解　説

仮想記憶を利用したシステムファイル†

登家正夫†† 辻　雄介†† 鈴木泰次††

1. はじめに

最近の情報処理システムはデータベース／データコミュニケーション処理，対話／タイムシェアリング処理，バッチ処理等の多用途に利用されている．それらの各処理形態が統合されたシステムでは，システムの大規模化，多様化と並んで，より高度なスケジューリングを行う管理・制御手法の実現が必要とされてきた．このため現代のオペレーティングシステムでは，その主要な任務である処理業務の遂行管理や制御，各種資源の有効利用を計るための管理や制御等，各種スケジューリング機構が，従来に比してより高度で多様なアルゴリズムを具備するようになってきた．そしてスケジューリングの要因となる情報は，プログラム，JCL（ジョブ制御言語），端末ユーザコマンドやセンタ操作員による指示情報の他，システムとサブシステムおよび業務プログラムの稼動状況，運転管理者や利用者の意志・計画等広い範囲に渡り詳細でしかも膨大な量になる傾向がある．その結果，システムコントローラが各種スケジューリングを行うとき参照するシステム制御表は，その個数およびそれぞれの大きさの肥大化を招いた．これらの情報の多くは，通常ファイルに格納されており，その管理と入出力動作のため，スケジューリングに費やすオーバヘッドが非常に大きなものになってきた．このような背景があり，システムコントローラは何等かの対策がなされなければならない．本稿では，その対策の1つとしてACOS-4/MVPで実現したシステムファイル用の仮想ファイルを紹介する．

2. ACOS-4/MVP (Multiple Virtual Processor) の仮想記憶制御

ACOS-4/MVPはデータベース／データコミュニケーション，タイムシェアリング，バッチの各次元の機能を共通にシステム制御プログラムとして組み込み，各次元へ共通の制御を行うように開発された統合多次元オペレーティングシステムである（図-1参照）．その仮想記憶方式はセグメンテーションにページングを備えた方式を採用している．ここで紹介する仮想ファイルは，この仮想記憶制御とファイル制御を組合せて実現している為，簡単にACOSの仮想記憶方式の概略と特徴を述べておく．

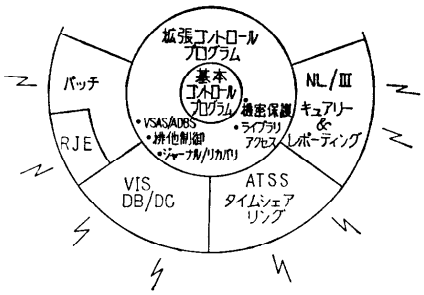

図-1　ACOS-4/MVPの統合多次元処理

・ページング方式を基に実現されている仮想記憶制御だけでなく，論理的分割管理単位であるセグメントの特長を活かしている．それは各タスクごとに持っているアドレス空間（多重仮想記憶空間）の優先度のみならず，それぞれの空間を構成しているコードセグメント，データセグメントが論理的に分けられ，それぞれの属性に応じて常駐性や常駐度等が配慮できる．

・高速化された演算処理速度と入出力装置の高密度化とのバランスのため，入出力バッファに必要なメモリが量的に大きな割合を占めるようになってきた．そこで入出力制御部にも二次元アドレス機構を設け動的チャネルアドレス変換機構を用いて，標準ファイルのバッファを通常のデータセグメントと同等に近い取扱いができるようにしている．

・VSAS (Virtual Storoge Access System) やデータベースでは，仮想空間上にバッファプールを持ち，このアクセスとアプリケーションに最も適した置換えを行い，同時にこの機構中でデータの保全のためのジャ

† System Files on Virtual Storage Concept by Masao TOKA, Yusuke TSUJI and Taiji SUZUKI (Basic Software Development Division, Nippon Electric Co. Ltd.).

†† 日本電気(株)基本ソフトウェア開発本部

ーナリングも自動的に行えるようにしている.

・システムファイルの仮想化

このようなシステム環境下においてシステムスケジューリングに使用するファイルに対し，主記憶が持つ高速性，二次記憶が持つ保全性，大容量性等の特長を活かして組合せ，専用の記憶階層を持ったファイルとして仮想待機結合編成ファイルを用意した．そして高度なスケジューリングが実装されても，システムオーバヘッドを増大させず，スケジューリングロスをシステムの許容範囲内に収める目的を持たせてある．したがって一般的な仮想化ファイルシステムを目指したものであるが，現在はシステムの内部用として最適化し実装している.

3. MVP のシステムファイル

システムがジョブ／ジョブステップ或いはコマンドを認識してスケジュールを行い実行するために用いられるシステムファイルの主なものは図-2 に示す通りである.

