目设置得太大,则是不经济的,因此,FoxBASE+ 提供的缺省值为 40。

DOS 中文件数的最小值为 8,这个值对一般操作系统环境而言,已经足够,然而在大多数 dBASE 与 FoxBASE 系统中,缺省值为 8 却为它带来不少困难,当 DOS 启动时有三个文件相继被装入,即 IO.SYS、MSDOS.SYS、COMMAND.COM 等,继之又启动 MFOXPRUN,如果你想查看数据库,而又启动了 FOXPRUN.OVL,但这时所打开的数据库文件又附带打开另外的索引文件,则文件的计数一直累加,可用的文件数很快就被用光了,因此如前面所提到的,如果内存的容量不够或是系统启动磁盘上的 CONFIG.SYS 文件有误,则文件的操作会因为没有空间存放设置而导致发生:

Too many files variable.....

假如上述情况发生,请将 CONFIG.SYS 中的 FILES 及 BUFFERS 重新调整或者在

CONFIG.FX 中调整 MVCOUNT 值。

以下各部分将描述这个简单但是却有弹性的配置文件,它可以允许用户在开始时修改内存的分配,而且改变启动 SET 指令系统内定值。

1.3.2 CONFIG.SYS

使用 FoxBASE+时,无论你使用的计算机属于哪一类 IBM-PC 及其兼容机,其基本启动条件都需要 560K 的内存,这些内存用来容纳所有任务,以便让程序顺利执行。一旦这些要求都存入内存之后,剩下的空间才可用于容纳用户程序。由此可知,最佳的使用条件为 PC 上至少需要有 640K 以上的内存容量,假如现有计算机内存容量不够的话,那么请增加内存容量。

如果上述问题不存在,也就是说,你已拥有一台内存容量达 640K 以上的计算机,但仍可能碰上另一奇特的现象,即出现如下提示:

Insufficient memory

上述提示意味着用户所使用的各类文件个数已超出上限,但其上限值是多少呢?上述现象是许多 dBASE 以及 FoxBASE+用户都会碰上的,为了避免这种现象产生,可以修改系统启动磁盘中的 CONFIG.SYS 文件内容,另一种方式则是购买或自行编写使用 RAMDISK 的方法。

要改变 CONFIG.SYS 文件的内容,实际上并没有改变 FoxBASE+由磁盘上取回数据的方式,而所改变的只是 FoxBASE+需要到磁盘上寻找数据的次数及增加可同时打开的文件和缓冲区的数量。

1.3.3 CONFIG.FX

在硬盘成功地安装 FoxBASE+之后,你可能发现有必要调整内存以达到所需的缺省数值。有一个详细、有用的设备可用来调整 FoxBASE+的内存使用量。

然而,在刚开始时,你可能不需要这份设备的全部功能。因此我们提供一些不同的标准配置文件,从这些文件中,你可以选择起始配置。

FoxBASE+系统本身所设置的系统配置需要用到 560K 字节的内存空间来执行程序。若 FoxBASE+在开始时找不到 CONFIG.FX 或 CONFIG.DB 文件,它就直接选用系统内定的配置。

若有一些少量的可用内存,那么必须调整 FoxBASE+配置以尽量避免浪费其内存使用空间。

FoxBASE+允许你通过一个简单但是有弹性的方法配置内存分配以及所启动的 SET 命令的系统内定值,这只需建立一个名叫“CONFIG.FX”的正文文件或是与 dBASE III PLUS 兼容的“CONFIG.DB”就可以实现。

当 FoxBASE+执行时,它就自动地在当前的工作目录中搜索 CONFIG.FX 文件。若在当前的工作目录中无法发现,那么 FoxBASE+就会由 MS-DOS“PATH”环境字符串中的目录搜索 CONFIG.FX 文件,请参阅 MS-DOS 技术手册中有关如何设置 PATH 的描述。

