

OS/omicon 上での追記型光ディスクを用いた世代管理ファイルシステムについて

横関隆 岡野裕之 並木美太郎 高橋延匡  
東京農工大学 数理情報工学科

本報告では、我々が開発を行った日本語OS、OS/0第2版におけるファイルの世代管理方式と、その問題点について述べる。

OS/0第2版では古くなったファイルの内容を、シーケンシャルにリンクし保存するという方式で世代管理を行っている。この方式では、削除したファイルの復活が不可能である等の問題点があった。我々はこの問題点を追記型光ディスクを用いる仮想ディスク方式により解決を計った。また次版OS/0では、世代管理のモデルと追記型光ディスクの扱いに関する再検討が必要であるという考察を得た。

A Version Control System on OS/omicon with Write-Once Optical Disks

Takashi Yokozeki Hiroyuki Okano Mitaro Namiki Nobumasa Takahashi  
Tokyo University of Agriculture and Technology  
Faculty of Technology  
2-24-16, Naka-machi, Koganei, Tokyo, 184, Japan

This paper describes a version control system on OS/omicon with write-once optical disks. The version control system holds old files by linking them sequentially, but does not maintain deleted files. They are kept in write-once optical disks and made accessible by the virtual optical disk method. This paper also describes problems of the system and presents our strategy to solve them in the next version of the OS/omicon file system.

## 1. はじめに

近年になって著作権や知的所有権の主張に関する議論が活発になってきた。その中で知的所有権の主張のために、「オリジナリティの証明」といった課題が話題になっている。オリジナリティの証明のために何が必要であるか、他人とどこが異なるかを議論するだけでは解決しないであろう。

オリジナリティの証明に有効な手段の一つとして、その開発過程（考案過程）の一部始終を公開(publish)することが考えられる。しかしそれを行うためには、米国における「研究ノート制度」のように、開発過程すべてを記録しておかなければならない。

こういったことから我々は、コンピュータ社会を考慮に入れて、計算機による支援システムの必要性を考えた。すなわち、OSの機能としてのタイムスタンプ付きの世代管理を用意することを考えた<sup>[1]</sup>。一方、現実的な問題として、将来的に世代管理が法的な手段として有効となるか否かは別としても、ソフトウェア開発や、ユーザインターフェースなど人間工学の研究において有用であることは確かである。

我々は日本語情報処理を指向したOSであるOS/omicon (以下OS/o)の研究・開発を行っている<sup>[2]</sup>。OS/o第2版には世代管理機能が実現されており、現在追記型光ディスクと合わせてこの機能が利用されている。

本報告では、OS/o第2版の世代管理機能の実現方法とその問題点、そして次版で実現する際に取るべき方針について述べる。

## 2. OS/o第2版における世代管理

OS/o第2版では、ファイルの世代管理機能を実現した。この機能はファイルを書き換えても、書き換える前の内容を消さずに保存するという単純なものであり、記憶装置として磁気ディスクを対象とした形で実現されている。

しかし現在はOS/o第2版ファイルシステムに追記型光ディスクを取り込み、世代管理機能は主に追記型光ディスクを用いた形で利用されている。追記型光ディスクを用いた世代管理機能を実現する際にとった方針は次の2点である。

a. OSの変更を最小限におさえる

b. 既存のアプリケーションで利用可能にする  
次にOS/o第2版での世代管理機能の実現方式を述べる。

### 2.1 OS/o第2版

我々は日本語情報処理を指向したOS、OS/oの研究開発を行っている。現在OS/o第2版が稼働しており、手書き文字認識(JOLIS)、日本語文書出力(JOSHIO)、日本語文章処理、また次版OS/oなどの研究、開発に利用されている。

OS/o第2版の、ファイルシステムを中心とした特徴は次のとおりである。

#### (1) ストリーム型のファイル

OS/o第2版のファイルは、read, write, seek SVC (Super-Visor Call) を基本とした、ストリーム型のファイルである。アクセス単位はユーザが自由に設定可能である(バイトに限らず、ワード、3バイトといったデータを最小アクセス単位とするストリームファイルが作成可能)。

