

■ 汤宝骥 祁惠民 丁大为 戴家振 编著

■ 施伯乐 张瑞蓝 审

实用 FoxPro 教程



上海交通大学出版社

实用 FoxPro 教程

汤宝骥 祁惠民
丁大为 戴家振 编著

施伯乐 张瑞兰 审

上海交通大学出版社

内容提要

本书较为详细地介绍了 FoxPro2.5 数据库管理系统的基本概念和使用。其主要内容有 FoxPro 运行环境、窗口及菜单的使用和操作方法。FoxPro 主要命令和函数的功能和使用。书中还介绍了 FoxPro 程序设计,并列举了几个应用程序实例供广大读者参考。

根据循序渐进的原则,本书注重实际应用,内容介绍深入浅出,兼顾普及与提高的需要。本书可以做为管理人员、计算机工作者、自学人员学习 FoxPro 的参考书,也可作为大专院校有关专业的教材。

责任编辑 陆义群

封面设计 朱天明

实用 FoxPro 教程

上海交通大学出版社·出版

(上海市华山路 1954 号 邮政编码 200030)

新华书店上海发行所·发行

常熟市印刷二厂·印刷

开本:787×1092(毫米) 1/16 印张:17.5 字数:434000

版次:1996 年 3 月 第 1 版 印次:1996 年 4 月 第 1 次

印数:1—5000

ISBN7-313-01656-5/TP·307 定价:22.00 元

前 言

近年来,在我国流行最广、应用面最大的微型计算机数据库语言是 dBASE III 关系数据库管理系统。这是因为 dBASE III 具有很强的数据管理功能和灵活的程序设计能力,编程很方便。它与高级语言相比,不但省时省力,而且程序与数据的独立性很好,易于维护、易于扩充,在数据处理、企事业信息管理、办公室自动化等方面常有事半功倍的效果,因而受到广泛的欢迎,得到普遍的应用。

但是,dBASE III 也有其固有的弱点,那就是计算能力较弱、处理速度较慢、没有数组和自定义函数、没有图形功能、没有多用户功能等等。这些弱点导致用 dBASE III 开发的应用项目往往只停留在单一项目或比较低的水平上。因此,人们很自然地期望有既能保持 dBASE III 优点,又能在功能、速度、用户界面等方面有个飞跃的数据库语言诞生。正是在这样的形势下,FoxPro 应时而生。

FoxPro 是美国 FOX 软件公司的最新产品,能与 dBASE、Foxbase 兼容。FoxPro 采用 Turbo 系列软件中方便快捷的菜单及按钮操作;提供了一个 Command 窗口,用户可以在这个窗口中输入指令;数据的输出输入可在 Browse 窗口中进行,用户可以随时将窗口开关、移动、改变大小,甚至可以分割窗口,使操作变得直观、方便和灵活;FoxPro 还提供了一个强有力的文件管理器 Filer,能进行集成化的文件操作。

FoxPro 具有快速查找能力,其查找速度比采用普通的查找方法快数百倍。FoxPro 还具有紧缩类型的索引,建立的索引文件缩短 20% 以上。

FoxPro 提供了高级外部程序接口 API,可以方便地实现与 C 语言和汇编语言的连接,这对于应用 FoxPro 编写大型、高水平、多功能软件提供了十分便利的条件。

FoxPro 所具有的优异性能,使它一问世就受到广大软件开发人员和计算机用户的高度重视,因此 FoxPro 的应用前景十分广阔。

本书第一章至第五章,第十四章至第十六章 16.1 节由祁惠民、丁大为执笔,第六章至第十章由汤宝骥执笔。第十一章至第十三章及第十六章 16.2 节由戴家振执笔。全书由汤宝骥通稿,施伯乐教授、张瑞兰高级工程师审阅了全书并提出了许多修改意见。在本书编写中,汪奕华教授、姚志清教授、邵祖英教授审阅了本书的编写大纲,并给予热情的支持与帮助,在此表示感谢。