図-2 では説明を単純化するために，バッチジョブの入力から出力までの各種スケジューリングに使用される主なファイルが記述されている．これらのスケジューリングに使用されるファイル上のデータには，システム上に滞在する期間，アクセスの分布や頻度等にさまざまな特徴がある．これらのシステムファイルのうち記憶階層を用いたファイルに適用して，最も効果が期待できるものは多頻度乱アクセス型のデータである．通常，このようなデータは多数の情報群からなり，各情報群の中のデータは比較的少量である．またこのようなデータは，その性質上，待機結合編成ファイルが使われてきた.

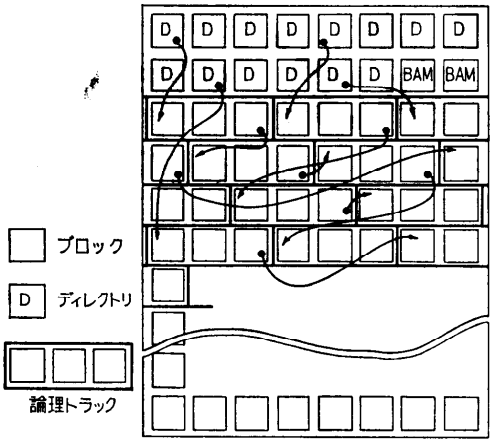

図-3 待機結合編成の概念構造

・待機結合編成ファイル

ACOS-4 系のオペレーティングシステムではスケジューリング情報を格納するファイルとしてライブラリ等にも適した独特の待機結合編成を基本として使用している．それは待機結合編成が，高い情報の分類性，速いアクセス性，効率の良いスペース管理を持ち，スペースの再使用性に最も勝れているからである.

図-3 は待機結合編成の構成を表わしている．待機結合編成ファイルは，ディレクトリ部とデータ部からなり，ディレクトリ部は物理ブロックをバケットとする乱アクセスが可能な構成とし，データ部は論理トラック（指定により１ブロックから数トラック分のブロックを１つの論理トラックとしスペース管理単位となる）に分割されている．各論理トラック内は物理的な順序に従ってアクセスされ，各論理トラック間はポインタで結ばれたリンク構造をしている．そして論理トラックの使用・未使用を管理するビットアドレスマップ (BAM) を持ってお

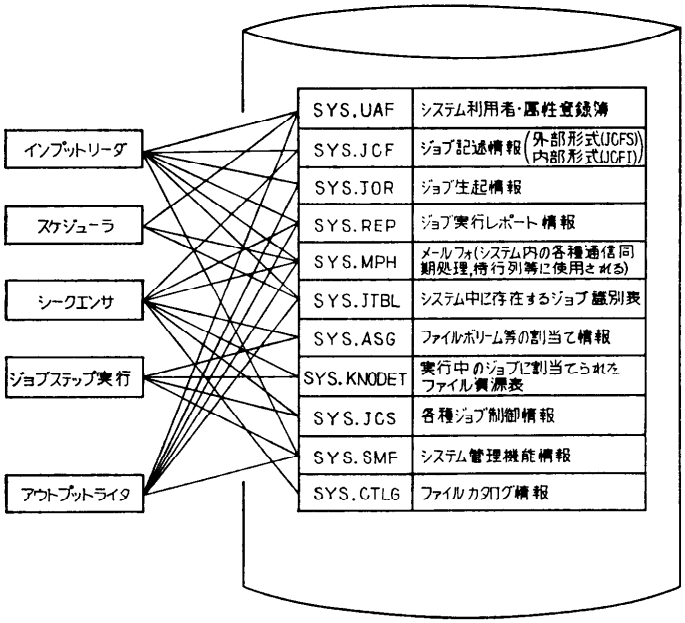

図-2 ACOS-4/MVP がジョブスケジューリングに使用するファイル

り，スペースの再使用性を高めている．各ディレクトリからは鎖状の論理トラックを指し，メンバとしている．そしてディレクトリ部はメンバ管理情報の他，255 バイトまでの利用者情報を持っている．したがってデータが 255 バイト以下の場合，ディレクトリ部だけの待機結合編成も可能である．（乱編成と同等レベルのファイルとなる．）

今回はこの待機結合編成ファイルを仮想記憶の技術を応用して階層化し，仮想待機結合編成として実現した．

4. 仮想待機結合編成ファイル

前章で触れたように，これらの対象システムファイルの制御データは，OS の非常に多くのモジュールで使用されており，またモジュール間で除々に移行して行くために次のようなポイントをおいた．

- OS 内部ユーザからのビジビリティ (visibility) を全く変えない．即ちこの階層化に起因して OS の構造／処理／プログラミングの変更は行わないようにした．
- 仮想／非仮想ファイルの選択は自由でファイルのアロケーションにより決まるものとする．
- したがって仮想ファイル（ファイルの概念）として実現しメモリアクセスのごとく一次元的なものとしない．

