

网络 MIS 数据冲突问题的解决^{*}

杨 幸^① 张银明^① 张谋东^②

(① 华侨大学电子工程系, 泉州 362011; ② 泉州烟草公司, 泉州 362000)

摘要 介绍在以 FoxPro for Window 为平台的网络环境中, 研制 MIS (Management Information System) 所要解决的重要问题之一, 即数据冲突问题的解决方法 .

关键词 网络, MIS, 数据冲突

分类号 TP 393

网络因数据可以共享而受欢迎, 越来越多的用户要求研制网络 MIS. 而 FoxPro for Window 具有很强的生成功能、极为令人心悦的界面及可用于多媒体而受到研制者的喜爱, 当然也深受 MIS 用户的青睐. 这使原来使用 FoxPro for DOS 的软件人员迅速地转 FoxPro for Window 作为平台进行 MIS 的研制. 然而, 数据共享可能引起的数据冲突是一个首先必须妥善解决的极其重要的问题, 也可以说是关键问题.

本文根据我们在研制中的经验和实际运行后证明有效的解决方法作些介绍.

1 隐式加锁问题

有些操作或命令在运行时是由系统自动加锁的, 这就是隐式加锁. 但是, 不能认为在程序中可毫没限制地使用这些命令. 相反, 应进行判别和必要的处理, 否则, 会使系统无法进行正常的运行而致使系统出错文件的记录大量增加. 因此, 在程序中需采取相应的措施.

1.1 APPEND BLANK 的正确使用

APPEND BLANK 是任何数据文件的记录登录时, 必然使用的语句. 按数据库管理语言的原定功能, 它具有自动对库文件头和当前记录加锁的能力, 但绝不是任何时候都可以随意地使用. 一旦该文件被其他用户锁住或独占时, 便会发生错误, 甚至无法正常运行. 为此, 在要对数据文件进行记录添加时, 必须先进行判断. 只有当文件没收有被他人独占或加锁, 而且经自己加锁成功后可进行 APPEND BLANK 的操作, 故需在程序中加上一段判断及加锁的处理.

假设要添加记录的数据文件为 DBFILE, 已在前面打开, 要添加的记录已存入数组 AD (N), 规定等待期限为累加数 M, 则该处理程序可编制为

```
SELE DBFILE
```

```
I= 1
```

```
DO WHILE .NOT. FLOCK( )
```

```
I = I + 1
IF I > M
    WAIT WIND "该文件已被其他用户占用! 请先进行其他操作!!"
    RETURN
ELSE
    IF FLOCK()
        EXIT
    ENDI
ENDI
ENDD
APPE BLANK
GATHER FROM AD(N)
UNLOCK
RETURN
```

这种处理方式, 即可以保证因数据文件被他人独占或加锁所引起的冲突, 而且可以解决 APPEND BLANK 及 GATHER FROM 两者所需加锁的要求. 如长时间被其他用户独占, 也可保证不至于盲目地等待. 但应注意, 记录添加后, 一定要立刻解锁, 以免影响别人的使用.

1.2 DELETED 及 REPLACE 的合理使用

居于 DELETED 及 REPLACE 使用范围不同, 需要加锁的范围也不同, 故应认真考虑. 一般而言, 在网络中要尽量体现数据共享的优越性, 亦即尽量让其他用户有更多的机会共享资源. 据此, 加锁的原则应该是能用记录加锁的, 便不用文件加锁, 能用文件加锁的, 不用独占. 因此, 程序中对记录的删除处理及修改记录部分字段内容而使用 REPLACE 语句, 可以使用记录加锁. 其算法如下: (1) 选择需处理数据文件的工作区及转向待处理的记录号; (2) 记录是否被他人加锁? 如果是, 转(3); 否则转(4); (3) 等待他人记录解锁, 如等待时间未超过规定时间, 转(2); 否则显示相应信息, 退出当前处理, 返回; (4) 对记录加锁, 进行 DELETE 处理或 REPLACE NEXT 1 操作; (5) 记录解锁, 返回.