由于作者水平有限,编写出版时间较短,书中错误缺点在所难免,恳请读者指教。

编 者

目 录

第一章 概述	(1)
1.1 数据库	(1)
1.2 数据库管理系统	(1)
1.3 FoxPro 基本概念	(2)
1.4 FoxPro 技术指标	(6)
1.5 FoxPro 运行环境	(8)
1.6 本书的印刷约定	(10)
第二章 FoxPro 的安装和运行	(11)
2.1 系统的安装	(11)
2.2 启动和退出	(11)
2.3 键盘和鼠标	(12)
2.4 菜单系统	(13)
2.5 窗口的使用	(20)
2.6 Help 功能	(21)
第三章 数据库文件及创建	(22)
3.1 数据库文件	(22)
3.2 数据库文件的建立	(23)
3.3 数据库文件的打开和关闭	(25)
3.4 数据库结构的修改	(26)
3.5 数据库文件结构显示	(27)
第四章 记录指针与记录显示	(28)
4.1 当前记录与记录指针	(28)
4.2 指针移动	(28)
4.3 条件定位	(30)
4.4 显示和打印记录	(30)
第五章 数据库文件的维护	(34)
5.1 增加记录	(34)
5.2 插入记录	(35)

5.3	编辑修改记录.....	(35)
5.4	Browse 方式下的编辑	(36)
5.5	替换修改记录.....	(40)
5.6	删除与恢复记录.....	(41)
5.7	数据库文件复制.....	(43)
第六章	数据库记录的选择处理	(46)
6.1	作用域子句.....	(46)
6.2	条件子句.....	(48)
6.3	FIELDS 子句	(50)
6.4	设置过滤器与预设字段表.....	(50)
第七章	排序与索引	(54)
7.1	排序.....	(54)
7.2	索引.....	(57)
7.3	索引文件的类型.....	(58)
7.4	索引文件的建立.....	(58)
7.5	索引文件的使用.....	(62)
第八章	统计、汇总与计算.....	(69)
8.1	计数操作.....	(69)
8.2	求和操作.....	(71)
8.3	求平均值操作.....	(73)
8.4	计算操作.....	(75)
8.5	汇总操作.....	(78)
第九章	内存变量与数组	(80)
9.1	内存变量与内存变量文件.....	(80)
9.2	数组.....	(82)
第十章	多文件数据库操作	(86)
10.1	工作区	(86)
10.2	数据库文件的联接关系	(88)
10.3	多个数据库文件之间的联接	(92)
10.4	View 窗口的使用	(94)
第十一章	建立和使用查询程序	(98)
11.1	查询程序	(98)
11.2	建立查询程序	(99)
11.3	使用条件来选择记录.....	(103)

11.4	建立总计查询程序.....	(107)
11.5	选择输出目标.....	(110)
11.6	修改一个现存的查询程序.....	(111)
11.7	其他查询方法.....	(111)
11.8	多数据库文件查询.....	(111)
第十二章	报表设计与打印.....	(115)
12.1	报表设计命令介绍.....	(115)
12.2	建立列表报表.....	(117)
12.3	进行设计.....	(121)
12.4	设置打印驱动程序.....	(122)
12.5	打印报表.....	(123)
12.6	加入 GROUP BAND	(124)
12.7	加入 SOMMARY 带	(125)
12.8	打印分组报表.....	(125)
12.9	建立格式报表设计.....	(126)
12.10	格式信件设计	(126)
12.11	多数据库文件报表	(128)
第十三章	FoxPro 函数	(132)
13.1	术语符号和规则.....	(132)
13.2	函数.....	(132)
13.3	用户自定义函数.....	(146)
第十四章	程序设计基本方法.....	(149)
14.1	程序的编辑.....	(149)
14.2	程序的基本结构.....	(150)
14.3	用户窗口设计.....	(157)
14.4	菜单程序的设计.....	(159)
14.5	输入和输出程序设计.....	(163)
14.6	程序调用方法及参数传递.....	(167)
14.7	程序的调试.....	(168)
14.8	程序的编译.....	(170)
14.9	程序设计的若干要点.....	(170)
第十五章	FoxPro 的其他功能	(172)
15.1	与其他数据库文件的信息交换.....	(172)
15.2	FoxPro 的多用户使用	(175)
15.3	FoxPro 的图形接口	(182)

