

講座

データベースの実際(5)

—運転計画伝達システムと汎用ファイルプロセッサ(GFP)について—

稲田伸一* 林 真理* 槻木公一**

1. まえがき

最近のコンピュータのハードウェア、ソフトウェア技術の進歩につれ、ユーザシステムも高度に複雑化、大規模化への道を歩み、今までは企業組織の1つの道具として存在するに過ぎなかったコンピュータ・システムが、その組織の中核となり、大量の貴重な情報を一手に集約しつつある。この時、重要な役割を果たすソフトウェア技術の1つがDBMS(データベース管理システム)である。

多種多様な情報をデータベースとして一括管理することの長所、短所については、個々のユーザシステムにおいて慎重に検討されねばならないが、今後の方向としては、いわゆる情報処理を主体とするシステムは、システムの信頼性、処理の迅速性、運用などの面から、大なり小なりデータベースを中心にして発展していくのは明白であろう。

データベースを含むオンライン・システムは在庫管理型(バンキング・システムや予約システム)、プロセス制御型、生産管理型(スケジューリングやMIS)などに大別され、データ量、データ構造、レスポンスの特性、障害時の回復方式からシステムの利用者層に至るまでそれぞれに特徴がある。従って、すべてのシステムに適した汎用のDBMSの開発は非常に困難であり、いくら汎用と言えどもその適用範囲が限定されるのは当然である。

ユーザシステムの開発において、もし、ある汎用DBMSの提供機能とユーザシステムの要求条件とが一致すれば使用上何ら問題はないが、そのようなDBMSがない場合、ユーザは希望する機能、効率の範囲内でサポートされているDBMSを利用するか、あるいはプログラムの複雑さと将来の機能拡張をあきらめ

て、OSに用意されている基本的なアクセス法を利用し、ユーザ自身で専用のDBMSを開発しなくてはならない。

国鉄における運転計画伝達システムの開発に際し、DBMSの状況は後者の様相であったが、専用のDBMSにはとどめず、汎用化の1つの方向を狙ってGFP(Generalized File Processor)と名付けた汎用ファイルプロセッサを開発した。

2. 運転計画伝達システムとそのファイル処理

2.1 システムの目的

国鉄では毎日約3万本、延べ走行距離200万kmにも達する多くの列車が運転されているが、これらの列車は旅客、貨物の輸送需要に応じて日々変動しており、安全かつ正確な運行のためには綿密な計画、正確な伝達指示、万全の準備手配が必要である。

この輸送計画の作成、輸送管理に関する情報処理は、現在まで大部分のところ会議と電話による人間主体のシステムによって行なわれ、関連箇所(機関区、電車区など)への伝達は局報と言う形式がとられている。これでは計画作成から実施までの計画準備期間が短縮できず、長期的な需要予測に基づいた輸送体制に対処するだけで、短期的で急激な輸送波動に必要な弾力的、即応的な輸送計画の作成が困難となってきた。そこで運転システムの近代化が計画され、そのサブシステムの1つとして運転計画伝達システムから開発が進められた。