若搜索完成后,CONFIG.FX 仍没有发现,那么 FoxBASE+就会重复这个过程以搜索另一个称为 CONFIG.DB 的文件。

若 CONFIG.FX 文件或 CONFIG.DB 文件都无法被找到,那么就要使用内定的系统配置。

请注意,若希望使用系统内定的配置,那么在当前工作目录中建立一个空的文件,将其命

名为 CONFIG. FX, 这样可以让 FoxBASE+ 快一点“开始”, 因为它不再需要在目录中搜索合适的文件。

由于现存的 dBASE III PLUS 中的 CONFIG. DB 文件都可以直接使用而不必进行改变, 因此, 若遇到一个无效的命令行, 那么 FoxBASE+ 将不会报告有错误, 取而代之的方法是这一行被忽略。被指定到数据文件的配置选择数据都会被检查, 以确定每一个项目的数值是否在可允许的数值范围内。对于不正确的数值都会自动地被改变到接近的可允许的数值。若提示的字符串长度超过 20 个字符, 那么它将被截断。

请注意, 在 FoxBASE + 1. XX 版本中还有以下三个用于内存分配的配置项, 但在 FoxBASE + 2. XX 版本中由于实现了内存的动态管理, 这三个配置项已无意义, 即使在 CONFIG. FX 文件中定义了也不起作用。这三个配置项为:

(1) BUFFERS

该值为 I/O 缓冲区的个数。每个缓冲区占 2K 字节, 其范围为 4—31, 系统缺省值为 31。

(2) MAXMEN

该值为共用内存的容量。值的单位为 1K, 值的范围为 8—64, 系统缺省值为 64。

(3) PCACHE

程序缓冲区的容量。值的单位为 1K, 值的范围为 8—64, 系统缺省值为 64。

对于提供少于 560K 字节内存的机器, 只要减小这些项目就允许执行 FoxBASE+。BUFFERS 以及 MAXMEN 将明显地影响系统的速度。这三个参数 BUFFERS, MAXMEM 以及 PCACHE 通常是决定 FoxBASE+ 内存需要的主要因素, 若它们都设置到最大的数值, 那么 FoxBASE+ 的工作效率最高; 当它们的最大数值被减少时, 工作效率也会相应降低。

然而, 改变系统配置非常简单。因此, 最好的方法可能是试验各种配置直到确定一个可以在机器上执行程序的最佳配置为止。

CONFIG 文件中含有如下格式的一行或多行:

<inem> = <value>

在 CONFIG. FX/DB 文件中配置项中, 大部分为各种 SET... TO 或 SET... ON/OFF 命令, 它们也可在启动后通过 SET、SET... TO 或 SET... ON/OFF 命令设置。但有些配置项只能在 CONFIG. FX/DB 中设置。下面我们分别对这些配置项予以介绍。

CONFIG 文件可以用任何正文编辑器建立, 包括与 FoxBASE+ 编辑器相似的产品。

1.3.4 系统配置的分配项目

以下列出有关配置及其意义:

COMMAND = <命令>

FoxBASE+ 启动时立即自动执行的任意 FoxBASE+ 命令。如果 FoxBASE+ 以 MFOXPLUS <文件名> 的形式启动, 则启动后忽略 COMMAND 命令, 而立即执行 <文件名> 所指定的命令文件。

BUCKET = <常数>

该参数为用于存储 @... GET 命令 (包括 PICTURE、FUNCTION、RANGE、VALID 等选择项) 的区域的总量, 以 K 字节表示。存储 @... GET 所需要的空间可以从 32 到 302 字节, 它根据 PICTURE 以及 RANGE (若有的话) 的设置而定。该值的单位为 2K, 其数值在 1 到 32 之间

改变,系统缺省值是 4(即 8K)。

BUFFERS=〈常数〉

这是输入/输出缓冲区的数目,每个缓冲区占 2K 字节。BUFFERS 的设置可以控制 FoxBASE+ 的输入/输出速度,BUFFERS 愈大愈好。然而,BUFFERS 越大,意味着用户程序可用内存越小,因此除非是非常大的数据库或数据库使用了很多记录,否则 BUFFERS 应设置一个比较合适的数值,最好在 10 到 20 之间。BUFFERS 可以在 4 到 31 范围变化,系统的内定值是 31。

