

基于文本文件的高级语言(软件)数据共享问题

太仓职业高级中学教务处 黄兴无

摘 要 本文对计算机常用软件(如 C 语言、FOXBASE、FORTRAN、BASIC、PASACL 及 COBOL、等)间的数据共享与参数传递作了较全面的介绍,并重点介绍了数据共享的具体软件方法。

关键词 计算机 软件 数据 共享

一、问题的提出

程序员为了充分利用各高级语言(软件)的长处,经常使用几种语言来混合编程。在编程的过程中,碰到最多也最难解决的就是多种软件的数据共享、参数传递的问题。高级语言大多有自己的库文件,对数据的存储格式有各自不同的严格的要求,因此它们间不能直接共享数据,从而造成数据的大量重复录入。有没有办法能让常用的高级语言的数据相互调用呢?笔者做了许多尝试,下面详细介绍常用软件间数据共享的原理及其具体操作方法。

二、数据共享的常用方法

数据共享一般在下面两种情况下进行:一是两种特定的软件间进行,二是数据被多种事先不确定的软件调用。不同的情况要采用不同的方法。

对于第一种情况,一般采用直接读取数据的方法,这样较为直观、迅捷,也减少了交换文件,节省系统资源;但其缺点是由于不同的高级语言有各自不同的出口入口语句,因而数据共享的接口要依据特定的软件而定,造成软件不通用。

对于第二种情况,主要通过文件(如大多数高级软件都能访问的 TXT 文件)进行数据传递的方法来解决,其优点是能生成一个文本文件供大多数高级语言和应用软件所接受、所应用,缺点是这个生成的文本文件一来它要占用有限的系统资源,二来要对其做进一步的修改而不能直接被某些高级语言或应用软件所接受,而这种修改必须受要调用软件所能接受的格式的约束。

考虑在进行大量数据输入和数据处理时多采用数据库管理系统,且多数高级语言和应用软件都有各自读取文本文件的入口语句和输出文本文件的出口语句,因此本文就基于数据库管理系统(FOXBASE)来考虑数据共享问题,如果某种高级语言不是与数据库共享数据而是与另一种高级语言共享数据,也可以通过数据库管理系统中转而达到数据共享的目的。

三、数据共享的基本原理

我们知道:由 FOXBASE 系统生成的以 TXT 为扩展名的文本文件能与高级语言共享数据。它们最基本的特点是:

所有数据全部以 ASCII 码写成;每条记录物理连续,并以回车(ODH)、换行(OAH)为尾标;整个文件以(1AH)为结束标记。

例如:有一个名为 HXW.DBF 的数据文件,其结构如下:

FIELD	FIELD NAME	TYPE	WIDTH	DEC
1	HH1	CHARACTER	4	
2	HH2	NUMERIC	5	
3	HH3	NUMERIC	6	2
4	HH4	NUMERIC	6	2
5	HH5	NUMERIC	6	2

HXW.DBF 文件的内容是:

RECORD#	HH1	HH2	HH3	HH4	HH5
1	A001	4567	29.11	29.11	54.25
2	A002	1240	142.78	35.32	47.67
3	B001	802	34.24	125.60	78.50

对 FOXBASE 中打开的 HXW.DBF 文件,采用不同的命令格式可以生成下面三种文本文件:

第一:采用命令 COPY TO HXWSDF SDF,生成名为 HXWSDF.TXT 的同样有 3 条记录的数据文本文件,其格式如下:

A001	4567	29.11	29.11	54.25
A002	1240	142.78	35.32	47.67
B001	802	34.24	125.60	78.50

我们称为“标准数据文件”。每条记录的长度正好是 HXW.DBF 中对应字段长度之和(不含尾标),而每项数据自身的长度、小数点的位置,又分别与原库文件对应字段的规定完全吻合。在标准数据文件中,记录的各数据项之间没有任何分割标记,只能靠准确的对位识别它们。

标准格式的数据文本文件,可与多种高级语言共享数据。

第二:采用命令 COPY TO HXWDLI DELI,生成名为 HXWDLI.TXT 的数据文本文件,它以逗号分割记录中相邻的数据项(字符型数据用双引号标定起止界限)。其格式如下:

"A001"	4567,	29.11,	29.11,	54.25
"A002"	1240,	142.78,	35.32,	47.67

"B001" 802, 34.24, 125.60, 78.50

在这个文件中, 3 条记录所显示的长度不一致。数值性数据的前导空格、字符型数据的尾空格均被压缩掉。

这种格式的数据文件, 主要用来与 BASIC 程序共享数据。

第三: 采用命令 COPY TO HXWBLA DELI WITH BLANK, 生成如下格式的数据文本文件(HXWBLA.TXT):

