第 1 篇　FoxBASE ＋程序设计

第1章 概 述

由美国 FOX 公司推出 FoxBASE+多用户关系数据库，运行速度比 dBASE Ⅱ高 5～7 倍，它具有 dBASE Ⅱ全部的命令和函数功能，并且引进新的命令和函数、过程、命令文件编译、多用户功能。因此，在国内外得到广泛的应用。本章介绍 FoxBASE+的硬件要求、技术指标及安装过程和启动。

1.1.1 硬件要求及技术指标

一、硬件要求

FoxBASE+ 需要内存容量为 1.0MB，如在多用户状态下工作，应有 4MB 的内存容量。

为了提高运行速度，需装有 80287 芯片，配备一个或二个软盘驱动器和一个硬盘驱动器，具有 DOS2.0 以上版本或 XENIX 操作系统，并带有各种汉字终端。分别可在单用户 DOS 下和多用户 XENIX 下运行 FoxBASE+。输出显示器为单显或彩显，并配 24 针打印机较好。

二、技术指标

FoxBASE+对一个数据库文件的最大记录数可达 10 亿个。
一个记录的最大字符数为 4000 个。
一个记录的最大字段数为 128 个。
一个字段的最大字符为 254 个，数值精度为 16 位。
一个字符串的最大字符数为 254 个。
一个命令行的最大字符数为 254 个。
每个报表标题的最大字符数为 254 个。
索引关键表达式长度<＝180 个字节。
内存变量个数<＝3600 个。
打开数据库个数<＝10 个。
同时打开文件数<＝48 个。
同时打开索引文件数<＝21 个，过程文件的过程个数<＝128 个。
程序嵌套重数<＝64。
数组元素最大数<＝3600。
MS/PC-DOS(单用户)最小版本 2.0。

MS/PC-DOS(多用户)最小版本 3.1。

1.1.2 FoxBASE +安装过程和启动

一、FoxBASE +(1.00 版本)安装

在操作系统 XENIX286 2.0 和 SYSTEM V 下运行。

FoxBASE+系统放在 2 张 5.25 "的软盘上(或 1 张 1.2MB 高密软盘上)。

安装步骤:

(1)在超级用户下安装 FoxBASE+。

#CD/(回车)

#(回到根目录下)

(2)在联机状态下,放入第一张软盘(或高密软盘),键入命令:

#TAR XVF(设备名)./TMP/INSTALL(回车)

表示将软盘上文件放在硬盘的 XENIX 中空目录进行安装。其中(设备名)对不同机型是有区别的。

对 IBM PC/AT 与 286、386、486 等微机,以每英寸 48 道(48TPI)、双面每道 9 扇区软盘设备名是:/DEV/RFDO48DS9

对 INTER 286/310 是:/DEV/RDV FO

当文件调出成功时,键入命令(3)。

(3)#./TMP/INSTALL(设备名)(回车)

(4)#(回车)——(第一张盘仍放在驱动器 A 上或按顺序号装入各张盘片)如果按 Q 表示退出。

(5)输入软件产品序列号和启动键号。

如需更新版本,再按"Y"(回车);如不需更新版本,回答"N"。

然后从软盘驱动器中取出盘片。通常为避免重写一些文件的操作错误,在普通用户("$"提示符)下使用 FoxBASE+较为安全。

于是,FoxBASE+(1.00)版本安装完毕。普通用户下输入注册帐号,即可调入已装入的 FoxBASE+系统文件。

二、FoxBASE +(2.00 版本)安装

在 CCDOS 2.00 以上版本下,将 FoxBASE+ 二张系统盘依次拷贝到硬盘中。

即:

C)COPY A:*.*(回车)　　(第一张盘)

C)COPY A:*.*　　(取下第一张盘,放入第二张盘)

如果建立一个子目录拷入 FoxBASE+步骤:

C)MD FOXP(回车)　　(建立子目录 FOXP)

C)CD FOXP(回车)　　(进入子目录 FOXP)

C)COPY A:*.*(回车)　　(将第一张盘拷入)

C)COPY A:*.*(回车)　　(取下第一张盘,放入第二张盘,拷入到子目录 FOXP 中)

于是,以后进入到子目录就可调用。

三、FoxBASE +(2.10 版本)安装

1. 加密 2 张盘片安装

C)MD FOXP(回车)　　(建立子目录 FOXP)