#### (2) 二重の木構造

OS/o第2版のファイルシステムは一般的な木構造(ディレクトリとファイルの組合せ)に加えて、ファイルの中にも木構造を持っている(図1参照)。ファイルアクセスの際には、ファイルの指定と、ファイル中の木構造の葉(メンバと呼ばれる)の両方を指定する必要がある。

このような構造にしたのは、文字コードと文字属性情報などの分離する応用を考えたためである。

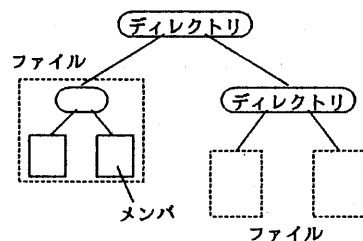


図1 OS/o第2版のファイル

(3) ファイル単位の世代管理

ファイル作成時にユーザが指定することにより、任意世代分のファイルをディスク上に保存しておくことが可能である。このときの管理単位は、ファイル内の木構造全体である(図1中の破線の中全体)。

2.2 OS/2第2版の世代管理方式

OS/2第2版における世代管理の実現方法を述べる前に、OS/2第2版ファイルシステムの構造について簡単に説明する。

ディスクへのファイル格納方式は図2に示す形をとっている(概略図)。ディスクの領域を図2(a)のように二つに分け、一つは基本ディレクトリと呼ばれるファイルアロケーション情報を管理するための構造体配列として使用し、もう一方の領域はページと呼ばれる大きさ(1kバイト)ごとに区切り、ファイル内データを実際に格納する領域として使用する。

ファイルは基本ディレクトリ一つと、データを格納するために必要なだけのページを使用し、図2(b)のような形態をとる(注:この図は基本構造を示したものであり、正確な構造ではない。例えば実際のファイルは基本ディレクトリを二つ使用する。詳細な説明は省略する)。これで一つのファイル(あるいはディレクトリ)を形成するが、基本ディレクトリの中に旧版ファイルへのポインタがあり、図3に示す形で旧版ファイルを行えるようになっている。

このリンクを利用し、OS/2第2版にはファイルを書き換えても書き換える前の内容を保存する機能が用意されている。世代管理に関連した動作は次のとおりである。

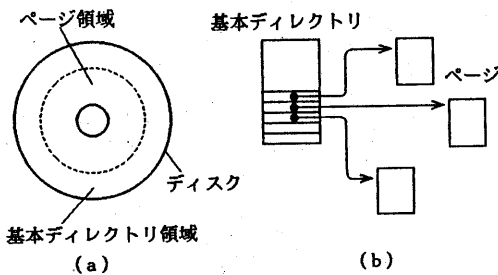


図2 OS/2第2版のファイル構造

(1) ファイルの作成

ファイルに対して世代管理を行うためには、ファイル作成時に世代管理を行うことを指定しなければならない。具体的にはファイルの作成を行う create SVCにより、何世代分の旧版ファイルを保存するかを指定する。世代管理の対象としないファイルについては、ここに0を指定する。

(2) 旧版ファイルの参照

リンクを形成し、ディスクに残っている旧版のファイルを参照する場合は、過去何世代目の内容を取り出すか、最新の世代を0として以下1, 2, ...という形で open 時に指定する。通常はここに0を指定することにより、最新の版を参照できる。

(3) ファイルの変更

ファイルの変更を行う場合、ユーザはそのファイルを書込みモードで open する。するとファイルシステムは図4に示すようにそのファイルのコピーを作り、旧版ファイルのリンクに接続する。この際旧版ファイルの数が create 時に指定された世代数を越える場合、一番古い版(図4中一番右の版)のリンクが切られ、使用されていた基本ディレクトリやページが解放される。

