5G－4

OS／omicron第3版ファイルシステムの設計と実現

横関 隆，岡野 裕之，並木 美太郎，高橋 延匡
（東京農工大学 工学部 情報工学講座）

1．はじめに

我々は日本語情報処理およびその環境に関する研究を行っている．これらの研究の一環として，筆者は日本語情報処理を指向したOS（OS／omicron）の研究・開発を行っている[1]．本報告ではOS／omicron第3版[2](以後OMICRON V3）のファイルシステムに関する設計と実現について述べる．

本研究の目的は次のとおりである．

(1) OS研究用のOSを実現する

我々はOSに関する研究を行うために，自分自身で手軽に手を加えることのできるOSが必要であると考えた．また認知心理学などの研究で，被験者の打鍵情報のデータを収集する仕組みをOSの中に組み込む応用などを考え，OS内動作に関するデータ収集を指向したOSの実現を行う．

(2) 系統立った世代管理機能を実現する

ソフトウェア開発や文書管理に世代管理（版管理）機能が有用であることは自明である．本研究では一次元の時間軸にそった世代管理だけではなく，ソフトウェアのプロトタイピングやデバッグを仮定した多次元の世代管理機能を実現する．

2．OS／omicron第2版の問題点とOMICRON V3の設計方針

OS／omicronは現在第2版が実用化されている．OS／omicron第2版（以後単に第2版）は次の特徴を持ったOSである．

a．シングルユーザ・マルチタスク
b．タスクフォースの概念を導入
c．内部文字コードとして全2バイトコードを使用
d．二重階層構造のファイルシステム

このうちd.の二重の階層構造は，図1に示すようにファイルの中にも階層構造を持つものである．この構造は文字コードと属性情報の分離を指向したものであったが，実際に使用した結果，次の問題点があった．

(a) ファイルのオープン操作が二重になり操作が煩雑

ファイルの入出力を行う際には，ファイルの特定と，メンバファイル（図1中）の特定という二段階のオープン手続きを踏む必要があり，ユーザに混乱を与えた．

(b) ファイルの論理的な構造と二重階層構造の不整合

属性の属性（外字コードと外字フォントなど）といった応用が，二重の階層構造では困難であった．ファイル内の論理的な構造を考えたとき，三重，四重の階層構造が必要であるという検討結果を得た．

またプログラム構造の問題点として次のものがあった．

(c) モジュール間のインターフェースが不明瞭

OS内部はいくつかのモジュールに分かれているが，各モジュール間の結合が密な関数結合であるため，OS全体を理解しないと保守・改造が困難であった．このことは，モジュールの交換などが事実上不可能になってしまうことを意味する．

以上の問題点を踏まえて，OMICRON V3の設計方針を次のように設定した．

(1) ファイルの構造として入れ子構造を導入

第2版のファイルシステム同様，文字コードと文字属性情報を分離して扱う応用を指向し，二重階層構造を拡張・一般化した入れ子構造（親子構造）を導入する．

(2) プログラム内部のモジュール化

ファイルシステムのプログラムをモジュールと呼ばれる単位に分割し，各モジュール間の結合は疎なもので統一することにより，単体デバッグ，モジュール交換（測定ルーチンが組み込まれたモジュールとの差替えなど）を容易にする．

(3) 世代管理に追記型光ディスクを利用

世代管理は大量のデータを蓄積する必要がある．この世代管理機能に，大容量で一度書き込んだデータの消去ができない追記型光ディスクを用いる．また追記型光ディスクの媒体管理には仮想ディスク方式[3]を採用する．

3．OMICRON V3ファイル構造

3．1 入れ子構造

OMICRON V3のファイルは，前述したとおり第2版の二重階層構造を拡張・一般化した入れ子構造を採用した．

いわゆる階層ディレクトリ構造を持ったファイルシステムでは，ディレクトリとファイルを区別し，それぞれ等価に名前を付けている．これに対し，入れ子構造は図2に示すようにファイル内部がファイル実体と子ファイルからなっている．子ファイルは親ファイルと同じ構造を持っており，ディレクトリとファイルの区別がない．

Design and Implementation of File System on OS／omicron Version 3
Takashi YOKOZEKI, Hiroyuki OKANO, Mitarou NAMIKI and Nobumasa TAKAHASHI
Department of Computer Science, Faculty of Technology, Tokyo University of Agriculture and Technology

