

B 1. 汎用ファイル処理システムについて

水 野 幸 男 (日本電気KK)

目 次

序 論

1. ファイル処理システムの現状及び今後の傾向

- (イ) ハードウェアの進歩
- (ロ) データ量の増大
- (ハ) データ処理業務の多様化と統合化

2. ファイル構成とその処理方式

- (イ) ファイル構成の基本型
- (ロ) コーボレイトデータファイル

3. ファイルシステムの実例

- (イ) CISS (Consolidated Information Storage System)

- (i) プログラム構成

- (ii) ファイル構成

- (iii) アクセス方式

- (ロ) その他のファイルシステム

- (i) DM-1

- (ii) IDS

- (iii) TDMS

- (iv) IMRADS

- (v) FFS

- (vi) GIS

4. ファイルシステムの問題点及び今後の課題

序 論

データ処理の分野で最近の発展としてもつとも注目すべきものは汎用ファイル処理システムである。磁気テープによるバッチ処理中心の第2世代の計算機から磁気ディスク、磁気カード等のマスメージックや各種オンラインターミナル及び豊富な遠隔処理装置を備える第3世代に遷り、リアルタイム及びタイムシェアリングに対して、又マネジングインフォメーションシステムを初め複雑多様化する適用業務の要求を満たすため、ファイル処理システムの重要性は増大する一方である。

本文では汎用ファイル処理システムの動向及びその問題点を検討する。

1 ファイル処理システムの現状及び今後の傾向

(イ) ハードウェアの進歩

データ処理のコストは急速に低下しておりCPUに於ける内部処理のコストは3年ごとに $1/3 \sim 1/5$ となつてゐるが、記憶コストにおいても過去10年間で $1/3 \sim 1/25$ になり更にこれを記録媒体についての見れば(磁気テープ、ディスクパック等記録媒体が装置から取替可能なものについては媒体のみのコスト) $1/30 \sim 1/500$ と下落した。また一方記憶容量については磁気テープで一千万字程度、ランダムアクセスの可能な磁気ドラムにおいては、高々数百字であつた。しかし現在磁気ディスク、磁気カードにより数億字の容量でしかも平均200m秒以内でランダムアクセスが可能な装置が開発された。

この様な進歩は今後とも期待されるが、これを今後5時間について考えれば、大容量記憶装置の主流となるものは、やはりエレクトロメカニカルなもので、これに対する改善は記録密度の増大とアクセスタイムの減少であろう。記録密度については磁気テープの場合過去10年間で200bpiから1600bpiと8倍になりさらに1970年までに2,000~3,000bpiになるであろうと予想される。

(ロ) データ量の増大

上述の如く大容量であり、ビット当りの記憶コストの安い装置の開発に対して実用上の要求はこれを上廻るものがあり、かつ格納されるレコードの長さも大きくなる一方である。米国での調査によると各企業での必要量は2百億~4百億であつたが、これも過少評価であると云われ、今後10年以内に中規模の企業でも1千億字程度になるであろうと思われている。従つて、多量のデータを高速に能率よく、しかもマルチプログラミングあるいは、マルチプロセッシングで処理を行えるファイル処理システムが必要されるに至つた。

現在、この様に多量のデータを高速にかつ安いコストで処理するために考えられていることは、二次記憶装置にハイラキーを成して格納する方式である。即ち各レベルは異つたアクセスタイムとコストを持ち、最も頻繁に使われるデータは最も速いアクセスの装置に

入れられその使用頻度でデータの配分を行う。また必要に応じて各記憶レベル間でのデータ移送が可能になることである。例えば、パッチ処理を行うときこの部分のデータを高速の装置に遷し、処理が終つたら低速の装置に戻し、かつユーザはデータが何処にあるについては全く配慮しなくてもよい。ただファイル名とパスワード等を知つていればよいという方式である。