仮想待機結合編成ファイルは次の 2 つの仮想化を行っている．

（1） ディレクトリ部分の仮想化

ディレクトリ部分は乱アクセスされる頻度が非常に高く，また全体量として少量のため，仮想ディレクトリとして空間を設け，その実メモリと二次記憶上との間の移動管理は仮想記憶管理 (VMM) のアルゴリズムを利用した（図-4 (1) 参照）．

まずディレクトリ部に対して，システムコントローラは予めデータセグメントを用意しておく，そして VOPENS, VCLOSES, VBUILD, VSTOW 命令は，このデータセグメント上にディレクトリの作成・消去・検索・更新等を行うための動詞であるが，通常のデータセグメントへのアクセスと同等となるため磁気ディスク装置上へのアクセスと比べて極めて，高速性のすぐれたものとなる．

（2） データ部の仮想化

データ部の仮想化はデータ量の多さ，データ自体の内容の完結性／保全性のため，次のような形態をとった（図-4 (2) 参照）．

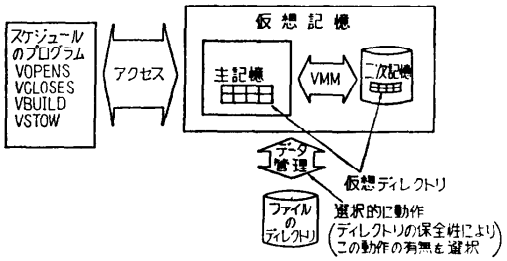

図-4 (1) ディレクトリ部の仮想化の概念

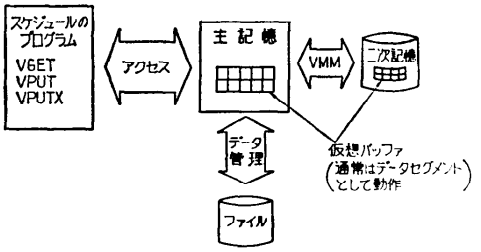

図-4 (2) データ部の仮想化の概念

- 仮想バッファを設け仮想バッファと実ファイルとの間の移動はデータ管理の制御下においた．これはデータの保全性（書き戻されたデータをファイル上に反映することを保証する）のためである．
- 仮想バッファはプール化し，このプール化されたバッファの仮想化は VMM の制御により行われる．
- さらに以前使用されたデータ自身の仮想バッファ上での再使用は CLOSES してメンバが閉じられた後，再度 OPENS で開かれたときでさえも有効とする．この動作を少し詳しく説明すると，ファイル上のデータ部から物理的な入出力動作を行うときは，一時的に当該ページを固定して行い，入出力動作完了後は通常のデータセグメントと同様な扱いを受ける．そして各々のメンバに対して N 個分のバッファ（すなわち，データセグメント）が確保できるので，以前アクセスしたデータブロックを再度読み直すといった無駄が，かなり少なくなる利点がある．データの出力・更新に関してはデータの保全性を考え，VPUT, VPUTX 命令と同期をとり，当該ファイルへ出力している．またディレクトリ部が仮想記憶上にあるため，各メンバをアクセスしている利用者やメンバのアクセス状況は，データ管理にとって，すべて同時に可視状態にある．このため，通常のファイルアクセスの場合，論理の整合性の都合上

OPENS A （バッファ確保）

```
GET      An  (データ入力)
CLOSES   A   (バッファ解放)
   ⋮
OPENS    A   (バッファ再確保)
GET      An  (データ再入力)
```

上例のごとく動作していたが，仮想待機結合ファイルでは，下記のようになり無駄な入出力がなくなるという利点がある．

```
VOPENS A   (バッファは予め準備させているデータセグメント)
VGET   An  (データ入力)
VCLOSES A
   ⋮
VOPENS A
VGET   An  (前の GET のデータ)
```

5. 仮想待機結合編成ファイルの適用

仮想待機結合編成は主として次のファイルで使用されている．

・SYS. JTBL：システム上に滞在するジョブに関する情報の集合で，ディレクトリ部のみからなる仮想待機結合編成である．

・SYS. MPH：一時的な使用と非一時的な使用の両方用意されており，ディレクトリ部のみを使用した小容量・高速通信同期機能とデータ部まで使用した大量のデータ通信同期機能がある．

・SYS. ASG：ジョブステップごとに使用する資源が登録されており，標準的な一定量をディレクトリ部の利用者データエリアに配し高速性を図っている．そして一定量以上の資源の要求はデータ部に登録される．

・SYS. KNODET：実行中ジョブに割当てられた資源表で，データ部のみからなるファイルであるが，バッファ数が十分に用意されており，非常に高い確率でメモリ上にヒットするようにしている．これはローカルな障害が発生した時使用していた資源を正しく戻す為にこの形式を選択している．

