

# 应用软件系统的故障恢复

张 银 明

(计算机科学(电脑)系)

## 摘 要

本文论述了一个应用软件系统可靠性的重要组成部分——故障恢复,并提出了实现故障恢复的两种方法——系统恢复及中断恢复。

## 一、前 言

计算机技术及电子技术的飞速发展,使硬件系统的稳定性和可靠性有了极大提高,这给软件系统提供了更为理想的运行环境。但并非硬件故障就此消失了。故障控制、诊断系统、容错纠错技术等正是针对可能发生的故障而研制的。而软件研制在经受了“危机”的震动之后,使人们重新研究程序设计的一些最为基本的问题。软件专家们为提高软件质量,确保系统的可靠性、可维护性和正确性等而作了一系列的研究,并建立了一些最基本的理论,提出了不少极有价值的方法。诸如程序正确性证明、算法正确性证明、软件可靠性评价的基本理论、软件质量度量、软件工程理论及模块化程序设计、程序测试方法、程序设计方法学等等。尽管软件的研制既有理论上的依据,又有方法上的指导,但至今为止,尚不能完全保证一个软件系统在投入运行时是正确无误的。即使是那些经过较长时间运行而被认为较为成熟的系统,也还可能含有错误。至于通常研制的应用软件系统,不可避免地含有某些尚未被测试出来的隐患。

这都说明,无论硬件或软件,故障发生的可能是存在的。因而给应用软件系统提出一个值得考虑的问题,亦即当硬件故障、软件出错、人为因素或外界环境所造成的故障一旦发生时,将会给应用软件系统带来什么样的后果,以及为消除这种影响所应具有的能力。这就是本文要讨论的应用软件系统的故障恢复。

## 二、可靠性的重要组成部分——故障恢复

软件质量度量SQM(Software Quality Metrics)虽对不同类型的软件,其质量要素和评

本文1987年2月10日收到。

价准则各不相同、有所侧重,但都把可靠性和正确性作为重要的质量要素。可靠性和正确性虽含义有别,但互为依存。正确性是可靠性的前提,可靠性是正确性的进一步保证。从软件工程的角度而言,贯穿其全过程的重要课题便是软件可靠性理论及其实践。应该说,一个应用软件系统的关键是可靠性,否则,其它要素再好,也是建立在不可靠的基础之上的。

一个系统的正确性指的是程序输出满足原定目标的程度。显然,要保证输出的正确先应确保处理对象的正确。在DBMS中要确保数据的完整性,而对一个数据处理系统,面对大量多样化的数据信息,要保证这些处理对象的安全正确,则是件重要且有必要认真考虑的任务。

在异常情况下的可靠性则表现为系统的健壮性,亦即要求系统在发生违反规范条件的情况下,程序仍能继续执行,则具有一定抵御恶劣环境条件的相应能力。目前不少程序设计者抱着迫切的心情希望能使所设计的系统尽早投入运行,以至较多地着眼于程序本身和处理结果的正确性,从而没有能在调试中设置各种故障情况,因而对在这种条件下保证系统正常运行的能力注意不够。一旦发生故障,就可能给系统带来灾难性后果。

实际上运行环境远非理想。硬件事故、突然断电、操作失误、强迫停机以及程序出错等因素都可能使系统运行中断,而这导致的后果可能有:(1)数据文件被破坏;(2)修改处理被中断;(3)计算统计或查询输出被中断;(4)程序系统被破坏。在一般情况下,后两种因不涉及数据文件中的信息更动,所以,只要重新启动或程序拷贝便可消除所引起的影响,而使系统恢复正常。但前两种如没有相应的恢复措施,将使整个数据系统引起混乱,甚至导致系统崩溃。

例如,在合同、销售与库存综合管理系统中,有配件库、订货合同、本厂合同、外购合同、内外进库单、各种出库单、退货、报废及合同注销等20多个数据文件。为进行综合管理,在对某个数据文件的数据进行更新时,与其有关的数据文件的相应数据也将作相应的更动。如对进库单文件进行记录的增删改维护时,将对配件库、本厂合同、订货合同等数据文件进行修改。若在修改进程中被故障中断,而系统又没有能力对此进行恢复,必将引起数据混乱。假如故障使某些数据文件遭到破坏,而系统又无此恢复功能,那么整个系统将无法继续使用,最终导致崩溃。

