オンライン・ファイルの障害防止処理について*

弓　場　敏　嗣**

Abstract

In this paper, procedures to prevent on-line file from obstacles are given, which should be observed during the developement phase of operating systems.

First of all, general patterns used in file structure are given in abstract form, and file obstacles growing out of system's dynamic stops are classified. Secondly, some fundamental principles to prevent these obstacles are described. These are as follows.

1) File structures must be in closed state at any instant.
2) Structure elements, which are objects of INSERT or DELETE operations, may run out of system's management. And so on.

Lastly, a sequence to alter file structures called "structure altering sequence (SAS)" is given on the basis of the above principles.

These discussions are based on the experiences obtained through the developement of ETSS (ETL Time-sharing System) and its experimental use.

1. はじめに

TSSを初めとするオンライン・システムにおけるオンライン・ファイルの重要性はいうまでもない．オンライン・ファイル構造に生じる障害は，システム全体の機能停止の原因ともなり，システムの実用性という観点からも深く考慮をはらう必要がある[2)-5)]．操作システムにおけるファイル管理システムの製作にあたっても，その種の障害の発生をできるだけくいとめるよう，処理手順に関する配慮がはらわれなければならない．すなわち，蓄積性をもつファイル情報（2次以下の記憶装置内に貯蔵されていて，端末から利用可能な情報とそれを管理するために必要な情報のすべて）の構造を変更する場合に必要なI/Oを含む一連の処理系列（**構造変形系列**[3)]，Structure Altering Sequence；SAS）に対し，処理の中断が情報構造に重大な障害をもたらさないような手順がなされなければならない．この種の問題は主記憶内の情報構造の変更処理に関してもいえる．しかし，I/Oを含まないのでハードウェア機構によって割込禁止状態にすることで構造の混乱は回避できる．また，パリティ・エラーなど中央処理装置に生ずる回復不能なエラーに対しては，主記憶内情報構造の凍結とシステムの再始動（restart）に備えての処理が行なわれる．そのとき，主記憶内情報構造自体に信頼がおけない状況の場合はそれらすべてを棄却し，システムを改めて始動（initial start）させることが一般には許容される．このように，主記憶内情報構造は基本的には制御情報構造であるため長い期間の蓄積性をもたない．したがって，SASに対する配慮は2次記憶内情報構造に対するものと比べて，実際的な必要性に相違がある．

以下では，まずファイル情報構造の基本的な形態，すなわち1つのSASが対象とする情報構造とSASの基本操作について考える．次に，システムの停止がSAS途中で生じた場合の障害について考察し，許容しうる障害と許容しえない障害の類列を行なう．

最後に，許容しえない障害に対する防止手順の基本原理と具体的な手順をファイル情報構造の基本型について示す．

2. ファイル情報の構造形態とその障害

2.1 ファイル情報構造の基本型

以下の議論を進めるに先だって，若干の言葉の定義を行なっておこう．ファイル情報I/O処理の基本単位としての物理的な情報担体を**物理レコード**（Physical

* On the procedures for preventing file obstacles in an on-line file management system, by Toshitsugu YUBA (Electrical Computer Division, Electrotechnical Laboratory)

** 電子技術総合研究所・電子計算機部

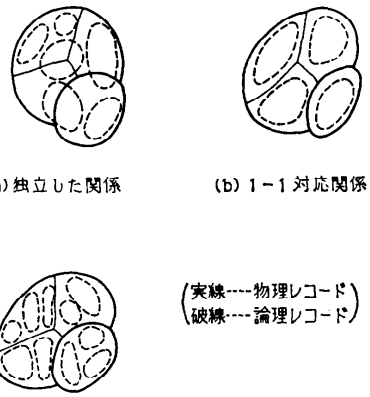


図 1　物理レコードと論理レコードの関係

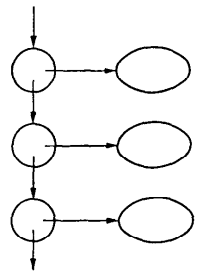


図 2　ファイル情報の基本構造（I）

record）と呼び，ファイル情報を扱う利用目的に有意な情報担体の基本単位を**論理レコード** (logical record) と呼ぶ．論理レコードの属性による分類として，あるファイル管理階層に着目して，そのレベルにとってのデータ情報をになう**データ・レコード** (data record) と，そのレベルでのデータ管理情報をになう**管理レコード** (associative record) の2つがある．物理レコードと論理レコードの関係は，その対応関係を管理するファイル管理階層上のレベル（基本ファイル・システム，Basic File System；BFS）の管理方式によって**図 1** で示すような形をとる．