(c) データ処理業務の多様化と統合化

データ処理を EDP 化する場合にその程度により、5段階に分けられる。

- (i) 給与計算のようにすでに確立された業務をそのまま計算機で行うもの。
- (ii) 上記業務の中で特に関連の深いものを若干統合して処理を行う。例えば在庫管理と生産管理を統合する。
- (iii) 多数の業務のデータを統合し集中管理を行い、データは各業務で共通に使用する。この場合は、各業務に対して互換性を有した標準化されたものでなければならない。
- (iv) 上記 i, ii, iii はデータ処理の部門でのその EDP 化及び統合化であるが、さらにこれを通じて実際の業務の一部に大幅な改善を行う。例えば在庫量を減じ、データ処理を単純化するために、製品系列の再検討と標準化を行い、又顧客へのサービスの向上を計ると云つたことである。
- (v) 基本的な業務は総て統合化され、これらの遂行は計算機と一体となつて行なわれる。

上記において従来は (i), (ii) の段階であつたが、最近の MIS 的考え方により (iii) 段階に遷ろうとしている。(i), (ii) 段階では、ファイルというものは各部門別にその処理に最も適した形成で作られた。この場合ファイル内のデータはほぼ同一形式のもので、そのファイル処理において技術的にさほど大きな問題とならなかつた。しかし (iii) 段階以降では部門別ファイルは作られず、総てのデータは統合化して共同で作られるため、ファイル内には多種多様なデータが存在し、かつその量も膨大なものとなる。

この種のファイルをコーポレートデータファイルと云うが、これは各部門のデータを統合して各プログラムと独立したファイルであり、プログラム又はデータ定義の変更の容易化、システム統合の促進などの利益を提供する。更にデータを統合化することにより、個々のプログラムでデータの保守を行う必要がなく、業務全体が簡素化される。さらにデータを共通で用いるために、一個のデータが各所に記録する必要がないため、記憶装置に無駄がなく、更新処理を一度行うだけでよく従来の如く各部門のファイルを総て更新する必要がないため処理は迅速に行え、かつ常に最新のデータが利用できる。しかし一方では、各プログラムに最適の形式のファイルができず、処理効率が落ちる場合がある。また各ファイルのユーザはファイルの一部を使用するのみであるから、ファイル全体の構成を知らなくとも使用できねばならず、またデータの定義が変わるとに全プログラムを再コンパイル

ルする必要をなくすことが望ましい。

過去においてプログラムとデータの定義の間には、非常に密接な関係があつた。しかし COBOL の出現によりデータの定義は再プログラミングの必要などに変更できるようになつたが、再コンパイルは現時点ではまだ必要とされる。従つて、これに対する解決策は、データの定義は実行時に翻訳を行う様にしてデータとプログラムを独立させることである。

ファイルシステムが高度化されるに従い、その設計と運用は専門化した領域になりつつある。これは非常に複雑な領域であるためデータ専門家という職務が生れるであろう。そして従来のシステム分析者やアプリケーションプログラマーは自分でデータファイルの設計は行なわず、その要求事項とファイルが如何に用いられるかの若干の説明をその専門家に提供して、その設計を委ねることになるであろう。従つてデータの専門家はファイルの設計に関して効率上の考慮、機密保護、カタログ作成、ファイルの拡張などに習熟したものでなくてはならない。

2 ファイル構成とその処理方式

(イ) ファイル構成の基本型

汎用ファイル処理システムについて述べる前にファイル構成の基本的な考え方を説明する。これには通常次の四種がある。

- (i) Sequential
- (ii) Partitioned Sequential
- (iii) Indexed Sequential
- (iv) Direct

これらはオペティングシステムの IOFCS (Input Output File Control System) で提供される機能であり、汎用ファイル処理システムでは、これをさらに複雑高度化したものであり、IOFCSのもとで働くように作られている。しかし一部にはファイルシステム自身がこの IOFCS 機能を有しているものもある。