假如有的用户需要在等待时间超过规定时限而退出时, 能提供一个 '重新加锁' 的机会, 以期待有新的成功机会. 要满足这种要求, 可将上述算法的(2) ~ (4), 改为(2) 记录是否被他人加锁? 如果是, 转(3); 否则转(5); (3) 等待他人对记录解锁; 如等待时间未超过规定时间转(2); 否则转(4); (4) 提示 "加锁失效", 并提供 "是否再作一次加锁的尝试(Y/N)?" 如选择 "Y", 转(2); 否则返回; (5) 对记录加锁, 进行 DELETE 处理或 REPLACE NEXT 1 操作; (6) 记录解锁, 返回.

2 文件独占方式

在网络中的 MIS 应尽量少用独占方式打开数据文件, 当一些操作不得不使用独占时, 也应力求使独占的时间最短, 必须以独占方式打开数据文件的有: (1) FoxPro 命令: PACK,

REINDEX, ZAP, INSERT[BLANK], MODIFY STRUCTURE, INDEX ON TAG(标签名);

(2) 专用文件; (3) 仅供用户个人使用的临时文件^[0,2]。

后两种居于专用性, 通常很少产生数据冲突, 因而我们主要讨论第一种情况, 其中最常用的是 PACK, REINDEX. 下面以 PACK 命令为例, 说明怎么编制较有效的程序, 即可使独占时间短, 又可解决冲突问题。

第一, 使用 PACK 命令对数据文件进行压缩处理时, 应力求使用集中方式, 如规定删除的记录数达到 M (根据需要而定) 时, 方进行 PACK, 尽量避免删除一个记录就进行一次压缩。这种集中处理方式可使文件独占的频率大大降低, 从而减少引起冲突的机会。

第二, 当删除的记录数达到 M 时, 在使用 PACK 命令的程序中要编制一段防止冲突而陷入死锁的处理程序。其算法如下: (1) 选择要处理文件的区号或别名; (2) 统计被删除的记录数是否大于等于 M , 如果是, 执行(3); 否则转(4); (3) 若发生出错, 执行 ERROR_PSC (出错处理) 程序; (4) 以独占方式打开要压缩的文件; (5) 如独占打开失效, 转(6); 否则执行(7); (6) 等待时间是否超过, 若尚未超过, 转(4); 否则转(8); (7) 进行 PACK 操作, 转(9); (8) 要否重试打开? 如要, 转(4); 否则继续; (9) 关闭当前文件; 返回。

这样处理可保证整个 MIS 的正确、可靠地运行, 不会因产生冲突而使系统运行中断或死锁。其中的出错处理程序 ERROR_PSC 将在后面介绍。

3 数据文件的正确打开

FoxPro for window 最多允许打开 255 个数据文件, 根据一般情况, 在 CONFIG.SYS 中设计 FILES = N 时, N 在 75 至 100 之间比较合适, 当然应根据系统的需要进行设置。

网络 MIS 是多用户的, 为避免两个或多个用户同时对一个记录或文件进行访问而发生冲突, 因而对数据文件的打开与访问应从整个 MIS 系统出发, 进行妥善的安排。一般而言, 比较有效的方法有三种。

(1) 整个系统公用的或使用频率较高的数据文件, 最好安排在前 N 个数据区, 如从 1 到 10 号数据区, 而且其打开的顺序在不同的程序中应该相同。例如, 有数据文件 DBFILE _{i} , DBFILE _{j} 是多个程序都要同时使用的, 则在使用这些文件的所有程序中皆以同样的顺序打开。比如按下列顺序

```
SELE  i                               SELE  i+ 1
USE DBFILEi                           USE DBFILEj
```

打开, 那么所有同时用到这两个文件的程序皆以这样的顺序打开, 而不以倒序的方式打开。

如果考虑到文件可能已被其他用户打开, 则可以使用下面的方式:

```
IF ! USED( 'DBFILEi' )                ELSE
  SELE 0                                SELE DBFILEi
USE DBFILEi alias DFILEi              ENDI
```

上述的别名 ALIAS 也可省略, 而直接使用文件名作为别名, 但文件名中不能含有中划线 ' - '。这种方式既可做到无须再次打开已被他人打开的文件, 又可确保按同一种顺序打开文件。

其次, 可以避免发生因一个文件多次打开而引起的

M FILE IS IN USE

的系统错误. 其中 M 是因一个文件被多次打开而被系统出错程序计算的次数. 这种错误虽不至于使系统产生严重的问题, 但则使出错处理程序的记录不断增多, 致使运行速度变慢.