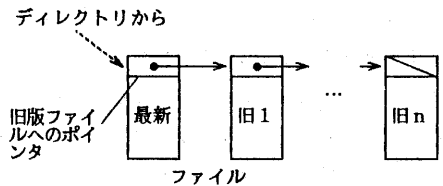


図3 旧版ファイルのリンク

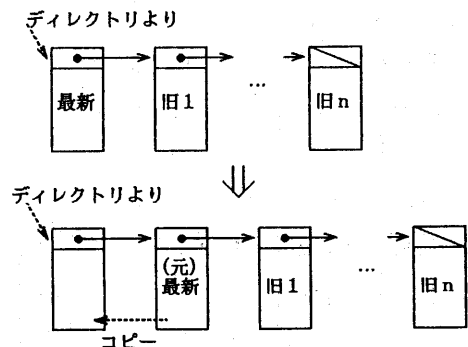


図4 ファイルの更新

る。

また旧版ファイルを書込みモードで open すると図 5 のようにその版が最新の版としてコピーされ、以後の変更はそのコピーに対して行われる。

#### (4) ファイルの削除

ファイルの削除を行うと、それまでに作られてきた旧版のリンクが解かれ、最新の版だけではなく、すべての版が削除される。ここで事実上、ファイルの復活は不可能となる。

### 2.3 追記型光ディスクの利用

OS/2 第 2 版の世代管理機能は、磁気ディスクを対象としているが、現在この機能と仮想ディスク方式<sup>[3], [4]</sup>と呼ばれる方式の組合せにより、世代管理は主に追記型光ディスクを用いて行っている。

仮想ディスク方式とは、本来書換えのできない記憶媒体である追記型光ディスクを、仮想的に書換え可能にするものである。この方式を採用することにより、追記型光ディスクを磁気ディスクとほぼ等価に扱うことができる（等価に扱えない部分については後述）。

仮想ディスク方式は、図 6 に示す多段の表を構成することにより、セクタ（ディスク入出力の最小単位）ごとに書換えの管理を行う。このようにしてできた、見かけ上書換えのできるセクタを仮想セクタと呼び、仮想セクタを一次元的に並べたものを仮想ディスクと呼ぶ。仮想ディスクへの入出力要求は、仮想セクタ単位で行われ、仮想ディスクのハンドラは、表をたどることにより、仮想セクタから物理セクタ（追記型光デ

ィスクの物理的なセクタ）への変換を行い、実際の入出力を行う。

仮想セクタの読出しの際には、表をたどり、目的の物理セクタを求め、そのセクタを読み出せば処理は終了する。書込みの際には、図 6 中における破線から実線への表構造の変更が伴う。このように表構造の変更が必要なのは、表の構成要素がすべて一度しか書込みができないためである。

図 6 中のメタ対応表 (meta mapping table) に関しては図 7 に示すように、追記型光ディスクの端から連続して順々に書き込んでいく。メタ対応表以外の要素用の物理セクタは、もう一方の端（図 7 では内側の端）から順々に一つずつ使用していく。

仮想ディスクには磁気ディスクと比べた場合に次の特徴がある。

#### (1) 過去の状態が取り出せる

仮想ディスクは、そもそも消去不可能な媒体の上で

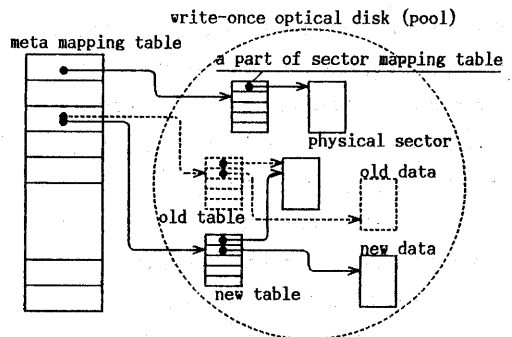


図 6 仮想ディスクの構成 (文献[4]より)