FILES=〈常数〉

该参数用于设置执行 FoxBASE+ 时可以同时打开的文件数目。值的范围为 16—48,系统缺省值为 16。超过 16 个时,每增加一个文件,内存开销就增加 130 字节。

在 MS-DOS 的 CONFIG.SYS 文件中,FILES 子句内设置的数目必须至少比本设置项大 10。

HISTORY=〈常数〉

该参数决定 DISPLAY/LIST HISTORY 可以显示的命令个数,即保存在命令历史表 HISTORY 中的命令个数。然而,存储命令的实际数目通常受 HMEMORY 设置的内存总量限制。这个数值可以从 0 到 16000 的范围之间变化。系统的内定值是 20。

HMEMORY=〈常数〉

该参数设置用来存储 FoxBASE+ 已执行过命令的历史表的总量,单位为 1K 字节,值的范围为 0—63,系统缺省值为 5。每存储一条命令,所需要的空间是它的长度加 8 字节。

INDEX=〈索引文件扩展名〉

该参数用于定义 FoxBASE+ 默认的索引文件扩展名。系统的缺省值为 ".IDX",如果希望 FoxBASE+ 的索引文件扩展名与 dBASE III 的索引文件扩展名相同,可以把该项设置为 INDEX=NDX。

MAXMEM=〈常数〉

该参数为共用内存的容量,单位为 1K 字节。这是可供文件处理器使用的内存总量,而且控制可用 MODIFY COMMAND 命令编辑的文件的大小。此外, SORT、INDEX、REPORT 命令以及其他许多函数可以使用这个区域作暂时存储;这个区域的大小将决定 FoxBASE+ 在执行 SORT 以及 INDEX 指令时使用何种策略,因而会显著影响其执行速度。这个数值可以在 8 到 64 的范围内变化,系统的内定值是 64。

MVARSIZ=〈常数〉

该参数为用于存储字符型内存变量的区域的总量,以 1K 字节为单位。每存储一个字符型变量,在该区域占用的字节数为 7+字符串长度。这个数值可从 1 到 64 的范围之间变化。系统的内定值是 6。

MVCOUNT=〈常数〉

内存变量的最大个数,这个数值可以从 128 到 3600 范围之间变化,系统的内定值是 256。

PCACHE=〈常数〉

该参数表示程序缓冲区的容量,以 1K 字节表示。该数值为分配给编辑程序的程序缓冲区的容量,它也可以用来表示任何单独的程序最大的空间,必要时,可以将该数值设置得很小,然而,如果该值被设置的太小,那么 FoxBASE+ 的工作效率将会大大降低。这个数值可以在 8 到

64 的范围之间变化,系统的内定值是 64。

PROMPT=<提示符>

这个项目允许用户修改 FoxBASE+ 的提示符。系统内定的提示符是一个圆点(.)后接一个空格。该参数为一长度不超过 20 的字符串,如果超过 20 的话,FoxBASE+ 启动时自动截去 20 位以后的字符串。

TEDIT=<字处理或文本编辑程序名>

设置一个执行 MODIFY COMMAND 指令时使用的外部字处理或文本编辑程序。使用本命令后,MODIFY COMMAND 命令将不会自动删除被修改的 .PRG 文件的下列两种文件版本(如果存在的话):

编译版本(.FOX 文件)

内存版本(已保存在程序缓冲区中的文件代码)

TIME=<常数>

为打印机等外部设备设置的重试次数。也就是说,若打印机没有准备好,那么 FoxBASE+ 就按照这个次数循环等待,并不断重试。若测试次数已经结束,那么 FoxBASE+ 就会发出错误信息“Printer not ready”(125 号错误信息)。次数的范围为 1 到 32767,系统的内定值是 6000。