ファイル情報構造の最も基本的な構造関係は**図 2** で示すような形をとる．管理レコード間の関係が物理アドレスで示される場合と，それらがブロック化されていてインデックスによってアクセスする場合（map 形式）とがある．また，データ部分と管理部分が同一論理レコード上に存在し，データ部分の直接アドレスを必要としない場合もある．

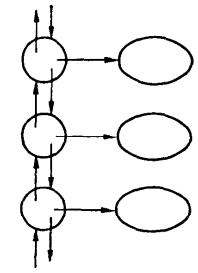


図 3　ファイル情報と基本構造（II）

構造変更系列 (SAS) を考察する場合のもう1つの基本的なファイル情報構造に，**図 3** で示す2方向の管理部分からなる形がある．さらに一般型として，n 方向の構造関係のものが考えられるが，SAS の考察対象の基本型としては図 **2**，図 **3** で示すもので充分であろう．

このような情報構造の基本型に対する構造変更処理の基本操作は，構造要素としてのデータをになう単一の論理レコードの追加 (append)，挿入 (insert)，削除 (delete) である．その場合，情報の共有ということを許す情報構造であれば，新たなデータの追加や挿入，あるいは実質的な削除の操作はなく，管理レコードを追加・挿入・削除し，データ・レコードには共有状態を示す情報の変更（書換え）によって情報構造の変更が行なわれる．追加と挿入の違いは，基本構造の終端に付加する場合と中間に挿入する場合の違いである．SAS を考察するにあたってその処理手順に相違があるが，挿入系列は一般に追加系列を含むので基本操作としては挿入と削除を考える．また，一般には複数個の論理レコードを挿入・追加・削除することもあるが，その場合は基本操作の組合せと考え，以下の考察の対象とはしない．処理手順に関しては，基本操作に対する類推が成り立つ．

なお，注意すべきは先に与えた基本構造における管理レコードとデータ・レコードの分類は，ファイル管理階層構造上のあるレベルに着目した場合のものであり，システム利用者にとって有意なデータとそれを管理するものと関係にとどまらないことである．たとえば，資源管理として2次記憶における自由領域の管理や，ファイル管理システム自体の管理ファシリティの管理などにおいても，管理するレコードと管理されるレコード（データ・レコード）の関係は存在し，それは上述の基本構造をもつ．

2.2 ファイル情報構造の障害

ハードウェアの異常やソフトウェアの虫(bug), さらにファイル情報構造の障害検出などの操作システム・レベルでのシステム停止条件の発生による計算機システムの動的停止 (dynamic stop) が, ファイル情報構造に対する構造変更系列 (SAS) の処理中に生じることによって種々の障害を生じる. 先に上げた情報構造の基本型について, その障害の状況を示したのが**図4, 図5** である. 以下に, 図中の番号にしたがって, 各障害の説明を行なう.

障害 1: 意図せざるデータの共有であり, それによってファイル情報の喪失と機密の漏れが生じる.