第十六章 应用程序设计实例	(184)
16.1 工资管理程序设计.....	(184)
16.2 通信录管理程序设计.....	(221)
附录 FoxPro 命令与函数一览表	(234)

第一章 概 述

FoxPro 是一个关系型数据库管理软件,它的职能是接受用户通过应用程序、键盘或鼠标命令提出的要求,完成各类数据处理和数据库维护功能。

1.1 数据库

数据库(Database)可以看作是一种存放信息数据的“仓库”,是结构化的数据集合,它根据数据的性质、内在联系,实行统一的、集中的、独立的管理。它不像早期的数据文件那样,只对应并依附于一个或几个应用程序。因此,数据库的数据一致性、共享性都较好,数据冗余度小,而且数据库独立于程序,易于维护、易于扩充。

目前比较流行的数据库有关系型、层次型、网络型等三种。关系型数据库与层次型、网络型数据库的主要区别在于它把每个数据子集都分别按同一方法描述为一个关系,并且让子集之间彼此独立,然后在使用时通过筛选、投影、连接等方法,使数据之间或子集之间按某种关系进行操作。因此关系数据库表示能力比较强,初学者易于理解,使用上也比较方便,而且容易实现通用的数据管理功能。

关系型数据库(Relational Database)中的每一个关系(即数据子集),都是一个二维表,其形式犹如日常生活中常见的学生成绩表、职工工资表、生产统计表等。二维表中的每一列就是这些数据中的一种信息项,称为字段(Field)。二维表中的每一行是由若干信息组成的数据,称为记录(Record)。二维表的记录数随数据的增减而改变,但它的字段数却是相对固定的。因此,字段的个数和每个字段的名称、类型、宽度等要素便决定了二维表的结构。在 FoxPro 中,这些二维表即关系是以文件的形式存储在磁盘上的,一个文件就是一个关系。这样的文件称为数据库文件(Database File),有时也简称为数据库(Database)。

1.2 数据库管理系统

通常把 dBASE III 或 FoxPro 称为数据库系统,但就其功能来讲,应该称之为数据库管理系统更为确切。

这类微机上使用的数据库系统,不仅能直接使用键盘命令对数据库进行基本操作,诸如数据库文件的建立,数据文件的打开和关闭,数据库纪录的增加、修改、删除,数据库的复制,数据库结构的修改,数据库的排序、索引和检索,数据的计算和数据输出等,而且具有与其他高级语言类似的程序设计语言,能调用功能很强的函数。用户可以通过编程方式对数据库进行有效的管理,设计出符合用户需要的人机界面和输出报表,实现一些比较复杂的功能。

因此这种以数据库为中心的管理系统能保持数据与应用程序的相对独立性,给用户带来很大的方便,具有广泛的应用前景。

1.3 FoxPro 基本概念

一、数据种类

FoxPro 数据有常量、变量、函数等 3 种。

常量 (Constant) 是固定不变的数据。变量 (Variable) 是其内容可能要变化的数据。函数 (Function) 是一种由其他数据根据某种规则运算或操作后得出的数据。

FoxPro 的变量有两种: 字段变量和内存变量 (Memory Variable)。

字段变量, 即数据库字段。

内存变量又分为标量变量 (Scalar Variable) 和数组变量 (Array Variable)。

1. 常量

FoxPro 的常量可以是数值型、字符型、日期型或逻辑型。

① 数值型常量: 可以是整数或小数, 不可用分数, 但可用科学记数法表示。如 100、-8.18、3.14152E2 等, 其中 3.14152E2 是用科学记数法表示 3.14152 乘以 10 的二次方。

② 字符型常量: 是用单引号、双引号或方括号等定界符括起来的数据, 又称为字符串 (Character String) 或简称字串 (String)。字符串的定界符必须成对匹配, 如 '数据库'、'FoxPro'、[123.45] 等。如果某种定界符本身也是字串的内容, 则需用另一种定界符来括起, 例如 'a.[NAME]' 等。应当指出的是, '' 也是字符型常量, 它表示长度为零的空字串 (Null String)。