・SYS. JCS：ジョブの実行を制御する為の各種情報(例，各種制御変数)が格納されており，本来の待機結合編成に近い形式をとっている，しかし参照の多い情報はやはりディレクトリの利用者エリアに配している．

これらシステムファイルへの適用と評価のために次のような簡単な工夫により変更を容易にした．

仮想ファイルへのアクセスを明示するために動詞にVをプリフィックスしてきたが，実際のプログラミングは従来通りVを付加していない．このためOPEN時，割当てられているファイルの属性（仮想化ファイルか否か）を調べ以後，当該ファイルへのアクセス経路のスイッチをセットする方法により，プログラミング上の独立性を得ている（**図-5参照**）．

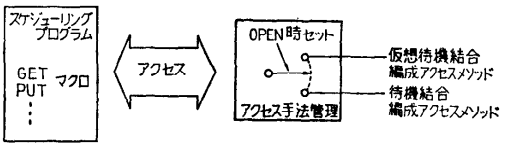

図-5　アクセス経路の変更

6. 評　価

今回のシステムファイルの記憶階層化による仮想化の導入は①ファイルの仮想化により，その使用者側に完全な互換性を保ち②情報の保全性／安全性を維持しつつ③システムの性能の向上を目的として，仮想ファイルの実用化への試行を行うことにあった．

システムの性能の観点では，次のようなデータが得られた．

1つのジョブステップに対してスケジュール／資源割当てを行い，実行させ／完了させるのに必要なシステムファイルの入出力回数を**表-1**に示す．

ジョブステップの集合としてのジョブの実行性能を“ジョブを構成する各ジョブステップが比較的小さい(10秒～30秒）業務”と“ジョブを構成する各ジョブステップが分単位（1分～3分）の業務”の2つに分け，その実測例を**表-2**に示す．

タイムシェアリングシステムにおけるセッション／

表-1　各システムファイルの入出力回数の比較

システムファイル	従　来	仮想待機結合編成
SYS. JCS	5	0
SYS. ASG	39	14
SYS. KNODET	30	0
そ の 他	3	3
計	77	17

表-2　ジョブの改善例

	従　来	仮想待機結合編成	改善度
ジョブステップ時間が10～20 秒のジョブ	2分53秒	1分53秒	34.7%
ジョブステップ時間が分単位のジョブ	7分30秒	6分25秒	14.4%

表-3 タイムシェアリングシステムにおける入出力回数の比較 (単位：回)

	従来	仮想待機結合編成
LOGIN	37	16
EDIT コマンド	34	5
COMPILE コマンド	57	8

コマンドの実行はバッチジョブのジョブステップの実行に OS の制御／管理の観点で酷似する．したがってこの仮想待機結合編成ファイルの適用は多端末同時処理の多いタイムシェアリングシステムにおける処理能力の向上にも非常に有効となっている．

現在稼動しているタイムシェアリングシステムにおける同様な入出力回数は表-3 の通りである．

待機結合編成ファイルの仮想化のうち効果の大きかった項目は，

① ディレクトリ部の仮想化，

② データ部の仮想化のうち次の OPENS 後への情報の引継ぎ，

の2項目であった．例えば SYS. KNODET ファイルでは①による効果が，SYS. ASG ファイルでは②による効果が顕著に表われている．

モジュール化構造でのオペレーティングシステムでは，制御情報が個々の機能モジュールにより少しずつ加工され流れて行く工程をとることが多い．この場合論理の完結性やデータの障害などの保全性から，そのデータの各工程（機能モジュール）でオープン／クローズすることが望ましい．このようなためにCLOSES後も仮想バッファの中のデータを有効にする手法が効果を上げている．

他方データの保全性については適用すべきシステムファイルの適切な選択により低下することを避けた．即ち保全性の必要なファイルは，その考慮のなされているデータの仮想化を適用し，一時的な情報を含むファイルや少量多種のデータは，保全性の低いディレクトリの仮想化を適用したことによる．

このような方法により，仮想化ファイルのユーザには何ら変更が発生しなかった．しかも適用すべきシステムファイルの選択／変更をシステムファイルのアロケーションだけで任意に変えられたことにより性能評価が容易であったのみならず，使用者の互換性が保証された．

7. おわりに

ここで紹介した機構は，システムコントローラ内部に閉じた範囲であり，しかも適用したファイルの性格も明確であったので，特に一部のファイル編成についてのみ行ったものである．この開発は一般かつユーザの永久ファイルに適用するために，まだ，より論理の汎用化と保全性の検討が必要であるが，かなり見通しが立ったといってよい．

(昭和55年1月24日受付)