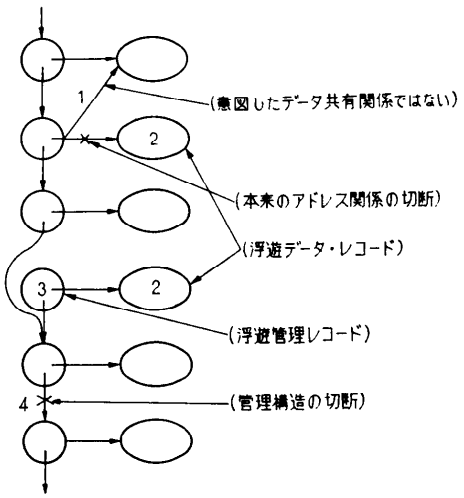


図 4　構造基本型（I）に対する障害

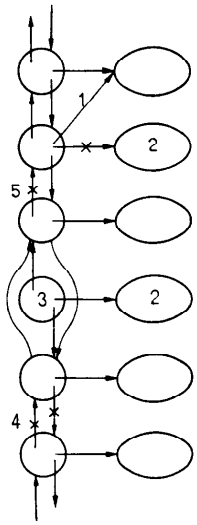


図 5　構造基本型（II）に対する障害

障害 2: どこからもアクセスされない浮遊の2次記憶領域が生じるが, 論理的な情報構造に対する擾乱要因とはならない.

障害 3: 障害2と同様であるが, ほかの浮遊領域に関する管理情報を保有している場合がある.

障害 4: データ情報を位置づける管理情報構造の切断であり, その箇所から先の情報はすべて喪失する.

障害 5: 両方向の管理構造をもっている場合の1方向のみが切断した場合で, 適当な回復処理（サルベージ; salvage)[1]によって修復可能である.

情報の共有を許す情報構造のときは, 共有されているという情報を逆方向アドレスの形でもつ場合と, 単に被共有数でのみ保有する場合がある. 前者については, 基本構造としては両方向の基本型IIと同じである. したがって, 発生する障害も先に示した障害 4, 5 などとなって現われる. 一方, 後者の場合は, 実際の共有数と共有されているレコードのもつ被共有数の不一致という障害が生じる. しかし, この場合もその発生過程を考えると, 基本的には障害5と同様である. したがって, 情報共有を許す情報構造の SAS を考える場合も, 先に与えた基本型I, IIに対して上述の障害の発生を防止するよう考慮すれば充分である.

ここで示した障害の中で, 障害 2, 3 は浮遊領域の回復処理（システム管理下にもどす処理; garbage collection）によって割合容易に修復可能なことから, 止むを得ない状況のもとでは許容できる. 許容できないのは障害 1, 4 である. したがって, 次節で述べる障害防止手順を考察するにあたって, 基本的な防止対象としては障害 1, 4 を考える. 以下での障害防止手順という言葉の“障害”は, それら許容しえない障害をさすものとする.

3. 構造変更系列 (SAS) と障害防止手順

3.1 障害防止手順の基本原則

SAS処理の過程で障害発生防止をはかるための**基本原則**は次のとおりである.

原則 1: ファイル情報の構造は常に閉じている*1（ただし, 原則3の場合を除く）.

原則 2: 基本操作対象の情報要素のシステム管理外への離脱を許す.

* 管理関係が正常であること, すなわち, 番地先のデータ・レコードがなかったり, 間違ったデータ・レコードをさしているという状況が存在しない状態をいう.

原則 3：両方向の管理関係（基本型Ⅱ）をもつ場合はその処理順序に管理情報の重要度*2 を考慮する．

これらの原則が SAS 処理過程のどの時点ででも保持され，守られなければならない．原則1は処理のある段階で情報構造が乱れた状態になることを禁止するものである．原則2は，回復可能な許容しうる障害(障害2, 3) の発生という犠牲において，許容しえない障害 (1, 4)*3 の発生を防止するという基本姿勢を示すものである．原則3は基本型Ⅱ，すなわち両方向の管理関係をもつ場合に許容しうる障害（障害5）の発生をやむをえないものとして，回復可能な構造状態にしておくことを示すものである．この場合は，原則1でいう閉じた構造関係は1方向において失われるが，ほかの重要度の高い関係が閉じた構造を保持するという原則1を守っている．