综上所述,一个应用软件,尤其综合管理或数据处理系统,若没有故障恢复能力,则该系统是不完善的,也难于称为可靠的。这就是说,故障恢复功能是系统可靠性不可忽视的重要组成部分。

### 三、故障恢复方法

由于故障恢复是应用软件可靠性和正确性得到保证的重要手段之一,所以在研制时,必须针对故障可能造成的后果采取相应的软件措施,这就是故障恢复方法。根据故障导致后果的分析,可把恢复方法分为两种:一是对引起数据文件受破坏的故障所作的恢复,叫做系统恢复或数据恢复;二是因故障致使处理中断的恢复,称为中断恢复。

#### 1. 系统恢复

系统恢复可使用日志文件与数据备份相结合的方法。一个应用系统的数据文件之间的关

系可分为两种：一是各自独立，记录的增删改同其它数据文件无关；二是互有联系，其记录的更动需修改其它数据文件。前一种关系的系统，当故障致使数据文件受损时，可根据遭破坏的不同程度而进行整个系统的恢复或部分恢复，亦即只对受破坏的数据文件进行恢复。因这种关系简单、恢复方便，故不多加讨论，而主要分析互有联系的数据文件所组成系统的恢复方法。

由于数据文件之间互有联系，所以在它们建立时有个顺序问题。如在合同、销售与库存综合管理系统中，建立数据文件的先后顺序应为配件库、本厂合同、外购合同、订货合同、内外进库单、各种出库单，最后是退货、报废与合同注销等等。其中有些可先后对调，如内外进库单。而有些则不能，如配件库必须在所有其它数据文件之前，出库单应在进库单之后。据此，在进行恢复时，也必须遵循相应的顺序，否则便无法进行正确的恢复。这就是说，如果在数据文件建立时要按一定的顺序，那么恢复时也应遵循相应的顺序。

居于各数据文件结构不同，处理涉及的对象不同，而且维护过程有插删改之别。因而记日志文件的较合理方式是对每个须输入和维护的数据文件各自建立日志文件，并对增删改分别加以标志。其记录的结构形式可以为：

FIELD 1, FIELD 2, ..., FIELD n, FLAG

其中FLAG为标志位，维护标志设为

FLAG = {	“I”	增加或插入；
	“D”	删除；
	“C”	修改。

为使恢复限制在一定时间范围之内，所以必须定期或不定期地对数据文件作备份。这个周期 $T$ 或时间区间的大小可权衡事务处理的频率、数量、存贮日志文件介质容量以及恢复可能耗费的时间等因素而定。 $T$ 如较大，备份次数则较少，但日志文件较大，所以恢复时间将长一些。反之， $T$ 取小，那么备份次数增多，耗时也多，但日志文件较小，因而恢复时间也将短一些。通常，这种破坏性故障发生的概率很小，所以 $T$ 可适当取大一点，以免为经常进行备份而花费大量时间。

一旦在时间 $i(i \in [1, T])$ 发生破坏性故障，那么，便可以使用数据备份及 $t=1, 2, \dots, i-1$ 时间单元期间的日志文件，按一定顺序进行系统恢复。假设系统有几个数据文件及对应的日志文件，分别表示为 $dfile(K)$ 及 $Rdfile(K)$  ( $K=1, 2, \dots, n$ )。则系统恢复的算法可描述如下：

(1)  $K \leftarrow 1$ ;

(2) 从备份盘把 $dfile(K)$ 复制到系统盘；

(3)  $K \leftarrow K+1$ ，若 $K \leq n$ ，转2；

(4)  $K \leftarrow 1$ ;

(5) 打开日志盘上的 $Rdfile(K)$ 及系统盘上的文件 $dfile(K)$ ；

(6) 读取 $Rdfile(K)$ 的一个记录；

(7) 若FLAG = “I”，则执行数据文件记录增加所应处理的过程，并把该记录加入 $dfile(K)$ ，转10；

(8) 如FLAG = "D", 则在dfile(K)中找到该记录, 执行记录删除相应的处理过程, 并删去dfile(K)文件的当前记录, 转10;

(9) 若FLAG = "C", 则在dfile(K)中查到该记录, 并以此记录各属性域之值与日志文件记录相对应的属性域之值的差值进行相应的修改处理, 且更新dfile(K)当前记录的属性域值;

