

木構造ファイルシステムの信頼性向上について

6Y-7

- 階層化OSに於ける媒体障害への対処 -

中原 慎一

NTT 情報通信処理研究所

1. はじめに

木構造を持つファイルシステムでは、ディレクトリ部の破壊はシステムに致命的な影響を与える。

本稿では、階層化OSを対象としてファイルシステムの信頼性向上について述べると共に、階層化OSによるマシンの機種に依存しない障害復旧処理について述べる。

2. 階層化OS

階層化OSとは、ハードウェアを論理化してみせる下位OS部分と、その論理化された資源情報を用いてAPに機能を提供する上位OS部分の2階層から構成されるOSである。

本稿では、下位OSとして入出力制御、上位OSとしてファイル管理を想定する。入出力制御は物理的な装置構成やチャネル番号などを論理的な装置識別子として、シリンダ番号やセクタ長を論理化した入出力ブロック番号として上位(ファイル管理)に見せる。[1]

3. 信頼性向上の方式について

3.1 障害処理の対象範囲

ファイルシステムにおいて、媒体障害検出時の対処としてユーザのファイルが障害となった場合とシステムファイルや管理情報(ディレクトリ)が障害となった場合とではその復旧処理が異なる。つまり、ユーザ所有のファイルについては障害の前後でその内容の変更が前提となり、障害対策としてはユーザ自身によるファイルの2重化やバックアップ、もしくはジャーナルによるリカバリが主体となる。

一方で木構造を持つファイルシステムにおいてはディレクトリの障害はユーザ及びシステム運転にとって致命的な影響を与える。またコマンド等のシステムファイルの障害はシステム運用に悪影響を及ぼす。

ここではユーザが対処できない管理情報(ディレクトリ/ファイルノード)の信頼性向上(ボリュームの二重化)について述べる。

3.2 従来方式とその問題点

入出力制御とファイル管理が一体化されていた従来OSでは、主として信頼性向上のため入出力制御にてボリュームの二重化が行われており、媒体障害検出時にアクセス対象ボリュームの切り替えを行っていた。

しかしながら、従来方式では次の点で問題がある。

- ①チャネル等の装置構成や物理情報を障害処理が意識しなければならず、適用対象の柔軟性にかける。
- ②物理情報を直接意識するため処理が複雑になる上、マシンに固有の処理となる。
- ③障害箇所がシステムのものか、ユーザのものかの判定が困難。

3.3 本方式とその効果

【方式説明】

(1)媒体フォーマット

上位OSであるファイル管理は、装置識別子にてボリュームを特定し、入出力ブロック番号にてディレクトリ及びファイルを特定する。

本方式では図1に示すような論理ボリュームフォーマットを採用した。つまり

- ・入出力制御により論理化されたボリューム(論理ボリューム)を管理情報域とファイル実体域に分割/局所化した。
- ・ボリューム一元管理領域ではディレクトリ、ファイルノード、ファイル実体域の各領域の先頭/最終入出力ブロック番号を管理している。

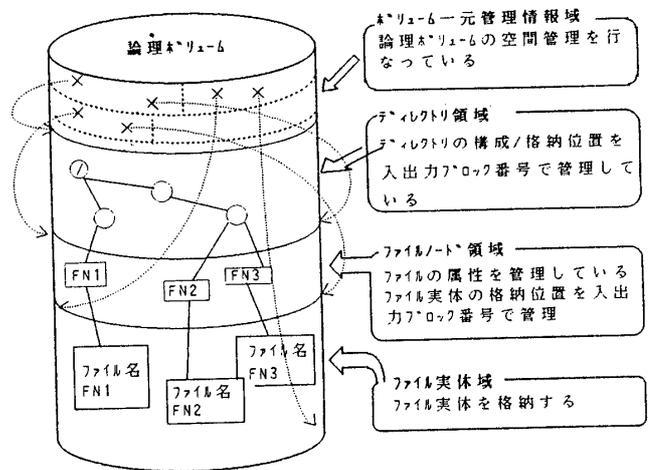


図1 論理ボリュームフォーマット

このフォーマットは次の点で有効である。

- (A)ファイル管理は、障害が発生した入出力ブロック番号元にその障害箇所が管理情報部かファイル実体部かを容易に判定できる。
- (B)管理部内でもディレクトリノード領域とファイルノード領域を分割することにより閉塞レベル(ボリューム閉塞/ファイル閉塞)の切り分けを可能としている。
- (C)ディレクトリ領域を局所化することにより管理域のみのバックアップを容易化している。

(2)障害処理(図2参照)

- ①まず媒体障害は入出力制御が装置アクセスを行ったときに検出される。
- ②ファイル管理は、入出力制御のシステムコールが異常終了することで媒体障害が発生したことを検知する。
- ③この時、ファイル管理は図1の論理ボリュームフォーマットと、エラーとなった入出力ブロック番号から障害箇所が管理部かファイル実体部かを判定する。
- ④ファイル管理は、障害部位に応じた障害処理を行う。
 - ・管理部障害の場合、OS側での障害処理を行う。ファイル管理が交替ボリュームへの切り替えを行う。(詳細は次節参照)
 - ・ファイル実体域障害の場合、ファイル管理が交替ボリュームへ切り替えを行う。交替ボリュームは現用ボリュームと同一のファイル構造を持つ。