2.2 システムの機能

運転計画伝達システムの詳細は参考文献^{1)~3)}に譲り、ここでは簡単にこのシステムの処理の流れ(図1)と主要機能を説明するにとどめる。

○データベースの管理

列車の運転に必要な列車データ、車両運用デー

* 日本国有鉄道

** 同 鉄道技術研究所

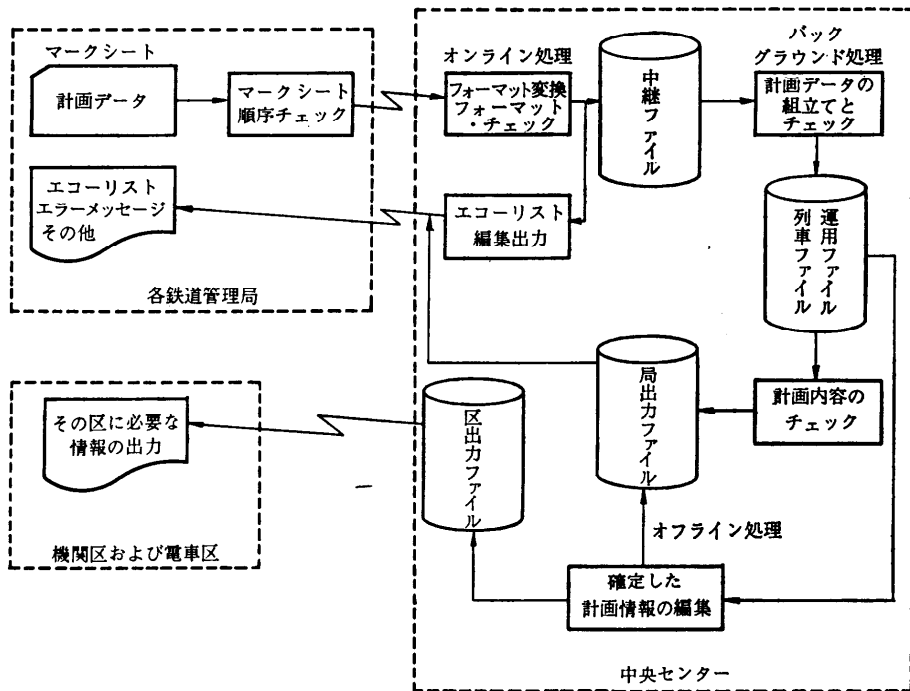


図1 システム処理の流れ

タ*, 乗務員運用データ**, 設備関係データなどの計画データを本社および各局***よりオンラインで入力し, データベースとして一括管理する.

○計画の精度向上

運転する列車に必要な車両, 乗務員が確保できるかどうかをチェックし, 計画の適否を判別する.

○情報の検索と伝達処理

必要となった運転計画情報をとり出し, 使用目的に合わせて編集し, オンラインで関連箇所へ伝達する.

○情報のステータス管理

計画の進捗に合わせて, 個々のデータのステータス(データ修正の可否, 現場への伝達の可否など)を管理する.

2.3 ファイル処理

2.3.1 データベースの構造

このシステムにおける主要ファイル群, すなわちデータベースは図2に示す構造になる. この中で, 列車ファイル, 車両運用ファイル(A運用ファイル), 乗

務員運用ファイル(B運用ファイルおよびC運用ファイル), レグファイル(各運用ファイルを列車単位に統合したもの)の各ファイルは可変長レコード・ファイルであり, レグファイル以外の4つはマルチ・インデックスを持つ簡単なラティス構造になっている. これ以外にも多くの中間ファイル, 作業ファイル, テーブル類が存在する.

2.3.2 ファイル処理の特徴

ファイル処理における最も大きな特徴は, このシステムの目的とする処理のため上記の5種類の可変長レコード・ファイルを構成せざるを得なかったことである. これら各ファイルの個々のレコードは単一の可変長データではなく, さらに可変個の可変長データから組み立てられた言わばサブファイルとも考えられる. 例えば, 列車ファイルの1つのレコードは, 同一列車番号を持つ複数個の可変列車データから成り, その他のファイルも同様である(図3).

一般的にはこのような場合には, 小単位の可変長データを1回のアクセス単位として考えてファイル処理するが, このシステムにおいては, 個々の可変長データに対する処理以外に, 1レコード内の可変長データ群に集中した処理が行なわれる. したがって, ユーザ