(i) Sequential

シーケンシャルなファイルとは図-1 の如く連続して記入されているもので、磁気テープの記録方式は総てこれで

ある。これは記録容量が有効に無駄なく使用できる利点があるが特定のレコードが必要なときに、ファイル

レコード	レコード	レコード	
------	------	------	--

図-1

全体を調べなければならず、またレコードの追加、削除等の場合、部分的な更新で済むことができない。

(ii) Partitioned Sequential

前記 Sequential なファイルをメンバと呼ばれる単位に分割して、各メンバの開始場所を示すディレクトリを作つたもの。

(iii) Indexed Sequential

ディレクトリ

Partitioned Sequential で各メンバの格納開始番地を示したのに対して各レコードの格納番地を Index で示したものである。

Index はレコードのキーとその格納番地より成つてゐる。この場合 Index のキーはある順序で連続に並んでいるが、実際のレコー

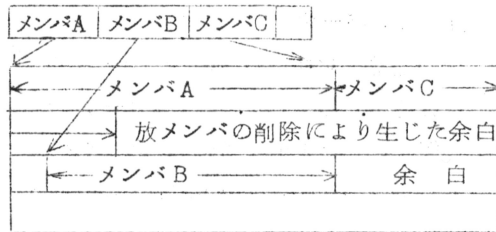


図-2

ドはランダムでもよい。この方式は インデックス

は Index によりファイル全体を調べることなくレコードを得ることができ処理もその部分についてのみ行えば済む。

この方式はインデックスにより Sequential な処理とダイレクトアクセスが同時にできるが、常に二回以上 (インデックスを多重

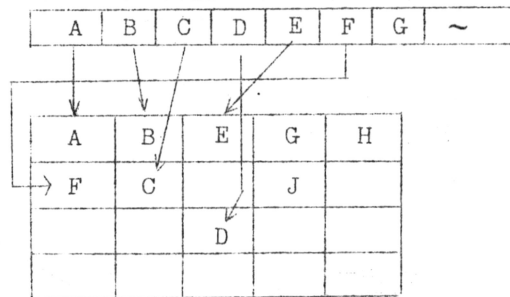


図-3

構造にすることができる) のアクセスが必要であり、かつ更新処理では、レコードとインデックスの両方の更新が必要となる。

(iv) Direct

レコードのキーを数学的に変換して格納番地とするもの。キーの変換式を知つていれば、キーが与えられたとき、これより直ちに格納番地を計算して求めることができ、インデックスを用いずにダイレクトアクセスできる。例えば、0 から N 番地の間にレコードを格納するときは、キーを N+1 で除して余りをその番地とすればよい。この場合、異なるキーが同一格納番地に割当てられる場合があり、これをできるかぎり減すためそのキーの性質により変換式を考えて定めねばならない。

(ロ) コーポレイトデータファイル

データを総合し、プログラムから独立させるコーポレイトデータファイルについてはすでに述べたが、このファイルシステムの特徴は強力なデータ検索であり、そのためファイル構成に柔軟な手段が与えられている。即ち通常のファイル構成では単一キーを基準に

mass storage の評価 $M = C \cdot R \cdot F$

$C = \text{capacity / cost}$

$R = T_1 R_1 / T_1 T_2 + \sqrt{T_1 T_2}$

$F =$

変動する移動距離

してファイルが作られているが、汎用ファイル処理システムにおいては如何なる基準からデータが得られるように内容による検索が必要である。

内容による検索は、インバーテッドファイル、二次インデックス、チェイニングの三通りがある。以下それぞれについて解説する。

(i) インバーテッドファイル (Inverted file)

通常のレコードをフィールド単位に分割し、同種のフィールドをまとめて記録したもの。この場合、元のレコードを成すものは互にその関連を示す情報を有する。例えば図-4をインバートファイルにしたのが図-5である。即ち社員コード、名前、本給年令、勤続年数、専攻の6つのフィールドを4通りに分割してファイルを

社員コード	名前	本給	年令	勤続年数	専攻
1207	伊藤 一郎	11,000	36	10	経済
1357	鈴木 誠	6500	25	3	機械
1388	山田 英二	7000	27	4	数学

