

浅析 FoxBASE + 中的容错处理技术

张克友 (河南信阳师范学院财务处 464000)

摘要:本文探讨了程序运行中可能发生错误的容错处理技术,包括对错误的检测方法和处理方式。最后给出了测试法容错处理的实例。

关键词:FoxBASE 容错处理

对于已经通过调试、试运行、并已投入实际应用的程序来说,一般不应再有错误。但这只是对程序本身而言。至于因运行环境、外部条件等方面的因素而引起的错误,则往往是在所难免的。程序在运行中发生错误,不仅会导致运行过程中断、数据库受到破坏等直接的后果,而且可能产生一些比较隐蔽的间接后果,对用户造成更具威胁性的危害。因此对一个好的程序来说,我们不仅要求它本身正确无误,而且要求它能防止或排除各处因外部因素引起的错误,不让这些错误影响它的正常运行,也不让数据库受到丝毫损坏,也就是说,它应有良好的容错处理能力。由此可见,容错处理的设计是程序设计中的一项十分重要的内容。

容错处理一般由对错误的检测和对错误的处理两部分构成。

一、对错误的检测

在程序运行中,可能发生的错误大致有以下几种:

1. 要访问的文件不存在。可能该文件已被误删除,或者在其他磁盘上。
2. 要访问的文件是错误的。例如数据库的索引文件被破坏后,与数据库不匹配。
3. 外部设备没准备好,例如打印机没准备好。
4. 外部存储空间不够。例如软盘上所存文件很多,剩余空间很少,无法再装入新的文件。
5. 数据溢出。这种错误是由数据本身而非外部原因引起的,因此在程序调试阶段就应当排除。但也有可能因客观情况的变化,超出原设计要求所致。数据溢出的情况一般有如下几种:数据运算的结果太大,以致输入数据时字段宽度不够,或者打印报表时该栏的格数不够;以

零或非常接近零的数作除数;函数自变量的值超出该函数的定义域。

6. 多用户环境中容易发生以下几种冲突:

- ① 试图打开一个已被其他用户以独占方式打开的数据库文件;
- ② 试图以独占方式打开一个其他用户已打开的数据库文件;
- ③ 试图对一个非独占方式打开的数据库文件,执行必须以独占方式运行的命令;
- ④ 对一个非独占方式打开的数据库文件,不经加锁就执行 READ 命令或(不带〈范围〉子句和 FOR/WHILE 子句的)REPLACE 命令;
- ⑤ 试图对已被其他用户加锁的数据库文件或记录,执行具有自动文件加锁或自动记录加锁功能的命令。

7. 用户在操作上的错误。例如误击键、未按规定的步骤操作等。此类情况导致的往往是不正确的结果,而处理过程的本身却是正确的,因而不会被 FoxBASE + 系统所发现。因此,此类错误比较隐蔽,所造成的危害往往也较大。

对上述各种错误的检测方法主要是测试法和陷井法。

① 测试法:通过 FoxBASE + 的各种测试函数,为将要进行的各种操作,检测其必需的前提条件是否已满足。如果未满足,就要采取相应的预防措施,否则该操作进行时就会发生错误。因此,这是一种对错误进行预测、预防的方法。这种方法是十分有效的,在一些重要的数据处理过程中,或者在预测可能出现错误的地方,都可用这种方法来预测和预防错误。

FoxBASE + 系统中的测试函数有四、五十之多,绝大多数都可用于对错误的检测和预防,其中最常用的函数

有:

FILE()	检测文件是否存在
BOF()	检测记录指针是否已到数据库文件头
EOF()	检测记录指针是否已到数据库文件尾
FLOCK()	检测数据库文件状态,并对其进行加锁
RLOCK()	检测数据库当前记录状态,并对其进行加锁
TYPE()	检测表达式类型
DISKSPACE()	返回磁盘自由空间的字节数
SYS(12)	返回当前可用内存的字节数
SYS(13)	检测打印机状态

测试法的不足之处在于检测能力受函数功能的限制,有些错误特别是多用户环境的错误,是无法用函数来预测的。

②陷井法:用 ON ERROR 命令设置捕捉错误的陷井,错误发生时程序就可抓住它,并通过几个专用的测试函数来了解出错情况,然后便可转入对错误的处理阶段。陷井法与测试法的不同之处在于,它所检测到的是已经发生的错误,而不是将要发生的错误;它是作为一种系统状态出现的,一旦建立后,只要不撤消就一直有效;它的检测能力很强,是多用户环境中必用的容错处理手段(在多用户环境中发生的冲突中,有些只有用陷井法方能检测到),但是它无法检测因用户的误操作而引起的“隐蔽错误”。

ON ERROR 命令的格式是:

ON ERROR <<命令>>

其中<命令>一般多采用 DO<错误处理程序>WITH<参数表>的形式,以便在错误发生后,自动转去执行错误处理程序。<参数表>中一般多采用下列函数,使错误处理程序得以正确判断出错情况:

ERROR()	返回出错号码
MESSAGE()	返回出错信息
MESSAGE(1)	返回出错的语句
SYS(16)	返回出错的程序名

错误处理一般多采用分支结构,各分支以错误号作为条件,根据不同的出错号,执行相应的程序段,执行后以 RETRY 或 RETURN 语句返回刚才出错的程序。RETRY 和 RETURN 的区别是 RETRY 把控制交回原程序后,重新执行刚才出错的语句,而 RETURN 则是执行刚才出错语句的下一个语句。