これらの原則は基本操作対象の存在する2次記憶領域という資源の喪失（原則2）の発生ばかりでなく，その媒体の有する情報そのものの消失を許容しうるものという前提に立っている．実際，構造の変更を要するような状況のもとでは，システムの停止によるある程度の情報の消失は許容しうるものである．すなわち，追加，挿入しようとする操作対象要素の保持する情報が，システムの停止という事故発生のために消失しても，実際問題としてはそれほどシステム利用者に不便を与えない．

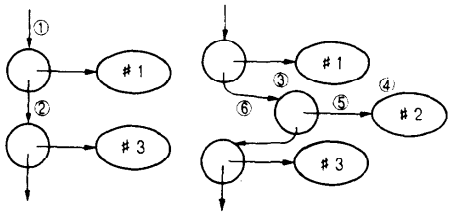
3.2 障害防止の基本原則を考慮した SAS

SAS の手順を考えるとき，情報構造管理方式の違いによって差異がでる．その基本的相違は，操作対象の情報構造が 'active' な状態*4 にあるとき，その期間にわたって構造情報が一時的に主記憶と2次記憶に共存するような管理方式をとる場合と，常に2次記憶内で閉じた情報構造を保つ管理方式の違いである（前者を開いた管理方式，後者を閉じた管理方式と略記する）*5．この相違は常に管理方式上の違いであって情報構造とは何の関係もない．以下では両者を分離して考察を進める．

さらに，論理レコードと物理レコードの関係によっても SAS の手順は影響をうける．独立した関係の場合は管理方式は常に閉じていなくてはならない．また，1-複対応関係の場合で管理レコードとデータ・レコードが同じ物理レコードに含まれるときは，データに関する管理情報とデータの実体が同一 I/O でもって処理可能であることから SAS 自体が簡単化される．管理レコードとデータ・レコードが同じ物理レコードに含まれないときは，1-1 対応関係の場合と同様に SAS の障害防止手順の最も一般的な考察対象となる．したがって以下では，管理レコードとデータ・レコードが別の物理レコードに存在し，それぞれの情報内容の変更に際して I/O が必要な事例について考える．

(A) 閉じた管理方式の場合の SAS

基本型Ⅰに対する挿入，削除の SAS および基本型



（SAS 前の状態）　（挿入 SAS 後の状態）

手順 1：挿入対象前要素および後要素の管理レコードの番地（図の①，②）の検知．
手順 2：挿入要素用の2次記憶領域（図の③，④）の獲得．
手順 3：挿入要素領域にデータ（図の #2）および後要素の番地（図の②）とデータ・レコードの番地（図の⑤）を書き込み．
手順 4：挿入対象前要素の次要素番地部分を挿入要素の番地（図の⑥）に書き換え．

図 6　基本型Ⅰに対する挿入時の SAS
——閉じた管理方式の場合——

手順 1：削除対象要素，削除要素，後要素の管理レコードの番地の検知．
手順 2：削除対象前要素の次要素番地部分を後要素の番地に書換え．
手順 3：削除要素用2次記憶領域の返還．

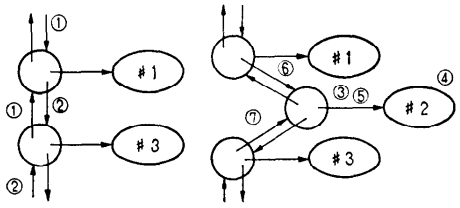
図 7　基本型Ⅰに対する削除時の SAS
——閉じた管理方式の場合——

*2 2方向の管理情報が異なる場合，正常な構造にもどす回復処理を行なうにあたって，どっちの管理情報を信用するかを前もって定めておかなければならない．また情報共有を許す構造の場合は，被共有データを保護するよう配慮しなければならない．

*3 障害1の場合は必ずしも回復不可能ではないが，回復に際して情報の語義 (semantics) の検討という人の介在を必然化する．

*4 1つのファイル情報の単位（たとえばファイル）に対して，一連の I/O 処理を行なうとき操作効率を考慮して処理の開始から終了までの期間その管理情報を主記憶内に保持する方式をとる．管理情報が主記憶と2次記憶内に併存する状態をいう．また，2次記憶内に主記憶内管理情報を書き込む処理を 'close' 処理という．