③ 日期型常量: 是用于表示日期的常量。在 FoxPro 中日期型常量无法直接表示, 一般是用字符串通过转换函数 CTOD() 来表示的, 如 CTOD('05/17/49') 表示 1949 年 5 月 17 日。其中字符串的格式为 "月/日/年" (MM/DD/YY)。用 SET DATE 命令可以改变该格式。

④ 逻辑型常量: 只有逻辑真和逻辑假两个值。逻辑真用 .T.、.t.、.Y. 或 .y. 表示, 逻辑假用 .F.、.f.、.N. 或 .n. 表示。

2. 变量

变量 (Variable) 的要素是名称、类型、长度、作用域、保存方式等。

变量的名称可以由字母、汉字、数字和下划线构成, 必须以字母或汉字开头, 名称的长度最多不能超过 10 个字符。

有关变量类型、长度、作用域、保存方式的规定, 在字段变量和内存变量中是有所不同的。

(1) 字段变量

字段变量的类型有数值型、字符型、浮点型、日期型、逻辑型、备注型、通用型 7 种:

① 数值型字段 (Numeric Fields): 包括整数或小数两种数值, 只能输入数字、小数点和负号 (-); FoxPro 中数字超过 1000 时不可用逗号隔开, 但可用格式定义来产生逗号。最多允许输入 20 位数字, FoxPro 可精确到 16 位, 所以, 除非进行精密科学计算, 都不必担心数字精度。

② 字符型字段 (Character Fields): 可容纳任何字符, 包括字母、数字、特殊符号和空格。一个字符型字段最多可容纳 254 个字符。

③ 浮点型字段 (Float Fields): 这是指有浮点小数的数值型字段。同数值型字段一样, 允许输入数字和负号, 并精确到 15 位。

④ 日期型字段 (Date Fields): 本字段用于存放日期。在日期型字段中允许输入 6 个数字, FoxPro

会自动插入分隔符(/)。在输入日子和月份时添足 0,如果不添 0 就必须输入/。

⑤逻辑型字段(Logical Fields):由表示真或假的单字母组成,T 或 Y 代表真(True),F 或 N 代表假(False)。

⑥备注型字段(Memo Fields):FoxPro 把记录中较大的文本块以固定格式存放于备注型字段中。一个无限长的文本理论上都可存放在备注型字段中(只是受硬盘空间的限制)。

⑦通用型字段(General Fields):此类字段可存放图画、声音、字处理文件、电子表格或其他种类的数据并与 WINDOW 实用程序相联系。

字段变量只存在于数据库中,是在生成数据库文件时建立的。由于字段变量的保存形式就是数据库文件,因此它的作用域随文件的打开而建立,随文件的关闭而撤消。由于在一般情况下,数据操作是针对当前选择的数据工作区中的字段进行的,因此,数据工作区也可看作是字段变量的作用域。不过 FoxPro 中绝大多数的数据操作也可使用非当前工作区的字段,但要在这些字段名前冠以所在的工作区名称或数据库别名。

对于一个数据库文件中的某一字段,该数据库文件有多少条记录,就有多少个值。因此,字段变量是一种多值变量,它的值随记录的不同而不同,只有记录指针当前所指记录的字段内容,才是它的当前值。

(2) 内存变量

内存变量是一种临时信息的存储单元,它独立于数据库文件而存在,使用时可随时建立。内存变量共有 5 种类型:数值型、字符型、日期型、逻辑型、屏幕型。

①数值型:存储数值型数据,即整数或小数,数值精度为 16 位有效数字。

②字符型:存储字符型数据,长度最大为 254 位。

③日期型:存储日期型数据,长度固定为 8 位。

④逻辑型:存储逻辑型数据,即 .T.、.t.、.F. 或 .f.,长度固定为 1 位。

⑤屏幕型:存储当前整幅屏幕信息。

内存变量中的标量一般是通过各种赋值命令建立的,这些命令同时完成变量的建立和赋值,并自动地把该变量的类型定义为相应的数据类型。

内存变量中的数组只能是一维或二维的,一般是通过数组定义命令(DIMENSION)建立的,数组中各单元的初始值是逻辑型.F.,通过各种赋值命令可以对整个数组同时赋值,也可以对各单元分别赋值并把它们的类型分别定义为相应的数据类型。同一数组中各单元的数据类型可以不同。

