

## 巧用内存变量文件及数组建立 MIS 通用显示模块

曲卫芬 夏方道 (青岛大学自动化系 266071)

在利用 FOXBASE 系列的关系型数据库建立数据库管理应用系统中,对数据库进行数据输入、输出、修改及查询显示是经常性的,也是必不可少的主要工作。对数据进行输入输出,一般是屏幕上给出提示,利用内存变量或字段变量进行输入或显示工作。通常是利用"@行,列 SAY '提出' GET 变量 PICT 格式符"语句完成的,这样程序灵活性差,通用性差。笔者在长期的实际应用中,摸索出利用内存变量文件及数组建立 MIS 系统的显示输入模块的方法,用于解决实际问题不失是一个很有用的灵活、通用的好方法,使程序的编写十分简洁明了,提高编程速度,收到很好的效果。

具体步骤如下:(举例说明)

## 1. 例如建立档案工资库 GZ.DBF 结构如下

字段名称 BH XM ZW JBGZ ZWBT JJIN FSBT FZ  
中文名称 编号 姓名 职务 基本工资 职务补贴 奖金 副食补贴  
房租  
类型长度 C(8) C(8) C(16) N(7,2) N(6,2) N(6,2) N(6,2)  
N(6,2)

## 2. 建立显示提示内存变量文件 TITLE11.MEM

```
DIME TITLE(8)
TITLE(1) = '编号:'
TITLE(2) = '姓名:'
TITLE(3) = '职务:'
TITLE(4) = '基本工资:'
TITLE(5) = '职务补贴:'
TITLE(6) = '奖金:'
TITLE(7) = '副食补贴:'
TITLE(8) = '房租:'
```

## 3. 建立格式符内存变量文件 MPROMPT11.MEM

```
DIME MFIELD(8)
MFIELD(1) = '99999999'
MFIELD(2) = ''
MFIELD(3) = ''
MFIELD(4) = '9,999.99'
MFIELD(5) = '999.99'
```

```
MFIELD(6) = '999.99'
```

```
MFIELD(7) = '999.99'
```

```
MFIELD(8) = '999.99'
```

## 4. 通用显示输入程序

```
PROCEDURE HGET
```

```
PARAMETERS @-WIN, @-SCRE &&-WIN 窗口号, @-SCRE 屏幕号
```

```
PRIVATE I, J, FCNT, LENGTH, TEMP, FSTART, KK, COL, ROW1, ROW2, COL1, COL2, MEMNAME
```

```
RELEASE MF
```

```
KK = @-WIN && 取数据库所在的工作区号, 设窗口号即是工作区号
```

```
SELECT @KK
```

```
FCNT = FD(VAL(@-WIN)) && 取数据库中的字段数
```

```
DIME MF(FCNT) && 定义数组用于存放一个记录
```

```
GO BOTTOM
```

```
IF .NOT.EOF()
```

```
SKIP
```

```
ENDIF
```

```
SCATTER TO MF
```

```
FSTART = F&@-WIN.(VAL(@-SCRE), 1) && 取该屏幕起始字段号
```

```
LENGTH = F&@-WIN.(VAL(@-SCRE), 2) && 取该屏幕需输入的字段个数
```

```
KK = F&@-WIN.(VAL(@-SCRE), 3) && 取内存变量文件号
```

```
I = 1 && 显示的起始行号
```

```
J = FSTART && 显示起始字段号
```

```
MEMNAME = 'MPROMPT' + KK
```

```
REST FROM @MEMNAME ADDI
```

```
MEMNAME = 'TITLE' + KK
```

```
REST FROM @MEMNAME ADDI
```

```
SET EXACT ON
```

```

ROW1 = 2
COL1 = 6
ROW2 = ROW1 + LENGTH + 3
COL = LEN(TITLE(1)) && 提示行的长度
COL2 = 60
@ROW1, COL1 CLEAR TO ROW2, COL2
@ROW1 + 2, COL1 + 2 TO ROW2 + 2, COL2 + 2
DOUBLE
@ROW1 + 1, COL1 + 10 SAY '请输入下列数据'
DO WHILE I <= LENGTH && 循环显示所需输入的全部字段
    TEMP = IIF(MFIELD(I) = "", "", ' PICT "' +
MFIELD(I) + '"')
    @ROW1 + 2 + I, COL1 SAY TITLE(I) GET MF
(J) & TEMP
    I = I + 1
    J = J + 1
ENDDO
CLEA GETS
I = 1
DO WHILE .T. && 循环接收用户的输入
    TEMP = IIF(MFIELD(I) = "", "", ' PICT "' +
MFIELD(I) + '"')
    @ROW1 + 2 + I, COL1 + COL + 1 GET MF(I + FS-
TART - 1) & TEMP
    READ
    RK = READKEY()
DO CASE
    CASE RK = 2. OR. RK = 258 && 用户按 Home 键
        I = 1
    CASE RK = 3. OR. RK = 259 && 用户按 End 键
        I = LENGTH
    CASE RK = 4. OR. RK = 260 && 用户按向上尖头键
        I = IIF(I = 1, 1, I - 1)
    CASE RK = 5. OR. RK = 261 && 用户按向下尖头键
        I = IIF(I = LENGTH, LENGTH, I + 1)
    CASE RK = 14. OR. RK = 270 && 用户按 ^W 或 ^End
键存盘退出
        -EXIT = .T.
        EXIT
    CASE RK = 12. OR. RK = 268 && 用户按 ^Q 或 ^Esc

```

```

键不存盘退出
        -EXIT = ..
        EXIT
        CASE RK = 15. OR. RK = 271) . AND. I = LENGTH &&
用户全部输入完毕存盘退出
        -EXIT = .T.
        EXIT
    OTHE
        I = IIF(I = LENGTH, LENGTH, I - 1)
        ENDC
    ENDD
    IF .NOT. -EXIT
        RETU
    ENDIF
    APPE BLANK
    GATH FROM MF
    SET EXACT OFF
    RETU
5. 模拟主程序
MAIN. PRG
SET TALK OFF
CLEA ALL
SET PROC TO HGET
SELE 1
USE GZ
DIME F1(1, 3)
F1(1, 1) = 1
F1(1, 2) = 8
F1(1, 3) = '11'
-WIN = '1'
-SCRE = '1'
DO HGET WITH -WIN, -SCRE
USE
SET PROC TO
RETU
    以上只是一个设计思想, 可以对此稍加改进, 以适
用于多屏幕、多数据库的输入、修改、查询显示等。笔者
用此思想设计了多个数据库管理系统, 收到了很好的效
果。

```

(来稿时间: 1996年11月)