

```

Attribute VB-Name = "Form1"
Attribute VB-Creatable = False
Attribute VB-Exposed = False
Private Sub Form_Load()
    Dim Index As Integer

    Outline1.AddItem "Root" ' 加入根节点
    Outline1.Indent(0) = 0 ' 层号为 0
    Outline1.PictureType(0) = 1 ' 用 PictureOpen 图标
    Index = 1 ' 表明记数器
    For i = 1 To 4
        Outline1.AddItem "A" & Trim(Str(i)) ' 加入第一级
        节点 A1~A4
        Outline1.Indent(Index) = 1 ' 层号为 1
        Index = Index + 1 ' 增加一项后计数器加 1
        For j = 1 To 4
            Outline1.AddItem "B" & Trim(Str(j)) ' 加入第二
            级节点 B1~B4
            Outline1.Indent(Index) = 2 ' 层号为 2
            Index = Index + 1 ' 增加一项后计数器加 1
            For k = 1 To 4
                Outline1.AddItem "C" & Trim(Str(k)) ' 加入
                第三级节点 C1~C4
                Outline1.Indent(Index) = 3 ' 层号为 3
                Outline1.PictureType(Index) = 2 ' 用 Picture
                Leaf 图标
                Index = Index + 1 ' 增加一项后计数器加 1
            Next
        Next
    Next
End Sub

Private Sub Outline1-Click()
    Label1.Caption = Outline1.List(Outline1.ListIndex) & "
    clicked" + Chr(13) + Outline1.List(Outline1.ListIndex) & "
    has "
    If Not Outline1.HasSubItems(Outline1.ListIndex) Then '
    如果该节点没有子项 Label1.Caption = Label1.Caption + "
    no "
    End If
    Label1.Caption = Label1.Caption + "subitems" + Chr
    (13) + Outline1.List(Outline1.ListIndex) + "'s full path is "
    + Outline1.FullPath(Outline1.ListIndex)End Sub
Private Sub Outline1-Collapse(ListIndex As Integer)
    If Outline1.PictureType(ListIndex) <> 2 Then ' 如果不
    是叶节点
        Outline1.PictureType(ListIndex) = 0 ' 则换成 Picture-
        Closed 图标
    End If
    Label1.Caption = Outline1.List(ListIndex) & " collapsed"
End Sub

```

```

Private Sub Outline1-DblClick()
    Outline1.Expand(Outline1.ListIndex) = Not Outline1.Ex-
    pand(Outline1.ListIndex) ' 切换打开/关闭状态
    Label1.Caption = Outline1.List(Outline1.ListIndex) & "
    dblclicked"
End Sub

Private Sub Outline1-Expand(ListIndex As Integer)
    If Outline1.PictureType(ListIndex) <> 2 Then ' 如果不
    是叶节点
        Outline1.PictureType(ListIndex) = 1 ' 则换成 Picture-
        Open 图标
    End If
    Label1.Caption = Outline1.List(ListIndex) & " expanded"
End Sub

Private Sub Outline1-PictureClick(ListIndex As Integer)
    Outline1.Expand(ListIndex) = Not Outline1.Expand
    (ListIndex) ' 切换打开/关闭状态
    Label1.Caption = Outline1.List(ListIndex) & "'s picture
    clicked"
End Sub

Private Sub Outline1-PictureDblClick(ListIndex As Integer)
    Label1.Caption = Outline1.List(ListIndex) & "'s picture
    dblclicked"
End Sub

```

(来稿时间:1996 年 10 月)

FoxBase + 中 RUN/! 命令的运用指错

郭建军 (晋东南师专计算机室 046011)

贵刊 1996 年第 4 期刊登的《FoxBASE + 中 RUN/! 命令的运用》一文,详细介绍了作为 FoxBASE + 和 DOS 交互调用命令 RUN/! 的运行环境和注意事项,并列举了在 FoxBASE + 状态下使用 RUN/! 命令的八种形式,对于使用 FoxBASE + 的用户来说,真是大开了眼界。但是笔者在对这八种形式的命令一一上机验证时,发现有些命令的使用受到一定的限制,不具有通用性,有些命令则无法执行。不知是印刷错误还是作者的一时疏忽。其实熟悉 DOS 命令和 RUN/! 命令使用的都知道,文中列举的所有形式的命令,除了在 FoxBASE + 状态使用的宏代换和内存变量以外,每一条命令去掉 RUN/! 后都可以在 DOS 状态下执行。