A001 4567 29.11 29.11 54.25

A002 1240 142.78 35.32 47.67

B001 802 34.24 125.60 78.50

文件中, 3 条记录所显示的长度也是不一致的, 各数据项(不论数值型还是字符型)之间一律以一个空格作为分割符。

这种格式的数据文本文件适应于与 C 语言共享数据。

纵上所述, FOXBASE 主要采用上述三种格式的数据文本文件与各种高级语言共享数据。至于高级语言在各自的规定下, 生成的二进制数据文件、混合型数据文件等, 均不能为 FOXBASE 接受。因此, 要实现数据共享, 要求高级语言所使用的文件工作方式必须选择那些以 ASCII 码读写数据的方式才行。

四、常用软件的数据共享问题

1. 与 BASIC 语言共享数据

BASIC 可使用顺序和随机两种文件工作方式, 以顺序文件方式工作时 BASIC 程序能读入和生成的文件格式可以控制成 HXWDLI.TXT 文件。

采用 BASIC 的顺序文件工作方式编程时, 在 OPEN 语句中, 以" I" (顺序读) 或以" O" (顺序写) 打开的文件, 对数据文件的记录长度不必作出规定, 在读/写语句中(INPUT # / WRITE#)也只要把对应变量的类型规定正确即可, 而不必考虑每个变量的精确长度。

读入方法:

10 OPEN " I", # 2, "HXWDLI.TXT"

20 IF EOF(2) THEN CLOSE: END

30 INPUT # 2, HA\$, HB, HC, HD, HE

40 PRINT HA\$, HB, HC, HD, HE

! 数据处理段落(略)

150 GOTO 20

反之, 以顺序写" O" 方式生成 BASIC 的数据文件, 只要以 WRITE# 语句写入, 即可生成 HXWDLI.TXT 的文本文件。

2. 与 FORTRAN 语言共享数据

我们知道, FORTRAN 的各种语句中, 只有格式化的读写语句能传送 ASCII 码数据及记录。其格式顺序文件及格式直接文件均能与标准数据文件通用。因此只要程序中采用有格式的文件方式, 并把格式规定得与被其共享的数据文件一致就能使数据共享取得成功。

读入请参考下列程序:

```
CHARACTER * 5 X
OPEN (2, FILE= 'HXWSDF.TXT', ACCESS= 'DIRECT',
      FORM= 'F-ORMATTED', RECL= 27)
READ (*, 10) J
10 FORMAT (12) J
READ (2, 20, REC= J) HA, HB, HC, HD, HE
WRITE (*, 25) HA, - HB, HC, HD, HE
20 FORMAT (A5, I4, 3F6.2)
25 FORMAT (1X, A5, 1X, I4, 1X, F6.2, 1X, F6.2, 1X, F6.2)
CLOSE (2)
STOP
END
```

这段程序采用了有格式直接文件方式打开了 FOXBASE 传来的数据文件 HXWSDF.TXT, 并在 OPEN 语句中给出记录长 RECL= 27(各字段长度之和)。程序中标号为 20 的格式语句, 要原 HXW.DBF 库文件中各字段的定义完全一样, 程序中的变量 HA, HB... 分别对应于数据库中的字段。

读出的原理与上相同, 但最好采用有格式顺序文件方式。

3. 与 C 语言共享数据

C 语言的流式文件是 ASCII 码文本文件, 并且是以空格为分隔符。因此它的数据部分可方便地与 FOXBASE 数据文本文件进行交换。

现给出 C 语言编的源程序, 实现接收 FOXBASE 所产生的 HXWBLA.TXT 数据文本文件, 并从打印机输出。

```
MAIN (ARGC, ARGV)
INT ARGC,
CHAR * ARGV[]
{
    CHAR HA[5], HB[6], HC[7], HD[7], HE[7],
    REC [30] * P
    INT FD, N;
    FD= OPEN( ARGV[1], 0);
    IF(FD= - 1) EXIT(0);
    PIUNTF("HH1 HH2 HH3 HH4 HH5 \ N");
    FOR(;;)
    {
        N= READ(FD, REC, 29);
        IF(N(29) BREAK;
        P= REC;
        MOV(P, 5, HA)
        MOV(P+ 5, 4, HB)
        MOV(P+ 9, 6, HC)
        MOV(P+ 15, 6, HD)
        MOV(P+ 21, 6, HE)
        PRINTF("%5S %4S %6S %6S %6S \ N", HA, HB, HC,
                HD, HE);
    }
    MOV(P1, N, P2)
    CHAR * P1, * P2
    INT N3
    {
        DO {
            * P2+ 1= * P1+ ;
        } WHILE( - - N);
        * P2= ' \ O';
    }
}
```

4. 与 COBOL 语言共享数据

COBOL 与 FOXBASE 只能通过标准数据文件进行共享,但 FOXBASE 的数据文件既没有高位零,且小数点也不隐含。为了实现它们间的数据共享,就要消除数据排列上的差异。即把数值变量用数值编辑型的 PIC 子句来描述,以求在输出记录上显示小数点,并去掉高位零。这就需要在过程部中把相应的数值型数据用 MOVE 等语句传给数值编辑型变量。

