MML環境における自動ファイル複製システム (2) 技術的側面

7E-5

平賀　瑠美　　村上　和隆　　中島　周　　黒沢　隆
(日本アイ・ビー・エム株式会社　東京基礎研究所)

1. はじめに

本システムでは、PC上のローカル・ファイルを自動的にホスト上のリモート・ファイルに複製する。自動複製は、日本語DOSの機能呼び出し（以後機能呼び出しと略す）が発行された際に、その情報をログと呼ぶ形式に貯め、後でそれを再び処理することで実現している。本稿では、ログの形式を中心に述べる。

2. ログの構成

2.1 ログとは？

ログは、発行された機能呼び出しを任意の時刻に実行するために情報を貯めたものを指す。ただし、機能呼び出しの都度ログが作られるとは限らない。

貯められた情報は、ホストーPCセッションが張られている時に、システムの複製実行部が空き時間を検出して[1][2]、対応する機能呼び出しに解釈して処理する。複製されたファイルと元のファイルの内容は、ログがすべて処理された時点では一致する。従って、ログを処理以前に失わないためにメモリ上ではなく、ファイルとして貯えた。

2.2 ログの形式

ログは、AFSLOGとAFSLTBLという二つのファイルから構成される（図1）。AFSLOGは、ログ単位の集まりである。ログ単位は16バイトで、機能呼び出しの際に設定されるレジスタ値そのものを容れてはいない。ファイル複製のための情報としてファイル名は不可欠であるが、ファイル名はAFSLTBL中の名前の項目へのポインタとして表されている。AFSLTBLは、名前の項目の集まりである。名前の項目は、ファイル名と複製実行部がログを処理するのに必要な情報を含む。　図2に、ログ単位と名前の項目の形式の例を示した。

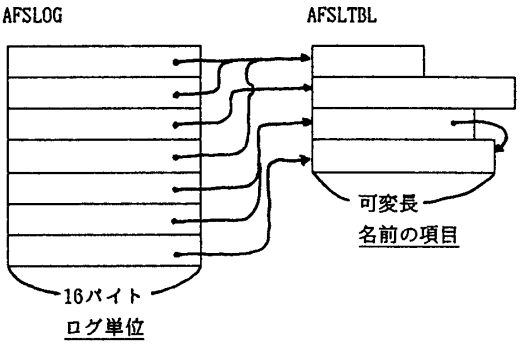
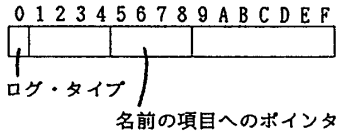


図1.　ログの構成

ログ単位の形式



ログ・タイプには、以下のものがある。
30h(ファイル作成),31h(ディレクトリ作成),
32h(ファイル削除),33h(ディレクトリ削除),
34h(ファイル名変更),35h(書込み),36h(ファイル削除),
37h(ディレクトリ削除),38h(0バイトの書込み)
61h,62h,63h(システム・コマンド)

ログ単位の形式中のその他のフィールドについては
ログ・タイプによって適当な値が設定される。

名前の項目の形式

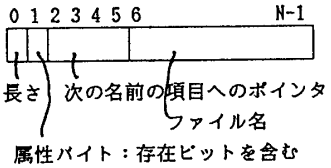


図2.　ログの形式

An Automatic File Replication System in MML Environment
(2) Technical Aspects
Rumi HIRAGA, Kazutaka MURAKAMI, Amane NAKAJIMA, Takashi KUROSAWA
IBM Research, Tokyo Research Laboratory

２．３　ログの例・・ファイルの更新

機能呼び出しに対してどのようなログが生成されるかをファイルＡの更新を例として述べる。

機能呼び出し	ログ・タイプ	名前の項目
Ａのオープン	作らない	作らない
Ａへの書込み	０Ｂ５ｈ	作る
Ａのクローズ	０３５ｈ＊	＊＊

＊ 書込みで作った0B5hのログ・タイプを035hにする
＊＊ 書込みで作った名前の項目を指す

ファイルの更新は、クローズ機能呼び出しで完了する。従って更新のログが完成するのもクローズ機能呼び出しが発行された時で、書込み機能呼び出しが発行された時には、クローズのログ・タイプの最上位のビットをたてて更新の未完了を示している。

また、ファイル更新のログ単位は、512バイトのブロックを単位として、更新の開始ブロックと変更するブロック数を含む。ログ単位は更新データを含まず、ログの処理時に、ローカル・ファイルを読み出してデータを取り出す。

２．４　ログの生成と処理の順序

ログは機能呼び出しに対応して生成されるので、生成の順序で処理されるのが自然である。こうするとリネームのログ処理で次の問題が生じる。

『ファイル名ＡをＢにリネームする機能呼び出しが発行されると、ファイルＡがなくなるため、以前に作られたＡに関するログは、Ｂに関するログとして処理されねばならない。そのためには、Ａに関するすべての未処理のログのファイル名をＢに替える必要がある。しかし、リネーム前と後のファイル名は可変のためログの書き直しのオーバーヘッドが大変大きくなる。』

この問題は、機能呼び出しの情報のうち、ファイル名とそれ以外の情報を分離し、それぞれをAFSLTBLとAFSLOGに容れることで解決した。

名前の項目を機能呼び出しの発生の順序とは無関係に存在させ、名前の項目の中には、次の名前の項目を指すポインタを設けた。リネームの場合は、このポインタをたどることで、古い名前と新しい名前とがわかり、ログの書き直しが不要となる。

また、複数のログ単位が同一のファイル名を必要とすることも多い（例えばファイル作成後にそのファイルに対して書き込みするなど）ため、名前の項目を複数のログ単位から参照して共通に用いることでログの大きさを節約できる。

２．５　ログの処理の効率化

ログの処理は生成順になされるが、生成と処理のタイミングによっては、ローカルに存在しないファイルをアクセスしてエラーを起こすことが考えられた。例えば、あるファイルについて

１．作成－＞２．書込み－＞３．消去

のようにログが作られたとする。３のログが生成された時点では、ローカル・ファイルは存在しない。一方、その後に２のログを処理すると、ローカル・ファイルを読み出してデータを取り出そうとするためファイル・アクセス・エラーとなる。従って、このような場合２のログは、処理してはいけないしする必要もない。いずれファイルが消去されることがわかっているならば、１のログも処理不要である。

本システムでは、名前の項目に存在ビットを用意してログの処理を効率化している。存在ビットは、消去のログを生成するときにリセットし、ログの処理時に存在ビットのリセットを見つけた場合にはそのログを処理しない。

２．６　ログ生成のオーバーヘッド

本システムを搭載するとログを生成する分だけ応答時間に影響する。ＤＯＳ内部コマンドの実行時間をシステム搭載した場合としない場合についてＰＳ／５５モデル６０で応答時間を測定したところ、一つのログ作成について約0.3秒かかっていた。コピー・コマンドではログが三つ作られるので、システム搭載時には未搭載の時よりも約１秒レスポンス・タイムが長くなる。また、コピーするファイル・サイズには余り関係ないこともわかった。

３．　おわりに

本システムのログは、自動的にファイル複製を行なうために、機能呼び出しに対応して生成され、生成順に処理される。ログはファイル名の情報とその他の情報に分けられている。複製されるファイル・データはログ中にはない。今後は、リモート・ファイルをローカルに複製できるように拡張していきたい。

参考文献

[1]村上他：MML環境における自動ファイル複製システム(1)概要，第37回情処全大(1988年9月)
[2]中島他：遅延実行によるマイクロ・メインフレーム結合の拡張とその適用，同上