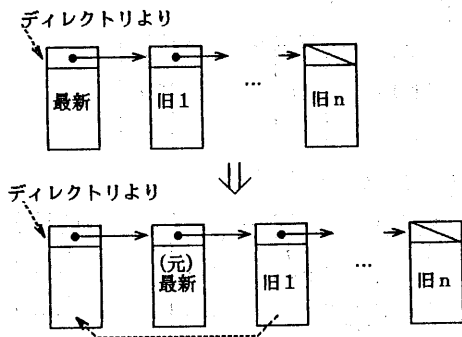


図 5 旧版ファイルの更新 (旧 1 の更新)

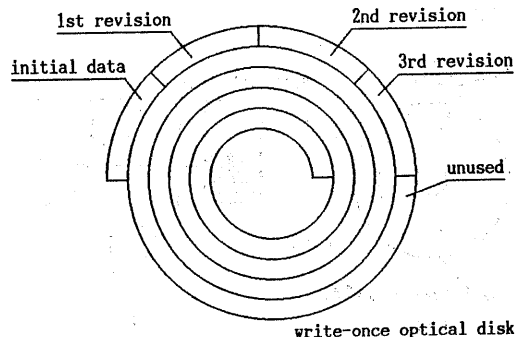


図 7 メタ対応表の書換え (文献[4]より)

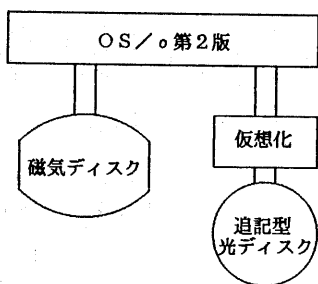


図8 追記型光ディスクの組込み

構成されているため、一度書き込まれたデータはすべて残る。したがって見かけ上書き換えられてしまったデータも、古いメタ対応表をたどることにより、過去の状態を取り出すことができる。

(2) 書き換えられる回数に限界がある

仮想セクタの書換えは、物理セクタの消去を伴う。したがって物理セクタの数が有限である限り、書き換えられる回数にも限りがあることになる。この制限があるため、仮想ディスクは磁気ディスクと完全に等価に扱うことができない。

2.4 追記型光ディスクを利用した世代管理

OS/2 第2版では、仮想ディスクを作り出すプログラムを図8のようにOS/2 第2版の光ディスクドライバに組み込むことにより、磁気ディスクと等価な形で追記型光ディスクを扱っている。

この形で実現を行うと、OS側が認識している空き領域と、実際に追記型光ディスク上で使用できる領域の大きさが一般に一致しない<sup>[4]</sup>。OS/2 第2版では、ディスク空き領域の計算を行う際に、磁気ディスクと追記型光ディスクとで別処理を行い、追記型光ディスクの場合は、仮想ディスク上における空き領域の大きさではなく追記型光ディスクの未使用領域（書込みの行われていない領域）の大きさを計算している。

また追記型光ディスクに書き込まれたファイルは、create時に指定する「保存する旧版ファイルの数」に上限値（約3万）が指定されており、事実上、旧版ファイルの削除は行われない形となっている。

ファイルの削除が行われた場合、仮想ディスク上からそのファイルは消滅してしまう。しかし、後で仮想ディスクの状態をそのファイルが削除される前の状態に戻すことにより、削除されたファイルの復活も可能となっている。このファイルの復活はツールとして実現されている。

3. OS/2 第2版世代管理機能の問題点

OS/2 第2版の世代管理機能は、世代管理のモデル確立が不十分であったため、ファイル変更の過程が残せ、後でその内容が取り出せるという最低限の機能しか実現できなかった。ユーザは、この最低限の機能を利用してファイル変更日時の一覧を表示するツールや、過去のファイルとの差分をとるツールなどを用意することになる。