(2) 统一分配整个系统数据文件的区号, 亦即如果系统有 N 个数据文件, 则在进行系统分析时, 不但要对各数据文件的结构、关键字表达式、索引文件名等进行说明, 而且要分配使用数据文件的区号, 就是第 i 个文件对应于第 I 个区号 ($i = 1, 2, \dots, n$). 则使用的形式为

```
SELE  I
```

```
USE DBFILEi
```

以后, 该数据文件无论在那个程序中要访问, 皆以

```
IF USED( "DBFILEi")
```

```
SELE  I
```

```
SELE DBFILEi
```

```
USE DBFILEi
```

```
ELSE
```

```
ENDI
```

这种整体分配的明显优点: (a) 决不会产生打开的顺序不同而引起的冲突; (b) 不会产生一个文件多次打开的系统错误; (c) 使数据文件的打开规范化、系统化, 从而给系统设计和系统调试带来极大的方便.

(3) 数据文件和记录加锁应遵循的一些基本原则: (a) 如果文件或该文件的某一记录被网络上的另一个加锁, 便不能对该文件加锁; (b) 如果文件或该文件的某一记录被网上的其他用户加锁, 则不能对该记录加锁; (c) 若加锁的文件关闭后需要再打开, 则文件打开后应立即上锁; (d) 以独占方式打开文件, 一旦所需操作执行完成后, 应立即转为共享方式; 同样对文件或记录加锁的命令执行完成后应随之解锁^[2,3], 以便让文件有尽可能大的共享性.

4 通用的出错处理程序

任何一个 MIS 系统, 不管大小, 在由设计、调试到运行过程中, 无论研制者如何尽心尽力, 还免不了隐含一些错误, 尤其是逻辑错误, 更难予觉察. 对多用户的 MIS, 还存在数据冲突问题. 因此, 设计一个“通用的出错处理程序”, 让它自动捕捉为记录系统出错信息而不中断系统运行, 一旦发生数据冲突时提示必要信息并作一定的等待处理, 而研制者又可从记录中查看发生的出错而进行及时的排错, 以保证系统的正确可靠. 因此, 一个 MIS 系统挂上通用的出错处理程序, 既是必要的, 也是具有明显的优越性.

4.1 出错处理程序的功能

该程序不妨命名为 ERROR_PSC.PRG, 其功能有: (1) 当发生文件或记录被他人占用时, 提示相应信息, 并进行重试; (2) 当文件需要独占又被他人占用时, 提示信息并进行等待, 如等待超过规定时间, 则提供处理选择; (3) 其他错误发生时, 进行出错信息的记录, 内容包括出错日期、时间、错误代码、出错信息、出错语句、所在行号、路径与程序名、最新发生错误的日期、出错次数等信息.

为记载这些出错信息, 需要设计一个数据文件, 其结构为

日期	时间	代码	出错信息	程序名及路径
<i>D</i>	<i>C</i>	<i>N</i>	<i>C</i>	<i>C</i>
8	8	4	40	30
COUNT	NEWDATE	NEWTIME	STATEMENT	LINENUM
次数	最新出错日期	最新出错时间	出错语句	新号
<i>N</i>	<i>D</i>	<i>C</i>	<i>C</i>	<i>N</i>
6	8	8	60	4

该文件可为研制者提供完整的系统信息, 以利于排错纠正处理.

4.2 出错处理程序的实现算法

(1) 由 ON ERROR 激活 ERROR _PSC 时产生的相关信息, 存入参数变量, 则 ERROR () ERRCODE, MESSAGE ERRMESS, SYS (16) ERRMODULE, MESSAGE (1) CODELINE, LINENO () LINEN.

(2) 如果 LEN (TRIM (ERRMODULE)) = 0, 则返回.