(10) Rdfire(K)指针加1, 如文件未结束, 则转6;

(11) 删去Rdfire(K)文件, 关闭dfile(K);

(12)  $K \leftarrow K + 1$ , 若 $K \leq n$ , 则转5; 否则结束。

显然, 系统恢复过程是从数据备份开始到受破坏时为止的时间( $t = 1, 2, \dots, i-1$ )之间处理过程的重演, 使整个系统的数据文件恢复到受破坏前的状态, 从而达到恢复的目的。当然整个恢复过程是较为费时的, 但它使用户不必重做该时间范围内所有输入和维护的处理过程而省下大量时间和精力。何况, 由于用户往往没有认真整理处理过的数据资料, 一些单据也非规范化, 致使重复执行原来的过程变得极为困难, 甚至无法实现或产生较大差错。而日志文件则如实地记下原来所有处理过程的信息, 因此可以恢复到受破坏前的系统状态。

## 2. 中断恢复

因故障而引起处理中断比导致数据文件受破坏的概率要大得多, 所以, 对中断故障更应采取相应的恢复措施。

由于数据文件之间的联系, 所以, 当一个数据文件进行记录的增删改事务时, 要对多个与其有关的数据文件进行相应的更新处理。若在更新过程中因故障而中断, 那么要靠人工分析找到断点几乎是不可能的。因而必须采用软件手段自动确定中断位置, 并由此开始继续执行尚未完成的处理任务, 这就是中断恢复。

比如要给MDF(Main Data File)文件增添 $L$ 个记录, 在输入时暂存在TDF(Transaction Data file)文件之中, 并由它的记录去修改 $M$ 个与之有关的数据文件之后加到MDF文件。那么, TDF文件每条记录在此处理过程中可能的状态有: (1)该记录尚未使用; (2)该记录正在使用之中, 亦即在修改 $M$ 个文件中的某一个; (3)此记录已用过并加进MDF文件; (4)在TDF中已对该记录作了删除标志。这样, 一条记录在整个处理过程至少有 $(M+3)$ 种不同状态,  $L$ 条便有 $L(M+3)$ 种状态, 假设 $L=100$ ,  $M=5$ , 便有800种不同状态。所以中断时要人工查找断点是不现实的, 尤其没有处理前后各数据文件的数据作对比, 也无法判断哪些文件的记录被修改过。所以, 只有使用中断恢复的软件手段才能得到解决。

中断恢复方法的基本思想是给事务(或临时)文件的记录设置一个标志位。刚登入的记录可置空, 当使用它去更新其它数据文件时, 每做完一步便标以一种标志, 而把该记录存入主数据文件后, 便置删除记号, 直至全部事务记录存进主数据文件后, 就删除事务文件的所有记录。这样, 一旦发生中断, 便可对标志位的判别来确定中断位置。当系统重新运行时, 可先判断事务文件是否存有记录, 如有, 则说明上次处理被中断, 便由标志位的分析找到断点位置, 再由此继续未完成的处理工作, 以实现中断恢复的目标。若事务文件不含记录, 说明前次处理正常结束, 便可进行其它所需的处理。

假设事务文件为TDF, 需更新的 $M$ 个文件为DFILE(I) ( $I=1, 2, \dots, M$ ), 最后存

入的主文件为MDF。TDF文件的记录标志位FLAG表示成:

$$\text{FLAG} = \begin{cases} \text{“ ”} & \text{刚登记的事务文件记录;} \\ \text{“I”} & \text{第I个文件已修改结束;} \\ \text{“*”} & \text{该记录已存入MDF文件。} \end{cases}$$

其中 $I = 1, 2, \dots, M$ 。

这样, 中断恢复的算法可描述如下:

- (1) 打开TDF文件, 若不含记录, 转12, 否则 $M_1 \leftarrow M$ ,  $RN \leftarrow 1$ ;
- (2) 文件中含有FLAG = “\*”的记录? 如没有, 转4;
- (3) 查找FLAG = “M”的记录。若有, 其记录号送RN, 转10, 否则转11;
- (4) 文件中含有FLAG = str( $M_1$ )的记录? 有, 转6;
- (5)  $M_1 \leftarrow M_1 - 1$ , 若 $M_1 \geq 1$ , 转4; 否则转12;
- (6) 含有FLAG = str( $M_1 - 1$ )的记录? 如有, 其记录号送RN,  $M_1 \leftarrow M_1 - 1$ ;
- (7) 打开Dfile( $M_1$ )文件;
- (8) 从TDF文件的第RN号记录开始, 用其数据对Dfile( $M_1$ )文件作相应修改, 并置TDF当前记录的FLAG = str( $M_1$ ), 重复直至TDF文件结束。关闭Dfile( $M_1$ )文件;
- (9)  $M_1 \leftarrow M_1 + 1$ , 若 $M_1 \leq M$ , 则 $RN \leftarrow 1$ , 转7;
- (10) 打开MDF文件。TDF文件第RN号开始的记录加入MDF文件, 并置TDF文件的当前记录FLAG = “\*” (或作删除标志);
- (11) 删去TDF文件的所有记录, 关闭文件;
- (12) 进行所需的处理或返回。

由于中断恢复的处理过程同正常情况下用TDF文件的记录更新其它数据文件的过程是一致的, 区别仅在于中断恢复的第一次更新从记录号RN开始, 而正常情况下从 $RN = 1$ 号记录开始, 算法中从RN号开始是为了两者共用处理程序, 其中的str( )是取字符函数记号。若用dBASE II或III编程, FLAG = “\*”可使用记录删除标志“\*”代替。

对于需要长时间的计算或统计过程, 可采用周期地或阶段性保存中间结果及有关参数的方式, 当发生中断时, 便可从保存的中间结果及参数作为继续进行计算或统计的起点以完成被中断的过程, 以免浪费不得不从头开始所花费的大量时间。

系统恢复和中断恢复的算法经实践证明是可行的。

#### 四、实 例

根据上述两个恢复算法的思想, 现以“合同、销售与库存综合管理系统”的内进库单恢复为例, 以说明算法的使用。

假设系统的配件库、订货合同、本厂合同以及内进库单等数据文件分别取名为FRDF (Fittings Reserve Data File)、OGCDF (Order Goods Contract Data File)、TFCDF (This Factory Contract Data File)、ICVDF (Inner Consignment Vouchers Data File), 并假定事务文件为TICV, 日志文件为DICV, 且约定系统盘为硬盘C, 备份盘和日志盘的

驱动器为A。

## 1. 系统恢复

从备份盘把数据文件复制到系统盘是容易实现的。若使用dBASE编程,那么只要使用命令

RESTORE A : C :

便可把原来使用命令BACKUP C : \* . DBF A : 进行备份的所有数据文件,依次复制到C盘,这当然可调用带有后缀.BAT的命令文件来实现。

现假定系统的全部备份的数据文件复制完成,那么,关键问题是如何应用日志文件把备份的数据文件复原成被破坏前的状态。由于要进行综合管理,所以,在内进库单进行增删改时,必须修改配件库,标志本厂合同完成情况,以及查询订货合同并输出必要信息。其恢复算法如下:

- (1) 打开日志盘上的DICV文件,若不含记录,则转10;
  - (2) 打开系统盘上的ICVDF文件;
  - (3) 读取DICV的一条记录,把该记录的机型十图号送TMT (Type of Machine Tool);
  - (4) 若FLAG = "I", 则按TMT查找FRDF文件的相应记录,将有关数据项加上DICV文件当前记录的相应数据;接着以TMT查TFCDF文件,根据进库的数据标志合同完成情况;再查询OGCDF文件的订货情况,根据需要输出发货信息;最后把记录添入ICVDF文件,转7;
  - (5) 如FLAG = "C" 则按TMT查到ICVDF文件的记录,并以两个记录对应数据项之差值修改FRDF文件中相应记录的有关数据;更改TFCDF文件原来所标志的完成情况;若修改结果数据值增大,便查询OGCDF文件,输出相应的发货信息;最后更新ICVDF文件的记录,转7;
  - (6) 若FLAG = "D", 则按TMT查ICVDF文件的相应记录,以此修改FRDF文件相应记录的有关数据;更改TFCDF文件有关记录的完成情况;删除ICVDF文件的当前记录, RM ← "D",
  - (7) 指针转向DICV的下一条记录,如果尚未结束,则转3;
  - (8) 删去DICV文件的全部记录;
  - (9) 若RM = "D", 则对ICVDF文件进行记录压缩;
  - (10) 关闭所打开的文件,返回。
- 其它数据文件只要按顺序进行相应的恢复过程,便可使整个系统恢复到被破坏前的状态。