C)CD FOXP(回车)　　(进入子目录 FOXP)

C)COPY A:*.*(回车)　　(将第一张盘拷贝)

C)COPY A:*.*(回车)　　(取下第一张盘,放入第二张盘,在驱动器 A 上继续拷贝)

于是,以后进入到子目录 FOXP 就可调用。

第一张盘经加密处理,在启动时需将第一张盘放在驱动器 A 上作为过渡引导才能启动;如果未进行加密处理,就可省略此步骤。

2. 加密 10 张盘片安装

(1)C)MD　FOXP(回车)　　(建立子目录 FOXP)

(2)C)CD　FOXP(回车)　　(进入子目录 FOXP)

(3)C)RESTORE A:C:\FOXP/S(回车)　　(根据提示,将前 9 张盘从驱动器 A 上拷入硬盘 C 中。)

(4)C)COPY A:*.*(回车)　　(将第 10 张盘片拷入硬盘 C 中)

第 10 张盘片是加密盘片,启动时仍放在驱动器 A 上;如果已去掉加密,则可省略此步骤。

四、启动汉字 FoxBASE +的步骤

1. CCDOS 引导。

2. C)CD FOXP(回车)　　(进入 FOXP 子目录)

3. C)FOXPLUS(回车)　　(单用户 FOXBASE PLUS 启动或 2.10 版本)

或 C)MFOXPLUS(回车)　　(多用户 FOXBASE PLUS 启动)

如果是加密过的软件,则需将加密盘片放在驱动器 A 上作为"钥匙"过渡。

当显示圆点"."提示符时,表示启动后进入汉字 FoxBASE+。于是,可进行各种操作命令。同时,也可将放在驱动器 A 上作为"钥匙"盘片取下。

小　结

1. 汉字 FoxBASE+ 可用于 IBM PC/AT 与 GW286、386、486 等兼容机。CCDOS 2.0 以上版本,如 CCDOS2.1、CCDOS3.0、CCDOS2.13A 等,512K 以上内存。

2. FoxBASE+比 dBASE Ⅲ在性能及运行速度上都有很大提高,并与 dBASE Ⅲ完全兼容。

3. 对于不同型号的 FoxBASE+ 装入方法也不同。如果有加密盘,必须将它放在驱动器 A 上作为"钥匙"过渡。但启动方法是相同的。

即:

C)FoxPLUS(回车)　　(单用户或 2.10 版本)

C)MFoxPLUS(回车)　　(多用户)

习题与思考

1. FoxBASE+ 对硬件有什么要求?
2. FoxBASE+ 比 dBASE Ⅲ 在功能上有什么提高?
3. FoxBASE+ 的各种版本如何安装?
4. FoxBASE+ 如何启动?

第2章 函 数

FoxBASE+ 比 dBASE Ⅲ 具有新的函数功能，使用起来非常方便，函数部分能独立操作，同时又是以后各章的基础。

1.2.1 数据和变量

数据是数字、字母及各种符号的集合。它存储在各种介质上(如磁盘、磁带等)，由计算机经过数据处理成各种有用的信息。

一、数据类型

具有相同类型的各种数据才能够进行计算机处理。在 FoxBASE+中可分为 6 种类型：

1. N 型(数据型)：由数字、小数和正负数组成，长度<＝19 位。
2. C 型(字符型)：由字符、数字、空格等组成，长度<＝254 个字符。
3. D 型(日期型)：由分隔符"/"、数字或小数点等组成，长度固定为 8 个字符。
4. M 型(备注型)：由较多字符说明组成。通常不显示内容，建立时需按 CRTL-PgDn 键才能写入内容。长度固定为 10 个字符 。
5. L 型(逻辑型)：取值. T.(TRUE 真或成立)和. F.(FAULSE 假或不成立)或取值. Y. 和 . N. ，长度<＝1 个字符。
6. S 型(屏幕型)：S 型数据保存当前显示屏幕信息，相同于键盘控制屏幕输出(SHIFT＋Prtsc)。