図-4

作つたものである。ここでそれぞれをSegmentと云い、各Segmentではフィールドは連続に格納されている。このファイルでの検索は、例えば社員コードか与えら

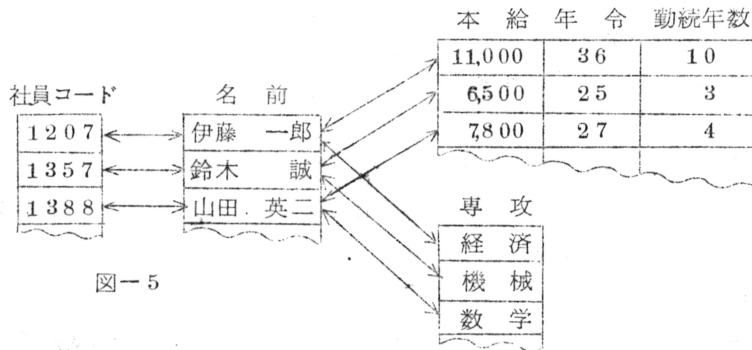


図-5

れてその名前と本給を求めるときは、社員コードのセグメントを調べ一致するものを求める。このフィールドには名前のセグメントでこれに対応する社員の名前の格納番地が示されており、これより名前を得ることができ、さらにこれより本給も得られる。またある特定の科目を専攻したいときには、専攻のセグメントでその専攻名を総て調べ、これに対をする氏名を得られる。また数種の条件を“AND”“OR”で結び、これを満すデータを得ることができる。

なお、ここですべてのフィールドを分割するのでなく“本給”、“年令”“勤続年数”をまとめた如くその性質により分割しないこともできる。

(iii) 二次インデックス

インバートドファイルは二次インデックスの特殊なものである。インデックスが必要なフィールドについてのみ、それに関するインデックスを作る。通常二次インデックスで最初に基準に合うレコードを探し、次にこれで得た結果を用いて一次インデックスを調べて求めるデータを得る。ここでどのフィールドについて二次インデックスを設けるべきかは一般的な問題への解はまだ未解決である。

(iii) チェイニング

相互に関連のあるレコードを結びつけるものである。

例えば右図でレコード A には B のレコードの番地を有し、B はさらに C の番地を有す。従つて関連するレコードを順次読出すことができる。ここで各レコードは任意個のチェーンに属することができる

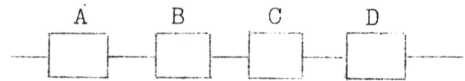


図-6

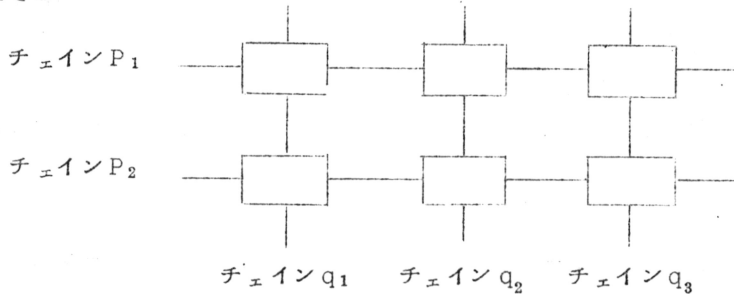


図-7

3 汎用ファイル処理システム

(1) CISS (Consolidated Information Storage System)

本システムは磁気ディスクを対象とした汎用ファイル処理システムであり、ユーザはファイルの構造を詳しく知らなくともファイルの読出し、更新が容易にできるものである。

(i) プログラム構成

・ CISS MONITOR

・ TRANSLATOR

・ FILE CONSTRUCTOR

FILE REORGANIZER

・ UTILITIES FILE MAP

LOAD/UNLOAD

(a) CISS MONITOR

ジョブプログラムとオペレーティングシステムの IOFCS との仲介をするものである。

(b) TRANSLATOR

ジョブプログラムは COBOL で書かれており、CISS 言語で書かれたものを COBOL に変換するのが TRANSLATOR である。

(c) FILE CONSTRUCTOR

ファイルはジョブプログラムと別個に作られるが、これを行うのが FILE CONSTRUCTOR である。これは各データのフォーマット、その相互関係及びこれらが装置内で物理的に如何に格納すれば効率よくできるかを考えて、パラメータを定め、データを入力すれば自動的にファイルができる。