このような形で実現された世代管理機能の問題点として次に挙げるものがあつた。

(1) 世代管理がシーケンシャルに行われる

OS/2 第2版の世代管理を実現する際には、単にファイルの変更過程をすべて保存しておき、任意の時点で過去の内容が取り出せれば良いと考えてきた。そのため、図3に示したように更新されたファイルの列はシーケンシャルなものになり、過去の内容に戻したい場合は、その過去の内容を最新のものと置き換えて、そこから変更しなおすというモデルであつた。

しかし実際は世代管理のモデルは、シーケンシャルなものではないことが経験的にわかっている。

(2) ディスクの利用効率が悪い

これは技術的な問題であるが、世代管理の単位がファイル全体となっているため、ファイル中のたった1バイトを変更するだけで、ディスク領域をファイル全体分の大きさだけ消費してしまう。変更していない部分の複製がいくつもできてしまうため、ディスクの利用効率が非常に悪い。追記型光ディスクはビット単価が安いとはいえ、再利用不可能な、限りある資源である。また、磁気ディスクを使用する場合には、効率の悪さは使いやすさにもびびいてくる要素なので、改善が必要となる。

(3) 変更場所・過程の管理が困難

OS/2 第2版のファイルはストリーム型であり、

OS側ではファイル内のデータ構造に関知していない。したがってファイル中のどこをどのように変更したかの情報を管理することが困難になっている。

例えば、ファイル中に1バイト挿入する作業を行う場合、OS/0第2版のファイルの場合、挿入場所より後ろのデータをすべて write しなおさなければならない(図9参照)。このため、たとえファイルに加えられた操作 (seek, write 等) を記録していても、挿入なのか変更なのかという情報は、新旧ファイルの比較 (しかも単純な比較ではなく、DPマッチングを使用するようなもの) を行わなければ取り出せない。

#### (4) 削除したファイルの復活ができない

ファイルを削除した場合、追記型光ディスクを使用している場合を除き、それまでの世代管理が無効になってしまいます。これでは、せっかく世代管理を利用していても、誤ってファイルを削除してしまった時のパニックがなくなるということになる。

### 4. OS/0ファイルシステムの今後

筆者らは現在、次版OS/0の設計を行っている。ここでは特に次版ファイルシステムの設計方針について述べる。

先に述べたとおり、OS/0第2版のファイルシステムの世代管理機能は、単純な機能の実現であった。そこで次版はファイルシステム全体を、世代管理機能を指向した形で設計しなおす方針である。具体的には、世代管理機能のモデルとファイル構造の見直しを検討中である。

#### 4.1 世代管理のモデル

世代管理機能を利用するうちに、ファイル更新の流

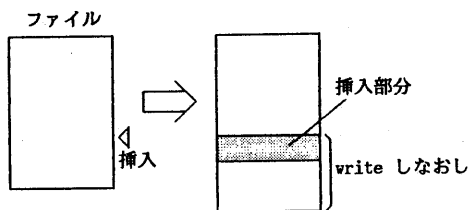


図9 ストリーム型ファイルの挿入操作

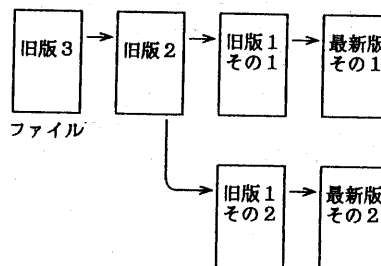


図10 パラレルワールド

れはシーケンシャルなものではなく、図10に示すように分岐を伴うものであることがわかってきた。すなわち、ファイルを過去の状態に戻すのではなく、過去の状態から更新の流れが分岐し、パラレルワールドを形成するのである。

このパラレルワールドは、ファイル単位だけではなく、ファイル群やファイル中の変更箇所といった単位でも存在し得る。例えば、プロジェクトを組み、ソフトウェアを開発する場合などには、ファイル一つ一つの世代管理と同時にそれらのファイルの世代が同期していること、つまり世代管理の単位がプロジェクトで作成したファイルすべてとなっていることが好ましい。さもなければ、あるファイルは最新の版を用いているのに他のファイルは何世代か前の版を用いてしまう、などの過ちを犯す可能性がある。