二、变量

在运行程序过程中，变化的量为变量，固定不变的量称为常量。变量分为两种：

1. 字段变量：建立数据库时的字段名。
2. 内存变量：在数据处理过程中用于存放动态数据的工作单元。

变量名可以是字母或汉字开头的字母、数字及下划线组成的序列，长度不超过 10 个字符。

变量类型由数据类型决定。

字段变量分为：C 型(字符型)、N 型(数据型)、D 型(日期型)、L 型(逻辑型)、M 型(备注型)。

内存变量分为：C 型(字符型)、N 型(数据型)、D 型(日期型)、L 型(逻辑型)、S 型(屏幕型)。

1.2.2 内部函数

1. ABS

格式:ABS(〈数值表达式〉)

功能:返回一个数值表达式的绝对值。

例:.? ABS(-5)

显示:5

.? ABS(15-10)

显示:5

2. EXP

格式:EXP(〈数值表达式〉)

功能:返回 EXP 的指数函数值。E=2.718

例:.? EXP(2.72)

显示:1.00

3. SQRT

格式:SQRT(〈数值表达式〉)

功能:求平方根

例:.? SQRT(4)

显示:2.00

4. INT

格式:INT(〈数值表达式〉)

功能:取整数

例:.? INT(20.18)

显示:20

5. LOG

格式:LOG(〈数值表达式〉)

功能:求以 E 为底的自然对数。

例:.? LOG(2.718)

显示:1.00

6. MAX

格式:MAX(〈表达式〉,〈表达式〉)

功能:求表达式值的最大的一个。

例:.? MAX(20,35)

显示:35

7. MIN

格式:MIN(〈表达式〉,〈表达式〉)

功能:求表达式值的最小的一个。

例:.? MIN(20,35)

显示:20

8. MOD

格式：MOD(〈表达式 1〉,〈表达式 2〉)

功能：求表达式 1 除以表达式 2 的余数。

例：. ? MOD(7,3)

显示：1

9. ROUND

格式：ROUND(〈表达式 1〉,〈表达式 2〉)

功能：由〈表达式 2〉给出小数位数，而对〈表达式 1〉进行四舍五入定出位数。

例：. ? ROUND(3.14159,4)

显示：3.14160

. ? (3.14159,0)

显示：3.00

. ? ROUND(3.14159,-1)

显示：0.00000

10. FILE

格式：FILE(〈文件名〉)

功能：如为. T.（真），表示文件存在；如为. F.（假），表示文件不存在。

例：. ? FILE("C:A.PRG")

如显示：. T.，表示 A.PRG 在硬盘 C 上；

如显示：. F.，表示 A.PRG 不在硬盘 C 上。

11. BOF

格式：BOF（〈表达式〉）

功能：如为. T.（真），表示记录指针移出数据库第一个记录之外；如为. F.（假），表示在数据库内，表达式为工作区。

例：. USE GS

. SKIP -1（当打开 GS 库时，指向第一个记录，又往上移动一个记录，指针就指向第一个记录前。）

. ? BOF()

显示：. T.（真，成立）

. SKIP +2（指针指向第二个记录）

. ? BOF()

显示：. F.（假，不成立）

12. EOF

格式：EOF(〈表达式〉)

功能：如为. T.，表示记录指针移出最后一个记录之外；如为. F.，表示在数据库内，表达式为工作区。

例：. USE GS（打开库 GS）

. GO BOTTOM（指针指向最后一个记录）

. SKIP +1（移出最后一个记录外）

. ? EOF()

显示:.T.(真,成立)

.SKIP -1(指针指向最后一个记录)

.? EOF()

显示:.F.(假,不成立)

13. DELETED

格式:DELETED(〈表达式〉)

功能:如为.T.(真),表示库中记录有暂时性删除标志"*";如为.F.(假),表示库中记录没有暂时性删除标志"*",表达式为工作区。

例:.USE GS

.LIST FOR DELETED( )

(如果没有记录删除,就不列出记录;否则列出删除记录内容。)

14. RECNO

格式:RECNO(〈表达式〉)

功能:返回文件当前的记录号数。表达式为工作区。

例:.USE GS

.? RECNO()

显示:1(表示第一个记录)

.SKIP +2(往下移动二个记录)

.? RECNO()

显示:3(表示第三个记录)

15. RECCOUNT

格式:RECCOUNT(〈表达式〉)

功能:返回库文件的记录数。

例:.USE GS

.RECCOUNT()