*5 閉じた管理方式の場合，ファイル情報構造の変更などの操作にあたって主記憶内におかれるのは，その操作に一時的に必要なポインタなどの情報であり，それらの情報がほかのファイル操作に使用されることはない．したがって，操作ごとに情報構造として参照されるのは2次記憶内のものであり，それは操作ごとに無矛盾 (consistent) である．開いた管理方式の場合，'active' な状態においてはファイル情報構造のパターンが主記憶内におかれ，それに関係したファイル操作は主記憶内情報を基にして処理される．つまり，ある時期にわたって，主記憶内情報構造と2次記憶内のそれ，および2次記憶内相互の情報構造は不一致状態 (inconsistent) になることがある．



（SAS 前の状態）　　（挿入 SAS 後の状態）

手順 1, 2: 基本型 I に同じ.
手順 3: 挿入要素領域にデータ（図の #2）および前要素後要素の番地（それぞれ図の①, ②）とデータ・レコードの番地（図の⑤）の書き込み.
手順 4: 基本型 I に同じ（ただし，図で下向きの方を重要度が高いとする）.
手順 5: 挿入対象後要素の後向き（図で上向き）次要素番地部分を挿入要素の番地（図の⑦）に書き換え.

図 8　基本型 II に対する挿入時の SAS
——閉じた管理方式の場合——

II に対する挿入の SAS を**図 6**, **図 7**, **図 8** で示す. 同じ手順内での異なる I/O 処理は並列に行なわれなければならない(パフォーマンスの観点から)しかし，各手順は直列に処理され，おのおの前の手順の終了を確認後に実行されることが必要である. 図 8 の手順 4, 5 は必ずしも直列に実行される必然性はない. しかし，手順 5 の処理が完了し，手順 4 の処理が未完である状態でシステムが停止した場合，回復処理 (salvage) によって重要度の高い前向き方向の情報を基本にして修復されるので消失領域が生じる. すなわち，浮遊領域の発生の機会をできるだけ少なくすることも手順を考える場合配慮されなければならない.

基本型に対する SAS は，管理レコードがいくつか 1 つの物理レコードに blocking されている，いわゆるマップ方式の場合にも当然のことながら成立する. たとえば挿入 SAS を考えよう. 手順 2 において，管理レコード用の 2 次記憶領域を獲得する場合，blocking された管理レコード群のある場所に挿入要素用の領域がとられなければならない. したがって，その物理レコード内の領域割付けを行なう管理機能が存在しなければならない. 挿入要素用に割り当てられた領域はその旨を示す表示子をたてておき，その部分が浮遊領域となった後もファイル情報構造に対する通常のアクセスに障害とならない（表示子のある部分はダミー領域としてスキップするなど）ようにする. この表示子は手順 4 において reset され，ここで初めて挿入要素が情報構造内に組み込まれる. なお，reset 操作は手順 3 におけるデータ・レコード番地の書き込みと同時に行なうこともできるが，その場合はデータの書込み終了に直列に行なわれなければならない.

図 6 で手順 3 と 4 を逆に行なった場合を考えよう. このとき，SAS 中にシステム停止状況が生じなくて正常にすべての手順 4 を完了し，手順 3 を終えない状態で停止すると，先に述べた障害 4 が発生する. また図 7 で手順 2 と 3 を逆に行ない同様の状況を考えると，障害 1 が発生する可能性がある. すなわち，自由領域管理下に返された領域は再び新たな獲復要求によって，別の管理下におかれる，前の管理下にもおかれているので，意図しない情報の共有という障害が生じる.

(B)　開いた管理方式の場合の SAS

開いた管理方式では，active な状態の間に SAS 処理を含めばある期間，主記憶内と 2 次記憶内での情報構造の不一致状態が生じる. 情報構造の更新は主記憶内構造に対して優先的に行なわれるが，2 次記憶内構造に対しては先に述べた原則，すなわち常に閉じた構造を保つという原則を遵守することが必要である. また，エラー発生後に主記憶内の情報構造に 2 次記憶内のそれを一致させる処理（無矛盾処理；consistency processing）が可能である場合がある. つまり，システム停止条件の発生後に事後処理として主記憶内情報構造の 2 次記憶への書込みを行ない，通常の close 処理に相当する処理を行なうことで，構造の更新状態を新しい時点のものに保つことができる.