* 個々の機関車, 車両の列車編成への割付けや行路を管理するデータ

** 乗務員の運転行路を管理するデータ

*** 地方鉄道管理局の略

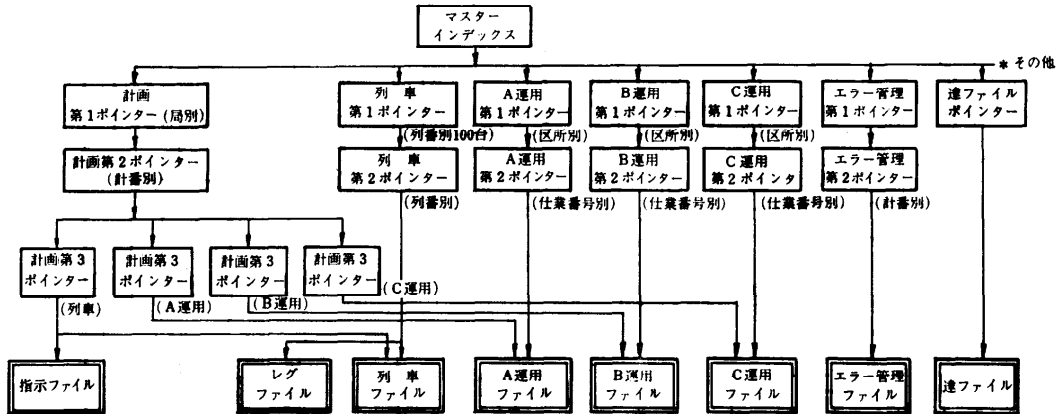


図2 運転計画伝達システムのデータベース

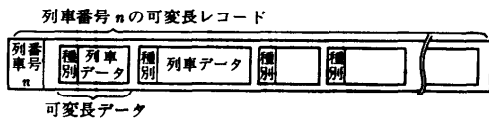


図3 列車ファイルのレコード構造

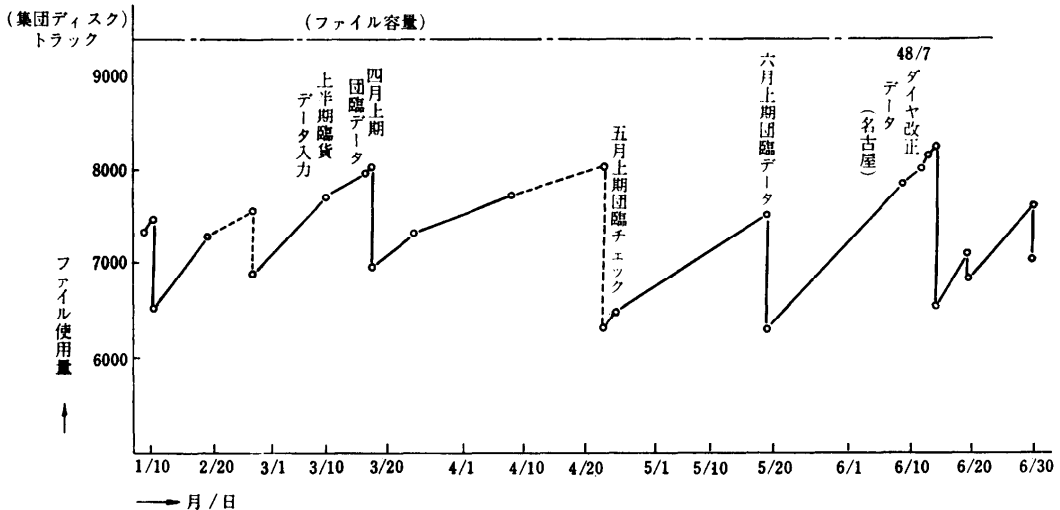
(応用プログラマ) にとっては、1レコードが1回のロジカルなアクセス単位であるのが望ましい。しかしその結果として、アクセス単位である1レコードは、数百バイトから数千バイト以上にもなる著しい可変長レコードとなる。

さらに、各可変長レコード・ファイルはシステムを中心となるファイルであり、図4に示すように格納、更新、消去など日々のファイル処理によって蓄積データ量が大きく変動し、各ファイルの最大量が定め難いのも特徴である。