(d) UTILITIES

・ FILE REORGANIZER

ファイルが作られた後、多くの更新処理が行われると次第に使用効率が低下して行くため、改めてファイルを作り直して効率のよいファイルとするもの。

・ FILE MAP

ファイルの TABLE 又は DATA 部に対して希望する部分をプリンターに編集して印字する。

・ LOAD/UNLOAD

ディスクのデータを磁気テープに出したり、逆に磁気テープよりディスクに格納納する。

(ii) ファイル構成

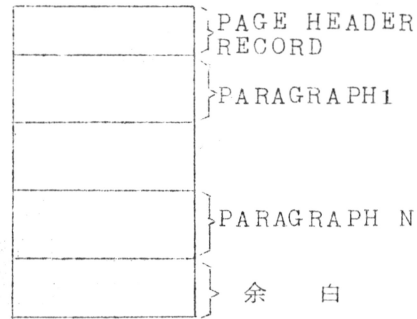
本システムのファイル内のデータは FIELD, RECORD, SEGMENT という単位でまとめられる。即ち FIELD とはデータの最小単位であり、氏名、年令、本給といったものはそれぞれフィールドである。RECORD はこの FIELD より成る。例えば各従業員の氏名、年令、本給を合わせたものが、それぞれ一個の RECORD である。なお関連する RECORD についてその関係を示すために CHAIN, LINK (後述) がある。

次に同種の RECORD の集合を SEGMENT という。例えば前述の氏名、年令、本給からなる RECORD の集合が一個の SEGMENT を作る。

次にファイルを入出力の単位で考えるとこれを PAGE という。これはファイル内ではすべて同一の固定長である。なお、この PAGE の内部は PARAGRAPH と云う単位で分割されている。PARAGRAPH とは同一 SEGMENT の RECORD より成り、その長さは PAGE 内で任意である。よつて PAGE のフォーマットは図

— 8 の如くなる。

即ち数個の PARAGRAPH と未使用の余白と PAGE HEADER RECORD より成る。ここで PAGE HEADER RECORD とは各 PAGE の始めにありその PAGE 番号、各 PARAGRAPH の開始位置、余白の文字数等 PAGE の状態を示すものである。



次に SECTION と云う単位がある。

これは各 SEGMENT の RECORD は分

図— 8

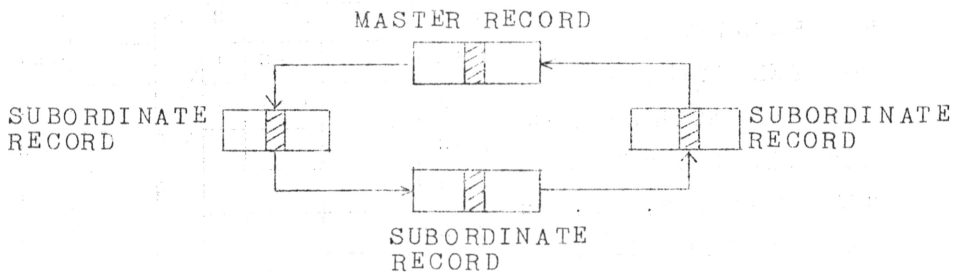
割されて PARAGRAPH と云う単位で PAGE 内に収められるが、この同一 SEGMENT の PARAGRAPH を含む PAGE の集合が SECTION である。従つて SECTION は数個の SEGMENT より成るとも云える。ここで SECTION 内の SEGMENT 数、即ち PAGE 内の PARAGRAPH 数に対する制限はなく、ファイル作成時に任意に与えられる。

・ CHAIN と LINK

SEGMENT 内外の RECORD の相互関係を示すために、RECORD 内に POINTER FIELD と云う次に続く RECORD の位置を示すフィールドを設けて CHAIN 又は LINK を作る。

* CHAIN

下図の如く各 RECORD は POINTER FIELD (斜線の部分) により、ループを作つたものを CHAIN と呼ぶ。



図— 9

この場合 CHAIN の終始端となる RECORD があり、これを MASTER RECORD と云い、他を SUBORDINATE RECORD と云う。各 SUBORDINATE RECORD は同一 SEGMENT の RECORD でなければならず MASTER