(3) 假如 ERRCODE = 108 或 ERRCODE = 109, 则提示 “文件或记录被其用户占用, 请等待!”, 并将累计时间置初值. 具体算法: (A) 累计等待时间, 若等待时间未超限, 转(B); 否则执行(C); (B) 进行加锁重试, 若加锁未成功, 转(A); 否则返回; (C) 提示 “加锁未能成功”, 并提供选择 “重试; 挂起; 忽略; 退出”; (D) 选重试, 则将累计时间置初值, 转(A); (E) 选挂起, 则执行 SUSPEND, 暂停程序执行; (F) 选忽略, 则返回; (G) 选退出, 则退出当前所执行的程序.

(4) 如果 ERRCODE = 1705, 则提示 “文件正被他人使用, 尚不能独占, 请等待”, 并置等待时间为初值. (A) 累计等待时间, 若等待未超时, 转(B); 否则执行(C); (B) 进行独占尝试, 如独占未成功, 转(A); 否则返回; (C) 提示 “独占打开未能成功!”, 提供选择 “重试; 挂起; 忽略; 退出”; (D) 选重试, 则将等待时间置初值, 转(A); (E) 选挂起, 则执行 SUSPEND; (F) 选忽略, 则返回; (G) 选退出, 则退出当前执行的程序.

(5) 其他系统出错, 保存当前执行程序的工作区号, 则保留下面的信息 SELECT () WKING, ‘SELECT + STR (WKING, 2) GOBACK. (A) 打开出错记录数据文件 ERROR _REC, 则

```
SELECT 0
```

```
USE ERROR _REC;
```

(B) 查找 ERRCODE = CODE 及 MODULE = ERRMOKULE 的记录; (C) 对 ERROR _REC 进行文件加锁; (D) 若加锁未成功, 转(C); 否则执行(E); (E) 若查到(B) 所找的记录, 则对记录进行更新替换; COUNT 置为 COUNT + 1, LINENUM 置为 LINEN, NEWDATE 置为 DATE (), NEWTIME 置为 TIME (); 转(G); 否则执行(F); (F) ERROR _REC 增加一个记录, 其内容为: 1 COUNT, LINEN LINENUM, DATE () DATE, TIME () TIME, ER-
RCODE CODE, ERRMESS MESSAGE, ERRMODULE MODULE, STATEMENT

CODELILE;(G) 文件解锁,并闭 ERROR _ REC 文件 ;(H) 恢复出错程序所选用的工作区,则 & GOBACK;(I) 返回 .

该出错处理程序执行中具有人工干预的选择,如认为无须这种干预,则可将这部分去掉.从上述算法中可看出这是一个较完善的,具有较强功能的通用出错处理程序.只要按算法写出相应的语句,便是一个完整的“通用出错处理程序”.限于篇幅,此不赘述.

4.3 ERROR _ PSE . PRG 的使用

任何 MIS 皆有总控程序,只要在总控程序头加上一句

```
ON ERROR DO ERROR _ PSC WITH ERROR(), MESSAGE(), ;
SYS( 16), MESSAGE( 1), LINENO()
```

便可使总控所调用的各个模块的下属程序全在此出错处理程序的作用下.它既可为系统的查错纠错提供极大方便,又为整个系统的可靠运行起到重要的保证作用.

5 结束语

数据冲突问题是多用户 MIS 的一个极为重要的问题,也是必然存在的问题.这个问题的不断探索和研究,将有利于软件研制人员编制出更完善可靠的多用户 MIS. 本文对此问题的探索希望能促进这个难题的有效解决.

参 考 文 献

- 1 刘金文,张 研. 高级编程技巧. 北京:中国铁道出版社,1995. 213 ~ 223, 248 ~ 282
- 2 史济民. Foxbase+ 1 及其应用系统开发. 北京:清华大学出版社,1995. 284 ~ 297
- 3 陈明德,倪 朝. NOVELL 网络实际操作. 北京:人民邮电出版社,1996. 166 ~ 264

The Solution of Data Clash for the Development of Management Information System in a Network Environment

Yang Sing^① Zhang Yinming^① Zhang Moudong^②

(^① Dept. of Electron. Eng., Huaqiao Univ., 362011, Quanzhou;

^② Quanzhou Tobacco Co. Ltd., 362000, Quanzhou)

Abstract For the development of management information system in a network environment with FOXPRO for Window as platform, the authors solve the the problem of data clash as one of important problems to be solved.

Keywords network, management information system, data clash