2.3.3 DBMS への要求条件

2.3.2 で述べたファイル処理の特徴を満足するDBMSの備えるべき機能を抽出すると、

- (1) データベースの構造として階層構造のみならず、ラティス構造でもサポートし得ること、
- (2) データベースとして多量の変長データの著しい



- このグラフは、主として運転計画データを収めているファイル (GFP. POOL 5) の推移を示したものである。(使用量はサポートプログラムを用いて測定したもので点線部分は推定値である)
- ファイルメンテナンス処理で施行日の過ぎた不要データは消去される。

図4 主要ファイル容量の推移

レコードから成るファイルでも作成し得ること、

(3) レコードの量的変動に対し、領域制限の生じ難い適切な空領域管理を行なうこと、

(4) ユーザにとっては可変長レコードも固定長レコードの処理に似た手続きで、容易に効率良く処理が行なえること、

(5) 幅広く応用性と汎用性をもつこと。

当時、要求されたこれらの機能を備えた汎用 DBMS は提供されておらず、運転計画伝達システムのための DBMS を新しく開発する必要があった。

以上のような条件を満足し、かなりの汎用性をもたせるにはホスト言語型の DBMS が望ましいので、各種の機能を実現するマクロ命令の集積としての汎用 DBMS として、GFP システムを開発した。

3. 汎用ファイルプロセッサ GFP について

3.1 開発方針

一般に、ファイルの処理システムには図5に示す次の5つの階層レベルの構成要素が考えられる。

1. 入出力制御手順 (IOCS),
2. 基本アクセス法,
3. 物理的ファイル構造 (領域管理),
4. 論理的ファイル構造,
5. ユーザプログラム (又は独自のデータ管理法).

データ構造が単純なオンライン処理、在庫管理を対象としてきた従来のファイルでは、第2レベルの

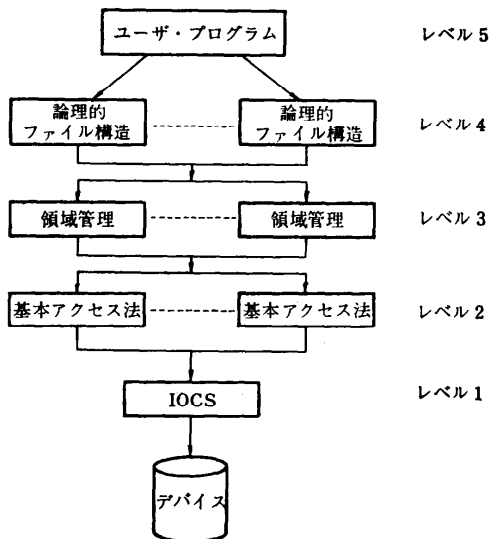


図5 ファイル処理における階層レベル

ISAM 編成でも十分に目的を果すことができた。しかしこのレベルでは、唯一性の保証できないキー、あるいは多重キーによる検索、多段にわたる階層構造、ラティス構造をもつファイルのサポートは不可能である。

一般の DBMS は、第3と第4のレベルが固定的に結合されユーザはファイル・イメージの論理的な構造のみを指定し、記憶空間へのマッピングについてはシステムが一方的に行なう方式をとったものがほとんどである。

そこで、GFP では汎用化の1つの方向として、第3レベルに属するGFP-Iと第4レベルに属するGFP-II とに明確に区分した階層構成とした。この構成により、ユーザは同一システムで使用目的に適したレベルを選択でき、さらに各レベルの構成要素を利用して希望する形の DBMS を作成することが容易になる。

すなわち、GFP における汎用化の狙いとは、各レベルをオープンエンドな形として DBMS 作成の道具を提供することにある。ただ現在の GFP では、効率上の問題から第2レベルの基本アクセス法は除かれている。

GFP の具体的な性能目的として、

(1) GFP-I では、ユーザが物理的な記憶空間への変長レコードのマッピングに必要なパラメータを直接設定することにより、2次記憶装置上に求めるべき領域効率、処理速度が確保できること。