RECORDは異なるSEGMENT でなければならない。図-9に於いてはCHAINは一方だけであつたが、次の様に逆方向及びそのMASTERを直接示すPOINTER FIELDを設けることができる。

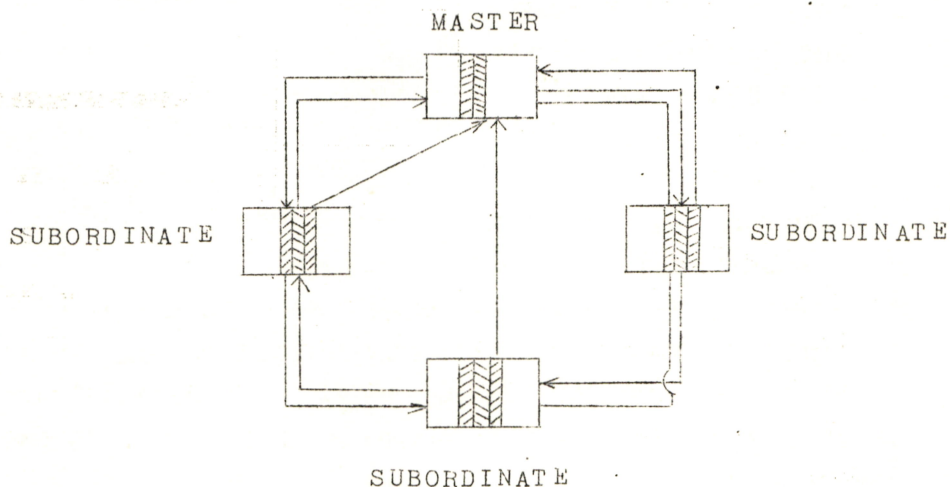


図-10

* LINK

LINKはCHAINと同様にPOINTER FIELDにより次のRECORDを示す。しかしループにはならず下図のようになる。

例えば上図に於いてMASTERとなるRECORD m はそのPOINTER FIELDでSUBORDINATEのRECORD n 及び3を示すことにより、 n より始まる3個のRECORD $n, n+1, n+2$ が m のSUBORDINATEであることを示す。従つてこれは m を

m	n	3	→	n
$m+1$	$n+3$	1	↘	$n+1$
$m+2$	$n+4$	1	↘	$n+2$
$m+3$			↘	$n+3$
$m+4$			↘	$n+4$
$m+5$				$n+5$

図-11

MASTERにするCHAINと同じことになる。しかしこの場合SUBORDINATEにはPOINTER FIELDが不要である。又これら二個のSEGMENTのRECORDは他のSEGMENTのCHAIN又はLINKのMASTER又はSUBORDINATEとなることが出来る。この場合の制限はCHAINのときと同様である。

(iii) アクセス方式

ファイル内のあるデータが必要なときは、それに関する条件を定めればCISS

MONITOR がこの条件と照合して一致したデータがジョブプログラムに与えられる。例えば年令35才以上で勤続年数が10年以下で経済か法律を専攻した人の氏名を求めるときは、

```
RETRIEVE NAME IF (AGE LE @35@ AND CAREER HE
  @10@) AND (MAJORMENT EQ @ECONOMICS@ OR
  @LAW@)
```

ここで、“NAME”、“AGE”、“CAREER”はFIELD NAMEと云われるものでファイル内で同一種類のフィールドに与えられる名称である、即ち氏名を示すフィールドはすべて“NAME”で代表され、年令を示すフィールドは“AGE”で代表される。従つて求めたフィールド及びその時の条件はFIELD NAMEで示される。なお、検索は“RETRIEVE”で行なわれ、条件は“IF”以下で表されるが、この時の条件は“AND”、“OR”で結ばれ、その順序はカッコで示される。また条件を示すフィールドの内容は等号、不等号及び大小関係により次の6通りが用いられる。

EQ(=), LL(>), HH(<), HE($\leq$), NE($\neq$)

が有る。

(ロ) その他のファイルシステム

(i) DM1(Data Manager-1)

AUERBACHの開発したものであり、その特徴はデータファイルとプログラムファイルの両者の管理を行うことである。与えられた条件のもとでのデータの検索が特殊なインデックスとテーブルを用いて行われ、同時に一連の処理に於いてあるプログラムが終了したとき、このプログラムの出力をファイル内のワークエリアに格納し、次にこの出力データを処理する次のプログラムをロードしてこれにこの出力データを入力として与えて実行させる。即ち一連の処理に必要なプログラムを順次呼出し、またこれらの出力データを次の入力データとして与える機能がある。

また、その言語はASSEMBLER-COBOL, JOVIALで利用できる。

(iii) IDS(Integrated Data Store)

GEで200シリーズ、400シリーズ、600シリーズ用に開発されたものでCOBOLの拡張型をしたファイル言語である。

ファイル構造はCHAINにより成っている。即ちレコードはMASTER RECORDとDETAIL RECORDがあり、MASTERを親にしてロジカルなループを作る。なおあるCHAINのDETAILが他のCHAINのMASTERになることより、トリート状のファイルができる。また各レコードは任意個のCHAINのMASTER及びDETAILとなることができる。

このファイルでのアクセスはまずダイレクトアクセスによりMASTER RECORD

をアクセスし、次にCHAINを辿ることにより求めるデータを得る。なおこの場合ユーザはファイルの構造については完全に知らなければならない。

(iii) TDMS (Time Shared Data Management System)

SDC (System Development Corporation) で開発したもので、以前バッチ用に開発したLUSIDをTSS用に改良し、また360用に修正したシステム。

このTDMSの特徴はファイルがデータセクション及び二次インデックスセクションの二つに分けられ、内容による検索を行う場合は、この二次インデックスにより、次にデータセクション内で一次インデックスによりデータを求める。

(iv) IMRADS (Information Management Retrieval And Dissemination System)

UNIVACで9000シリーズ用のシステムでCOBOLで使用する。各データファイルはSelf-Contained形式である。即ち各ファイルの始めにそのファイルについてのコントロールデータを有する。その内容は

- (イ) ファイルが如何にセグメントされているか。
- (ロ) レコードに関するデータの定義
- (ハ) ファイル保持のlogic
- (ニ) レポートの定義

またIMRADSの特徴の一つとして各ファイルから必要な部分を取り出して新しいファイルを作成することができる。この場合、同一データは互にリンクされ、一方を更新すると自動的に他方も更新されるように作られている。

内容による検索は

- (イ) ダイレクトアクセス
- (ロ) ファイルの一部を走査する。
- (ハ) 二次インデックスの使用

(v) FFS (Formatted File System)

IBM 1410/7010米軍との共同開発されたもので次に示すGISの基礎となった。

各ファイルはFile Format Tableを有し、これによりファイルの保守、選択、抽出、レポート作成に必要なデータを得る。またこれは現在ファイルと共に使用することができる。

(vi) GIS (GENERALIZED INFORMATION SYSTEM)

IBMが360用に開発した汎用ファイル処理システムでBASIC GISとMULTI-TASKING AND TELEPROCESSING GISの二通りがある。

GISの機能はファイル創造、ファイル保守、検索、レポート作成、問合せへの

応答等である。又これは、FFSと同様現存するファイルといつしよに用いることが出来、新しい業務を急速にEDP化するのに用いるのによい。又、ファイル構成においてはLINK, INDEX が用いられる。

4 ファイルシステムの問題点及び今後の課題

前述の如く各種汎用ファイル処理システムが開発されている。しかし、ファイル設計の問題を総て解決するものでなく、まだ多くの問題を残している。即ちファイルの設計について確定的な方法が発見されず、数学的に解析する手段もない。シミュレーションの利用ということも考えられるが、現在のシミュレーション言語はこの方向に活用するのは不向きであり、又良いシミュレーションモデルが組めるほど十分な情報が得られない。

従つて、ここしばらくは試行錯誤の段階をくり返し、その過程においてよりよいものが開発されて行くことになろう。

本 PDF ファイルは 1968 年発行の「第 9 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>