

汎用ファイル処理システムの性格とその発展*

竹 下 亨**

1. はじめに

1967年の秋以来、日本では MIS に積極的に取り組んでいる向きが少なくない。しかしながら、真の意味での高度の MIS の実現は、必ずしも容易なことではないとされている。MIS を企業に導入するには、1) どのような情報をどのような形で、経営者や中間管理層に提供するか、2) そのような情報をどのような装置で、どのような形に貯え、どう処理するか等の2つの面をカバーする必要がある。1)が決まり、その体制が仮にできても、それを支えるソフトウェアがなければ実現は困難であるし、もちろん、その逆もしかりである。本稿では MIS に使われるソフトウェアと考えられる汎用ファイル処理システムについて、その歴史的展開、機能、特徴、見通しなどについてまとめてみたい。

情報システムの最近のものとして、1968年1月に新聞、ラジオ、テレビで報道されて大きな注目を浴びた四国電力の EMERS なる管理者用ファイル処理システムがあり、また同年4月に IBM が発表した汎用化情報システム (GIS)、情報管理システム (IMS)、客先情報制御システム (CICS) の3つがある。このほかに米国の2つのソフトウェア会社が開発中の DM-1 や TDMS などがある。これらは様々な形で受け取られ、解釈されていて、その区別も判然としてないようである。そこで、これらの性格、適用性や占める位置なども明確にすべきだと思われる。

ファイルやデータ・ベースなるものに情報を貯え、その更新・照会・処理するためのシステムは、その使用目的から

経営者用や中間管理者用情報システム

日常業務運用情報システム

の2つに大別できる。前者は各層のマネジメントとスタッフからの自然発生的、もしくは予期しない要求に

応ずるためのものである。後者は取引データなどを大量に扱い、日常業務——受注業務・販売管理・生産管理・在庫管理等——に使用されるシステムである。

ここでは、前者、すなわち汎用の情報システム——従来は汎用ファイル処理システムと呼ばれているもの——を主として取り扱う。

2. 汎用ファイル処理システムとは

データ処理のアプリケーションが多様化すれば、個々ばらばらに処理することは、多くの面で不都合となり、さらに MIS に近づこうとすれば、統合化された処理が必要となる。そのためには、どうしても1つのファイルが多目的に使われ、逆に多くのアプリケーションが共通のファイルをアクセスすることになる。そこで考えられるのは、こうしたファイルの処理に使える汎用のソフトウェアである。

この意味の汎用ファイル処理システム (general purpose file processing system) は、汎用情報管理システム (general purpose information management system) なる名称で置き換えることができるし、ファイル管理システム (file management system) やデータ管理システム (data management system) とも呼ばれている。また、最近ではデータ・ベースなる概念が導入され、データ・ベース管理システム (data base management system) という表現も使われるようになった。そして、ファイル処理のための言語を情報言語 (information language) やデータ言語 (data language) などと称している。

ここでいうファイル処理システムのおもな機能は何かというと、次の4つである。詳しくは後で説明する。

ファイルの生成 (file creation)

ファイルの更新 (file maintenance)

情報検索 (information retrieval)

レポート作成 (report generation)

汎用ファイル処理システムの適用範囲はその機能や性能により自然と変わってくる。あるものは、在庫管理や給与計算などにも向いているし、あるものは中間

* General Purpose File Processing Systems, Their Characteristics and Development by Toru Takeshita (IBM Japan, Ltd.)

** 日本アイビーエム株式会社

層以上の管理者用の情報システムとして利用できる。融通性の高い汎用システムであれば、経営事務計算の過半数はこれでカバーされるともいわれている。

さて、汎用ファイル・システムの設計概念は何であろうか。多くのものは辞引 (dictionary) の概念をその中心として採用している。これは人間の言葉に近い形で与えられた表現を辞引を見ながら翻訳して、内部の表現——機械語——になおして実行するやり方である。辞引を使用した初期のものの代表としては1959年ごろに SHARE のメンバーにより開発された709/7090 の 9PAC がある。このシステムでは、各ファイルの頭にその中のレコードの中味の記述を持つ。

データ・ファイルの辞引の内容は、ファイルの構造を示すもので、レコードやフィールドの名前、レコードの書式、データ・フィールドの性格などを含む。内容的には、COBOL のデータの部に類似しているが、COBOL の場合と違って、全部のデータが説明されている。それだけではなく、処理時点に与えられるパラメータと対応する計算・処理法まではいっているものもある。こうした辞引——登録簿 (directory) とも呼ばれる——は、それぞれのファイルの頭部についているか、すべてのファイルの分がひとまとめにされて補助記憶装置の1箇所にはいついて、ファイル名やファイル番号で取り出せるようになっている。こうした辞引の採用により、プログラマはレコードやフィールドの位置や書式に対応した細かいコーディングをその都度しなくともよくなり、プログラムの変更・修正も容易であり、かつファイルの中のレコードの位置や書式が変わっても、プログラムの変更を必要としなくなる。かくして、ファイル処理のプログラムの融通性や一般性を増すことになる。

次に、汎用処理システムの言語——システムに対するユーザの指示の書き方——であるが、

表形式 非表形式

の2つの形がある。

表形式 (tabular form) というのは、9PAC や RPG や MANAGE のように、定められた表形式の用紙に所要項目——ファイル名、レコード名、フィールド名、演算の種類、書式など——を記入するやり方である。後で説明する EMERS や MARK IV などは、要求された出力レポートの内容とほぼ1対1対応するような形の表に記入するので、素人でも使いやすい。許されるパラメータがすべて印刷してあって、その中から必要なものに印をつけるという選択方式もある。このよう

な方式は入出力装置が映像表示装置になった場合にきわめて有効である。たとえば、NHK の TOPICS ではこの方式を巧みに利用している。

非表形式 (non-tabular form) は、普通の言語の書き方と類似しており、単語をある文法に従って書き並べる形をとる。項目と項目との境界は空白やコンマなどの句読点で示す。この形式は多くのプログラミング言語に採用されているやり方であり、それを採用している汎用ファイル処理システムとしてはGISがその代表的なものである。オンラインのタイプライタ式の端末装置を入力とする場合には、PL/I のような自由形式の (free form) の書き方が望ましい。

さて、ファイル処理の内容にはどのようなものがあるか、以後の説明に重要であるので下に列記する。

- ファイルのラベル——ボリューム・ラベル——のチェック、すなわち、正しいテープやディスク・バックかどうかその見出しラベルのチェック
- データの読込み——カード、テープ、ディスクやドラム、通信回線からの入力
- データの変換——入出力データの表現が計算機コードと異なる場合の変換
- ファイルの生成——主としてカード、テープや通信線より入力
 - 固定長レコード、可変長レコード、長さを定義してないレコードがある。
 - 直接アクセス記憶装置では、順次 (sequential)、区画 (partitioned)、索引順次 (indexed sequential)、直接 (direct) の4種類のファイル編成がある。
- ファイルの更新——レコードの変更、追加、削除
- ファイルよりデータの抽出
- データの分類、グループ分け
- 見出しの印刷——または映像表示
- 加減乗除その他の計算
 - 小合計、合計、平均、標準偏差
 - 最適化、号則、評価など
- データの配列替え、編集 (editing)
- 数量項目の印刷——または映像表示
- グラフの印刷、図形の表示
- 新しいファイルやファイルの組合せの生成
- 監査用ファイルの出力——受入れられない入力レコードや主ファイルより削除されたレコードなどの書出し
- ドキュメントの索引作成

- 処理時間、頻度などの統計作成
- 制御表や辞引の内容の追加、修正、削除

3. データ・ベースについて

前節で汎用ファイル処理システムは、データ・ベース管理システムとも呼ばれることを述べたが、多目的に使われる高度化したファイルは、最近ではデータ・ベースと呼称するようになっており、さらにデータ・プールやデータ・バンクということもある。

従来のデータ・ファイルというのは、在庫管理ファイル、給与ファイル、製造計画ファイルなど個々のアプリケーション用に設計されている。各データ・ファイルはテープやディスクやドラムの上にそれ専用のスペースをとっている。そして、多くの場合、これらのデータ・ファイルに重複した情報はいっており、1つのファイルが更新されても、他は古いままで、相互に時間的ずれを生ずることがしばしばあった。

また、アプリケーションが異なれば、本来は同じデータが違った形で別のファイルにはいっていることも少なくない。アプリケーション・プログラムの入出力の部分や処理の一部は、特定のデータ構成や物理的装置の特性に合わせて、いちいちプログラムを書く必要がある。データの構成が変わったり、新しいデータ管理手法や装置が出ると、アプリケーション・プログラムを書き替えなければならないので、アプリケーション・プログラムをたえず変更せざるを得ない。

このような問題はデータ・ベースの出現により解消することになる。データ・ベース (data base) とは、「1つまたは複数のアプリケーションにより処理可能な相互関係のあるデータ項目の重複のない集合」と定義される。

その大きな特徴は、共通のデータの統合と共用を可能ならしめることである。それにより、重複したデータやその保守が不要になるし、企業内各部門による同一データの使用に互換性を持たすことができる。アプリケーション・プログラムは記憶装置や媒体の中でのデータの物理的な形態から、独立した形にしやすいので、データの構成や装置が変わっても、プログラムの書き替えは少なくなる。こうしたことから、アプリケーションのコストや記憶装置のコストと処理経費全体を減らすことができる。

4. ファイル処理プログラムの発展

ファイル処理のためのプログラミングおよびオペレ

ーション上の助けとなるソフトウェアとしては、大きく分けて下記のようなものがある。

- a) 汎用プログラム補助
- b) 特定アプリケーション用の処理プログラム
- c) データ・ベースやオンライン入出力の情報制御システム——日常業務運用の情報管理システム
- d) 汎用ファイル処理システム——管理者用汎用情報システム

これらはいずれもファイル処理のシステムといえるが、真の意味の汎用ファイル処理システムを掘り下げて論じる前に、それぞれの歴史的展開、特徴、利点と問題点などを整理することになろう。

まず、最初に汎用のプログラミング補助であるが、特にファイルに限定せずに一般にプログラミングの補助となるものである。その最も初期のものは汎用の入出力ルーチンである。1950年代の中期から後半にかけての真空管の計算機では、入出力を行なわせるために、主プログラムの中にこれらを組み込んで使うことが多かった。1950年代の終わりごろ——トランジスタの計算機の時代——になると、入出力と中央演算処理装置の操作を平行させ、いくつかのレコードを1つのブロックとして、一度に入出力することが考えられた。これらを制御する入出力制御システム (input-output control system, IOCS) が出現し、ユーザはマクロ命令を使って、自分のプログラムにそれを容易に組み込むことになる。固体素子集積回路を使った第3世代の計算機になると、これはデータ・セット制御 (data set control) ——実際の番地を指定せずとも、記号名でデータやプログラムの入出力を可能にし、テープやディスクのラベルをチェックし、直接アクセス装置に対してはスペースの自動割当てを行なうもの、ファイル制御ともいう——の機能を加えたデータ管理 (data management) に発展している。かくして、記号語 (symbolic language) ——アセンブラ語 (assembler language) ——でファイル処理のプログラムを書くのに、ユーザは入出力の部分にたいへんな苦勞をしなくてもよい。

他方、ファイルの生成、更新、レポートの作成が中心である経営事務計算のプログラミングを容易にするため、Commercial Translator, COBOL, RPG や PL/I などの高水準言語が開発されてきた。

こうした事務計算用の言語でファイル処理プログラムを書くことは、記号語によるプログラミングより、ずっと容易である。しかも、COBOL にはレポート・

ライター (Report Writer) やソートやすでにライブラリに入れてあるファイルの記述をそのまま組み込む COPY の機能が導入されているし、PL/I には、COBOL の COPY 以上の機能を持つコンパイル時の機能 (compile-time facilities) がある。

こうした高水準言語に飽き足らないユーザは、ファイル処理の手順をもっと簡単に表現できる言語を考え、それで書いたものを既存の高水準言語に直してからコンパイルするいわゆる **プリコンパイラ** (pre-compiler) を開発した。その一例は後述する BL/I で、これは判断表 (decision table) を入力として COBOL に翻訳する。

こうした高水準のプログラミング言語を使えば、機械語で書くより時間、労力、コストがはるかに少なくてすむし、機械内部や機械語について詳細を知らなくともよい。そのため、保守や変更もそれほど困難ではない。若干複雑な処理や例外の取扱いも可能である。しかし、それでもまだかなりの事務的作業が必要であり、高度の専門的知識や技術を要し、入出力関係——特に作表——のコーディングが面倒である。それに、何回かコンパイルしたり、テストをしなければならないので、プログラム完成までに時間がかかる。保守や修正についても同様である。記号語で書いたものに比べて、目的プログラム (object program) の効率がかなり低いことがある。

経営事務計算のアプリケーションの中には、どの企業でも行なわれるようなもの、たとえば、料金調定、給与計算、経理業務、在庫管理、生産管理、伝票業務などがある。これらについて、多数のユーザが共通に使えるようなプログラム——**特定アプリケーション用のファイル処理プログラム**——をメーカーが提供している場合がかなりある。たとえば、IBM の在庫管理用の BM プロセッサ (Bill of Material Processor), IMPACT (Inventory Management Program and Control Technique), 生産管理用の PICS (Production Information Control System) などは広く利用されている。

これらは、そのアプリケーションについては、入出力の取扱いやオペレーションがやさしく、しかも、通常は記号語で組まれているので性能がよい。上にあげた例はたまたまよく使われているプログラムである。しかし、聞くとところによると、これまでのこの種のアプリケーション・プログラムは必ずしもさほど受けはよくないようである。そのおもな理由は、書式とパラメータの制限がきつくと、ユーザはそれにファイル、入

力、出力の書式や構成を合わせる必要があり、他のファイルやプログラムとの互換性がなく、融通性に乏しい。しかも、例外処理や複雑な処理ができない。また、1つのアプリケーションしか使えず、他のプログラムはやはり別にする必要がある。他と統合して総合化することがむずかしく、全体のデータ処理の問題解決にならない。

これと逆の行き方をしているのが、オペレーティング・システムのデータ管理機能をさらに拡大したもの、すなわち、**データ・ベース**か**それとオンラインを含む情報制御システム**であり、日常業務運用の情報管理システム (operational information management system) に利用できる。このシステムでは、データ・ベースのサポート——辞引、生成、読み書き (access), 更新——やオンラインの入出力関係の諸種の制御とサービスが提供され、各ユーザはアプリケーション・プログラムの内部のデータの計算や加工の部分だけを書けばよいようになっている。IDS はデータ・ベースだけだが、IMS や CICS では、端末装置との入出力、メッセージの種類の解読、処理プログラムの読込みなどは制御ルーチンが行ない、ファイルの処理をバッチからオンラインに拡大している。

このシステムは、企業の末端で発生する細かいデータ (transaction data) を扱い、明確に定義され、あらかじめ設定された日常業務に関係した繰返し作業に適用される。

このようなシステムでは、アプリケーション・プログラム中の入出力を除いた部分だけをコーディングすればよく、その作成も容易であり、かつ性能も悪くない。データ・ベースを使っているので、かなりの標準化とデータ処理、コストの抑制が可能である。問題点としては、次のようなことがあげられる。処理プログラムはすべてテストが完了したものを機械語の形でオンラインの補助記憶装置に入れておく必要があり、プログラムの修正や保守にやはり多少負担がかかる。新しい種類のデータはすぐには受け付けられないし、急に新しい形のレポートを求められても直ちに出不せないので融通性は乏しい。定形化した日常業務には最適であるが、汎用性に難点がある。

上記の業務運用システムは取引中心 (transaction-oriented) で、このシステムで取り扱われるアプリケーションはあらかじめ計画されているものである。これに対して、いつどんな要求がきても受け入れられるようにしたのが、真の意味で**汎用の処理プログラム**であ

る。

このようなものの始まりはユーティリティ・プログラムであり、制御カードにより、カードからテープ、テープから印刷、テープからカードなどの操作を直ちに実行する。装置を変えたり、テープ・レコードのブロッキングやテープやディスクのファイルのラベルの変更も可能である。ずっとレベルの高いものとしてソート/マージのプログラムがある。これも制御カードにより制御フィールドや分類順序などを指定すれば、すぐに処理操作が行なえる（もっとも、これらの情報を入れて機械語のプログラムを生成してから、実際の仕事を行なうジェネレータ方式のものもある）。こうしたものを飛躍的に高度化させたものが、汎用ファイル処理プログラムと考えてもよいと思う。この種とみなされるプログラムはその設計思想の相違により、直接実行方式、編成翻訳方式、解釈方式などに分かれ、しかも、その機能や性能は大きく相違する。

こうした汎用処理プログラムは、当然ながらある程度の融通性を備えている。多数のプログラムの役割を1つのプログラムが果たすとなると、プログラム作成の時間と機械使用時間が短縮し、プログラム・ライブラリのスペースが節減できるばかりか、多くは専門的熟練を不要にし、素人の受けもよくし、プログラムの保守の手間も省く。また、ファイルの処理手順の標準化が可能であるし、処理時間の計算も前もって容易にできる。直接実行方式のものは処理効率もよい。他方、問題点としては、あまり複雑なものや例外の処理がむずかしいし、当然ながら芸の細かい融通性は期待できない。そして、専用に作られたアプリケーション・プログラムと比較すると、記憶装置のスペースや処理時間をより多く必要とする。

第1表 ファイル処理プログラムの種類

汎用プログラム補助	入出力ルーチン、IOCS、データ管理、高水準言語、プリコンパイラ
特定アプリケーションのプログラム	料金調定、給与計算、在庫管理 (BM Processor, IMPACT)、生産管理 (PICS) など
データ・ベースやオンライン入出力のサポート	CISS, IDS, DM-1, Multi-List IMS/310, CICS/360 など
汎用のファイル処理プログラム	ユーティリティ、ソート/マージ、9 PAC, EMERS, MARK IV, GIS, TDMS, DM-1 など

4. 汎用ファイル処理システムによる処理

汎用ファイル処理システムは様々な内部構成を持っているが、プログラムの読み込みと初期状態の設定 (initialization) がすんだ後、大よそ次のような3つの段

階を経て要求された処理を行なう。第1段階は、パラメータの読み込みである。この段階では、処理の仕方を示すのに必要なパラメータ——ファイルの名前、レコードやフィールドの名前、計算のやり方、見出しや出力の書式など——が読込まれ、内部表現に直されて所定の場所に記憶される。その際に誤りが検出される。

第2段階は、前の段階で読んだパラメータから目的プログラムを準備する作業の全部または一部を行なう。純粋に機械語の命令を作り出すのは 9 PAC, BEST, MANAGE, IDS, ASI-ST, MARK IV, RGP などがある。EMERS のように機械語 (core image) になっているプログラムを読み込むだけのものもある。ここで、いきなり機械語のプログラムを出さずに、他の言語翻訳プログラム——コンパイラ——の原始プログラム (source program) を出すものもある。このようなものに BL/I や CISS がある。もう1つのやり方は、解釈方式的に実行するのに必要な一連のパラメータのようなものを次の段階に渡すようにしたものである。

最後はファイル処理を実行する段階である。機械語になっているプログラムが実行される場合と、前の段階から渡されたパラメータや表があって解釈方式に実行する場合がある。ここで、指定されたファイルからデータを取り出して分類、グループ分けし、加減乗除を行なって、要求されたレポートを編集・出力する。

以上の3つの段階は、システムによっては必ずしも明白に区切られてなく、第1段階と第2段階がいっしょに行なわれることがあるし、第2段階のあとに機械語に編集翻訳するステップがはいることがある。第3段階は関連するレコードの抽出・分類・計算・編集・印刷などのいくつかのジョブ・ステップから構成されるものもある。

なお、ファイル処理システムがいくつかの処理モジュールと表 (table) で構成されているときは、上記の3つのステップとは別に、必要となった新しい機能の挿入、制御表の更新や、システムの中に含まれるファイル仕様——ファイルの辞引、データの書式表——の追加・削除・変更を行なうことがある。

5. 汎用ファイル処理システムの例

前節で汎用ファイル処理システムの内部処理の仕方について一とおり解説したが、以下は4つのグループに分けた代表的な実例である。その中には辞引やデータ・ベース的な考え方を使っていないものもあるが、ファイル処理を主目的としているので同類としてあげ

ておく。

コンパイルしてから実行するシステムの例である 9 PAC は、データ・ファイルの生成と保守およびレポート作成に使うもので、ファイル・プロセッサとレポート・ジェネレータの2つがあり、いずれも表形式の用紙に記入された原始プログラムから機械語のプログラムを生成する。これらのプロセッサは各ファイルの頭につけられるファイルの辞引を参照する。プログラムはファイル内の情報を——絶対的な位置や長さの代わりに——レコード・タイプとフィールド番号で指定できる。ファイル・プロセッサは辞引または変更のパラメータのカードを入力としてファイルの生成か保守を行なうための目的プログラムを生成する。レポート・ジェネレータはレポート作成の方法を指定するパラメータ・カードを読み込んで、レポートを出すための機械語のプログラムを生成する。入力ファイルは1つに限るが、条件により異なる取扱いや、出力の書式の指定が可能で、しかも、算術演算を行なう場合は、その部分は、SCAT という記号語で書いたものを受け付ける。

9 PACとは全く別の手法の例には、GEが開発したIDS (Integrated Data Store) がある。これは大容量記憶装置(mass storage)におかれたファイルの構成と取扱いを行なうソフトウェアで、ファイルは連鎖(chain)方式を採用している。その言語はCOBOLにデータの定義とファイルの処理機能を加えるものである。

人間が書きやすい形で書いた記述——記入表を使う場合や文章のような形式——を、一度コンパイルして実行させる他の例としては、第2表のものがある。

第2表 コンパイルしてから実行させるファイル処理システムの例

名称と制作者	機 種	発表時期	完成時期
BEST (Business EDP Systems Technique), NCR	NCR 304 NCR 315	— 1964年	1962年 1963年
NFOL (Information Oriented Language), CDC	CDC 3600	1965年	1966年
MANAGE, SDS	SDS 9300 SDS 900シリーズ	1965年	1966年
ASI-ST, Applications Software, Inc.	IBM システム/360 SDS シグマ 5/7 RCA スペクトラ70		
		1967年	不明

日本でもこうしたシステムはユーザによって作られている。三井生命の藤田氏のところで IBM 7070 用に作られた RGP (Report Generation Program) ——1966年発表、同年完成——は、COBOL に似た言語で与えられた抽出条件および集計、編集の指示から報告書作成のプログラムを生成し、目的プログラムを主記

憶装置において、そのまま実行させる。

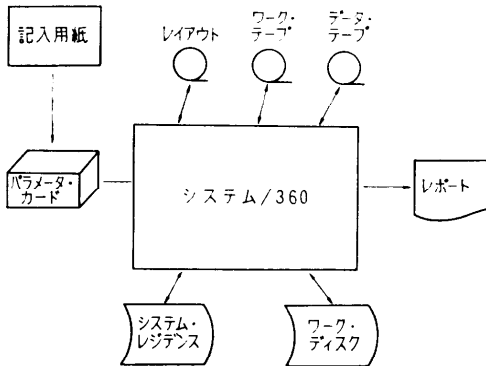
既成のコンパイラはそのまま利用することにし、その原始プログラムを作り出すプリコンパイラがいくつかのユーザにより試みられている。

旭化成の梅田氏や古川氏が開発された BL/I (Business Language One) ——1967年完成、同年発表——は、個々のプログラム特有の条件を単純化した形式で表現し、基本論理——処理に必要な一連のルーチン——はプリプロセッサによって組み込まれる。BL/I プリコンパイラは OS/360 の PL/I で書かれており、いくつかのテーブルをCOBOLのステートメントに変換し、基本論理と組み合わせて COBOL プログラムを作り出す。BL/I プリプロセッサの入力となる原始プログラムには、コントロール項目、コントロール・レベル、内部カウンタ、出力レコードの形体 (image)、作業区域、索表のキー (key)、細かいプロセスなどが記述される。細かい処理は判断表 (decision table) の形で与えられるのが特徴である。

以上とやや趣を異にする CISS (Consolidated Information Storage System) なるシステムを日本電気の永野氏が1968年1月の箱根プログラミング・シンポジウムで発表されている。これは次のような構成になっている。ファイルはファイル・コンストラクタにより生成され、CISS 言語で書かれたものを COBOL に変換するトランスレータがあり、ジョブ・プログラムとオペレーティング・システムの IOFCS (Input Output File Control System) との仲介は CISS モニタが行なう。それにファイル・リオーガナイザ、ファイル・マップ、ロード/アンロードのユーティリティがある。

コンパイルしてから実行するのではどうしても時間がかかるので、機械語になったプログラムを直接実行するやり方も採用されている。その代表的なものは、EMERS と MARK IV でないかと思われる。これは、内部処理のやり方がパラメータにより変えられる目的プログラムであって、ほとんどの場合、数多くの機能モジュール——計算や編集を行なうサブルーチン——を結び付け (link) たり、番地部などが未定になっているいくつかの命令にパラメータから読み取ってはめ込む方式をとっている。こうするとパラメータを全部読み取るとすぐに処理が実行されるので最も効率的である。

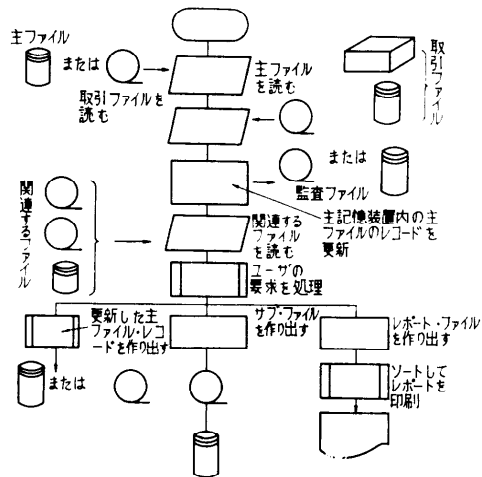
EMERS (Easy Management and Executive Reporting System) ——1967年開発——は、四国電力の奥田氏ほかが開発したもので、日本



第1図 EMERS のシステム流れ図

的できわめてすぐれたシステムである。通常の報告書の様式をほぼ同じ表形式の記入用紙を使用して、ファイル名、報告書に必要な項目および計算内容を指定し、これをパラメータ・カードにせん孔して計算機に入れば、コンパイルやアセンブルなしに、ただちに所要の報告書を出すことができる。したがって、誰でも簡単に使えるし、時間がかからないので、あらゆる階層からの情報の要求に即刻応ずることができる。特に、合計や平均だけでなく、標準偏差などの統計的計算や棒グラフの印刷などを選択することができるので、管理者が見やすい検討・判断資料を提供する。このシステムはCOBOLで書かれており、カードやテープのファイルを取り扱う。ディスク・システムに切り換えることもさほど困難ではないであろう。

同様の思想により設計されたシステムはアメリカでも開発されている。たとえば、インフォーマティクス社 (Informatics Inc.) が、GIRLS (Generalized Information Retrieval and Listing System) —IBM 7090用に作られたもの—や MARK I, MARK II, MARK III —いずれも IBM 1400 シリーズ用—を拡張して MARK IV を1965年より開発した。これはシステム/360の OS および DOS をベースとするファイル管理用システムである。有料で客先に提供されており、SE サービスがある。これも EMERS と同じように表形式の用紙に、ユーザは所定項目を記入すればよい。このシステムでは直接アクセス記憶装置 (direct access storage device, DASD) —ディスクやドラム—も扱い、ファイルの生成、更新、他の関連するファイルの同時処理が可能であり、レコードやその中の抽出、分類、グループ分け、小合計や合計、見出しと数量の印刷などが可能である。そして、ユーザが指定

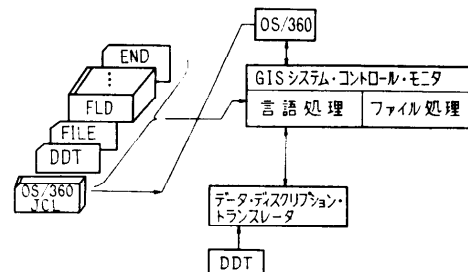


第2図 MARK IV のシステム流れ図

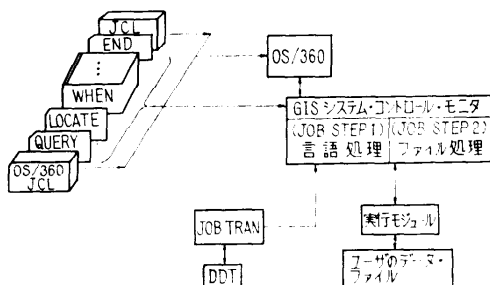
する処理内容を DASD に登録しておけば、同じものをもう一度入れなくとも、必要な都度取り出して所定のレポートを作成させることができる。

さきに INFOL や ASI-ST はコンパイルされてから実行されるシステムとして紹介したが、これらも実行時にはパラメータを解釈しながら処理を進める部分も存在するようである。ASI-ST は特にオンラインで使う場合にしかりである。こうした解釈方式で実行する部分が多いものに GIS や IMRADS をあげることができると思われる。この方式ではユーザの要求に対して、通常のコンパイルをせずに処理ができるので、融通性も高い適用性が広いといえる。

IBM がシステム/360用に開発している GIS (Generalized Information System) —1965年に発表、完成予定は基本的なものは1969年6月、テレプロセシングとマルチタスクを含むものは、その後の予定—は、特定のアプリケーションにも適用できるが、主目的は中間層およびトップの管理者を対象とする情報システム



第3図 GISのタスク指定：データの記述を入れた場合



第4図 SISのタスク指定：手続の記述 (QUERY)を入れた場合

ムで、利用度が高い包括的な汎用システムである。ファイルの生成、更新、情報検索、レポート作成、照合処理などを行なう。なお、このシステムは米空軍と、IBM が共同で 1410/7010用に開発した FFS (Formatted File System) に基礎をおくといわれている。

なお、公式製品ではないが IBM が開発した **MIS/360**——初版は1967年6月完成、第2版は翌年6月完成——があり、IBM 社内およびいくつかの客先により使用されている。これは構成されたデータ・ファイル (structured data file) のデータ検索を端末装置から行なう情報システムである。

IMRADS (Information Management Retrieval and Dissemination System)——1966年発表——はユニパックがその1108用に COBOL を使って開発しているといわれているシステムで、各データ・ファイルの頭に制御データを持ち、EXECUTIVE VM なるオペレーティング・システムの支配下でバッチおよびオンライン処理を行なうとされている。

また、オンライン対話方式のサポートも含む **DM-1** (DATA Manager-1) がアウアバック社 (Auerbach Corp) によりシステム/360 とユニパック M 1218 用に開発されている。

最後にタイム・シェアリングのシステムであるが、**TDMS** (Time-Shared Data Management System)——1966年発表——がシステム・ディベロプメント社 (System Development Corporation) によってシステム/360用に開発されている。これらは、大容量記憶装置、高速通信回線を使った対話方式の融通性の高いシステムといわれている(注)。

6. 汎用ファイル処理システムに望ましい機能

数多くのファイル処理システムを調べ、まだ理想的とされるシステムを考えると、汎用ファイル処理システムに望ましい機能として、次のようなものをあげることができる。ただし、多分に主観的、かつ思いつき程度のもので、確たるものでは決していない。

まず、いずれの場合も**基本的機能**としては、システムにユーザが与える指示——すなわち、制御情報——の解釈が絶対必要な機能であり、その際に間違ったパラメータなど人間の誤りをできる限りチェックする検査機能がなくてはならない。そうして各ファイルについてその仕様や辞引を内部に持つシステムでは、新しい仕様や辞引の読み込みと、その補助記憶装置に貯蔵と、さらに更新の機能が要求される。当然ながら、ファイル処理システムにはファイルの生成と修正、更新——追加・変更・削除——が基本的な必須機能である。そうしてできたファイルを使ってデータを検索したり、照会することになる。レポートを作成するには、必要なデータを抽出し、分類し、項目の計算——合計、平均、標準偏差——を行ない、出力データ項目を編集 (edit) して印刷する。数値のみでなく、少なくとも項目の見出しぐらいは印刷できなければならない。

使いやすくなるための機能としては、次のようなものをあげることができる。システムに与える指示は、簡単な記入用紙に素人でも容易に書けるようにし、しかも、通例的なもの——多くの場合にそうであるもの——はいちいち記入しなくともよいようにする。それに何回も行なわせるファイル更新やレポート作成については、毎回そのための指示をいちいち入れなくともすむようにシステムの中に——ディスクやドラムの中に——登録しておいて、たとえば、1枚の制御カードで呼び出せるようにする。既存のファイルが大きくなったり、新しいファイルが加わったり、逆に不要になったファイルを取り除くことがおきるが、このための補助記憶装置のスペースの割当てや整理も欲しい機能である。

以上の能力を備えれば、一応使えるシステムになるが、システムの**融通性を高める機能**としては、次のようなものが考えられる。ファイルはカードや磁気テープのみでなく、ディスクやドラムやデータ・セルに入れることができ、しかも、データ構成には順次編成のほか、直接編成や索引順次編成のいずれも選択可能

(注) 本節に出ている数例の発表と完成時期は、参考文献 5) より引用させていただいた。

とする。しかもユーザは装置の番地でなく、記号名によるファイルの指定ができて、いわゆる“装置と独立”に処理の指示が与えられることにする。これらのファイルはユーザが別途作成するプログラムでも容易にアクセスできるようにしたい。それに一部は別の媒体——たとえば、一部は磁気テープ、一部はカード——にはいつているファイルを読んで、1つのファイルに連結 (concatenate) することもある。データの内部や外部表現は自由に選択・変更が可能で、レポート上のデータの長さ、位置、項目数、書式が任意に変えられるとよい。また、論理的条件により処理のやり方が決まるような場合、手掛り語 (key word) を使うなどして表現しやすくする。次に、システムの内部構成であるが、いわゆる積木方式 (modular system) の構成にして、要求された機能に応じてルーチンが選択されるようにし、かつ、これらルーチンの変更・追加を容易にしておく。そして、システムにはいついていない特別の機能——例外処理など——を行なわせたいときには、ユーザが任意の言語で組んだルーチンを組み入れられることにする。

さらに**高度の機能**となると、これまたいろいろ考えられる。1つはチェックポイント/リスタートなどの事故の際の復元・再開の手順で、最初からやり直さなくても、少し前にもどって再び始められるようにすることはきわめて重要である。それに、いわゆるテスト・モード (testing mode) を許して、主ファイルの中味を変えずにテストすることや、ファイルの一部だけを使つてのテストを可能にする能力や、エラーの検出を容易にする追跡ルーチンが欲しい。また、会計ルーチン (accounting routine) があって、使用頻度と各種の時間やその他の統計的データが自動的に得られるとよい。ファイルは階級構造をしていて、高いレベルのデータから低いレベルにたどれることも望ましい。主記憶装置内の同一の区画で複数の処理の要求を平行的に同時処理——いわゆるマルチプル・ユーティリティ的なオペレーション——をすることも考えられるが、複数のオンライン端末装置から指示情報を受け付けるとすると、この方式がぜひ必要となる。この場合、共用ファイルの保護——所要の位置に移動させたアクセス機構が、データ転送操作の終了前に他のレポート処理の入出力のため、別の位置に移動することがないようにするなどの手段——や、優先処理の問題が生じてくる。人事や財務の情報など特定な人しかアクセスできないファイルがあるときは、合言葉 (pass word) な

どによる機密保護の手段を講ずべきである。オンライン端末装置が使えて、対話方式の操作ができるとユーザの能率は大幅に上がる。さらに、端末として映像表示装置が使えれば、人間と計算機の対話はよりやさしく、効率的になる。出力も文字のみでなくグラフや図形が出ると一層効果的である。このほか、エラーを生じたときに、その状態の自動記録とユーザが任意に指定した処置を行なわせることも考えるべきである。

7. 汎用ファイル処理システムの利点と見通し

前節で説明したような機能を備えた汎用ファイル処理システムは、次のような**利点**を持つといえよう。

かなり広い適応範囲において、プログラミングについての知識・経験が不要であり、素人でも容易に使え、プログラミングに関する労力、時間、コストが減り、プログラミングの人員は少なくてすむ。そればかりかファイルの構造や取扱いが標準化される。しかも、ファイルのためのスペースが少なくてすむし、プログラムのコンパイルやテストの時間がいらなくなるので、機械使用時間の減少によるコストの低下をももたらす。専門のプログラムはもっと高度のアプリケーションに時間と労力を傾注できる。経営者・管理者は必要とする情報を短時間で入手し、現状分析や意志決定に役立てる。かくて、企業内の生産性や経営の質の向上に大いに貢献する。すなわち、MISを支えるソフトウェアとしての役割を果たすことになる。

こうした汎用ファイル処理システムでは、次のことが**問題**である。第1は性能である。どうしてもこの種システムは大きな主記憶装置を必要とし、処理速度もアプリケーション別の専用のプログラムに比べて、かなり差があることはやむを得ない。このほか、小回りがきかないということ、例外処理や特殊処理を受け付けられないことと、機械が持っている特別な機能が十分に使えないことなどがある。

この辺で、これまで汎用ファイル処理システムがどう展開してきたかをまとめ、今後の見通しを考えてみることにする。1957年には、すでに IBM 702 を使用した汎用ファイル処理ルーチンができていたが、1959年ごろにはファイルの辞引を使う 9 PAC が開発された。以後種々なアプローチがなされてきたが、その主流は辞引を使う手法の拡大であり、データ・ベースの概念の誕生と、その支えとなるプログラムの開発である。第3世代の計算機のオペレーティング・システムには、データ・セット制御の機能が導入され、ユーザ

のプログラムの中でファイルをその名前ですせるようになり、**装置より独立** (device independence) にプログラムの作成が可能になった。大きなもので可能になったものが、小さなものでも実現することが少なくない。ファイルより小さい単位は、その中に含まれるレコードやセグメントやフィールドであるが、絶対または相対位置と実際の長さに従って、細かい処理ステップをいちいち昔く従来の方式を脱却して、データ項目もその記号名だけで指定できるシステムがでてきた。これで、“**データ形式より独立** (data format independence)”——ファイルの構造より独立——が可能となった。こうした傾向は益々強くなっていくであろう。

オンライン・システムの急速な発展により、ファイル処理システムもバッチにとどまらず、オンライン化が当然進められ、トップや中間層のマネジメントやそのスタッフが端末装置を自分の席の近くにおいて、より身近に必要な経営情報を得るやり方も実際に使用されつつある。そして、ここ二、三年急にクローズ・アップしてきた映像表示装置が、汎用ファイル処理システムの端末装置として使用されれば、表示された項目を選択したり、必要なパラメータのみを選択し、対話方式で情報の要求ができるなどマネジメントがごく気軽に使えるようになる。

そして、同一企業内の MIS だけでなく、通信線で結んだ大形計算機の不特定多数のユーザによる利用——こうなるとデータ・バンクの共用ということになるか——にも、この手法は応用されるのではないか。こうなれば、どのユーザにも応答時間にできるだけ公平を期し各ユーザの効率を上げるために、タイム・シェアリング方式が当然であろう。

なお、こうしたファイル中心の汎用情報システム (file-oriented information system) の発展により、RPG, COBOL, PL/I などのプログラミング言語の重要性はほとんど低下しないであろうし、また、データベースに関するサービスやオンライン入出力情報制御のみを提供する日常業務運用情報管理システムも効率のよいシステムとして使われであろう。しかし、丁度科学技術計算と経営事務計算の両者のプログラミングの要求に応ずる PL/I 言語が出現したように、経営者・管理者用情報システムと日常業務運用情報管理システムが統合化された 1 本の汎用情報システムもいずれは作られるのではなからうか。

(昭和43年9月27日受付, 昭和43年12月10日再受付)

参考文献

1) J. Becker and R.M. Hays: Information Stor-

- age and Retrieval: Tools, Elements, Theories, John Wiley & Sons, New York, 1963.
- 2) P.W. Howerton, General Editor: Information Handling: First Principles, Spartan Books, Washington, 1963.
- 3) C.T. Meadow: The Analysis of Information Systems, John Wiley & Sons, 1967.
- 4) R.V. Head: Management Information Systems: A Critical Appraisal, Datamation, May, 1967.
- 5) D.H. Sundeen: General Purpose Software, Datamation, January, 1968.
- 6) Data management: Generalized File processing Software, EDP Analyzer October, 1965.
- 7) Data management: File Organization, EDP Analyzer, December 1967
- 8) Data management: Functions, EDP Analyzer, January 1968.
- 9) P. J. Dixon and J. Sable: "DM-1—a generalized data management system", Proceeding of the 1967 Spring Joint Computer Conference, Thompson Book Co., Washington D.C.
- 10) Session on data management systems, Proceeding of the 1967 Spring Joint Computer Conference
- 11) Proceeding of Second Symposium on Computer-Centered Data Base Systems held at SDC, 1965, Clearinghouse for Federal Scientific and Technical Information, Virginia.
- 12) J. F. Nolan and A. W. Armenti: "An Experimental On-Line Data Storage and Retrieval System," MIT, 1965, Clearinghouse for Federal Scientific and Technical Information, Virginia.
- 13) A. H. Vorhaus: A New Approach to Data Management, Systems and Procedures Journal, July-August 1967.
- 14) J. Spiegel and J. K. Summers: "AESOP" 上と同じ定期刊行物.
- 15) R. P. Larsen: "Data Filtering Applied to Information Storage and Retrieval Applications," Comm of the ACM, Nov. 1966.
- 16) W. M. Carlson: "A Management Information System Designed by Managers," Datamation, May, 1967.
- 17) 水野幸男: 汎用ファイル処理システムについて, 情報処理学会第9回プログラミング・シンポジウム報告集, 1968年1月.
- 18) J.A. Postley: The Mark IV System, Datamation, January, 1968.
- 19) J.R. Ziegler: Program Generator, Computer and Automation, July, 1968.
- 20) System/360 Generalized Information System Application Description Manual, IBM, 1968.
- 21) Information Management System/360 Application Description Manual, IBM, 1968.