(2) GFP-II では、レコード内の各アイテムが全て検索のためのインデクスとすることができ、各アイテム間には物理的に次の3つの関係を指定し得ることを目的とした。

1. Immediate association (または local association)

各アイテム間が物理的に位置関係で定まるもの。例えば、アイテムが連続して並べられたもの。

2. Direct association (または address association)
各アイテム間がアドレスポインタによって示された関係。

3. Indirect association (または key association)
キーによる検索が可能な関係。

3.2 GFP-I

3.2.1 領域管理

可変長レコードを取扱う場合、一般的に記憶空間でのレコードの格納配置および空領域を管理する必要がある。

GFP では基本的には、一定サイズのブロックをチェーンする方式を採っているが、このチェーン方式は、そのまま何の細工もせずと使用すると処理速度が悪くなってしまう。そこで以下に説明するプール、コンテナと言う領域概念を取り入れた。

○プール (Pool)

プールとは、同一タイプの2次記憶装置上にフォーマット・ライトされた長さの等しい未使用のフィジカル・フレーム (フィジカル・レコード) の集合であり、ユーザはこのプール名、領域サイズ、フレーム長などを設定する。このフレームはコアと2次記憶装置との I/O アクセス単位であり、フレーム長の異なった複数個のプールを設定することができる。

○コンテナ (Container)

コンテナとは、レコードを格納したフレームの集合を空領域管理、データ管理の1グループとみなす抽象的な概念である。一般に、領域管理は等長の固定長ブロックをチェーンで結んだ方式をとっているが、多種類の性質の異なったファイルを扱う時、単純な方法では領域効率、処理速度を向上することはできない。領域効率に重点を置き短いブロックを作れば、処理速度が低下するし、また長いブロックにすれば、領域効率が悪化する。

GFP では、同一システム内でもファイルの性質、使用条件に合わせてかなり自由に組合せを選ぶように、コンテナにタイプAとタイプBを作った。

タイプAのコンテナの特徴は、フレームをさらに細分割したブロックを単位として可変長レコード、空領域を管理する。図6で説明すると、ユーザレコードが5ブロック使用する大きさであれば、1フレーム内の空ブロックが5個あるフレームを見つけて、このレコードを格納する。もし適切な空ブロックのあるフレーム

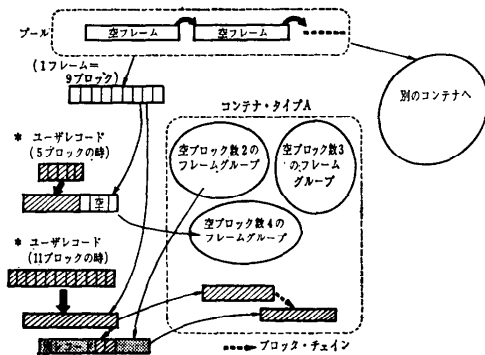


図6 コンテナ・タイプAのレコード格納処理

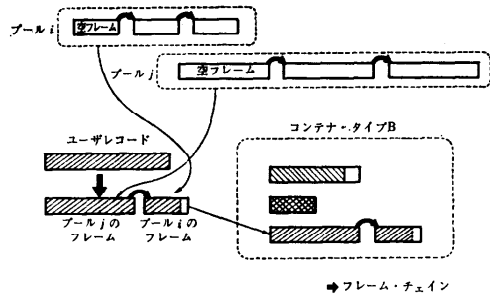


図7 コンテナ・タイプBのレコード格納処理

ムがコンテナになれば、プールから新しい未使用フレームを切出して使用し、残りの空ブロックは、コンテナの同数の空ブロックをもつフレームグループに組込まれる。また、レコード長が大きく1フレームでは格納できない時は、複数個のフレームにまたがって、レコードを格納するのに必要な数のブロックがチェーンされる。