WP=<文本编辑程序名>

设置一个外部的文本编辑器,它可以用来编辑注释字段。

下面,我们依照字母顺序列出 CONFIG. FX/DB 的其他项目,关于这些项目的详细使用方法,请参见本书所讲述的 SET 命令部分:

项目	数值	系统内定值
ALTERNATE	<文件名>	ON
BELL	ON/OFF	4
BUCKET	<1 to 64>	4
BUFFERS	<4 to 31>	31
CARRY	ON/OFF	OFF
CENTURY	ON/OFF	OFF
COLOR	<彩色属性>	
COMMAND	<命令>	
CONFIRM	ON/OFF	OFF
CONSOLE	ON/OFF	ON
DEBUG	ON/OFF	OFF
DECIMALS	<0 to 14>	2
DEFAULT	<驱动器/路径>	
DELETED	ON/OFF	OFF
DELIMITER	ON/OFF	OFF
DELIMITER	<字串表达式>	冒号

DEVICE	SCREEN/PRINT	SCREEN
ECHO	ON/OFF	OFF
ESCAPE	ON/OFF	ON
EXACT	ON/OFF	OFF
FUNCTION	数值表达式	
FILES	<16 to 18>	16
HEADINGS	ON/OFF	ON
HELP	ON/OFF	ON
HISTORY	<0 to 16000>	20
HMEMORY	<0 to 63>	5
INDEX	<extension>	IDX
INTENSITY	ON/OFF	ON
MARGIN	<1 to 254>	0
MAXMEM	<8 to 64>	64
MEMOWIDTH	<8 to 256>	50
MENU	ON/OFF	ON
MVARsiz	<1 to 64>	6
MVCOUNT	<128 to 3600>	256
PATH	<path>	
PCACHE	<8 to 64>	64
PRINT	ON/OFF	OFF
PROMPT	<字符串表达式>	""
SAFETY	ON/OFF	ON
SCOREBOARH	ON/OFF	ON
STATUS	ON/OFF	ON
STEP	ON/OFF	OFF
TALK	ON/OFF	ON
TEDIT	<文本编辑程序名>	
TIME	<1 TO 32767>	6000
TYPEAHEAD	<0 to 32000>	128
UNIQUE	ON/OFF	OFF
WP	<字处理器名称>	

1.4 将 dBASE III PLUS 转换成 FoxBASE+

因为 FoxBASE+ 本身和 dBASE III PLUS 兼容,所以在 FoxBASE+ 中执行 dBASE III PLUS 程序时,不必进行任何改变。但是,下列说明是相当重要的。

FoxBASE+ 和 dBASE III PLUS 的兼容性是非常完全的,包括:

- 原始程序(.PRG 文件)
- 数据库文件(.DBF 文件)
- 备份文件(.DBT 文件)
- 内存变量文件(.MEM 文件)
- 屏幕格式文件(.FMT 文件)
- 报表格式文件(.FRM 文件)
- 标签定义文件(.LBL 文件)

这些文件不论哪一类,FoxBASE+使用它们之前都不需作任何改变。

但是,用来存取 FoxBASE+数据库的索引通常比 dBASE III PLUS 的索引要小得多而且格式也不同。这是由于 FoxBASE+使用较新的索引技巧,因此显然有助于提高 FoxBASE 的速度。

在 FoxBASE+中,若想用 USE 或 SET INDEX TO 建立一个 dBASE 式的索引文件,FoxBASE+将马上自动地将数据库重新索引并且建立一个相应的 FoxBASE+索引文件。

由系统内部确定的这些新的索引在文件名称后都附上 .IDX。这种作法可以让 FoxBASE+和 dBASE 索引和平共处。

当 FoxBASE+遇到一个 dBASE 索引时,就会输出下列信息:

```
dBASE III index --rebuilding
```

并且开始重新索引。当然,若数据库很大,那么重新索引的过程要花上一定时间。但是自动地重新索引通常只要一次操作即可,FoxBASE+执行重新索引的速度相当快。最重要的是不必使用其他特殊手段就可以开始执行。

