

資 料

汎用ファイル処理システム SIMPS の特徴と機能*

森 田 章**

Abstract

SIMPS (Simple and mnemonic file processing system) has been developed by Kyoei Information Processing Service Center Co., Ltd.

This system run under FACOM 230-25/35 BOS and require a minimum of 8K bytes of memory. SIMPS consists of a interpreter and nine separate generators. The generation is done in main frame and the run is done in any frame. In this system we adapted simplified tabular form for use by non-computer personnel. In this paper, I describe the design and features of this system. Finally, discussion on advantageous features of SIMPS are given in comparison with the program prepared on the language of COBOL.

1. はじめに

単純なバッチ処理の基本的な内容はファイルの生成、更新、データの分類、組み合わせ、編集、抽出、計算、および作表である。ここでいうファイルの生成には、データの変換、つまりコード変換、データの記録媒体の変換、それに記録内容の矛盾を検出するデータ・チェックが含まれる。アプリケーションによって、組み合わせ方と実行順序が異なっているに過ぎないことが実証されるならば、上記の基本的な処理内容を遂行するプログラムを用意することによって、単純なバッチ処理のほとんどがカバーされるはずである。アプリケーション専用のプログラムを必要の都度作成することを不要にするためには、汎用性が重視されなければならないことは当然である。そのためには変数と関数の関係を明確にすればよい。ここでいう変数とはアプリケーションごとに異なる要素のことであり、レポート・プログラム・ジェネレータについて考えてみても、入出力書式や演算内容はアプリケーションごとに異なっている。これらは固定できない要素である。ここでいう関数とは変換のルールであり、与えられた入力（パラメータおよびデータ）にどのような働き

きかけをして出力するかということ、いいかえれば変換の機能である。一例として、SORT/MERGE のプログラムをとりあげてみると、キー・フィールドや分類順序をパラメータとして与えることによって、オブジェクト・プログラムを生成した後、所望の分類操作を行なう方式のものがある。このことを分類・併合プログラムだけにとどめず、バッチ処理に共通な機能すべてについて、このようなプログラムを作成すれば、汎用ファイル処理プログラムができあがることになる。そこで機能とパラメータ決定に当って、協栄計算センター所有の事務計算用プログラム 2,500 個（調査対象は COBOL で作成されている全アプリケーション・プログラムで、調査実施は 1969 年 10 月である）を入力から出力への変換内容別に分類整理することにした。その結果 32 種のパターンに集約することができ、さらにこれを単一な機能をもったプログラムに分解した結果 9 種のものの組み合わせによって 80% カバーできることが確認できた。ただし、この 9 種の中にはソート・マージは含めてない。理由は、既存のものを利用することにし、さじ当って、開発対象にしなかったからである。どのように 32 種のパターン分解したか、その一例をあげれば、ファイルの振り分けは条件 A を満足するレコードと条件 A を満足しないレコードをそれぞれ別のファイルに出力することであるが、そのためには抽出プログラムが用意されていればよ

* Features of SIMPS (Simple and mnemonic file Processing system) by Akira Morita (Kyoei Information processing Service center Co., Ltd.)

** (株)協栄計算センター

い。まず条件Aを満足するレコードを抽出し、その次に条件Aを満足しないレコードを抽出すれば、振り分けたと同じ結果が得られる。

振り分けのプログラムはジョブ・ステップが1であった。それに対して抽出プログラムの利用によって同一結果を得ようとするとき、ジョブ・ステップが、上記の例であると、2になる。多重処理用機械にマッチしたオペレーションを考慮すれば、このことはコンピュータの使用効率を必ずしも低下させることにつながらないが、たとえ、オペレーション効率が COBOL を使用する場合と比較して、10% 程度低下したとしても開発の価値があるという結論であった。そこで、最高3ジョブまで多重処理可能なFACOM 230-25/35 BOS (Batch Operating System)用に汎用ファイル処理システム・SIMPS (Simple and mnemonic file Processing System) の開発を 1969 年 11 月に着手した。1971 年 1 月から一部ずつ実用化して、1971 年 10 月に初期の目標であった磁気テープ・ベースのものを完成し、現在、ディスク・ベースの SIMPS 開発を続行している。本稿では SIMPS の特徴、機能、プログラム構造さらに、COBOL との比較例について述べる。

2. SIMPS の構成と機能概要

SIMPS では辞引きの働きをするファイル・レイアウト登録簿を磁気テープまたはディスクでもつようにしている。日常用語に近い形で与えられたパラメータとこの辞引きから相対形式のオブジェクト・モジュールを選択編成して目的の機械語のプログラムを得る(ジェネレータ方式)ように設計してある。したがって、SIMPS はこのファイル・レイアウト・ディレクトリリー・プログラムと 9 個のプログラム・ジェネレタによって構成されている。9 個のジェネレタはいずれもパラメータを読みこんで目的プログラムを生成するとき、必要なファイルの登録内容を自動的に索引してきて、その詳細内容を参照する。

SIMPS を構成しているプログラムの名称と機能概要是次の通りである。

2.1 レイアウト登録プログラム (LDP-LAYOUT DIRECTORY PROGRAM)

ファイルのレイアウトを磁気テープまたはディスクに登録、変更削除、追加、登録内容の印刷、それに登録媒体の変換(磁気テープ—ディスク)を行なう。

登録内容はファイル名、ブロッキング・ファクタ、レコード長、フィールド名、フィールド位置、フィー

処 理

ルド長、記録モード、小数点の桁位置、編集形式、それにグループ・フィールド表示である。このプログラムだけは解釈方式であり、目的プログラムを生成するジェネレータ方式ではない。

2.2 變換プログラム・ジェネレタ (CON-CONVERSION PROGRAM GENERATOR)

紙テープまたはカードに穿孔されたデータを入力書式そのままのかたちで磁気テープ(ただし、ディスク・ベースの SIMPS についてはディスク)に記録するプログラムを生成する。紙テープ・データは固定長、可変長とも処理可能であり、マルチ・カードも処理可能である。入力コード系の指定、チェック・ポイントの指定もできる。ファイル更新用トランザクション・ファイル作成用プログラムもこの CON によって生成する。

2.3 データ・チェック・プログラム・ジェネレタ (GCP-GENERALIZED CHECK PROGRAM GENERATOR)

CON で生成したプログラムによって作成した記録内容の矛盾を検出するプログラムをこの GCP が生成する。この GCP によって、桁ズレ、記録モード、年号、年月日、データコードの範囲、定数、符号、重複レコード、同一レコード内のフィールド間の関連をチェックするプログラムを生成する。チェック・アウトされたレコードをどうするかは指定による。つまり、別ファイルに出力するか、完全に無視するか、あるいはエラー内容の内訳別集計レポート作成用ファイルを出力させるか、いずれも指定できる。

2.4 編集プログラム・ジェネレタ (EDT-EDITING PROGRAM GENERATOR)

フィールドの配列がえ、記録モードの変換、レコードの集約、リテラル挿入、コード翻訳を行なわせるプログラムを生成する。

2.5 組み合わせ・照合プログラム・ジェネレタ (MCP-MERGING AND CHECK PROGRAM GENERATOR)

入力ファイル 2 個を組み合わせて 1 個のファイルを出力するプログラムを生成する。あるいは、入力ファイル 2 個間の対応レコードおよび内容について矛盾を検出するプログラムを生成する。チェック・アウト・レコードについては GCP 同様指定によってどのような取り扱い方をするか決定される。

2.6 演算プログラム・ジェネレタ (CPG-COMPUTATION PROGRAM GENERATOR)

同一レコード内での四則演算、複合演算、算術平均、

百分率計算を行なうプログラムを生成する。演算について四捨五入、切捨てなどの指定も可能である。

2.7 更新チェック・プログラム・ジェネレータ (TCP-TRANSACTION CHECK PROGRAM GENERATOR)

既述の GCP は入力ファイル 1 個に対し、1 レコード内のフィールド内容についてのエラーを検出するプログラムを生成する機能をもったジェネレータである。これに対して、TCP は更新しようとするマスター・ファイルの内容とトランザクション・ファイルの対応レコードを照合して、矛盾する内容があればチェック・アウトするプログラムを生成する。

2.8 更新プログラム・ジェネレータ (MPG-MAIN-TENANCE PROGRAM GENERATOR)

トランザクション・ファイル内容にもとづいて、マスター・ファイルの追加、削除、変更を行なうプログラムを生成する。更新内容を確認するためのレポートを必要とするときはレポート作成用ファイルの出力指定が可能である。

2.9 抽出プログラム・ジェネレータ (SLT-SELECTION PROGRAM GENERATOR)

マスター・ファイルから指定条件を満足するレコードを抽出するプログラムを生成する。最高 30 個までの複合条件を指定することができる。

2.10 レポート印刷・プログラム・ジェネレータ (ROP-REPORT OUTPUT PROGRAM GENERATOR)

見出し印刷 3 行まで、縦の集計は 6 レベルまで、1 レコード 6 段印刷まで選択でき、明細印刷、集計印刷、明細集計同時印刷いずれも可能なプログラムを生成する。リ・スタートの機能、一部処理の機能ももっている。ジェネレーション終了後、イメージ・プリントを出力させることも可能であるし、実処理を開始する前にテスト・プリントを出力させることも可能である。

3. セットアップ・プログラムの構造

SIMPS の各ジェネレータの基本構造は共通であり、セットアップ・プログラムとオブジェクト・プログラムの 2 つからできている。それぞれの機能、および構成について述べる。

3.1 セット・アッププログラムの機能

セットアップ・プログラムは、パラメータ記入用紙に記入され、カードまたは紙テープに穿孔された制御文 (パラメータ文) を読み、その内容をチェックし、

内容に誤りがなければ、オブジェクト・プログラムで使用するテーブルを主記憶装置内につくり出す。制御文すべてをテーブル化した後、テーブル間に矛盾がないかをチェックし、誤りがなければ、テーブルを判別して、必要なオブジェクト・ルーチンを組み合わせ、1 個のオブジェクト・プログラムを生成する。もし、制御文中に誤りがあれば、タイプライターにメッセージを印刷して停止する。正常に生成されたオブジェクト・プログラムはメイン・ルーチン 1 個といくつかのサブルーチン群とそれぞれに対応する、テーブル群によって構成されている。

3.2 セットアップ・プログラムの構成

セットアップ・プログラムは次のようなルーチンとパラメータによって構成されている。

・メイン・ルーチン

メイン・ルーチンの基本機能は、制御文の種別によって必要なサブルーチンをロードし、テーブルをそのサブルーチンに作成させることである。そのために必要なパラメータを準備し、レイアウト・ディレクトリーをシステム・ドラム上に展開したり、制御文をシステム・ドラム上にスタックするルーチンを働かせる。その他共通ルーチンを必要な時点で主記憶装置へロードする働きもある。

・パラメータ

メインルーチンとサブルーチン間の連絡、またはサブルーチンとサブルーチン間の連絡に使用される。基本内容は共通ルーチンのアドレス、共通エリアのアドレス、テーブルのアドレス、受け渡しデータ、共通作業領域である。

・共通ルーチン

共通ルーチンはサブルーチン群が共用しうるルーチン群である。用意されている共通ルーチンはレイアウト・ディレクトリー・サーチ・ルーチン、制御文読み込みルーチン、エラーメッセージ・ルーチン等である。

・サブルーチン

サブルーチンは制御文をチェックし解析して、テーブルを作り出すルーチンである。制御文 1 種類につき 1 つのサブルーチンが用意されている。またテーブル作成後テーブル間に矛盾がないかチェックするサブルーチンが 1 つある。なお、SIMPS の制御文は全部で 50 種ある。

・ジェネレート・ルーチン

ジェネレート・ルーチンはサブルーチンが作成したテーブルに従って必要なオブジェクト・ルーチンを選

択決定し、さらにジェネレート連絡データを付加して1つのオブジェクト・プログラムを完成し、名前部・辞書部ともにシステムドラム上に書き出し、メッセージをタイプライターに出力する。このメッセージにはオブジェクト・プログラムの大きさと入出力バッファの大きさが与えられており、実行時に必要な主記憶領域の大きさを知らせる。実行フレームの決定基準が利用者に知られることになる。

3.3 セットアップ・プログラムのメモリー

3.1で述べた各ルーチンやパラメータは、それぞれが使用されたり、作成されたりするタイミングや相互の関連、それに主記憶使用可能領域8KB以内という制限から、主記憶領域に常駐するものと、オーバーレイさせるものに分けてある。常駐部分は、メイン・ルーチン、パラメータ、共通ルーチンからなり、第一領域を占める。オーバーレイ部分はサブルーチンとジェネレートルーチンであり、第二領域にオーバーレイする。テーブル部分は、第三領域でここにテーブルが作り出される。以上のこととを図示したのが図1である。

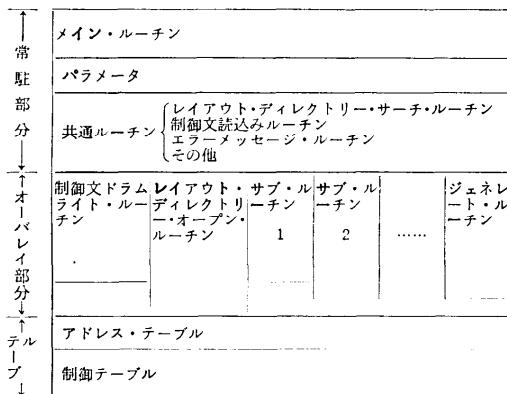


図1 セットアップ・プログラムのメモリー・レイアウト

4. オブジェクト・プログラムの機能

オブジェクト・プログラムは処理すべきデータ・ファイルを読み、セットアップ・プログラムで用意されたテーブルにしたがって、それぞれ必要な処理を行なう要求された出力ファイルを作成する。そのとき行なう処理内容は各 SIMPS プログラムごとに異なっている。たとえば、データ内容をチェックしたり、演算したり、2つのファイル間のレコードを合成したりなどを行なう。その処理内容によって出力されるファイルも、種々のものがあり、たとえば、演算結果を出力す

るファイル、チェック済レコードのみを出力するファイル、エラーレコードの出力ファイル、あるいはエラー内容レポート用ファイルなどがある。これらの機能は各 SIMPS プログラムごとにも異なるが、1個のジェネレータが有している機能も、そのままで必要とされるわけではなく、セットアップ時に与えられた指定用紙によって作られたテーブルに従って選択決定される。そのためオブジェクト・プログラムは実行時に必要な機能のみで、不要な機能はすでに切捨てられている。

4.1 オブジェクト・プログラムの構成

オブジェクト・プログラムは次のようなルーチン、パラメータ、テーブルおよび領域によって構成されている。

・メイン・ルーチン

メイン・ルーチンはオブジェクト・プログラム全体の進行を制御する機能をもっており、その働き方から区分すると2つの部分からできている。その1つはデータ処理を開始する前に1回だけ働く“イニシアライズ・セクション”であり、いま1つはデータを処理するために繰り返し働く“メインループ・セクション”である。イニシアライズ・セクションは必要な1回の処理が終了すると、他のルーチンによってオーバーレイされるように設計してある。そのため1度しか働くかないがオーバーレイの不可能な部分はメインループ・セクションに含めている。

・パラメータ

パラメータはメイン・ルーチンとサブ・ルーチン、あるいはサブ・ルーチン相互間での連絡に使用される。基本的な内容はテーブルのアドレス、サブ・ルーチンが使用するメイン・ルーチン内の特定領域のアドレス、共通ルーチンのアドレス、受け渡しデータ、連絡スイッチである。

・共通ルーチン

サブ・ルーチンが共通に使用する目的でメイン・ルーチンの中に用意してあるルーチン群で、CPG でいえば加算ルーチン、乗算ルーチンそれに除算ルーチンなどがある。

・入出力ルーチン

磁気テープおよびディスクのオープンルーチン、リード／ライト・ルーチンそれにクローズ・ルーチンである。これらのルーチンはメイン・ルーチンのイニシアライズ・セクションと同じ領域にオーバーレイされる。

・サブ・ルーチン

データに対する処理を行なうルーチンで、セットアップ・プログラムの作成されたテーブルの指示に従って、それぞれの処理を実行する。サブ・ルーチンは処理内容ごとに1つずつ作成されている。

・テーブル

これはセットアップ・プログラムで作成したテーブルでアドレス・テーブルと制御テーブルである。一例をあげると、制御テーブル中に含まれている磁気テーブルの読み書きのために必要なパラメータはメイン・ルーチンのイニシアライズ・セクションで補完される。

・入出力バッファエリア

入出力用バッファエリアはメイン・ルーチンのイニシアライズ・セクションズで、1バッファか2バッファかのどちらかを使用者の指示によって決定し、必要な大きさが確保される。

4.2 オブジェクト・プログラムのメモリー

4.1 で述べた各ルーチン、パラメータ、それにテーブル等は、使用的なされ方、大きさおよびオブジェクト・プログラムの状態によって主記憶装置上の位置が決定される。主記憶領域に常駐する部分はメイン・ルーチンのメインループ・セクション、パラメータ、それに共通ルーチンであり、さらにそのオブジェクト・プログラムに必要なサブルーチン群とテーブルである。オーバレイ部分はメイン・ルーチンのイニシアライズ・セクションと入出力各ルーチンである。以上のことを見図示したのが図2である。

5. SIMPS の特徴

SIMPS のいくつかの特徴と実用化後の反省点をまとめてみるとつきのようになる。

(1) 社内利用のみならず有料で客先に提供することをねらいに開発したものであること。

(2) プログラムを利用対象にしないこと。使いやすさを重視したこと。

(3) 性能・品質を重視したため、すべてアセンブラー (FACOM 230-25/35 BOS-FASP) で作成したこと、部分チェックを厳重にすることと、メンテナンス作業を迅速容易にすること等のためにモジュール構造を採用したこと。SIMPS 開発専用に各モジュールテストのためのデバッグギング・エイドを用意したこと。

(4) SIMPS の各プログラムごとにオペレーションズ・マニュアルを完備したこと。マニュアルには基本的な利用例について、パラメータの与え方、オペレーションの仕方をすべて実例で収録したこと。利用者

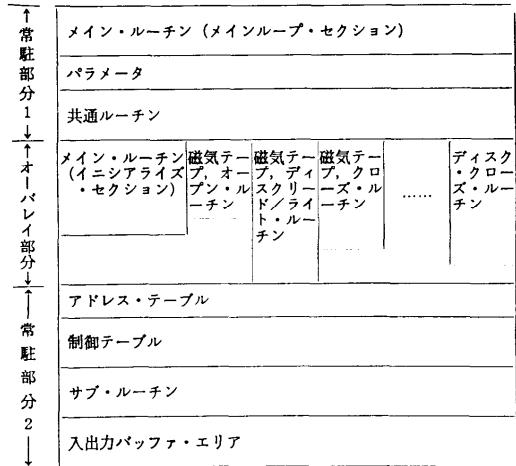


図2 オブジェクト・プログラムのメモリー・レイアウト

はこれによって、確実に動くことが確認できる一方、所望する処理内容に類似した例題を参照しながら利用できるよう配慮したこと。

(5) 利用上必要な機器構成は最小システムでもよいよう設計したこと。

(6) SIMPS の各プログラムは個別に利用することが可能であること。たとえば、あるアプリケーションで 10 個のプログラムが必要であるとする。そのうち 8 個のプログラムについては SIMPS を利用し、残りの 2 個については COBOL でプログラムを作成することも可能であること。

(7) 処理時間を COBOL で作成したものと、SIMPS で作成したものとの実測比較を行なった限りではプラスマイナス 20% 以内であること。

(8) プログラムを利用対象外とすることから、SIMPS の制御用紙はすべて表形式であること。そのためパラメータ内容をジェネレーション時タイプライターに出力させても、記入用紙を参照しない限りその意味が理解しにくいくこと。

(9) 表形式で、できるだけ記入上の注意事項も指定用紙に印刷しておくことによって、マニュアルを使用の都度参照しなくとも記入できるよう配慮したためと、9 個のジェネレータに分割したために SIMPS 全体で 50 種の記入用紙を設計することになった。これは多すぎる嫌いがあること。

(10) 前項の理由から、オペレーション上の観点に立てば、ジョブ・ステップ数が多くなりすぎる嫌いがあること。

6. COBOLとの比較

SIMPSを利用して、ある売上げファイルから倉庫コードが2であるレコードを抽出する。これと同様のことをCOBOLを用いて行なう場合について比較してみると次のようになる。まず、SIMPSSLDPで作成した処理対象(入力)ファイルのレイアウトダンプを図3で示す。この図3にもとづいて、SLT制御指定用紙と選択条件指定用紙に図4、図5のように記入する。ただし、ここではジェネレーションに統いて実行させることにし、また入力ファイル名はブランクであるため、SLT制御指定用紙の目的プログラム名欄と入力ファイル名欄には何も記入しない。この例で必要なパラメータはこの2種類である。これを実行させたときのタイプライター出力は図6の通りである。図6でPROG. 001795W. BUFF 004899Wと出力されているメッセージの意味はジェネレーションした結果得られたオブジェクト・プログラムの大きさを示す。次の行に出力されているメッセージは入出力領域を1バッファで実行せると6595ワードの領域が必要であり、2バッファで実行せると11395ワードの領域が必要であることを示す。オペレータは実行時に使用可

能な領域を判断してバッファ1で実行させるかバッファ2で実行させるかのどちらかを指示する。マルチ・ファイルにするかシングル・ファイルにするかについても同様のやり取りをタイプライターを通じてオペレータが行なう。SIMPSSLTで行なった同じことをCOBOLで実行させるには図7で示したプログラムが必要である。このプログラムを実行させたときのタイプライター出力を図8で示す。この例ではSIMPSの処理時間の方が早い。

レポート作成のためにCOBOLを使用する場合100行の記述を必要とするものでもSIMPSROPを利用すれば、わずか5種類の記入用紙の空欄をうめるだけよい。

7. おわりに

アプリケーションの多様化、プログラマ需要の増大あるいはプログラマの技術力向上の必要性のため等等、この種のものの開発利用の背景は一意ではない。歴史的な観点からいっても決して新しい試みではない。COBOL等の高級言語にくらべて普及していないことも事実である。その理由としてまず開発工数が大きいことがあげられる。そのため機種ごとに用意す

FILE-NAME =		B,F	30	R,L	T 160	72-02-04	PAGE 1
FIELD NAME	MODE	FLD	POS.	LENGTH	EDIT		
E-KUBUN	Z		0	2	99		
KUBUN	Z		2	2	99		
HANBAICD	Z		4	3	999		
TOKUICD	Z		7	8	99999999		
SHOHINCD	Z		15	7	9999999		
URIAGEBI	Z		23	6	999999		
DEN-NO	Z		28	4	9999		
SOKOCD	Z		32	1	9		
URIAGE	Z		33	7	9999999		
SURYO	Z		40	1	9		
KINGAKU	P		41	31	999999999999999999		
SHOHIN	Z		59	23	99999999999999999999999999999999		
TOKU	C		80	40	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
HANBAI	C		120	40	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		

図3 SIMPSSLT 例題用入力ファイル・レイアウト

SIMPS SLT (セレクション プログラム ジェネレーター)

SLT制御指定用紙

カード種別	<input type="checkbox"/> S <input type="checkbox"/> L	目的プログラム名	業務名 作業名 作成者 作成年月日 年 月 日																												
入出力装置指定欄	入力ファイルMT <table border="1"><tr><td>E</td><td>01</td></tr><tr><td>13</td><td>14 15</td></tr></table>	E	01	13	14 15	出力ファイルMT <table border="1"><tr><td>E</td><td>03</td></tr><tr><td>16</td><td>17 18</td></tr></table>	E	03	16	17 18	LD(レイアウトディレクトリ)-MT <table border="1"><tr><td>E</td><td>05</td></tr><tr><td>19</td><td>20 21</td></tr></table>	E	05	19	20 21																
E	01																														
13	14 15																														
E	03																														
16	17 18																														
E	05																														
19	20 21																														
(注) MTは磁気テープ(MAGNETIC TAPE)略																															
データファイル名指定欄	入力ファイル名 <table border="1"><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>29</td></tr><tr><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr></table>							29	22	23	24	25	26	27	28	出力ファイル名 <table border="1"><tr><td>S</td><td>C</td><td>2</td><td>F</td><td>I</td><td>L</td><td>E</td></tr><tr><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td></tr></table>	S	C	2	F	I	L	E	30	31	32	33	34	35	36	
						29																									
22	23	24	25	26	27	28																									
S	C	2	F	I	L	E																									
30	31	32	33	34	35	36																									
LDファイル名指定欄	入力ファイルのLDファイル名 <table border="1"><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>45</td></tr><tr><td>38</td><td>39</td><td>40</td><td>41</td><td>42</td><td>43</td><td>44</td></tr></table>							45	38	39	40	41	42	43	44	(注) 省略すると入力ファイル名と同一とみなされる															
						45																									
38	39	40	41	42	43	44																									

(株) 協栄計算センター S45.11.20 SLT1/2

図 4

FACOM 230-25/35

SIMPS SLT (セレクション プログラム ジェネレーター)

選択条件指定用紙

カード種別	<input type="checkbox"/> S <input type="checkbox"/> C	条件記号 (○でかこ)	条件値	結合記号 (○でかこ)
選択条件記入欄	(注) 左右カッコ "("と右カッコ ")" は必要な時のみ記入。②AND条件はOR条件より優先する。			
条件フィールド フィールド名 ()	5 6 7 8 9 10 11 12 13	= ① ≠ ② ≤ ③ ≥ ④ > ⑤ < ⑥	フィールドデータ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100	AND OR /
条件フィールド フィールド名 ()	28 29 30 31 32 33 34 35 36	= ① ≠ ② ≤ ③ ≥ ④ > ⑤ < ⑥	フィールドデータ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100	AND OR /
条件フィールド フィールド名 ()	51 52 53 54 55 56 57 58 59	= ① ≠ ② ≤ ③ ≥ ④ > ⑤ < ⑥	フィールドデータ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100	AND OR /

(株) 協栄計算センター S45.11.20 SLT2/2

図 5

ることが困難であること。設計思想の相違により、機能や性能が一様でないこと。これまでのプログラム作成過程からみて、ドキュメンテーションが十分でなかったことも指摘できよう。

```
* END IPL *
*W L000 SIMPSLT.LIB/HIT-E07
*W L000 PROCESS FMD
*W XEXEC
*W /SL E01E03E05 SC2FILE"
*W /END
*W SIMPS SLT * PRCG_001795W, BUFF 004800W
*W SIMPS SLT * TOTAL 006595W(16), 011395W(26)
*W JOB TIME 0M13.3S
*W YJEND

*W XFEF
*W SIMPS SLT * TYPE IN BUFFER 1 CR 2
*W SIMPS SLT * TYPE IN MT SINGLE-FILE=1 OR MULTI-FILE=2
*W SIMPS SLT * INPUT RECORD 080155
*W SIMPS SLT * CPUTUP RECORD 018847
*W JOB TIME 5M31.0S
*W YJEND
```

図 6 SIMPS SLT 例題のジェネレーション
および実行時のタイプライター出力

FACOM 230-25/35 COBOL -7106164 0015-02	
1	ISN
2	SEQN, A-B
3	001010 IDENTIFICATION DIVISION.
4	001020 PROGRAM-ID. SLT-1.
5	001030 AUTHOR. YUKI-MIYAMA.
6	001040 ENVIRONMENT DIVISION.
7	001050 CONFIGURATION SECTION.
8	001060 SOURCE-COMPUTER. FACOM 230-25.
9	001070 OBJECT-COMPUTER. FACOM 230-25.
10	001080 INPUT-OUTPUT SECTION.
11	001100 SELECT ITAPE ASSIGN TO MT-E1.
12	001110 SELECT OTAPE ASSIGN TO ME-E2.
13	001120 DATA DIVISION.
14	001130 FD ITAPE.
15	001140 BLOCK CONTAINS 30 RECORDS : LABEL REC.
16	001150 DATA RECORD IS O-REC.
17	001170 01 I-REC.
18	001180 C2 F-KUBUN PIC 99.
19	001190 02 KUBUN PIC 99.
20	001200 02 FILLER PIC X(18).
21	001210 02 URITAGEBI PIC 9(6).
22	001220 02 DEN-NOD PIC 9(4).
23	002010 02 SOKOCD PIC 9.
24	002020 02 FILLER PIC X(127).
25	002030 FD OTAPE.
26	002040 BLOCK CONTAINS 30 RECORDS : LABEL REC.
27	002060 DATA RECORD IS O-REC.
28	002070 01 O-REC.
29	002080 02 FILLER PIC X(160).
30	002081 WORKING-STORAGE SECTION.
31	002082 77 CI PIC 9(6) VALUE 0.
32	002083 77 CO PIC 9(6) VALUE 0.
33	002090 PROCEDURE DIVISION.
34	002100 DISPLAY *** START OF SLT-1. ***.
35	002110 OPEN INPUT ITAPE OUTPUT OTAPE.
36	002120 LTR.
37	002130 READ ITAPE AT END GO TO CL.
38	002131 ADD 1 TO CI.
39	002140 IF SOKOCD = 2 GO TO OM.
40	002180 GO TO LTR.
41	002190 OM.
42	002200 MOVE I-REC TO O-REC.
43	002210 WRITE O-REC.
44	002211 ADD 1 TO CO.
45	002220 GO TO LTR.
46	003010 CL.
47	003011 DISPLAY INPUT RECORD CI.
48	003012 DISPLAY OUTPUT RECORD CO.
49	003013 DISPLAY *** END OF SLT-1. ***.
50	003020 CLOSE ITAPE OTAPE.
51	003030 STOP RUN.
52	999999 ENDCOBOL.

図 7 SIMPSSLT 例題に対応する COBOL プログラム

磁気テープ・ベースの SIMPS は、実用化を開始してから 9 カ月経過したが、ほぼ見込み通りの成果である。できあがりの総ステップ数は 75,000 であるが、

```
*M *JOB
*M *COBOL LIST
*M CB, COMPILE END
*M JOB TIME 0M13.0S
*M *JEND

*M *EXEC
*M LD, UNI = 5863
*** START OF SLT-1. ***
* INPUT RECORD 080155
* OUTPUT RECORD 018847
*** END OF SLT-1. ***
*M JOB TIME 6M33.0S
*M *JEND
```

図 8 SIMPS SLT 例題に対応する COBOL プログラムのコンパイルと実行時のタイプライター出力

プログラム設計にはモジュール転用を重視したため、実質上開発したステップ数は 40,000 弱である。開発工数は 120 人月でコンピュータ使用は 750 時間である。詳細設計担当者は 1 人で、システムプログラミング経験者であり、プログラミングの主力要員はプログラマ 1 年生 6 名である。分担作業が比較的スムーズであった理由は、各担当者の努力もさることながら、投入可能要因の技術レベルにマッチした設計方式であったといえよう。

最後に、BOS 制御のもとにいすれのフレームでも SIMPS が作動するように設計してあるが、BOS を開発した富士通ソフトウェア技術部の方々に、インターフェース上の諸点につき技術的な指導を得たこと、このレポートをまとめるに当たって、中村清司、深山幸高両氏の協力を得たことに感謝する。

(昭和 46 年 12 月 7 日 受付)

(昭和 47 年 3 月 21 日再受付)