显示:10(表示有10个记录)

16. RECSIZE

格式:RECSIZE(〈表达式〉)

功能:测定库中某一条记录长度。表达式为工作区。

例:.USE GS

.SKIP +2

.? RECSIZE()

显示:65(表示第三个记录长度为65个字符)

17. FOUNT

格式:FOUNT(〈表达式〉)

功能:测定库中字段数。表达式为工作区。

例:.USE GS

.? FOUNT()

显示:5(表示GS库中有5个字段)

18. FOUND

格式：FOUND（〈表达式〉）

功能：如为.T.，表示寻找命令成功（如FIND、COUNTINUE、LOCATE、SEEK）；如为.F.，表示寻找命令不成功。表达式为工作区。

例：.USE GS（打开库GS）

.INDEX ON 姓名 TO IGS（对姓名索引）

.SET INDEX TO IGS（打开索引文件）

.FIND 王刚（寻找王刚）

.? FOUND()

显示：.T.（显示.T.表示找到）

19. UPDATED

格式：UPDATED()

功能：如果最近的READ命令改变与之相联的GET语句的任何数值，返回.T.；否则返回.F.。

例：CLEAR（清除屏幕）

@5,0 GET 奖金（输入奖金值）

READ

IF UPDATED()（奖金变化，取值.T.）

USE GS（打开库GS文件）

REPLACE 补助 WITH 奖金（奖金替换补助）

ENDIF（IF条件句终结句）

20. DISKSPACE

格式：DISKSPACE()

功能：给出当前驱动器上可用的字节数。

例：IF DISKSPACE()>204800（如果磁盘空间大于200K，进行拷贝）

DO COPY

ELSE（如果磁盘空间小于200K，显示拷贝失败）

@2,1 SAY "无足够磁盘空间，拷贝失败"

ENDIF

21. SELECT

格式：SELECT()

功能：确定所指定工作区号。

例：.SELECT 4 （选择工作区4）

.USE GS（打开库GS）

.? SELECT()（测试出工作区号为4）

显示：4

22. LEN

格式：LEN(〈字符串表达式〉)

功能：测定字符的个数。

例：.? LEN("BOOK")

显示:4 ("BOOK"为 4 个字符)

.? LEN("中国杭州")

显示:8 (一个汉字相当于 2 个字符)

23. TYPE

格式:TYPE((表达式))

功能:测试表达式为 C 型(字符型)、N 型(数字型)、L 型(逻辑型)、D 型(日期型)、M 型(备注型)或 U 型(无定义)。

例:. X=10

.? TYPE("X")

显示:N (数值型)

. T="10"

.? TYPE("T")

显示:C (字符型)

.? TYPE("T/2")

显示:U (无定义,因字符不能被数字除)

24. ISALPHA

格式:ISALPHA((字符串表达式))

功能:用于测试字符字母开头。如是. T.,表示字母开头;如是. F. 表示不是字母开头。

例:.? ISALPHA("EF11")

显示:. T. (表示字母开头的字符串)

.? ISALPHA("11EF")

显示:. F. (表示不是以字母开头的字符串)

25. ISUPPER

格式:ISUPPER((字符串表达式))

功能:测定字符串,如果是大写字母开头为. T.;否则为. F.。

例:.? ISUPPER("ABC")

显示:. T. (表示大写字母开头)

.? ISUPPER("aBC")

显示:. F. (表示不是大写字母开头)

26. ISCOLOR

格式:ISCOLOR

功能:测试彩色显示器为. T.;否则为. F.。

例:. SET COLOR TO GR/B (设置蓝底黄字)

.? ISCOLOR()

显示:. T. (判定在彩色方式下运行)

27. LUPDATE

格式:LUPDATE()

功能:确定数据库最后一次修改日期。

例:. USE GS (打开库 GS)

.? LUPDATE() (显示最后修改日期)

显示:10/05/91

28. FLOCK

格式:FLOCK()

功能:测试数据库加锁。如为.T.,表示加锁成功;否则为.F.。

UNLOCK 将一个锁住的库文件解锁。

例:.USE GS (打开库 GS)

.IF FLOCK() (如库加锁为.T.)

.COPY TO GS1 (复制 GS 到 GS1 文件)

.UNLOCK (解锁)

.ELSE (如库未加锁为.F.)