- ・ファイルノード/ファイル実体域障害の場合ユーザへ障害通知を行う。
ファイル管理は、そのシステムコールの異常終了によりユーザに媒体障害であることを通知する。
(復旧処理はユーザ依存)

【効果】

- 本方式で提案したように上位OSであるファイル管理で障害処理を実現することにより、次のメリットがある。
- ・管理部障害時には、ユーザに障害を意識させずに簡単な障害処理が可能となる。
 - ・上位OSで障害処理機能を提供することによりマシン機種/装置構成に依存しない障害処理が可能となる。

これにより、3. 2節で示した問題点は解決される。

4. 障害処理の実現性

媒体障害の時には一般的に次の処理が行われる。

- (1)障害部位の検出
- (2)装置の閉塞
- (3)装置の切り替え
- (4)装置の切り替え通知
- (5)被障害ボリュームの復旧

4. 1 障害部位の検出について

3.3 節で述べたように媒体上の管理フォーマットを工夫することによりファイル管理にて障害箇所がディレクトリか、ファイル実体域かを容易に判定できる。

4. 2 装置閉塞について

本方式では障害部位がディレクトリであれば、自動復旧するので原則として装置閉塞は必要ない。

システムからの要求条件として、装置閉塞を行う必要がある場合には、ファイル管理がファイルシステムを占有することで、ユーザからのアクセス要求を保留する。システムによってはファイル管理より上位で障害処理を行っているものもあるため、保留ではなくファイル管理システムコールを異常終了とすることもできる。

4. 3 装置の切り替えについて

ファイル管理はルートノードの存在するボリュームと交替用のボリュームの装置識別子を予め知っている^{*)}。また、装置識別子は、メモリ上だけで管理しており、媒体上には一切持たない。ボリューム上でのルートノードの入出力ブロック番号は特定値に固定しておく。

このようにすることでファイル管理は、ディレクトリの障害検出時に被障害ボリュームの装置識別子を交替用のものにメモリ上で変更するだけで容易にボリュームの切り替えを実現できる。この場合、他のディレクトリノードやファイルの入出力ブロック番号は一致させる必要はない。

*) ファイル管理への通知方法としては、システムたち上げ時の決定(SG)またはルートノード変更用のコマンドを提供することによる。

この時留意しなければならない点を次に示す。

- (1)ディレクトリをメモリ常駐している場合、ボリューム上での入出力ブロック番号はルートノード以外保証されないため、常駐情報のクリアが必要となる。システムの信頼性という観点からはクリアしても何等問題無い。

- (2)システムファイルのうちログ格納ファイルやODP用ファイルは書換え型である。これらは管理元があるため、ファイル管理はシステムコールの異常終了による障害発生通知のみ行う。
 - ・ログファイルはログ管理の配下にあり、障害時ログ管理が自動的にファイルのスワップ制御を行う。
 - ・ODPファイルはメモリ管理の配下にあり、障害時にはファイルの切り替え/プロセスサポート等の対処がメモリ管理によりなされる。
- (3)システムファイルとユーザファイルが混在している場合、管理情報障害が発生したボリュームの切り替えによりユーザファイル資源の引継ぎができなくなることがある。これについては、システムファイルとユーザファイルを別ボリュームとすることで最大限のユーザ情報の確保が可能となる。

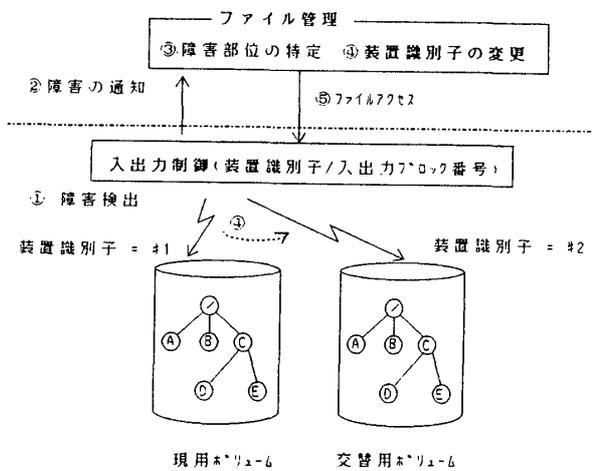


図2 ファイル管理でのボリューム切り替え

4. 4 切り替え通知

ユーザへの通知は不要。システム管理者に装置障害が起こったことを通知する必要があるが、それは入出力制御/ファイル管理のどちらかで実現しても技術的に問題無い。

4. 5 被障害ボリュームの復旧

ファイル実体が障害となった場合には、バックアップファイルを用いてファイル管理を使用して従来方式で復旧可能。ディレクトリ障害となったボリュームは入出力制御を使用した復旧となるが、図1に示したディレクトリ領域部分のバックアップを保持しておくことにより復旧が容易となる。

5. おわりに

これまで述べてきた障害処理は、すべてファイル管理内の処理として実現可能なものであり、切り替え処理を実現するのに他のプログラムに影響を与えることはほとんど無い。しかもマシンの種別/構成に依存しないという点で従来の方式より優れている。

今後検討課題としては次の点が考えられる。

- ①切り替えボリューム中のユーザファイル資源の引継ぎの容易化。
- ②システムファイルの一元的障害復旧処理の実現

【参考文献】

[1]原典CTRON大系 入出力制御インタフェース オーム社