内存变量的作用域有两种:全局和局部。全局变量建立后,除非删除它,否则始终存在并起作用。局部变量只在建立它的程序(包括被该程序调用的程序)内有效,一旦退出该程序,即自行删除。

内存变量的保存形式是内存变量文件,内存变量可以存入该文件,也可以从该文件中恢复到内存中。

(3) 字段变量和内存变量的关系

字段变量和内存变量只要类型相同,可以互相传递,并可参与相应类型的数据操作。如类型不同,则必须通过类型转换函数把类型统一后,方可传递或运算。数据传递的方式有以下几种:

①字段变量传递给标量变量。

②标量变量传递给字段变量。

③字段变量传递给数组变量。

④数组变量传递给字段变量。

字段与数组相互传递时,一个字段对应数组中的一个单元,一条记录或一个文件对应一个数组。

字段变量和内存变量的不同之处,主要表现在以下几点:

- ① 字段变量隶属于数据库文件,不能独立存在。内存变量独立于数据库,自由度较大。
- ② 字段变量的保存形式是数据库文件,内存变量的保存形式是内存变量文件。
- ③ 内存变量的类型和长度,随所赋数据的类型和长度而定,因此只要对其重新赋值,便可予以改变。字段变量的类型和长度则是相对固定的,除非改动数据库文件结构;否则无法改变。
- ④ 内存变量是一种单值变量,只要不重新赋值,它的当前值就保持不变。字段变量是一种多值变量,即使不重新赋值,只要记录指针一移动,它的当前值就会随之改变。
- ⑤ 在处理的顺序上,字段变量比内存变量优先,即两者如果同名时,系统只认字段变量,而不认内存变量。为区别同名的内存变量与字段变量,应在内存变量名之前冠以“M→”或“M.”。

3. 函数

函数(Function)也是一种数据,而且这种数据的值是由其他数据通过某种特定的操作后得到的。因此,函数一方面是一种不同于常量和变量的特殊数据,另一方面它又代表了一些特定的数据操作功能,每取一次函数的值,就是进行一次数据操作。FoxPro 中具有 200 多种函数,因此它也就有 200 多种特殊数据或功能。

函数只能在表达式中使用,它的值也有一定的数据类型:数值型、字符型、日期型、逻辑型。

每个函数对其自变量的个数、类型、值域均有一定的规定,调用函数时必须遵守这些规定。在遵守这些规定的前提下,可以用表达式作自变量,如在该表达式中还含有函数,便可实现函数的嵌套调用。

FoxPro 的函数按其功能或用途可分数学运算函数、日期时间函数、字符操作函数、类型转换函数、数据库函数、测试函数等以及用户定义函数。

无论是 FoxPro 已有的函数,还是用户定义函数,它们的表示形式都是:在函数名称后跟一对圆括号,圆括号内给出函数的自变量,例如:INT(3/10),DBF(),WEEK(DATE())等等。只有宏代换函数(&)是个例外,直接在函数名 & 之后加自变量,如 &X、&M→ab。

二、运算符

运算符(Operator)是表示数据之间运算方式的符号。FoxPro 的运算方式有 4 种:数学运算、字符串运算、关系运算、逻辑运算。因此,运算符也有 4 种:数学运算符、字符串运算符、关系运算符、逻辑运算符,各种运算方式和运算符说明如下。

1. 数学运算(Mathematical Operation)

数学运算的对象是数值型数据,运算的结果仍是数值型数据。数学运算符按其运算的优选级别从高到低排列如下:

()	括号		
+	正号	-	负号
^ 或 **	乘方		
×	乘	/	除
+	加	-	减

2. 字符串运算(String Operation)