図 9 は，管理レコード，データ・レコードともに主記憶内にある場合であるが，管理部分のみを主記憶内におくこともある. また，図は対象ファイル情報の 3 要素分の look-ahead の場合であるが一般には 1 つ以上，数個の要素を blocking した形で主記憶内におかれる.

開いた管理方式の場合の挿入および削除 SAS を**図 10**, **図 11**, **図 12** で示す. 図中の破線以下は，いわゆる

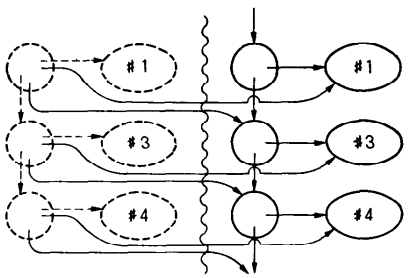


図 9　開いた管理方式における active な状態
——基本型（I）の場合——

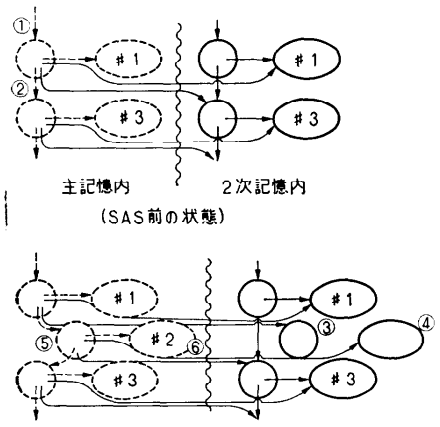


図 10　基本型Ⅰに対する挿入操作の前後状態
——開いた管理方式の場合——

手順 1：挿入対象前要素および後要素の主記憶内での管理レコード番地（図の①，②）の検知．
手順 2：挿入要素用の次記憶領域（図の③，④）および主記憶領域（図の⑤，⑥）の獲得．
手順 3：主記憶内の挿入操作．
手順 4．挿入要素領域にその主記憶内容，すなわちデータ（図の #2）および管理レコード（図の⑤の内容）を2次記憶内に書き込み．
手順 5：挿入対象前要素の管理部分の主記憶内容を2次記憶内に書き込み．

図 11　基本型Ⅰに対する挿入時の SAS
——開いた管理方式の場合——

手順 1：削除対象前要素および後要素の主記憶内での管理レコード番地の検知．
手順 2：主記憶内の削除操作．
手順 3：削除要素用主記憶領域の返還．
手順 4：削除対象前要素の管理部分の主記憶内容を2次記憶内に書き込み．
手順 5：削除要素用2次記憶領域の返還．

図 12　基本型Ⅰに対する削除時の SAS
——開いた管理方式の場合——

close 処理時に行なわれる手順である．したがって，active な状態での挿入，削除操作はおのおの手順3までである．また，close 処理を能率よく行なうために，2次記憶に書込みが必要なもののみを処理すべきであり，そのための表示子が主記憶内の各レコードにおかれることが望ましい．さらに，削除 SAS の場合の手順5，すなわち，削除要素用2次記憶領域の返還をまとめて行なうために，1回の active 状態で生じたそれらを stack にしておかなければならない．

実際問題としては，主記憶の使用領域が限られており，併存する要素の数，あるいは上述の stack の大きさなどに制限がおかれる．したがって，オーバフローが発生するので，その時点で close 処理を行なう必要が生じる．

以上の手順は，先に述べたエラー事後処理としての無矛盾処理が確実に保障されるならば少し異なったものになる．すなわち，常に2次記憶内の情報構造が閉じていなくてはならないとする原則が必ずしも守られる必要はなくなる．しかし，現実には事後処理の不可能なエラーの発生に対しても充分な予防が必要であり，したがって，無矛盾処理は一段上のエラー回復処理とみなすべきであろう．