このいずれの形をとる場合でも、常にレコードのアクセス回数が最小になるよう考慮され、可変度の大きいレコードの領域管理が、それ程処理速度を落さずに行なえる。

タイプBのコンテナは、1フレームを1ブロックとしこれ以上細分化された管理単位は持たない。したがって、ユーザレコードは1フレーム、あるいはレコード長が大きい場合は複数個のフレームをチェーンして格納される。このタイプのコンテナを複数プールから構成した場合、1コンテナにサイズの異なったフレームを持つことができ、図7のようにシステムはレコードを格納するとき、より最適にフレームを組合せる。処理によって生じた空フレームは、タイプAと同様に再びプールに返却される。

タイプAとタイプBの違いを性能面にとらえれば、タイプAは主として領域効率、タイプBは処理速度に重点を置くが、これら効率はプール、コンテナの各パラメータの設定により幅広く変化し得るものである。

運転計画伝達システムでは、処理する可変長レコードが大きいためタイプBを使用しているが、このフレーム長に対する主要ファイル容量と、I/O 処理時間の変移を考察したのが図8であり、どの点が最適であるかは、種々の外的要因を考慮して決められる。

3.2.2 定義機能

ユーザは GFP を用いた各処理の実行に先立ち、領域管理に必要なプール、コンテナの各パラメータを与

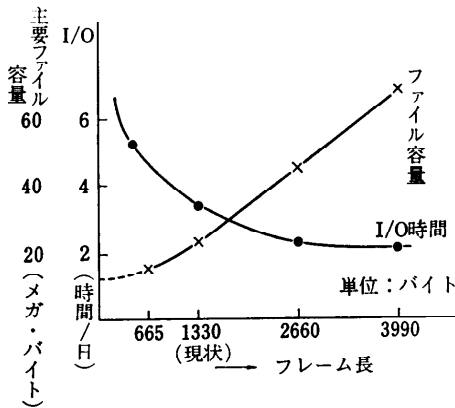


図8 フレーム長に対する主要ファイル容量とI/O処理時間の変移

えなければならぬ。これは定義機能のマクロ命令によって、プール・テーブル、コンテナ・テーブルから成る管理テーブルが作成され、同時にフレームの初期化が行なわれる。

3.2.3 処理機能

ユーザの可変長レコードの処理に必要な基本的なリード、ライト命令を COBOL, アセンブラのサブルーチンとして用意したものである。

このレベルでは次の4つのマクロ型の処理機能もっている。

- BRINGREAD, GET に対応する。
- STOREWRITE, PUT に対応する。
- CHANGE.....REWRITE に対応する。

ERASE不要となったレコードの削除命令で、生じた空領域は再使用のため、所属するコンテナ、あるいはプールの空領域管理に戻される。

格納命令である STORE を例にとると、パラメータとしてコンテナ名、ユーザレコードの作業領域を指定するだけで、GFP が最も処理速度の速い空ブロックあるいは空フレームのシリーズを見つけ出し、このレコードを格納する。その結果、格納アドレスがユーザに通知され、その管理はユーザに委ねられる。

また、更新命令 CHANGE では、レコード長が変っても格納アドレスは変化せず、より上位のファイル、インデックスのポインタの変更は不要である。その他、タイプBのコンテナの各フレーム単位のリード、ライト処理を行なう PBRING, PSTORE などがある。

3.3 GFP-II

3.3.1 データベースの作成

等質なレコードの集合をファイルと定義した場合、

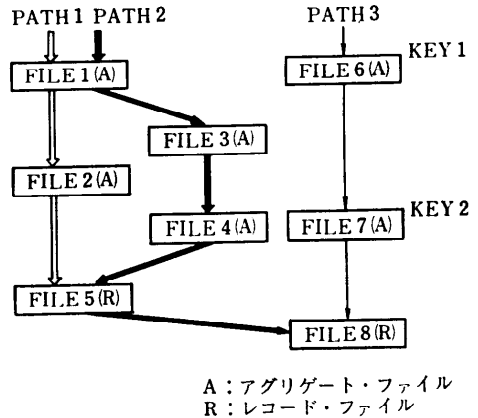


図9 GFP-IIによるデータベース構造

データベースとは相互に何らかの関連をもつファイルの集合と言えよう。

ファイル間の関連づけは個々のレコード間の関連をベースとし、IDSではチェイン名、CODASYL・DBTG案ではセット名として論理的に設定されるが、GFPではパス (path) と言う概念によって階層属性を表現する。パスとは、物理的には同一の direct association をファイル間に定義するポインタの総称である。