为了保证贵刊的实用性、可读性及更受到广大读者的欢迎,笔者冒昧地把验证的结果介绍如下,不妥之处请同行多多批评。

一、受到限制的命令

1. 文章中(二)建立子目录

(3)在程序中随时建立子目录的程序段:

```
STORE ' ' TO ML
@ 3,10 SAY '请输入要建立的子目录名:' GET ML
READ
RUN MD &ML
```

如运行该程序,只能建立子目录名是一个字符的子目录。因为在用格式化输入语句给字符型变量赋值时,输入的字符宽度是由原来字符变量的宽度所决定的。程序中的 STORE ' ' TO ML 应改为 STORE SPACE(8) TO ML。

2. 文章中(五)拷贝文件

(3)在程序中执行由用户指定文件程序段:

```
STORE ' ' TO NAME
@ 3,10 SAY '请输入要拷贝的文件名:' GET NAME
READ
NAME = TRIM(NAME)
RUN COPY &NAME A:
```

如运行该程序,拷贝的文件名只能是一个字符,扩展名无法输入。程序中的 STORE ' ' TO NAME 应改为 STORE SPACE(12) TO NAME,才能在拷贝文件时选择不同的文件名。

3. 文章中(五)拷贝文件

(4)在程序中拷贝,自动生成文件名程序段:

```
STORE ' ' TO P
STORE ' ' TO NY
@ 1,1 SAY "请输入要拷贝文件的标记(年月):" GET NY
@ 3,1 SAY '从 C 盘拷贝到'
@ 3,18 GET P
READ
NAME = C + ":" + GZ + NY + ".DBF" + P + ":"
COPY &NAME
RETU
```

如运行该程序,年和月的标记无法输全,应根据文件的具体标记宽度修改 STORE ' ' TO NY,假设年标记为 2 个宽度,月标记为 2 个宽度,则应改为 STORE SPACE(4) TO NY,另外在 ".DBF" 中 F 与 "之间应加一个空格,以保证拷贝的文件名和目标盘盘符分开,否则会出现提示为 "File cannot copy onto itself" 的错误。

二、无法执行及执行有错的命令

1. 文章中(三)PRG 文件和 TXT 文件的屏幕显示与打印

(1)显示 GZ.PRG 文件:

. RUN TYPE GZ 应改为 . RUN TYPE GZ.PRG

如用文章中的命令,当前目录下有 GZ.PRG 文件,屏幕上也会显示 "File not found" 的提示。

(2)打印 GZ.PRG 文件:

. RUN TYPE GZ > PRN 应改为 . RUN TYPE GZ.PRG > PRN

如用文章中的命令,会出现与上述同样的提示。因为在 DOS 下,显示和打印文件时,输入的文件名都要加上扩展名。

2. 文章中(六)查看目录、删除文件及文件误删的防止

(4)删除 A 盘上的所有前两位为 GZ 的备份文件:

. SC = 'GZ * .DBF' 应改为 . SC = 'GZ * .BAK'

如按原命令删除的不是备份文件而是数据库文件。

(5)安装新软件时,防止同名的文件被误删:

```
STORE 'Y' TO ZZ
IF FILE (C:GZ.DBF)
WAIT '已有 GZ.DBF 文件存在删除吗(Y/N)?' TO ZZ
ENDI
IF UPPE(ZZ) = 'Y'
RUN A:GZ.DBF C:
ENDI
RUN COPY A: * .EXE C:
```

运行此程序,用户选 Y 删除文件时无法执行,程序中的 RUN A:GZ.DBF C: 应改为 RUN DEL C:GZ.DBF。

3. 文章中(七)执行外部编辑软件和高级语言程序

(1)编辑修改 GZ.PRG 后直接执行

```
. RUN EDLIN GZ
. RUN GZ
```

以上用 EDLIN 编辑命令建立的文件为 GZ,而不是 GZ.PRG,运行时无法执行,不能和在 CONFIG.FX 文件中设置了 TEDIT = EDLIN 相提并论,直接调用 EDLIN 编辑文件时,必须加上文件的扩展名。因此,. RUN EDLIN GZ 应改为

. RUN EDLIN GZ.PRG

4. 文章中(八)硬盘大文件的复制和恢复

(2)拷贝成批文件:

RUN C: * .DBF A: 应改为 RUN COPY C: * .DBF A:

如用原命令是错误命令,拷贝文件命令行中没有 COPY。

顺便提一点,在 FoxBASE+ 状态用 RUN/! 运行 DOS 命令,对于内存需求大的命令,如 DISKCOPY 最好不要使用,因为该命令把内存作为信息交换的中转站,而汉字系统和 FoxBASE+ 都驻留内存,会使此命令运行速度缓慢。

(来稿时间:1996 年 12 月)