.?"不能使用数据库" (显示"不能使用数据库")

.ENDIF

29. LOCK

格式:LOCK()或 RLOCK()

功能:测试记录。如为.T.,有读、写权,表示加锁成功;如为.F.,仅有读操作,表示加锁不成功。

同样,可用 UNLOCK 来解锁。

例:.SEEK "王刚" (寻找王刚)

.IF RLOCK() (如加锁为.T.)

.SET FORMAT TO GS (打开格式文件 GS)

.READ (读入)

.ELSE (如加锁未成功为.F.)

.?"姓名记录不能打印" (显示不能打印)

.ENDIF

30. ERROR

格式:ERROR

功能:显示出错号码,便于程序调试。同时需要激活 ON ERROR 命令才能得到出错号码。

例:C)TYPE MAIN.PRG (主程序调用过程 EP,激活 ON ERROR,带参数 ERNO)

```
PROCEDURE EP (过程 EP,参数 ERNO)
PARA ERNO
.
.
.
IF ERNO=54 (如出错号为 54 ,显示出错内容、字符串)
@5,1 SAY "出错内容 "+SUBSTR(MESSAGE(),4)
ENDIF
.
.
.
```

31. MESSAGE

格式:MESSAGE()

功能:由 ON ERROR 得到出错信息的字符串。MESSAGE(1)表示最近引起错误的行源代码。

32. IIF

格式:IIF((逻辑表达式),(表达式 1),(表达式 2))

功能:逻辑表达式值如为.T.,执行表达式 1;如为.F.,执行表达式 2。

例:.Y=IIF(X>0,1,-1)(如 X>0,Y=1,X<0,Y=-1)

.? Y(输出 Y 值)

33. SYS

格式:SYS((数值表达式))

功能:返回各种系统值。值为字符型。

例:SYS(0) 返回网络分配计算机名。

SYS(1) 返回当前系统日期。

SYS(2) 返回系统开锁的时间(以秒为单位)。

SYS(3) 建立临时文件名。

SYS(5) 返回当前隐含驱动器名。

SYS(6) 返回当前隐含打印机设备名。

SYS(9) 返回 FoxBASE+ 系列号。

SYS(12) 返回剩余内存的字节数。

SYS(13) 返回打印机的状态(开或关)。

SYS(17) 返回正使用的处理器名(8086/88,80286 或 80386)。

SYS(100) 返回由最后的 SET 语句产生的控制台设置状态(ON/OFF)。

34. ROW

格式:ROW()

功能:显示光标所在行的位置。

例:.USE GS (打开库 GS)

.GO 3 (指向第三个记录)

.? ROW() (显示第三行)

显示:3

.@ROW()+2,5 SAY "请输入奖金:"(光标下移两行,显示在 5 行、5 列位置输入奖金)

.? ROW()

显示:5

35. COL

格式:COL()

功能:显示光标所在列的位置。

例:.USE GS (打开库 GS)

.GO 3 (指向第三行)

.? COL() (显示第 5 列)

显示:5

.@ROW(),COL()+2 SAY "补助"(列数加 2 列,显示"补助"在 3 行、7 列位置)

.? COL()

显示:7

PCOL()和 PROW()分别表示打印机当前列和行的位置。

36. &

格式:&〈内存变量〉[.〈字符串表达式〉]

功能:用 & 后的内存变量值替换内存变量名,能实行灵活的查询和程序控制,提高程序通用性。

如在字符串中插入字符,在变量名后要用句号作为终止宏替换;& 也可以嵌套使用。

例:.姓名="王刚"　　(赋值)

.FIND &姓名　　(寻找王刚)

.X="*"　　(赋值)

.STORE "9&X9" TO P　　(将 9*9 赋值给 P)

.? P　　(显示 9*9)

显示:9*9

37. AT

格式:AT(〈字符串表达式 1〉,〈字符串表达式 2〉)

功能:寻找表达式 1 在表达式 2 中的位置。

例:.? AT("TER","COMPUTER")

显示:6　　(在第 6 个位置开始)

.? AT("浙江","浙江财经学院")

显示:1　　(在第一个位置开始)

.? AT("财院","浙江财经学院")

显示:0　　(没找到)

38. SUBSTR

格式:SUBSTR(〈字符串表达式〉,〈开始位置〉[,〈字符个数〉])