例えば、図9の構造をもつデータベースを設定するには、FILE 1 から FILE 8 までの個々のファイルの定義と同時に、FILE 1, 2, 5 を結ぶ PATH 1, FILE 1, 3, 4, 5, 8 を結ぶ PATH 2, FILE 6, 7, 8 を結ぶ PATH 3 の3つのパスを定義する。

パスの定義により、レコードそれぞれにポインタ・エリアが確保され、格納、更新処理時に自動的に下位レコードへのポインタが挿入される。

GFP がサポートするファイルには次の2種類がある。

○アグリゲート・ファイル (Aggregate file)

共通の属性をもつ固定長レコードの集合であり、ユーザにとっては連続した記憶位置に配置されたものである。単純なものはキー部だけで構成されるインデックスから、キー部と固定長レコードから成るシーケンシャル・ファイルを総称したものである。すなわち、キーによる indirect association をもつファイルで、single hit, index, fifo など6つのモードが選べる。

○レコード・ファイル (Record file)

immediate association をもった2つ以上のアイテムが集合したもので、レコード長は固定でも可変でもよい。換言すれば、キー付き可変長レコードのファイ

ルであり、より上位に少なくとも1つのアグリゲート・ファイルが必要とする。

3.3.2 定義機能

前述のアグリゲート・ファイル、レコード・ファイル、パスなどを定義するマクロ命令群であり、論理的なデータベースの構造を記述する FST (File Structure Table) を作成する。

3.3.3 処理機能

GFP-II の処理命令も、論理的にファイル进行操作するパラメトリックな形をした COBOL、アセンブラのサブルーチンである。たとえば、図9の FILE 8 のレコードを PATH 3 に沿って、FILE 6 の KEY 1、FILE 7 の KEY 2 によって求めるには、

RREAD (FILE 8, PATH 3, KEY 1, KEY 2,
読出し領域名)

の形をしたリード命令を発行するだけでよい。

このキーによるリード命令以外に、ファイル・モードによるシーケンシャル・リード命令、これらに対応するライト命令、および更新、削除命令、さらにファイル名などの変更、指定ファイルのコア常駐化、ポインタの直接操作命令などがある。

3.3.4 保護機能

GFP が使用された運転計画伝達システムは、一般の不特定ユーザにデータベースを開放するものではなく、含まれる情報も厳密なロックが必要な性質のものでもない。むしろ、機密保護の機能より、不注意による破壊、書き込みを禁止する機能が必要である。この機能は、GFP-II の段階でファイル相互を関連づけるパスによってなされる。

ファイルに入るパスには3つのモードがあり、定義時に定められる。プライマリ・パス (primary path) は、1つのファイルに1本だけ指定可能でこのパスを経由してファイルの内容を変更できる。リードオンリ・パス (read-only path) は読出し専用パスであり、フリー・パス (free path) はこれらの制限条件のないパスである。この組合せにより、保護モード、非保護モードが選定できる (図10)。

