

FOXBASE 调用 PASCAL 数据的接口技术

何亚群 (中国矿业大学)

一、问题的提出

笔者在开发选煤厂数据库管理系统时,需要使用为选煤厂预测选煤效果,模拟选煤过程而开发的“选煤软件包”的数据,该软件包用 PASCAL 语言编写。由于 FOXBASE 语言无法直接调用 PASCAL 程序产生的大量数据,我们设想在 FOXBASE 环境下,用 FOXBASE 提供的 RUN 命令外部调用编译成 EXE 文件后的 PASCAL 程序,使之产生所需的数据文件。再用 FOXBASE 语言将其读入对应的数据库,供数据库管理软件调用。这一过程要求:

1. PASCAL 产生的数据文件名需体现日期信息。
2. 数据文件读入数据库后能从盘上被自动删除。
3. 检查磁盘有无所需数据文件,若有则用 FOXBASE 程序实现将数据文件写入对应的数据库,若没有数据文件则保留数据库中原有数据。
4. 写库前应先检查数据库中有无对应日期的同类型数据,有则需先删除数据库中该数据后再写入新数据,否则直接向数据库写入数据。

二、形成数据文件的协定

数据库管理系统中,数据日期往往是数据的一个重要内容。我们在用 PASCAL 语言建立数据文件过程中,将同一类型,同一日期的数据写入一个数据文件。形成文件名时,在文件名的后四位字符上体现日期信息,文件名其它部分由编程者确定。如文件名可取为:‘OUTA’+DATEB’+.DAT’,其中 DATEB 为体现日期信息的变量,根据不同的日期做相应处理后赋值,如 93 年 7 月的数据,DATEB 赋值为 9307,这样该文件读入数据库前,以及读完文件后便于在磁盘上查找和删除。

为实现与其它高级语言的数据交换,FOXBASE 提供了 APPEND FROM 语句,把磁盘数据文件读入相应

的数据库。通常可读入三种格式的数据文件:

1. 标准数据格式,文件类型选择项为 SDF。
2. 通用数据格式,由空格分隔各数据变量,类型选择项为 DELIMITED WITH BLANK。
3. 通用数据格式,由逗号分隔各数据变量,类型选择项为 DELIMITED WITH <定界符>,其中<定界符>可与 WITH 一起缺省。

在实际处理问题时,标准数据 SDF 格式要求非常严格,用其它高级语言形成符合该格式的数据文件非常困难,除非数据文件就是用 FOXBASE 语言的 SDG 选择项形成。为此,我们在形成数据文件时是用通用数据格式,即用逗号分隔。尽管这种方法建立数据文件时较麻烦,但它对格式的要求较低,各数据变量间只要用逗号隔开,可不考虑形成文件时所产生的多余空格。

以下是用 PASCAL 语言形成数据文件的例子:

```
ASSIGN(FOUT,'C:\HY\OUTA'+DATEB+'.DAT');
REWRITE(FOUT);
FOR M=1 TO MDO
BEGIN
    WRITE(FOUT,AL(I):6:2,',',A2(I):6:2,',');
    WRITE(FOUT,AL(''),DATEA:8,'',',');
    WRITE(FOUT,AL(''),BJ(I):2,'',',');
    WRITE(FOUT,AL(''),LB:2,'');
    WRITELN(FOUT);
END;
```

CLOSE(FOUT);

其中:A1(I),A2(2)是两个实型数组。

DATEA 是字符型变量,表示日期,通常是 93.07.15 的形式。

BJ(I)是一字符型数组,作为标记。

LB 是一字符型变量,作为类别。此处 LB='A'。
形成的文件格式为如下二维矩阵。

```

OUTA9307.DAT
A1(1).A2(1),"93.07.15","BJ(1)","A"
A2(1).A2(2),"93.07.15","BJ(2)","A"
...    ...    ...    ...    ...    ...
A1(M).A2(M),"93.07.15","BJ(M)","A"

```

三、数据文件写入数据库过程

数据文件形成后,即可用 FOXBASE 语言编程,将数据文件读入相应的数据库。针对上述数据文件,库结构设计为:

A.DBF STRU

FIELD	FIELDNAME	TYPE	WIDTH	DEC
1	A1	N	6	2
2	A2	N	6	2
3	日期	C	8	
4	标记	C	4	
5	类别	C	2	

由于在写库前需检查磁盘上某日期数据文件是否存在,因此在 FOXBASE 环境下同样需要一个日期信息。这个日期信息在运行软件包输入数据时已经输入,所以用上述步骤传入一个日期变量,即在磁盘上建立一个只含有日期信息的数据文件,然后将它写入一个对应的临时数据库,经处理后在 FOXBASE 环境下仍还原为变量 DATEA 和 DATEB。此时即可检查磁盘,确认该日期数据文件是否存在,进而检查对应数据库内是否已有同日期、同类型数据,若同类数据存在而认为需要更新,程序要将原有数据从数据库中删除,这是因为 APPEND FROM 语句只能在数据库尾部添加新数据,而不能覆盖原有的旧数据。数据库中同日期、同类型的数据被删除后即可将数据文件写入数据库,该过程由以下程序实现:

```

SET DATE ANSI
IF FILE('OUTA'+DATEB+'A.DAT')
USE A
LOCATE FOR YEAR (DTOD(日期)) = YEAR
(DATEA) .AND .MONT (CTOD (日期)) = MONT
(DATEA) .AND.LTRIM (类别) ='A'
IF .NOT.EOF()
DELE FOR YEAR (CTOD (日期)) = YEAR (DATEA)
.AND. MONT (CTOD (日期))= MONT (DATEA) .AND.

```

LTRIM (类别) ='A'

PACK

STOR'OUTA'+DATEB+'A.DAT'TO NAME

APPEND FROM NAME DELIMITED

USE

ENDIF

在数据文件写库过程中,日期变量只能作为字符变量处理,所以必须用 FOXBASE 提供的 CTOD()函数转换。当数据文件中字符变量的长度与库结构中对字段长度非精确匹配时,还必须用 LTRIM()函数去掉字符串左边的空格。

四、其它问题

数据文件读入数据库后,仍存于磁盘上,若不及时删除,它们将在磁盘上无休止地增加。事实上,这些数据文件已完成其使命,存于磁盘上是不合理的,因此程序在写库后应有相应的删除功能。由于 FOXBASE 语言只能删除文件名已知的外部文件,而在运行软件包输入数据时还产生一些磁盘文件,这些文件供软件包数据库处理时调用。文件名部分地由用户自定义输入,如名为 INXXXXXXX.DAT,其中 IN 后面的六个字符对程序来说是未知的。因此,我们用 PASCAL 语言编写了一段程序,以完成删除文件名为 OUT*.DAT 与 IN*.DAT 数据文件。该程序调用了 DOS 单元的 FINDFIRST 和 FINDNEXT 过程。程序编译为 EXE 文件后,在 FOXBASE 环境下用 RUN 命令运行,主要部分如下:

```

VAR
DIRINFO:SEARCHREC
F:FILE OF REAL
BEGIN
FINDFIRST('C:\HY\*.DAT',ARCHIVE,DIRINFO);
O);
WHILE DOSERROR=0.DO
BEGIN
ASSIGN(F,'C:\HY\'+'DIRINFO.NAME');
ERASE(F);
FINDNEXT(DIRINFO);
END;
END.

```