なお，開いた管理方式の場合の考察対象として，基本型Ⅱに対応するものは省略した．

4. おわりに

本稿では，まず一般のファイル情報構造の基本型を与え，それに対する障害を類型化した．本論文で対象とする障害の発生過程と障害状況を明確にした上で，構造変更を伴う操作（挿入，削除）に対する障害防止手順の原則を与え，その原則にのっとった構造変更系列 (SAS) を具体的に示した．以下では，いままでに述べた障害防止処理に付随した若干の問題点について考察する．

まず，本稿で述べた基本型が現実の場合に，そのままあてはまらないという問題がある．しかし，論理的な関係においては基本型の範囲を一般にはできないので，障害防止処理を抽象的に扱うことは意義があろう．また，ファイル情報構造は，ファイル管理システムの階層構造の各レベルに対応して存在するものであり，障害防止処理は当然のことながら各レベルで配慮されなければならない．ファイル管理システムの設計においても，そうした処理がやりやすい管理構造と管理方式がとられなければならない．

次の問題点として，こうした障害防止処理の有効性の限界について考えよう．たとえば，SAS 実行中に2次記憶装置が誤動作してファイル情報構造を乱すとか，既存の情報構造領域がハードウェアの経年変化などによって I/O できなくなるなどの障害に関しては無効である．こうした障害に対しては別途の信頼性を高めるための処理（信頼性処理[3]，Reliability Management），たとえばバックアップ (back-up) 処理が用意されなければならない．

オンライン・ファイルの信頼性は，TSS を初めとするオンライン・システムの有用性を決定する重要な要因である．システム全体の信頼性を下げる要因を刻明

に分析し，それに応じた処理を体系的に行なう心要がある．一方で信頼性処理は複雑化する傾向があり，信頼性保証の程度を設定し，重複のないよう機能分類を行なってそれに応じた処理を施されなければならない．

オンライン・システムの設計当初それほど重視していなかった問題で，製作・運用の過程でその重要性を認識させられるものの一つとして，オンライン・ファイルの信頼性の問題を考えることができる．その意味で障害防止処理の必要性は，システム製作者の誰もが経験するものであり，いわば 'know-how' として蓄積されているものであろう．筆者の場合も，ETSS (ETL Time-sharing System) の製作，運用の過程でこの問題について考え，ETSS という実際のシステムを素材として本考察を行なったこと，さらに本稿で論じた抽象モデルに対する障害防止処理がファイル管理システムの各階層で構じられる必要があり，かつ，それが有効である[3] ことを付言しておく．

謝 辞

ETSS 開発・運用の責任者であられる相磯秀夫計算機方式研究室長，淵一博主任研究官には日ごろいろいろな面でお世話いただいている．また，ETSS ファイル管理モジュールを担当された飯塚 肇，横井俊夫の両氏を始め，宮川正弘氏には討論を通じて幾多の有益なご意見をいただいた．その他，芝分室の皆様にあわせて深く感謝の意を表したい．

参考文献

1) R C. Daley, P. G. Newmann: A general purpose file system for secondary storage, Proc. FJCC 1965, pp. 213-229.
2) G. Oppenheimer, K. P. Clancy: Considerations for software protection and recovery from hardware failures in a multiaccess, multiprogramming, single processor system, Proc. FJCC 1968, pp. 29-37.
3) 弓場，宮川：オンライン・システムの信頼性処理について，電総研彙報 Vol. 34, No. 8, pp. 699-710.
4) P. C. Lockemann, W. D. Knutsen: Recovery of disk contents after system failure, C. ACM, Vol. 11, No. 8, Aug. 1968, p. 542.
5) 広沢，久保，益田，本林：5020 TSS のファイル・システムの管理，昭和 45 年電気四学会連合大会予稿集，情処 (1)，2578.

(昭和 45 年 11 月 30 日受付)