また、処理機能のパスのオープン、クローズ命令は、必要なファイルだけをオープンし、保護機能の1つとなっている。

3.4 GFP によるファイル処理の結果

運転計画伝達システムのデータベースの処理に際し、GFP-II は開発時期の関係上使用できず、やむを得ずこれらのファイル、インデクスはすべて GFP-I

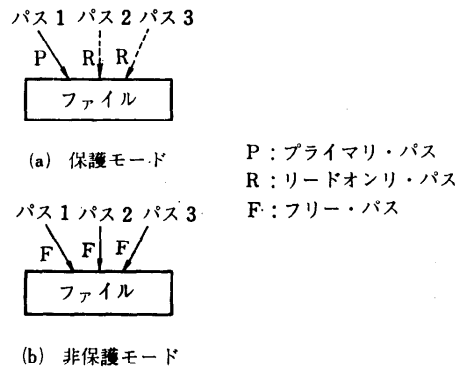


図10 パスによる保護機能

により処理されている。そのため、GFP-II に備わった論理的なデータベース操作機能が活用できず、ポインタ処理はすべてユーザサイドで慎重に行なったにもかかわらず、やはりシステム全体の安定に対し、1つのマイナス要因となったことは否定できない。

しかし、可変長レコード・ファイルを中心とした個々のレコードの処理においては、GFP-I による可変長レコードの取扱いの容易さ、プール、コンテナの概念による領域管理方式は、ユーザにとって極めて有効なものであった。もし、可変長レコードが取扱えず、すべて固定長レコード・ファイルとすれば、各ファイルは5~10倍程度に領域を拡張せざるを得ないだろう。

一方、可変長レコードの中には、レコード長が20kバイトに及ぶものも発生し、コアのI/O エリア不足から GFP だけでは対処しきれず、ユーザサイドでさらにレコードの分割、統合処理を行なっている。このような特異なデータの存在は、全体の処理速度を悪化させる大きな原因でもあり、今後 GFP サイド、システムサイド両方での再検討が必要である。

また、チェイニングによるデータ管理は、GFP の方式でも長期間になるとやはり処理速度が低下するので、ユーザレコードの再配置ユーティリティによるファイル・メンテナンスも不可欠である。

4. あとがき

運転計画伝達システムのため開発した GFP は、これ自体でクローズした DBMS でなく、ユーザの要求にマッチした DBMS 作成のための1つの道具としてオープンエンドなシステムとすることを目的とした。そのため、機密保護や回復機能はサポートせず、ユー

ザに個々のシステムに適した補強を委ねることにより、幅広い汎用化の形をとった。

ユーザシステムのファイル処理は、その要求する性能や機能にそれぞれ特徴があり、汎用化された DBMS の存在を拒んでいる。したがって、すべての機能を含み完成された DBMS を追求するだけでなく、DBMS は単にアクセス法の延長として種々の基本的な機能の集合としてサポートし、その中でユーザが使用目的に合わせて独自のデータベース管理を行なえるような融通性のある開発の方向も、今後もっと考えられてもよいのではないかと思う。

末筆になるが、GFP を開発するにあたって、プログラム作成に協力していただいた日立製作所のシステム開発研究所、ソフトウェア工場の方々、またこの実用化にあたり、実際面から種々の問題を指摘され、欠点を改良された国鉄オペラン・プロジェクトチームのメンバーに対して謝意を表したい。

参考文献

- 1) 林真理：国鉄における運転計画伝達システム、情報処理学会データベース研究会資料 (1973).
- 2) 稲田，林，他：運転計画情報管理システム (OPERUN-D)，鉄道サイバネティクス・シンポジウム論文集，第7回，pp. 239～247 (1970).
- 3) 大谷昌弘，他：オペランDにおけるファイル処理方式，同上，第10回，pp. 196～201 (1973).
- 4) 稲田，根木：運転情報処理におけるファイル管理システム，同上，第10回，pp. 202～206 (1973).
- 5) 稲田，松岡：汎用ファイル・プロセサの一方式，情報処理学会第11回大会予稿集，p. 339 (1970).
- 6) 根木，稲田，他：汎用ファイルプロセサ GFP-II の製作と実動テスト，同上第13回大会予稿集，p. 247 (1972).
- 7) N. Inada: A generalized file processor as applied to railway scheduling information system, First USA-JAPAN Computer Conference, pp. 616～620 (1972).

(昭和49年1月18日受付)