错误处理程序的典型形式如下:

```

* * * 错误处理程序:ERR.PRG * * *
* * * 陷井:ON ERROR DO err WITH ERROR(),
MESSAGE(), MESSAGE(1), SYS(16) PARAMETERS
err no, mess, source, prgfile
.....
DO CASE
CASE err - no <错误号码 1>
<错误处理程序段 1>
CASE err - no <出错号码 2>
<错误处理程序段 2>
.....
CASE err - no <出错号码 n>
<错误处理程序段 n>
.....
ENDCASE

```

ON ERROR [<命令>] 应当用在可能出现错误的程序之前,如果在 ON ERROR 中不指定<命令>,则撤消已设置的 ERROR 陷井。

③对“隐蔽错误”的检测。上述测试法和陷井法对于由用户的误操作而引起的“隐蔽错误”往往不能奏效,需要编制特定的程序段或子程序,根据预定的规则来检验当前或先前的处理的结果是否符合要求。例如:如果规定从键盘输入变量 ZH 的字符必须是字母 A - D,那么用户违反规定的误操作可以用语句 @... GET zh VALLD LEFT(ZH, 1) \$ "AaBbCdDd" 中的 VALID 子句来检测,却无法用陷井法来发现。

二、对错误的处理

在 FoxBASE + 中对错误的处理方式大致有以下几种:

1. 自动处理:对于因数据库本身而非外部因素引起的或将要引起的错误,例如字段宽度不够,索引文件与数据库不匹配等,一般可由程序自动进行处理,如放大字段宽度,重新进行索引等,消除出错因素,便可继续正常的运行。这一处理方式可以使应用程序不仅具有很强的容错能力,而且具有很强的适应性,可以减少程序的适应性维护工作。

2. 人工干预:对于因外部设备的原因而引起的或将要引起的错误,例如打印机没联机,磁盘的自由空间太小

等,一般可在屏幕上通知操作者,请他进行人工干预,例如将打印机联机,重新更换磁盘等,消除出错因素后,便可恢复正常运行。

3. 等待许可:如果引起出错的原因可能消失,那么可以不作任何处理而等待该因素消失,到允许正常运行时再恢复运行。所谓等待,实际上就是不停地进行检测,以便及时知道出错因素是否消失了。这一方式一般多用于多用户环境中的冲突问题。等待方式又有无限等待和有限等待两种,后者是考虑到如果出错因素一直不消失,总不能一直等下去,因而规定一定的检测次数,到时如还不消失,便放弃引起或将要引起错误的操作。

4. 放弃操作:如果引起出错的因素在当时情况下不可能排除,或者等待了一段时间仍未消失,或者引起错误的操作无关紧要,那么就可考虑放弃操作。放弃操作的做法有以下四种:

①只放弃引起出错的那个语句的执行,条件是放弃后不会影响后续语句的执行;

②放弃引起出错的那个程序模块的执行,条件是放弃后不会影响调用该模块的上级程序的继续执行;

③直接退回最高一级的主程序,条件是返回后不会影响主程序的继续执行;

④如果上述三种选择都不可行,也就是说,已发生的或将要发生的错误是无法排除的,或者继续执行将造成严重后果的,或者其他需要立即终止运行的情况,就应用 CANCEL 命令终止整个应用程序的运行,返回到交互命令状态,或者用 QUIT 命令返回到操作系统状态。

上述四种处理方式可以分别使用,也可以一起使用,有时对同一类的错误,由于发生时的环境不同,或者用户的要求不同,所采取的处理方式也不同。因而往往需要对环境、条件或用户的要求进行判断,并随之选择相应的处理方式。这一判断选择工作,可以由程序自动进行,也可以通过人机对话形式由操作者来完成。

这四种处理方式都可以与前述的测试法和陷井法构成完整的容错处理功能。

三、测试法容错处理的实例

1. “自动处理”的实例

.....

USE gz

IF! ("GZ.IDX") && 打开索引文件前先检验其是否存在。如不存在

INDEX ON 编号 TO gz && 则建立索引文件
ELSE

SET INDEX TO gz

ENDIF

.....

2. “人工干预”的实例

.....

DO WHILE SYS(13) = "OFFLINE"

WAIT "请把打印机准备好,然后按任意键!"

ENDDO

SET PRINT ON

.....

3. “(无限地)等待许可”的实例

.....

DO WHILE .NOT. FLOCK()

@ 4,10 SAY "当前文件正被他人锁闭,请通知他尽快解锁,我们正在等候!"

ENDDO

BROWSE

.....

4. “(有限地)等待许可”的实例

.....

n=0

DO WHILE .NOT. RLOCK(). AND. N < 999 & 最多重试 999 次

@ 1,10 SAY "当前记录正被他人使用,请稍候片刻!"

n=n+1

ENDDO

IF! RLOCK()

? "当前记录正被他人占用,不再等待!"

RETURN

ENDIF

.....

(来稿时间:1997年9月)

★ 书讯 ★

《AS/400 开发工具集》(第二集)已出版,每本定价 360 元,另加邮资、包装费 10 元,共计 370 元。

欲购者请汇款:

开户行:工行北京市支行海淀镇分理处

帐号:891537-80

户名:中国计算机用户协会 IBM 机分会

地址:北京市 2719 信箱 IBM 协会办公室

联系人:张燕萍 邮码:100080

电话:(010)62554390 传真:(010)68533376