言い直せば，一般に言うディレクトリとファイルが一つずつ組となったものを，OMICRON V3では一つのファイルとして扱っている．

この構造における入出力モデルは既存のモデルと異なっている．基本動作は，

(1) ファイルのオープンによるファイル記述子の取得
(2) ファイル記述子を使用した入出力
(3) ファイルのクローズ

と，変わりないが，ファイルのオープンが「既にオープンされている親ファイルを起点としたもの」である点が異なっている．つまりファイルオープンのSVCが，

fd＝open(親fd，パス名，入出力モード)；

という形になる．したがって，タスクを起動する場合は少なくとも一つのファイルが事前にオープンされていなければならない．

このモデルにより，OS内部から「カレントディレクトリ」の概念がなくなった．なぜならば，オープンされているファイルは，すべてそのタスクにおけるカレントディレクトリ（ファイル）になり得るためである．

3．2 世代管理機能

OMICRON V3ではファイルを単位とした世代管理機能を実現している．第2版までのOS／omicronでは，一次元の時間軸に沿ってファイルを保存する世代管理機能を実現してきたが，現実には図3に示すように，各ファイルごとに時間軸とは別の次元の版が存在することがわかった．我々はこれをパラレルワールドと呼んでいる．パラレルワールドは主にソフトウェアの試作（プロトタイピング）やデバッグ時に発生する．また，多機種に対応したソフトウェアにおける，機種依存部分もパラレルワールドと考えることもできる．

このパラレルワールドを扱うためにOMICRON V3のファイルは，旧版ファイルとパラレルワールドファイルへの二つのリンクを持っている．また一つのファイルを複数の親から参照する形態として，LinkとDuplicateの二つを用意した．Linkは一つの実体を複数の親から共有する形態である．これに対しDuplicateは，Duplicateされた時点では，Linkと同様に実体を共有した形態をとっているが，変更が加えられるとリンクが切り離され，別ファイルとして独立する．この形態では共有されたファイルは，同じ内容を持つファイルが複数存在することと等価に扱われる．

世代管理機能のうち，旧版の管理はLinkが，パラレルワールドの管理はDuplicateが基本機能となる．

4．おわりに

本報告では，疎に結合したモジュールで構成したOS研究用のOSと，パラレルワールドの管理に対応した世代管理機能を実現したことを述べた．今後は，OS内部モジュールのデータベースなどを作成し，OSのプロトタイピングや，個人用のOSや専用OSを手軽に作成できる環境を実現した上で，本題であるOS各部の性能測定・評価や，認知心理学の研究への応用を行う予定である．

参考文献

[1]高橋：“研究プロジェクト総説：OS／omicronの開発”，情処学OS研資39－5，pp．1～15，1988．6．
[2]岡野，横関，並木，高橋：“OS／omicron第3版の設計と実現”，情処学OS研資45－11，pp．1～7，1989．11．
[3]横関，並木，中川，高橋：“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”，電情通論文誌D－I，Vol．J72－D－I，No．6，pp．414～422，1989．6．

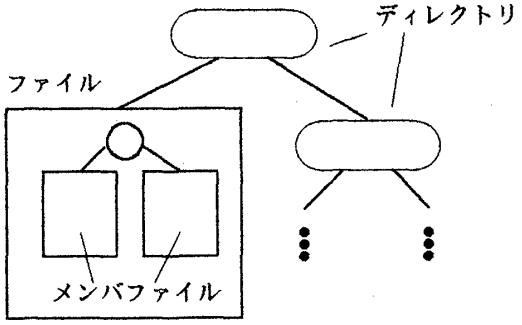

図1　二重階層構造

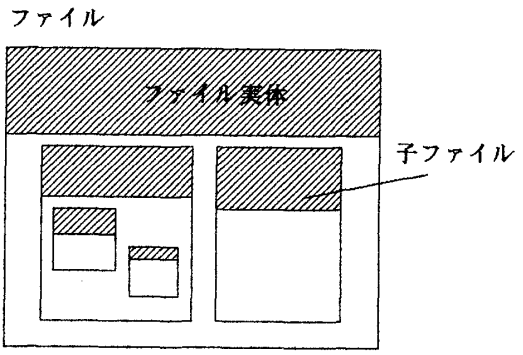

図2　入れ子構造

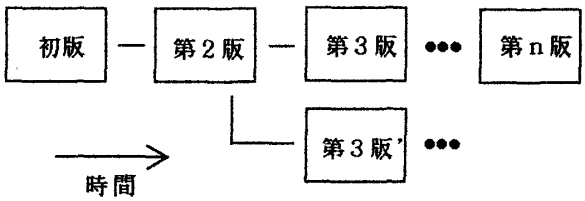

図3　パラレルワールド