下列程序的程序数据部作了如下的描述,它所写成 HXWSDF.TXT 文件就可以传送给 FOXBASE 的 HXW.DBF 库文件的标准数据文件。

```
FD OUT- FILE LABEL RECORD IS STANDARD
VALUE OF FILE- ID C:HXWSDF.TXT
01 O- RECORD
01 O- HA PIC X (4)
02 O- HB PIC- - 9.
02 O- HC PIC- - 99.9.
02 O- HD PIC- - 99.9.
02 O- HE PIC- - 99.9
```

程序中 O- HA、O- HB……各项对应原库的字段。它们构成的组合项 O- RECORD 与 HXWSDF.TXT 的记录格式完全相同。

同样我们只要对从 FOXBASE 传来的数值数据进行小数点的隐含,并填满高位零,就能使其在 COBOL 中实施运算处理。

由于 FOXBASE 的数据变量或字段均自动保留小数点占一位并消除高位零。因此,要隐含小数点并填满高位零必须利用字符串变量来作替换,即对于 HXWSDF.TXT,在数据交换开始之前,必须用一个中间文件作为替换的媒介,替换时,利用 FOXBASE 的 SUBS 函数及 STR 函数构成复合转换函数来去掉小数点及填充高位零,从而完成数据排列上的转换,其转换公式为:

REPL PX WITH SUBS(STR(SX/10N, N+ 2, N), 3, M)

式中: SX- - 转换的源变量

PX- - 转换的目标变量

N- - - SX 的结构描述的整数位数

M- - - SX 处小数点外的字长(即 SX 字长- 1)

但若传送的数据有负数,上述程序则不能完成符号传输。此时要在 COBOL 程序中对要接收的每条记录中的所有数据均分别按其长度,一律以字符型方式实现。然后在过程部中使用 UNATRNG(折串)语句将原数值数据折成整数、小数及符号三部分,再用 INSPECT(检测)语句把整数部分的高位空格或负号均置换为零。最后用 MOVE 及 COMPUTE 语句把原数值数据恢复成 COBOL 认可的数值数据。

5. 与 PASCAL 语言共享数据

PASCAL 与 FOXBASE 共享数据时,只能采用文本型文件(TEXT)为工作文件,两者共用标准型数据文件(SDF)。

下列程序片断,它将生成 HXWSDF.TXT 文件,其记录

可向 HXW.DBF 中追加:

```
PROGRAM PADBFILE(INPUT,OUTPUT);
VER
HA: STRING(4);
HB: INTEGER;
HC,HD,HE: REAL;
SX: TEXT;
FILENAME: STRING(12);
BEGIN
FILENAME:= 'HXWSDF.TXT';
ASSIGN(SX, FILENAME);
REWRITE(SX);
BEGIN
READLN (HA,HB,HC,HD,HE);
Writeln (HA,HB,HC,HD,HE);
Writeln (SX,HA,HB:5,HC:6.2,HD:6.2,HE:6.2);
END;
CLOSE(SX);
END
```

程序中 SX 定义为文本型,ASSIGN(SX, FILENAME) 语句打开了名为 HXWSDF.TXT 的文件,REWRITE(SX) 语句规定向文件写入记录。它由键盘输入数据,并按写语句 Writeln 规定的格式作为一条记录写入指定的 HXWSDF.TXT 磁盘文件中。

经分析生成的 HXWSDF.TXT 文件记录,发现各数值型数据之间都存在空格,这是因为上述程序是企图把读入的 HB、HC、HD、HE 直接作为整型数及实型数而不是先作为字符串读入后再进行转换,因此这种空格是必要的。为此,当我们要将 FOXBASE 的 SDF 数据文件转向 PASCAL 时,只要对上述程序略作修改,并满足数据项之间留有空格就能做到。

五、FOXBASE 程序与高级语言程序的交替执行

为使 FOXBASE 程序与高级语言程序按需要自动交替执行,一般采用以下几种方法:

1. 使用批处理文件

在 DOS 操作系统下,调用以 .BAT 为扩展名的文件,能轮流执行高级语言的可执行程序及 FOXBASE 的命令文件。

例如:我们需要先执行一段 HXW1.EXE 程序(一段 FORTRAN 的可执行程序),生成标准数据文件(SDF)后,转入 FOXBASE 下交给 HXW2.PRG 命令文件对数据继续处理,最后又要求把它新生成的有关数据文件的数据传送给 HXW3.EXE(C 语言的一段可执行程序)继续执行。为此我们可以编写下列内容的批处理文件(文件名为 SJJH.BAT):

HXW1

FOXBASE HXW2

HXW3

这样,我们只需在操作系统下键入 SJJH 就达到目的。

2. 在 FOXBASE 状态下使用 RUN 命令

格式: .RUN <DOS 命令或可执行文件>

使用 RUN 命令要有一个前提:即计算机必须有足够的内存,必要时还要对 FOXBASE 配置文件 CONFIG.FX 中的内存说明项(MAXMEM)加以说明,规定 FOXBASE 系统使用的最大内存,以保证其 RUN 命令能运行较大的高级语言程序文件。若内存达不到要求,则 RUN 命令无法执行。