字符串运算是指把两个字符型数据连接起来,形成一个新的字符串。字符串运算符有两种:

十 两字符串相连

— 两字符串相连,并把前串尾部空格移到字符串的尾部

3. 关系运算(Relational Operation)

关系运算是指比较两个数据的大小或前后。关系运算只能在数值型数据之间、字符型数据之间、日期型数据之间进行。

①数值型数据之间:按数值的大小进行比较。

②字符型数据之间:按字符的 ASCII 码的大小进行比较。汉字进行比较时,是按其内码的 ASCII 码值进行的。从外观上看,一级字库按其拼音字母的排列顺序,二级字库按其笔划(首尾码)的顺序进行。

字符型数据的比较还分为精确匹配和不精确匹配两种,采用前者时,字符串"AB"不等于"AB",采用后者时,"AB"等于"AB"。

字符型数据之间还有一种关系运算是子串比较,它用来检测一字符串是否为另一字符串的子串。

③日期型数据之间:日期在前的为小,日期在后的为大。

关系运算后的结果是逻辑型数据,非真即假。关系运算符有如下几种:

=	等于	#或<>	不等于
>	大于	<	小于
>=	大于等于	<=	小于等于
==	字符串精确匹配比较	\$	子字符串比较

4. 逻辑运算(Logical Operation)

逻辑运算只能对逻辑型数据进行,它的结果依然是逻辑型数据。逻辑运算符按其运算的优先级别从高到低排列如下:

()	括号
.NOT. 或!	逻辑非
.AND.	逻辑与
.OR.	逻辑或

多种运算符同时出现时,括号的优先级最高,最内层的括号最优先,其余运算符的优先级别从高到低排列如下(相同优先级的运算依从左到右的顺序进行):

数值运算符或字符运算符

关系运算符

逻辑运算符

三、表达式

表达式(Expression)是数据之间运算关系的表达形式,因此可以由常量、字段变量、内存变量(包括标量和数组)、函数等数据和运算符组成。作为一个特例,单个数据也是表达式。

表达式的每一部分,不管是常量、字段还是内存变量,都被看成表达式的一个元素。表达式的所有元素必须是同类型的。例如,在同一个表达式中不能把字符字段和日期字段混合使用,除非用函数将日期转变成字符表示。

表达式经过运算,总是得出一个具体的值。表达式的值也是数据,它的数据类型有数值型、字符型、日期型、逻辑型等4种。表达式一般是按其值的数据类型分类的,因此FoxPro的表达式有:数值表达式(Numeric Expression,简称:EXPN)、字符表达式(Character Expression,简称:EXPC)等4

种。逻辑表达式因为多用来作为数据检索或程序分支的条件,因此又称为条件(Condition)。

表达式在 FoxPro 的各类数据操作和运算中,起着非常重要的作用。它不仅常常用于规定各类数据操作的对象,而且用于限制数据处理的范围,定义数据筛选的条件,设计数据输出的格式,设置系统状态和参数,甚至可以用一个表达式来描述一个复杂的数据管理事务或过程,因此,正确地应用表达式往往可以收到事半功倍的效果。

四、命令

命令(Command)是 FoxPro 用以完成数据操作、程序控制、系统状态或参数设置等功能的指令,它是 FoxPro 的重要组成部分。FoxPro 中几乎所有功能都是由命令(或者通过命令调用各种函数)完成的。

FoxPro 的命令一般是由命令动词(Command Verb)和限定该命令动词的若干个子句(Claue)组成的。命令动词用以规定该命令所要完成的“动作”,子句用以说明该“动作”的对象、范围以及在何种条件或前提下才允许该“动作”进行。有些命令的对象、范围或条件是固定的,因此不带子句。

命令动词一般采用英文术语表示,如:AVERAGE(求均值)、LIST(列表)等。

命令中的子句一般由功能短语、表达式等构成。功能短语也采用英文术语表示。在各类命令中最常用的子句有如下几种:

- (1) 范围子句(Scope Clause)——用以限定数据操作的范围。
- (2) 条件子句(Condition Clause)——用以指定数据操作的对象必须符合的条件。
- (3) 字段表子句(Field List Clause)——用以规定数据操作的对象。

命令可以由用户直接从键盘键入执行,也可以编成程序来执行。命令单词取前 4 位即为有效。

1.4 FoxPro 技术指标

常用的 FoxPro 有 WINDOWS 下的 FoxPro 及 DOS 下的 FoxPro 2.5 标准版和增强版,表 1.1 提供的是这几种版本系统技术指标:

表 1.1

	FoxPro 2.5 标准版	FoxPro2.5 增强版	WINDOWS 下 的 FoxPro
每个数据库中 最大记录数	10 亿个	10 亿个	10 亿个
每条记录中所 含字符数	65000 个	65000 个	65000 个
每条记录中最 多字段个数	255	255	255
每个字段的 最大字符数	254	254	254
最多工作区个数	25	225	225
每个 IDX 索引关 键字表达式长度	100 个字符	100 个字符	100 个字符

续表

	FoxPro 2.5 标准版	FoxPro2.5 增强版	WINDOWS 下 的 FoxPro
每个 CDX 索引关键字 表达式中字符数	240	240	240
每个数据库可打开索引 文件的最大个数	只受可用内存与可用文件柄的限制		
关系表达式的 最大长度	不限	不限	不限
字符型字段的最大长度	254	254	254
字段名中最多字符个数	10	10	10
数值计算的精度 (小数位)	16	16	16
缺省内存变量个数	256	256	256
最多内存变量个数	3600	65000	65000
数组的最多个数	3600	65000	65000
每个数组中元素的最 多个数	3600	65000	65000
源文件中最多程序行数	不限	不限	不限
单个过程或自定 义函数的长度	64K	64K	64K
每个文件中过程或 自定义函数的最多个数	不限	不限	不限
DO 命令的最多嵌套层 数	32	32	32
READ 命令的最多 嵌套层数	5	5	5
结构化程序命令的最多 嵌套层数 (如 IF/ ENDIF, DO WHILE/ENDDO...)	64	64	64
可打开窗口的最大个数	只受可用内存和文件柄的限制		
可打开的 Browse 窗口 的最多个数	25	225	225
每个程序行最多字符 个数	2048	2048	2048
可打开文件的最 多个数	99	只受 MS-DOS 限制	只受 MS-DOS 限制
每个键盘宏的最多击 键次数	1024	1024	1024

续表

	FoxPro 2.5 标准版	FoxPro. 5 增强版	WINDOWS 下 的 FoxPro
一个 SQL SELECT 命令 的最多可选字段数	255	255	255
每个颜色组中颜色模 式的最大个数	24	24	24
颜色组的最多个数	不限	不限	不限
每个颜色模式中最多 颜色数	10	10	10
用户可定义的颜色 模式个数	8	8	8
报表格式定义中的 最多目标个数	只受可用内存和文件柄的限制		
报表格式文件中最多 程序行数	255	255	255
报表格式的最大层数	20	20	20

1.5 FoxPro 运行环境

目前 FoxPro 2.5 FOR DOS 有标准型和增强型两种版本,增强型版本是一个真正的 32 位产品,它可以充分使用 80386 和 80486 微处理器,并可使用所有可用的扩展内存。表 1.2 为 FoxPro2.5 FOR DOS 的两种版本的基本配置。

表 1.2

	标准版本	增强版本
处理器	8088 或更高	386SX 或更高
RAM	640K(建议使用 2MB)	3MB
鼠标	建议使用	建议使用
MS-DOS 版本	MS-DOS 3.1 或更高	MS-DOS 3.1 或更高

当 FoxPro 被正确地使用在合适的硬件平台上时,它可提供异常快的数据处理速度。当然它也受到扩展内存(EMS 内存、XMS 内存),DOS 和局域网操作系统的相互作用。为了高效地运行 FoxPro2.5,不仅要建立最有效的 FoxPro 的系统配置文件 CONFIG.SYS,而且需要调整 DOS、内存管理器、磁盘超高速缓冲器以及局域网的配置和选择,以使 FoxPro 和用户程序获得尽可能大的常规内存空间(即基本内存空间),从而使 FoxPro 的性能处于较佳状态。