功能:截取字符串。如缺省字符个数,则截取开始位置至尾部。

例:.? SUBSTR("信息系 90 级",7,2)

显示:90　　(从第 7 个字符开始截取 2 个字符)

.? SUBSTR("信息系 90 级",7)

显示:90 级　　(从第 7 个字符开始截取至尾部)

.? SUBSTR("中国浙江杭州",9)

显示:杭州　　(从第 9 个字符开始截取至尾部)

39. LOWER

格式:LOWER(〈字符串表达式〉)

功能:将表达式从大写字母转换成小写字母。

例:.? LOWER("ABC")

显示:abc

40. UPPER

格式:UPPER(〈字符串表达式〉)

功能:将表达式从小写字母转换成大写字母。

例:.? UPPER("abc")

显示:ABC

41. LEFT

格式:((字符串表达式),(数值表达式))

功能:从字符表达式最左边开始,选取由数值表达式所定个数的字符。

例:.? LEFT("BOOK,PEN",4)

显示:BOOK

.? LEFT("中国杭州",4)

显示:中国

42. RIGHT

格式:RIGHT((字符串表达式),(数值表达式))

功能:从字符表达式最右边开始,选取由数值表达式所定个数的字符。

例:.? RIGHT("BOOK,PEN",3)

显示:PEN

.? RIGHT("中国杭州",4)

显示:杭州

43. TRIM

格式:TRIM((字符串表达式))

功能:删除表达式中尾部的空格。

例:.STORE "信息系" TO 单位 (赋值)

.STORE "计算机教研室" TO 部门 (赋值)

.? TRIM ("单位")+部门 (删除单位尾部空格)

显示:信息系计算机教研室

44. LTRIM

格式:LTRIM((字符串表达式))

功能:删除表达式中左边空格。

例:.STORE " 杭州" TO X

.? TRIM("X")

显示:杭州

45. RTRIM

格式:RTRIM((字符串表达式))

功能:删除表达式中右边的空格,等价 TRIM 作用。

例:与 TRIM 相同。

46. SPACE

格式:SPACE((数值表达式))

功能:产生空字符,数目由表达式定。

例:.?"中国"+SPACE(5)+"杭州"

显示:中国　　杭州

47. REPLICATE

格式:REPLICATE(〈字符串表达式〉,〈数值表达式〉)

功能:由数值表达式的值决定字符串表达式重复的次数。

例:.? REPLICATE("中国杭州",2)

显示:中国杭州中国杭州

48. STUFF

格式:STUFF(〈字符串表达式 1〉,〈起始位置〉,〈字符串个数〉,〈字符串表达式 2〉)

功能:用于置换字符串。表达式 2 是置换表达式 1 的字符串。由表达式 1 的起始位置及字符串个数而定。

当起始位置大于表达式 1 的长度,即将表达式 2 增加到表达式 1 尾部。当字符个数等于 0,即将表达式 2 插入到表达式 1 中,而不删去字符。当表达式 2 是空字符串,即删除表达式 1 字符个数。

例:.? STUFF("SUNDAY",1,3,"MON")

显示:MONDAY (替换)

.? STUFF("BOOK",1,2,"")

显示:OK (删除)

.? STUFF("DOG",4,3,"CAT")

显示:DOGCAT (添加)

.? STUFF("中国",3,0,"立")

显示:中立国 (插入)

49. TRANSFORM

格式:TRANSFORM(〈表达式〉,〈字符串表达式〉)

功能:允许用户选择字符和数值变量格式,而不使用@...SAY 命令。通过使用?,??,LIST,DISPLAY,LABEL,REPORT 等命令输出字符或数值的 PICTURE 格式。

例:.? TRANSFORM("CHINA","@!") (转化为大写)

显示:CHINA

.USE GS

.LIST TRANSFORM(基本工资,"#,##,##.##")

显示:RECORD# TRANSFORM(基本工资,"#,##,##.##")

(小数位取 2 位,整数 2 位加",号)

1	1,50.00
2	1,40.00
3	1,10.00
4	1,00.00
5	90.00

50. TIME

格式:TIME()

功能:显示系统时间(时,分,秒)。

例:.? TIME()

显示:08:30:40 (8 时 30 分 40 秒)