現在このモデルをファイルの木構造と併せて、どのように表現するか (ユーザに対し、どのように見せるか) 検討中である。

#### 4.2 ファイルの構造化

OS/0第2版では一次元的なストリーム型のファイルシステムを実現したが、この構造ではOS側でキメ細かな世代管理を行えなかった。そこで考えられるのが、ファイルの構造化、つまり、ファイルシステムがファイル中のデータ構造に関知する構成である。

データベースが扱うような複雑なデータ構造が扱えなくとも、ファイルの入出力SVCとして、insert, delete といったものが用意されているだけでも、ある

程度細かな世代管理が可能である。また別の見方をすると、アプリケーションプログラムの負担も軽減されることが考えられる。

OS/οはプログラミング環境だけではなく、真の意味での情報処理（特に日本語情報処理）環境を指向している。特定のアプリケーションを仮定した場合、プリミティブを提供して「何でもできるOS」とするよりも、「これがしやすい」といった面を持たせ、情報処理者（≠プログラマ）の負担を減らすことは重要であると考えている。

#### 4.3 追記型光ディスクの位置付け

OS/ο第2版では、仮想化した追記型光ディスクを磁気ディスクと同様な記憶装置として扱った。この実現方法は、実現が容易であることと、既存のOSで追記型光ディスクを利用できるという利点を持っている一方、追記型光ディスクの有効利用ができないという欠点を持つ。

そこでブロック型記憶装置の抽象度を上げ、各装置に適した管理方法を用意する方針を取る。追記型光ディスクは、仮想ディスク方式により管理し、仮想ディスクへの書込みは、書換えではなく追記を基本とすれば、追記型光ディスクの利用効率が上がる。このような形態では、追記型光ディスクを仮想化する意味がなくなってしまうと思われるが、仮想ディスク方式は、追記型光ディスクを見かけ上書換え可能にするだけでなく、追記型光ディスクの物理的な書込みエラーの対処ができるという側面を持っている<sup>[1]</sup>（磁気ディスクにおける代替セクタの機能を含んでいる）。そこで仮想ディスク方式を追記型光ディスクの物理的な管理手段として利用する。もちろん、ディスク上のファイル管理用のデータ（管理表など）は、書換えを必要とするため、仮想的な書換えの機能も利用することになる。

ただし、ここに述べたことは技術的な見地から見た方針であるので、前節に述べた世代管理のモデルによって、追記型光ディスクの利用形態もより細かく決っていくだろう。

#### 5. おわりに

本報告では、仮想ディスク方式と既存OSとの組合

せによって実現された世代管理機能の問題点と、世代管理を指向した場合、ファイルシステムはどのような形をとるべきかの考察を行った。

今後は、世代管理のモデルをより細かく検討し、次版OS/οの設計・実現を行っていく予定である。

#### 参考文献

- [1] 林努, 高橋延匡, “NC68000 用OSの基本設計(1) —ファイルシステムの設計—”, 情報処理学会マイクロコンピュータ研究会資料 25-3, 1982.12
- [2] 高橋延匡, “研究プロジェクト総説: OS/οmicronの開発”, オペレーティング・システム研究会資料 39-5, 1988.6
- [3] 横関隆, 中川正樹, 高橋延匡, “追記型光ディスクを用いた世代管理ファイルシステムの開発”, オペレーティング・システム研究会資料 36-1, 1987.9
- [4] 横関隆, 並木美太郎, 中川正樹, 高橋延匡, “追記型光ディスクの仮想的な書換えと世代管理機能の実現”, 電子情報通信学会論文誌 D-I, Vol. J72-D-1, No.6, pp.414-422, 1989.6