下面将介绍如何有效地配置计算机系统、局域网系统以及进行内存管理。

一、系统配置文件 CONFIG.SYS

无论使用的是 DOS 的哪种版本,也不管使用的计算机是单机运行还是作为局域网上的一个

工作站,只要机器启动,DOS总要在计算机的缺省驱动器的根目录中寻找 CONFIG.SYS 文件。CONFIG.SYS 文件中有两项设置的值对 FoxPro 的正常运行至关重要,它们是 FILES 参数和 BUFFERS 参数。在 CONFIG.SYS 文件中可以看到两项参数的设置:

```
BUFFERS=nn
```

```
FILES=nn
```

这里的 nn 代表一个整数值。

1. 打开文件个数参数 FILES

CONFIG.SYS 中的 FILES 参数定义了同一时刻可以打开的文件柄的个数。DOS 须使用 3 到 5 个文件柄,这取决于 DOS 的版本。

FoxPro 2.5/DOS 的增强版须使用 9 个文件柄,因为它不使用覆盖文件,而在它的内部完成覆盖交换。

每打开一个数据库,就要增加一个文件柄。如果打开的数据库包含一个或多个备注型或通用型字段,则 FPT 文件还需要一个附加的文件柄。如果正在使用 IDX 类型索引文件,无论是标准的还是紧缩的,每个 IDX 文件都要有一个文件柄。而 SQL 和 SQL CURSOR 也需要额外的文件柄。

一般来讲,DOS3.3 及其以上版本支持的 FILES 参数范围为 1 到 255,但这个范围也可能因具体机器的不同而不同。DOS3.3 以下的版本一般只支持最多 99 个文件柄。我们发现,将 FILES 参数取为 75 到 100 之间的值,基本上能适合大多数应用程序的需要。

2. 缓冲器参数 BUFFERS

一个缓冲器,或称为磁盘缓冲器,是一块内存单元(一般为 512 到 532 字节,具体值取决于 DOS 版本)。DOS 用它来存储最近从磁盘上读出的或写入磁盘的数据,将数据读出或写入内存的速度要比不断地移动硬盘磁头读写信息快得多。

CONFIG.SYS 文件中的缓冲器参数设置得太大或太小都将减弱计算机的性能。如果该参数设置得过大,DOS 将把大量的内存用于缓冲器,这样将减少 DOS 可为 FoxPro 提供的内存空间,最终降低了机器的性能。

实践表明:具有 1M 以上内存的计算机的最佳 BUFFERS 参数是 20 至 25,当然这种最佳值要随着机器的不同而改变。

二、局域网与文件柄

与单机或单用户环境相比较,局域网环境下的文件柄参数配置是一个复杂的问题,但对使用多用户应用程序的用户来讲,它又是一个关键问题。

1. NOVELL 286 和 NOVELL 386 局域网

除了 DOS 的 CONFIG.SYS 文件,NOVELL 286 或 NOVELL 386 局域网上的工作站还要求一个 NOVELL 配置文件,这个文件视不同版本叫做 NET.CFG 或 SHELL.CFG 文件。当启动 NOVELL LAN,系统就要寻找 NOVELL 配置文件。如果 NET.CFG 和 SHELL.CFG 文件同时存在,则 NOVELL 使用 NET.CFG 文件。NOVELL 配置文件应当和 NOVELL 驱动器放入同一个子目录。

SHELL.CFG 或 NET.CFG 文件中至少应包含 FILE HANDLES 参数,在概念上类似于 CONFIG.SYS 文件中的 FILES 参数。一般将 DOS 的 FILES 参数取为 100,将 NOVELL 的 FILE HANDLE 参数也取为 100 较为合理。

2. 基于 DOS 局域网

基于 DOS 的局域网,如 NOVELL NETWARE 要求装入 DOS SHARE。这种以及其他基于 DOS 的