## 2. 中断恢复

若在对内进库单进行记录增添或维护过程中发生故障而中断,便须进行中断恢复。其算法如下:

- (1) 打开TICV文件。若文件为空则转12; 否则 $M_1 \leftarrow 3$ ,  $RN \leftarrow 1$ ;
- (2) 文件中含有FLAG = "\*" 的记录? 如无,则转4;
- (3) 还含有FLAG = "3" 的记录,则把该记录号送RN, 并转9; 否则转10;
- (4) 文件TICV中含有FLAG = str (M<sub>1</sub>, 1, 0) 的记录? 如有,则转6;
- (5)  $M_1 \leftarrow M_1 - 1$ ; 若 $M_1 \geq 1$ , 则转4; 否则转12;

(6) 如文件TICV中含有 $FLAG = \text{str}(M_1 - 1, 1, 0)$ 的记录, 则把记录号送RN, 转 $6 + M_1$ ; 否则转 $7 + M_1$ ;

(7) 打开FRDF文件。从TICV文件的RN号记录开始, 以机型十图号查找FRDF的相应记录, 并作相应修改; 且置TICV文件的当前记录的 $FLAG = "1"$ ; 重复此过程, 直至TICV结束; 关闭FRDF文件,  $RN \leftarrow 1$ ;

(8) 打开TFCDF文件。从TICV文件的RN号记录开始, 以机型十图号机找TFCDF文件的相应记录, 标志合同完成情况; 并置TICV文件当前记录的 $FLAG = "2"$ ; 重复此过程, 直至TICV文件结束; 关闭TFCDF文件,  $RN \leftarrow 1$ ;

(9) 打开OGCDF文件。从TICV文件的RN号记录开始, 按机型十图号查询订货情况及库存量, 输出有关发货信息; 置TICV文件当前记录的 $FLAG = "3"$ ; 重复此过程直至TICV文件结束; 关闭有关文件,  $RN \leftarrow 1$ ;

(10) 打开ICVDF文件。从TICV文件的RN号记录开始, 把记录添加入ICVDF文件, 并置TICV文件的当前记录的 $FLAG = "*" \text{ 或作删除标志}$ ; 反复此过程, 直至TICV文件结束; 关闭ICVDF文件。

(11) 删去TICV文件的全部记录, 关闭相应文件;

(12) 返回。

从内进库单恢复例子中可看到算法的具体实现。设计一个应用系统时, 可把系统恢复作为一个功能模块。根据系统规模及恢复耗时估量来决定采取一次性恢复或分阶段恢复, 并由一个控制程序进行顺序控制及其它处理, 如恢复后建索引文件等问题。为具有中断恢复能力, 凡记录需要增删改的数据文件都应设计一个中断恢复子模块, 由控制部分自动判断、自动调用和自动恢复。例中的算法描述是采用易于理解的表现形式书写的, 也就不必多加解释。

## 五、结 束 语

应用软件的恢复系统——系统恢复和中断恢复的算法是经过实际检验的。

恢复系统可使应用系统具有更高的可靠性和实用价值, 它不仅给用户方便, 而且大大增强用户使用的安全感和信心, 从而给应用软件的进一步推广树立信誉, 因而是十分必要的, 它应作为应用系统可靠性的组成部分和质量要素加以考虑。

## 参 考 文 献

- [1] 何克清, 计算机软件工程学, 武汉大学出版社, (1983)。
- [2] 潘绵平, 软件开发技术, 上海科学技术文献出版社, (1985)。
- [3] 史维文, 可诊断系统的最优设计, 计算机学报, 1 (1985)。
- [4] 朱三元、周庆隆、蒋瑞青, 软件质量度量, 计算机应用与软件, 1 (1987)。
- [5] 杨培森, 关系数据库RDBMS—1中的恢复系统, 计算机应用与软件, 4 (1986)。
- [6] 孙裕, 程序交换中存储故障校正系统, 计算机学报, 6 (1982)。

## Failure Recovery in Application Software System

Zhang Yinming

### Abstract

This paper discusses failure recovery which is an important part of reliability for application software system. It puts forward two means, system recovery and interrupt recovery, to realize failure recovery.