若曾经使用 dBASE 库和 dBASE 的索引而且希望确保所有反映数据库中现状的 FoxBASE+索引,那么只需删除所有的 .IDX 文件。然后,在 FoxBASE+中执行命令时,所有的索引又会自动被建立。

有一些命令可能需要索引文件的文件名称后加上 .NDX。对于这类用户,请将命令:INDEX=NDX 加到 CONFIG.FX/CONFIG.DB 文件中就可以强行使用 .NDX。

当然,若选择这项可选功能,在自动重新索引过程中,原有的 dBASE 索引将会被破坏。

FoxBASE+运行时提供与 MS-DOS 兼容的路径(PATH)。通过使用 SET DEFAULT 和 SET PATH 命令,FoxBASE+允许用户在搜索文件时自己设置一组目录。

SET DEFAULT 命令可以用来将系统内定的磁盘驱动器名称改成另一台不同于操作系统内定的磁盘驱动器名称。有一点非常重要,尽管所有的 FoxBASE+运算是在由 SET DEFAULT 命令所指定的磁盘中进行,操作系统所内定的磁盘驱动器仍然是一样的,除非通过使用 RUN 命令执行 MSDOS 的 CHDIR 改变操作系统的缺省磁盘驱动器。

SET PATH 命令可以用在文件搜索时设置一组目录。目录可以用相关的或合适的路径名称来设置。路径名称可用双圆点(.表示其上一级目录)、单圆点(.代表当前子目录本身)或反斜线(\第一个反斜线代表根目录,以后的反斜线作为分隔符)开始,如下所示:

```
SET PATH TO \SYSTEM\DATA
```

也可以磁盘驱动器名称开始,如下所示:

```
SET PATH TO C:\SYSTEM\DATA
```

那么,这个路径名称将被认为是完全合法的。若这个路径名称仅以目录名称开始,如下列命令:

```
SET PATH TO DATA
```

那么这个路径名称将被认为是当前工作目录的子目录,如下所示:

```
SET PATH TO .\DATA
```

当 FoxBASE+ 试图搜索一个文件时,它首先会搜索由系统缺省的磁盘驱动器或由 SET DEFAULT 命令设置的磁盘驱动器中的工作目录。若搜索不成功,那么 FoxBASE+ 将按 SET PATH 路径顺序进行搜索。

1.5 FoxBASE+ 数据类型

1.5.1 数据类型

许多 FoxBASE+ 命令在被处理之前需要额外的信息,这些信息以多种类型形成的数据表达式来提供。下面先介绍数据类型。

(1) 字符型

字符型的数据值包含 ASCII 集中 1 至 254 个字符。

(2) 数值型

数值型的数据介于 $-2 * 10^{63}$ 和 $2 * 10^{64}$ 之间,最多可显示 19 位(1 位保留为小数点)。

(3) 逻辑型

逻辑型的数据值为 True(真)或 False(假)。

True 值是以 .T.、.t.、.Y.、.y. 来表示。

False 值是以 .F.、.f.、.N.、.n. 来表示。

(4) 日期型

日期型的数据值介于 1/1/1900 到 12/31/1999 之间。

日期以月/日/年格式存储,如下所示:

- 月从 1 到 12。
- 日则视月份而定,可从 1 到 31。
- 年则有二位或四位,二位的年则代表 20 世纪的年份,如 89 代表 1989。

(5) 常量

常量是最基本的数据表达式,要键入这些值,必须遵守不同的程序来标记 FoxBASE+ 的数据值。

字符型的数据值如同字符串一样,最多不超过 254 字符,这种字符串必须用单引号、双引号、方括弧将它括起来。下列是有效的字符型值:

```
"NOW is the time"
```

```
[project # 1]
```

```
'city pao chicken'
```

日期型数据值不能直接在点提示符下键入,他们必须如字符串一样被括住后以 CTOD() 转换函数将它转为日期型,下列的例子说明了一日期值: