

M 345 文字表示装置支援ソフトウェアの作成

渡部 哲夫*・石原 好宏*

Implementation of a Supporting Software for Simple and Easy
Utilization of the M345 Colored-Character Display Unit

Tetsuo WATANABE and Yoshihiro ISHIHARA

Abstract

This paper describes supporting software which permits the facile utilization of the M345 colored-character display unit which is connected to the MELCOM 9100/30F in our department. The basic software which the computer maker has provided us enables us to utilize M345 only when programs are written in assembly language. Programming in this language is rather troublesome job. Thus we designed and built supporting software which enables us to write programs in FORTRAN utilizing M345 directly and to command it in job control language. This supporting software aims primarily at facilitating conversational operation utilizing the M345. Secondary aim is the simplification of operations such as input and output of data, file editing, and console operation through the M345.

1. まえがき

近年、情報処理システム（電子計算機）を能率よく利用するための道具として、文字表示装置や図形表示装置の有効性が広く認識され、利用されるようになってきた。その主な理由は、カードリーダーやラインプリンターのような単能の装置と違って、これらが次のような複数の機能を合せ持っていることであると考えられる。

- (1) 会話型処理
- (2) データの入出力
- (3) ファイルの編集
- (4) コンソールオペレーション

ところで、本学には情報処理システム MELCOM 9100/30F が導入されており、これに文字表示装置 M345 が接続されている。しかし、上記の機能を実現するためにメーカーが提供しているソフトウェアはごく基本的なものだけである（2. 参照）。従ってこのままでは、プログラミングのたびにアセンブリ言語によるかなり面倒で注意深い細かな作業が必要となる。これではプログラミングの作業能率が低下するだけで

なく、利用方法の習得そのものもかなり困難を伴う。これらの状況を克服するためには、上記の任意の機能が、FORTRAN などの高級言語のプログラムから容易に引用でき、キー操作などにより簡単に実行できることが望まれる。以上のような見地からわれわれは、M345 文字表示装置支援ソフトウェアを作成した。その際にわれわれが当面の目標としたことは、第一に文字表示装置 M345 を会話型処理の入出力端末として容易に利用できること、第二にユーザープログラムのための基本データの補助ファイルへの入力・編集、処理結果の出力、それらの制御などが文字表示装置から容易に行えることであった。

そこで本稿では、まず文字表示装置 M345 の制御方式とメーカー提供の基本ソフトウェアの概略を本ソフトウェアの作成に必要な範囲で述べる。次に今回作成した支援ソフトウェアの概要を、更に本ソフトウェアの各プログラムモジュールの使い方と応用例を述べる。なお、今回作成した各プログラムモジュールの機能の一覧表は付録にまとめる。

2. M 345 の制御方式と基本ソフトウェア¹⁾

文字表示装置 M345 は、MELCOM 9100/30F の本体とは、マルチプレクサーチャネル MPX、ラインコ

* 工業短期大学部情報処理工学科

ントロールアダプター LCA を経由して接続されている。又、この回線をポーリング方式で制御するためのプログラムとして LCP 345 がメーカーから提供されている。LCP 345 は、オペレーティングシステムが管理するタイムスケジューラルーチンによって、一定の時間間隔で起動される。

LCP 345 を介して M 345 を利用する方法としては、回線アクセスサブルーチン L¥EXIO を用いる場合と、I/O WRITER を用いる場合が可能である。しかし、前者の方が用途が広いので、本稿では前者についてだけ考慮する。その場合、回線制御のためのデータ領域として、一つの M 345 当り一つの UCB (Unit Control Block) が、又一つの LCA 当り一つの LCB (Line Control Block) がとられ、これらが CRT 用 GCA (Global Common Area) としてまとめられている。

起動された LCP 345 は、対応する LCB の情報をもとに割当てられた UCB を探索して読み書きのリクエストの有無を判断し、リクエストがあれば決められた伝送制御手順に従って端末へのアクセスを開始する。又、たとえ読み書きのリクエストがない場合でも、アテンションキーによる M 345 からの入力がありうるので一定時間間隔のポーリングを行う。

ユーザーが M 345 を利用するためには、行わせるべき仕事に応じて LCB と UCB の内容を設定してやる必要がある。このためにメーカーはプログラム L¥EXIO を提供している。このプログラムは、ユーザープログラム内の DCB (Data Control Block) とユーザーバッファ UBUF の情報を利用して LCB と UCB の内容を書き換える。なお、DCB の内容は、読み書きの区別、UBUF の先頭番地、UBUF 内の文字数など、一方 UBUF の内容は画面表示用の制御情報および表示データである。以上のことから、ユーザーは自分が書くプログラムの中で、DCB と UBUF を作成し、これを引渡し情報として L¥EXIO と LCP 345 が必要な手続きを実行して目的が達せられることになる。アセンブラ言語で CRT 画面に表示する例を Fig. 1-(a) に示す。

なおここで注意すべき点は MELCOM 9100/30 F の本体側で EBCDIK コードを使っているのに対し、M 345 側で JIS-C-6220 及び ISO (カナ入り) に準拠したコード (以下、本文では単に ISO コードと呼ぶ) を使っていることである。従って本体側と M 345 側との間でデータの受渡しをするときにはコード変換が必要である。ところがメーカー提供のコード変換プログラム ETAC¥, ATEC¥ ではカナ文字のコード変換を無視する。又、CRT の画面上に同時に表示可能な

文字数は 1 行当り 80 文字で 25 行、合計 2000 文字である。

以上の諸点を考慮して、3. で支援ソフトウェアを作成するための基本事項をまとめる。

3. 支援ソフトウェアの概要

まず支援ソフトウェアの基本的なものとして、カナ文字も含めた全記号を EBCDIK から ISO へ、又 ISO から EBCDIK へコード変換するプログラムを作成する。次いで、M 345 文字表示装置を、会話型処理、データの入出力、ファイルの編集、コンソールオペレーション、の 4 種類の操作端末として利用するための支

```

                REGIN  EX1
*              5 *ヨコ 10カラム カラ アキID ノックリック
*              'YAMAGUCHI' ト エヨウシズト
*
                REXT  L¥EXIO
-----
DCB            DC      X"0000" * コネクト ナックル *
                DC      X"0002" * OUTPUT リクエスト *
                AC      UBUF * テータ アドレス *
                DC      S"16" * テンク エシズ *
                AC      ENDERR * エラー リターン アドレス *
                DC      X"0000" * エラー ステータ *
                DC      X"0000" * エヨウ エシズ *
*
* UBUF          DC      X"1B0B" *          ESC   VT
                DC      X"2429" *          (5)  (10)
                DC      X"0F61" *          SI   PH
                DC      X"5941" *          Y    .
                DC      X"4D41" *          M    A
                DC      X"4755" *          G    U
                DC      X"434B" *          C    H
                DC      X"4903" *          I    ETX
*
* EX1          EQU     *
                BAL     L¥EXIO
                AC      DCB
*
* ENDERR       EQU     *
                ORIGIN
                SVC
                DC      X"0005"
                END     EX1
    
```

Fig. 1-(a) A program in assembly language for displaying on CRT the data defined in the program.

```

1      MAIN  EX2
2      C
3      C      シックロクカラム CRTOUX ショウイ
4      C
5      INTEGER RUF(40),RED
6      DATA RED/420F61/
7      ENCODE(9,1,BUF)
8      1 FORMAT('YAMAGUCHI')
9      CALL CRTOUX(5,10,9,RED,BUF)
10     STOP
11     END
-----
OBJECT WORDS      83
    
```

Fig. 1-(b) A program which describes the same job as that shown in Fig. 1-(a). This is written in FORTRAN and uses the supporting program CRTOUX.

援プログラム群を作成する。ここではそのために必要な基本的事項を整理する。なお、これらに基づいて実際に作成した支援プログラムの名称と機能説明の一覧表は付録にまとめる。

3.1 コード変換プログラム

EBCDIK コードでは英数字、カナ文字、特殊記号がすべて単純構造の8単位符号で表わされる。一方、ISO コードでは、英数字中心のシフトイン側とカナ文字中心のシフトアウト側とに分れた、いわゆる2段シフト構造の7単位符号となっている。従って両コード系の変換を行う場合、ISO コードにおける反転符号に起因するデータ長の伸縮を考慮した処置が必要である。コード変換プログラムは、直接ユーザーが使うことはないが、3.2以降に述べる各支援プログラムの中で不可欠の基本的なプログラムの一つである。

3.2 会話型処理

これを実現するためには、表示動作、入力動作、編集動作をうまく組合せる必要がある。このためのプログラミングを直接ユーザーがその都度アセンブリ言語を用いて行うのはかなり煩雑である。(Fig. 1-(a)参照)しかしそのようなプログラムをよく観察すると、いくつかの基本的な手続きにまとめることができ、しかもそれらの中で問題に応じて書き直しを要するのは少数のデータだけであって、手続きそのものは一般に共通であることが分かる。従って、ここでは共通のかつ基本的な処理手続きをサブルーチン型のオブジェクトモジュールとし、その中で問題ごとに書き直すべきデータを引数としてFORTRANなどで書かれた親プログラム側と受渡しができるようにする。

3.2.1 表示動作

このためにユーザーが指定すべきデータは、CRTの画面上の、(1)カーソル位置(行および列)、(2)色および点滅の有無、(3)表示文字数、(4)表示すべき文字列の格納領域名、である。これらを次の5つの引数N1~N5として与える。

- N1; 先頭文字の行位置(1~25の整数または整数変数)
 - N2; 先頭文字の列位置(1~80の整数または整数変数)
 - N3; 表示したい文字数(許容範囲—後述—内の整数または整数変数)
 - N4; 点滅の有無および色コード
 - N5; 表示したい文字列の格納領域名
- なお、N4についてはユーザーがDATA文などを

使って直接に以下のコードを指定する。X“—”は、—が16進コードであることを示している。

- (1) 赤で点滅させる ……X“0E 61”
- (2) 緑で点滅させる ……X“0E 62”
- (3) 白で点滅させる ……X“0E 63”
- (4) 赤で表示し点滅はさせない ……X“0F 61”
- (5) 緑で表示し点滅はさせない ……X“0F 62”
- (6) 白で表示し点滅はさせない ……X“0F 63”

次に表示動作のための手続きの概略を述べる。

ステップ1) 引数N1, N2を用いて、表示文字列の先頭文字位置コードを合成し、UBUF+1番地にセットする。

ステップ2) 引数N4をUBUF+2番地にセットする。

ステップ3) 引数N5の内容はEBCDIKコードで表記されているので、これをISOコードに変換し、その結果をUBUF+3番地以降に格納する。このとき引数N3を利用する。この変換で、表示すべき文字数はシフトイン、シフトアウトの符号が挿入された分だけ増加する。

ステップ4) ステップ3で得た新しい文字数に7を加えた数値をDCB+3番地にセットする。なお、7は画面表示用の制御情報、機能符号などが占める文字数である。

ステップ5) DCBを引数として回線アクセスサブルーチンL \forall EXIOへブランチする。

以上のデータ設定および手続きをアセンブリ言語で書いた一例がFig. 1-(a)である。これらの手続きを標準化して表示動作の支援プログラムモジュールとしている。(Fig. 1-(b)参照)その際、UBUFの占有領域の経済性と表示速度の向上を考慮して、支援プログラムモジュールとしては表示可能文字列が1行以内でカナ文字込み、4行以内でカナ文字込み、8行以内でカナ文字なしの3種類を作成した。なお、表示動作中に何らかのエラーが発生したときは、エラーが発生したプログラム名をシステムタイプライターに表示させる。

3.2.2 入力動作

入力動作は、文字データ入力か数値データ入力かによって細部が異なるが、基本的には次の3つのステップから成る。

ステップ1) オペレーターインプットフィールドを作成する。この後このフィールドを介したデータの入力が可能となる。

ステップ2) キー操作によって、オペレーターインプットフィールド内に実際にデータを作成する。

これにより ISO コードによるデータが UBUF 内に入力される。但し、この場合も機能符号、画面位置情報などが入力データの頭に自動的にセットされる。

ステップ3) UBUF 内の機能符号、画面位置情報などを除去した後、入力データを EBCDIK コードに変換して、親プログラムへ転送する。

なお、次のようなハードウェア上の制約がある。

制約1) オペレーター入力フィールドは行を越えて設定することができない。

制約2) 一定時間(約3分)を越えてキーイン操作を継続することができない。しかしわれわれはこの時間をソフト的にその3倍の時間まで許すことにした。その警告はシステムタイプライターへ出力される。

以下、文字データ入力用と数値データ入力用とに分けて、それぞれの場合の支援システムの作成に必要な基本事項を整理する。

3.2.2.1 文字データ入力の場合

ユーザーが指定すべきデータは次の5つの引数 N1 ~N5 である。

N1; オペレーター入力フィールドを設定する行 (1~25の整数または整数変数)。

N2; オペレーター入力フィールドが始まる列 (1~77以内であって 80-N3-2 を満足する整数または整数変数)。

N3; 入力予定文字数 (1~77の整数または整数変数)。

N4; 点滅の有無および色コード (3.2.1参照)。

N5; 入力データが転送される親プログラム内の領域名。

この中の N1~N4 を使ってオペレーター入力フィールドが Fig. 2 のように作られる。

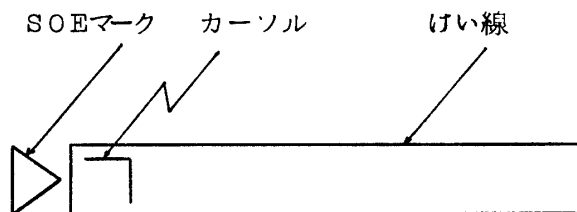


Fig. 2 Non-numerical data input field.

なお、入力予定文字数より少ない文字数を入力した場合には残りのフィールドには自動的に空白コードが詰められる。

3.2.2.2 数値データの入力の場合

ユーザーが指定すべきデータは次の5つの引数 N1

~N5 である。

N1; オペレーター入力フィールドを設定する行 (1~25の整数または整数変数)。

N2; オペレーター入力フィールドが始まる列 (1~62又は66の整数または整数変数)。

N3; 点滅の有無および色コード (3.2.1参照)。

N4; 入力数値データが倍長整数型として転送される親プログラム内の領域名。

N5; 入力数値データが倍精度実数型として転送される親プログラム内の領域名。

この中の N1~N3 を使って数値データ入力用のオペレーター入力フィールドが Fig. 3 のように作られる。なお、フィールドの大きさは10桁および14桁の2種類に限定する。又、このフィールドで許される字種は0~9の数字、小数点、+-符号で、これらが正しい整数または実数の表記形式になっておればよいものとする。

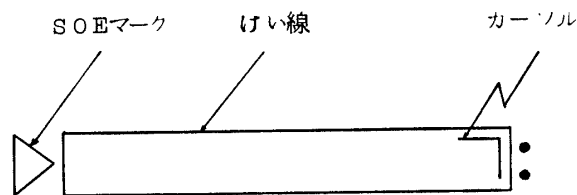


Fig. 3 Numerical data input field.

3.2.3 ライトペンによるメニュー選択

3.2.3.1 ユーザーがメニューを作る場合

これは、CRT 画面上に作り出されたメニューの中から任意の1個の選択肢をオペレーターにライトペンで選ばせた後、その選択肢の先頭からの順序数を、指定した領域にセットしてくれる機能である。これを実現するための手順の概略を次に述べる。

手順1) メニューを IBUF 上に作成する。その書式は次のようなものである。

```
@@項目1@@項目2.....項目n¥¥コメント@
IBUF 上にこれを作成する方法は、穿孔カード、DATA 文、文字表示装置の会話型処理による入力動作など任意のものでよい。但し、IBUF の大きさは80文字分とするが、有効文字数は66文字以内と考えて支援プログラムを作成する。
```

手順2) IBUF の内容のうち、@と¥は機能符号に、又項目1~項目nは EBCDIK コードで表記されている場合にだけ ISO コードに変換する。

手順3) ユーザーから別に与えられる位置情報 N1 及び N2 (3.2.1の引数 N1, N2 に同じ) を使って画面上にメニューを表示する。オペレーターが

ライトペンによって選択肢の1つをたたくと、その位置情報が先頭からの順序数として親プログラム内の領域 M にセットされる。

メニュー選択機能の場合、手順1, 2, 3のそれぞれが1つの支援プログラムモジュールを形成する。なお、手順2が終わったとき画面上に表示されるメニュー（これを正しくはライトペンディテクタブルフィールドと言う）の例を Fig. 4 に示す。

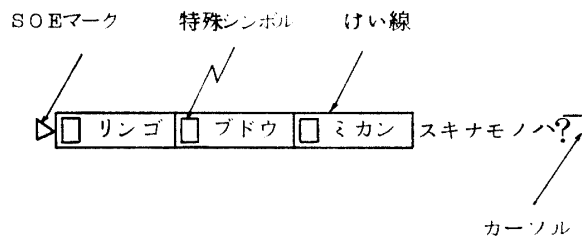


Fig. 4 Light-pen detectable field.

3.2.3.2 汎用として準備するメニューの場合

ユーザーがメニューを逐一作るのが煩わしい場合を想定して、2分岐および6分岐のライトペンディテクタブルフィールドを作成する支援プログラムモジュールを準備する。その引数としては次の3種類をこの順番に並べるものとする。

N1; { 3.2.1で述べた引数 N1, N2と同じ意味をも
N2; } 1つ。

B1~Bn; 一般に m 番目の選択肢が選ばれたときの分岐先の文番号を Jm とすれば、Bm は JmS と表記する。なお、n は2分岐の場合は2, 6分岐の場合は6でなければならない。

3.2.4 画面の編集

画面の編集としては当面、画面の消去、巻上げ、巻下げを考えるとそれぞれに支援プログラムを作成する。

3.2.4.1 画面の消去

これは、全画面を消去する場合と、画面の一部を消去する場合に分けて別個の支援プログラムを作る。前者の場合は問題ごとに新たなデータを指定する必要はない。しかし後者の場合は、次の5つの引数 N1~N5 に基づいて部分消去を行う。

N1; 消去開始の行, N1; 消去開始の列

N3; 消去終了の行, N4; 消去終了の列

N5; 消去コード

ここで、N1 と N3 は1~25の、N2 と N4 は1~80の整数または整数であること。又、消去コードは消去に伴ういくつかの指定を行う (Table 1 参照)。

Table 1 Codes for display erasing or clearing

種類	機能	コード
Display Clear	指定位置内の画面を消去し SP を入れる。	6E6F
Protect Display Clear	指定位置内の画面を消去しプロテクトされた SP を入れる。	6D6F
Display Erase	指定位置内の画面をセンターフィールドを除いて消去し SP を入れる。	6E61
Protect Display Erase	指定位置内の画面をセンターフィールドを除いて消去しプロテクトされた SP を入れる。	6D61

3.2.4.2 画面の巻上げと巻下げ

この機能は次の3つのデータ N1~N3 を用いると一つの支援プログラムで実現できる。

N1; 巻上げ又は巻下げの開始行 (1~25の整数または整数)。

N2; 巻上げ又は巻下げの幅を行数で指定する。

N3; 巻上げの場合 2HUP, 巻下げの場合 2HDW と指定する。

しかし、この機能は N1~N3 をプログラムの中で引数として渡して実現する場合の外、会話的に M 345 端末からのキー操作 (N1 と N2 の指定) とライトペン操作 (巻上げか巻下げかの選択) で実現する場合を想定すると便利である。従ってここでは上記のそれぞれについて別個の支援プログラムを作成する。

3.2.5 画面読取りのための時間待ち

会話型処理では人間が一連の処理の流れの中に直接介入する。しかも人間の行う処理時間は機械のそれと比べるとはるかに遅い。従って画面に表示されたものを読取るためにも相応の時間を確保する必要がある。ここではこのために2種類の支援プログラムを作成する。一つは、引数で指定された時間 (秒単位で $2^{15}-1$ 秒以内) だけ無条件に時間待ちするもの、もう一つは9分以内でしかもトランスミットキーが押されるまで時間待ちするものである。

3.3 データの入出力

外部記憶装置または入出力装置と本体の主記憶装置との間でデータの入出力を行うために、FORTRAN 言語では READ 文と WRITE 文を準備している。しかしこれらを用いて文字表示装置 M 345 からのデータ入出力を行うことは、われわれの所の MELCOM 9100/30F のシステム生成では想定しなかった。従って直接にはそれは実現できないが、外部記憶装置上の特定ファイルを経由すれば間接的ながら M 345 を READ 文、WRITE 文の対象装置とすることができる。

そこで、M345 から特定ファイルへのデータの入力、及び特定ファイルから M345 へのデータの出力、を行うための支援プログラムをロードモジュールで作ることとした (Fig.5 参照)。これにより、ユーザーは、バッチ処理の過程で文字表示装置を、カードリーダー又はラインプリンターとほとんど同じイメージで利用できることになる。

なお、ここで言うデータの入出力は、バッチ処理に必要なデータの全体をひとまとめにして行うものであって、会話型処理の際に当面の処理に必要な少量データの入出力とは文字表示装置の利用の仕方が根本的に異なる。

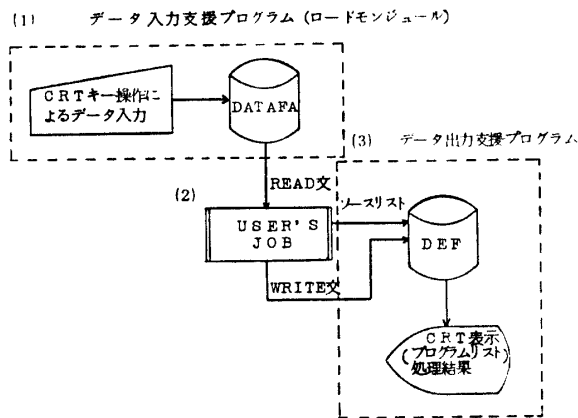


Fig. 5 Design of supporting programs for input and output of data through CRT.

3.3.1 データ入力操作の支援プログラム

この支援プログラムは、3.3で述べたことから、文字表示装置 M345 のキーボードから穿孔カードのイメージで任意のデータを外部記憶装置上の特定ファイル DATAFA に入力する作業を支援する。その仕様の概要を述べる。

- (1) プログラム名 ; CRTDSK
- (2) 特定ファイル名 ; DATAFA
- (3) レコード長 ; 80バイト
- (4) 最大レコード数 ; 4,000レコード
- (5) ファイルの形式 ; 順編成ファイル
- (6) プログラムの起動方法 ; 制御文

この支援プログラムでは、起動がかけられるとまず文字表示装置の画面に4行分のデータ入力フィールドを表示する。この1行が1レコードに相当する。そこでオペレーターはこのフィールド内にカード穿孔機によるデータカード作成と全く同じ要領でデータを作成する。誤りがないことを確かめた後トランスミットキーを押下すると、データはDATAFAへ転送され、続いて画面右下には更にデータの入力操作を継続する

か否かを問うライトペンディテクタブルフィールドを表示する。従って必要に応じていずれかの選択肢をライトペンで選択すると、入力操作終了あるいは更に新たにデータ入力フィールドが表示されて入力操作が継続できることになる。

こうしてDATAFAに入力したデータをユーザープログラムで利用するためには、READ文の中のファイル参照番号をDATAFAに割当てただけでよい。

3.3.2 データ出力用の支援プログラム

この支援プログラムでは、MELCOM 9100/30 F のオペレーティングシステムのもつ出力留保機能を利用する。この機能は、バッチ処理されたジョブのソースリスト、処理結果、処理経過情報、会計情報などを、新たに出力要求があるまで外部記憶装置上の特定ファイル DEF に貯えておくためのものである。従って、この支援プログラムでは、処理結果の外にソースリストなどラインプリンター用紙に出力できるものはすべて文字表示装置にも出力される。但し、表示する画面の大きさの制限から1行当りの表示可能文字数は80であり、それを越える分は捨てられる。以上のことから、この支援プログラムでは、特定ファイル DEF にチェーン構造でしかもコンパクトに納められている出力情報を1行当り80文字の順編成となるよう再編集して更にそれをCRTに表示する作業を行う。

従って、ユーザーは、プログラミングの際処理結果の出力がラインプリンター用紙の左から80文字以内で行なわれるよう注意するだけでよい。但し、ラインプリンターのような重ね表示は不可能である。この支援プログラムの使い方は4.2で述べる。

3.4 ファイルの編集

文字表示装置のキー操作で行うべきファイルの編集機能としては、ソース形式のプログラム又はデータが登録されているファイル SCFILE を対象として、内容の表示、追加登録、挿入、置換、削除、転送を考える。これを実現する処理手続きは次の5つのプログラムモジュールで構成する。

モジュール1) ソースファイル SCFILE の内容の全部を作業ファイル WFILE 1 に転送する。

モジュール2) 作業ファイル WFILE 1 の内容を20レコードを単位として連続的に、指定された範囲だけ CRT の画面に表示する。但し、画面の切換えと表示の終了はライトペンで指示する。

モジュール3) まず編集用情報(編集の種類や位置を表す制御情報、追加、挿入、置換のために必要な新たなデータなど)を文字表示装置からキー操

作で作業ファイル WFILE 2 へ入力する。次に、作業ファイル WFILE 2 の編集用情報を先頭から順次調べていく。その結果、編集の種類が追加登録、置換、削除のいずれかであれば、作業ファイル WFILE 1 を対象として必要な作業を直ちに実施する。しかし編集の種類が挿入の場合には別の作業ファイル WFILE 3 へ、挿入すべき生データを作業ファイル WFILE 2 から転記する。実際の挿入作業は、別に主記憶上に記録した挿入位置、挿入データ量の情報を参照しながらモジュール 4 で実施する。

モジュール 4) まず 2 つの作業ファイル WFILE 1, WFILE 3, 及び主記憶上の挿入情報を利用して混ぜ合せの作業を行い、その結果を WFILE 4 に作り出す。これによって挿入と削除が完了する。次いで、WFILE 4 の内容を WFILE 1 へ複写する。
モジュール 5) 編集作業を終了させるための作業である。その手続きは、作業ファイル WFILE 4 の内容をいずれかのファイル (例えば SCFILE) へ転送することで実現できる。

ここで作成するファイル編集機能では、モジュール 1 及び 5 をそれぞれ開始と終末の手続きとして固定する外は、モジュール 2, 3, 4 でライトペンによって任意の順序で選択できる。又、この編集作業で使われるファイルのうち、ユーザーが指定すべきものは、SCFILE とモジュール 5 で必要な転送先のファイルであって、いずれも順編成ファイル、1 レコードが 80 バイト、かつ全体が 6000 レコード以内でなければならない。

3.5 コンソールオペレーション

MELCOM 9100/30 F にはコンソールオペレーション用の装置としてタイプライターが接続されている。文字表示装置からのコンソールオペレーションとしてはこれを補足する意味で、当面次の 3 種類の機能を果たす支援プログラムをロードモジュールで作成する。

(1) システムファイル DEFSF の内容表示

DEFSF には出力留保用のファイル DEF に登録されたジョブネームと識別番号が登録されている。この内容を CRT の画面に表示する作業である。

(2) ディスクの内容のダンプ

カートリッジディスクのセクターアドレスによる開始番地と終了番地をキー操作で、ディスク装置の機番 (#0 又は #1) をライトペンで指定することにより、ラインプリンター用紙に 16

進符号と文字の両方で、指定されたディスクの内容を出力する。

(3) システム定数の表示と変更

ここではシステム定数として、ラインプリンター用紙の使用ページ数制限、処理時間制限、ジョブネームチェックの有効桁数を考え、これらの CRT 画面への表示、キー操作による変更を行う。なお、実際の表示、変更のためのコマンドの書式は文献 2) の p. 84 と p. 85 のものと全く同じとする。

4. 支援ソフトウェアの使用法および応用例

ここでは 3. で述べた支援ソフトウェアの使用法の概略を述べる。又、この支援システムを応用して作成したファイル間データ転送システムの概要も述べる。なお、今回作成したプログラムはいずれも M 345 のキー操作で起動させることは今のところできない。従って以下に述べる使用例中の制御文はいずれもカードリーダー又はシステムタイプライターから入力される必要がある。

4.1 会話型処理の場合

このプログラム群は FORTRAN 言語によるソースプログラムからは CALL 文によって自動呼出しされる。但し、このプログラム群を呼び出すソースプログラムを翻訳した後の制御文としては、結合編集と実行をまとめたカタログド・プロシジャー CRTIOW を使う必要がある (Fig. 6 参照)。

```
//JOB,FAA00C.
//FORT.

1 C
2 C      カイワカタ フ*02*56 カンフ*
3 C
4      DIMENSION IBUF(40)
5      INTEGER AKA,MIDORI,SHIRO
6      DOUBLE INTEGER IDATA,JDATA,KDATA
7      DOUBLE PRECISION FDATA,GDATA
8      DATA AKA,MIDORI,SHIRO/420F61,420F62,420F63/
9 C
10 1 CALL CLEAR0
11 FNCODE(80,10,IBUF)
12 10 FORMAT(' カイワカタ フ*02*56 カンフ*')
13 CALL CRTOUX(5,10,19,AKA,IBUF)
14 CALL NSDATA(5,40,AKA,IDATA,FDATA)
15 CALL NSDATA(5,60,SHIRO,JDATA,GDATA)
16 KDATA=IDATA+JDATA
17 ENCODE(80,2,IBUF) IDATA,JDATA,KDATA
18 2 FORMAT(M10,' = ',M10,' = ',M10)
19 CALL CRTOUX(20,5,36,MIDORI,IBUF)
20 FNCODE(80,3,IBUF)
21 3 FDRMAT('123 456789 0123456789 0123456789 0123456789')
22 CALL PENASC(IBUF)
23 CALL PENCN(25,5,IBUF,M)
24 GO TO (1,99),M
25 99 STOP
26 END

//CALL,CRTIOW.
```

Fig. 6 Sample program for conversational processing, where the supporting program modules CLEAR0, CRTOUX, NSDATA, PENASC, PENCN are used.

4.2 データの入出力の場合

この場合、入力用と出力用の2種類のロードモジュールプログラムがある。ユーザーが作成するソースプログラムは翻訳・結合編集・実行のすべての制御文を伴って、それらの間にはさまれる (Fig. 5 参照)。実際に利用する場合の制御文等の構成は Fig. 7 に示す。その際の注意事項を以下にまとめる：(a) で OM=DEF とする (出力留保機能の使用)。 (b) でファイル参照番号をユーザープログラム内の READ 文のそれと一致させる。なお、(1) では M 345 から DATAFA へのデータの入力、(2) では DATAFA のデータの処理、(3) では処理結果の CRT 画面への出力をそれぞれ行う。(3) で //CALL, HISDEF. 又は //CALL,

```

//JOB, ジョブネーム, OM=DEF.      ←(a)
//EXEC, CRTDSK.                    ←(1) 入力用
//LASGN, Un, FILE=DATAFA.         ←(b) Un:ファイル参照番号
                                   n=1~99
↑
ユーザー プログラム              ←(2)
(翻訳, 結合編集, 実行の制御文含む)
↓
//JOB, ジョブネーム, {OM=DEF},
//CALL, HISDEF.                    ←(3) 出力用
    {又は, //CALL, DEFCRT, }
//SKEND.
    
```

Fig. 7 An example of input and output operations of data. (Each number m ($m=1, 2, 3$) in parentheses corresponds to that in Fig. 5.)

DEFCRT. となっているのは、出力用のロードモジュールプログラム HISDEF 又は DEFCRT がカタログド・プロシジャーの中で呼出され実行されるためである。

4.3 ファイルの編集の場合

このためのロードモジュールプログラム EDITDT もカタログド・プロシジャーの中で呼出され実行される。そのための制御文の前で、編集または表示すべきデータが納められている転送元ファイルのファイル参照番号を u7, 編集処理の結果を納める転送先ファイルのそれを u20 とする必要がある。ファイル編集のための制御文の書き方を次に示す。なお、SNO とはファイルが存在するディスクの物理機番である。

```

//JOB, ジョブネーム,
//LASGN, u7, FILE=転送元, SNO=物理機番.
//LASGN, u20, FILE=転送元, SNO=物理機番.
//CALL, EDITDT.
//SKEND.
    
```

これにより EDITDT に起動がかけられると、最初は Table 2 の第一段階に示す3つのサブコマンドを選択すべくライトペンディテクタブルフィールドが表示される。DISPLAY を選ぶと、引続いて DISPLAY 又は、DISPLAY-END を選ばせるライトペンディテクタブルフィールドが表示される。EDIT を選んだときは@で始まる4組のサブコマンド及びデータのうち必要なものをキー操作で入力すればよい。第二段階の終了はライトペンで指示され、その結果第一段階の表示が再び行われる。全体の操作の終了は EDITEND が選択され指定ファイルへのデータの転送が終了した時点である。

Table 2 Sub-commands for file editing

第 一 段 階	第 二 段 階	
*DISPLAY ソースファイルの表示	*DISPLAY	表示を行う。
	*DISPLAY-END	表示の終了。
*EDIT 編 集	@ ADD. { データ } @ END.	ソースデータファイルの最後尾に追加する。追加するデータのレコード数は特に制限しない。
	@ DELET, XXXXX-XXXXX.	行番号 XXXXX-XXXXX の範囲を削除。
	@ REPLAC, XXXXX. データ・1レコード分	行番号 XXXXX の内容を次の行のデータ1レコード分で置き換える。
	@ INSERT, XXXXX. { データ } @ END.	行番号 XXXXX の次に @END の直前のデータまでを挿入する。挿入データのレコード数は特に制限しない。
*EDITEND	ファイル転送の後終了。	

4.4 コンソールオペレーションの場合

今のところ次の3つの場合で、いずれも //EXEC コマンドにより起動がかけられる。

(1) システムファイル DEFSF の内容表示

// EXEC, DSPDEF.

(2) システム定数の表示と変更

// EXEC, TVCONS.

なお、これらのプログラムが起動された時にオペレーターが行うべき事柄については3.5で述べた。

4.5 ファイル間データ転送システム

文字表示装置 M 345 を使うための支援ソフトウェアが作成されたことで、ファイルのメンテナンス作業はかなり簡便化された。但し、3. で述べた範囲では、作業の対象となるファイルは支援ソフトウェアの作成時に決められた特定のものである。しかし実際の作業では問題ごとにユーザーが指定する任意のファイル間でデータの転送を行うことも必要となる。更にそれらのデータを M 345 から編集したり、M 345 やラインプリンターへ出力したりできれば便利である。そこで、3.3で述べたデータ入出力用の支援プログラム CRTDSK, DEF CRT, ファイル編集用の EDITDT などを用い、更に新たなプログラムも追加してファイル間データ転送システムを作成した。その構成を Fig. 8 に示す。これらの支援プログラムは、以下に述べる1～3行の制御文によって起動される。まず、必要に応じてファイル参照番号(転送元は u7, 転送先は u8 に固定)、ファイル名、ファイルが存在するディスクの物理機番などを指定する ASSIGN 文 (// LASGN, ~.) を置き、次にプログラムモジュールをカタログ・プロシジャーで呼出す制御文 (// CALL, ~.) を置けばよい。

なお、このデータ転送システムで転送等の対象となるデータは、1レコード80バイトの順編成ファイルであって、FORTRAN 言語の READ 文, WRITE 文から参照可能なファイルの内容となっているものとする。

このデータ転送システムが作成されたことで、従来カードリーダーだけからデータ入力することを想定して作られていたバッチ処理用のユーザープログラムも、比較的簡単な修正を行えば、M 345 のキー操作によるデータの入力も行なえることになった。その一例として以前に作成した日本語終端表現生成システム³⁾の一部を次のように変更して原文データと処理結果が M 345 を通じて入出力できるようにした。すなわち、単語辞書その外の機能辞書類は予めディスク内のユー

ザーファイルに登録しておくこと、一方原文データはその都度 M 345 から別の作業ファイルへ入力すること、処理結果は別のユーザーファイルに出力すること、更にこれらのファイルは生成システム内の READ 文および WRITE 文から参照できるように必要な修正を行うこと、などである。

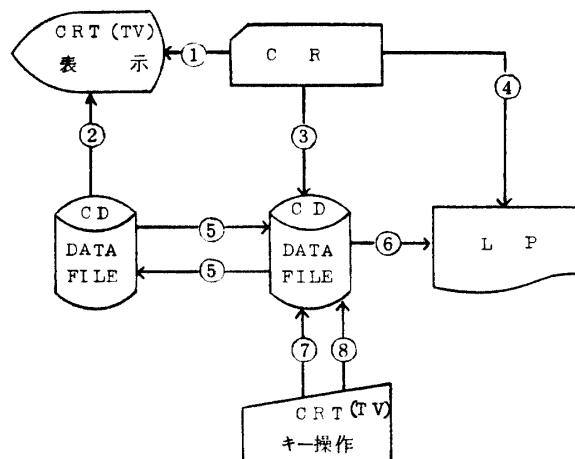


Fig. 8 The supporting program (load module) for each data transference indicated by circled figure is as follows:

- | | | |
|----------|----------|----------|
| 1 CRTOTV | 2 CDTOTV | 3 CRTOCD |
| 4 CRTOLP | 5 CDTOCD | 6 CDTOLP |
| 7 TVTOCD | 8 EDITDT | |

5. 検 討

本稿のような支援ソフトウェアを作成する際に念頭に置くべきことは、ソフトウェアの有用性、汎用性、使い易さなどであろう。使い易さのパラメーターとしてはプログラムモジュールやその中で必要な引数の個数の少なさ、引用の容易さなどが考えられる。これらの要求の中には相反する部分もあるので個別に考えることは難しい。又、ソフトウェアの作成目的と切離してはもちろん考えることができない。目的の設定の仕方によって種々の異った内容と機能を備えた支援ソフトウェアを考えることができるからである。

ところで、われわれが今回設定した目的はまず第一に会話型処理機能の整備であった。これは問題処理の過程で必要なデータあるいは処理手順の分岐指令を、その都度プログラム自身からの要求でユーザーが M 345 のキー操作で入力して処理を進行させる機能である。第二にはカードリーダーとラインプリンターだけに頼っていたデータ入出力、ファイル編集が M 345 のキー操作および CRT 画面表示によって会話的に実施できるようにすることであった。

われわれはこの度作成した支援ソフトウェアが上記の諸観点からみて適切に作られているかを検討するために実際に種々のテストプログラムを作り、又、4.5に述べたような日本語終端表現生成システムの修正を行い、M345を使って動かしてみた。それらに基づく検討の結果を1.で分類した4つの機能別に簡単に述べる。

- (1) 会話型処理機能に関しては、数値の出力方法に工夫の余地を残すが、モジュールの設定の仕方やその引用の仕方、引数の個数などを含め、かなり適切なものができていると言えよう。ただ数値出力の場合、数値から数字列の変換を現在ユーザープログラムの中で行うことにしているが、これは数値出力用支援プログラム内に埋込んで、ユーザープログラムの負担を軽減できる。
- (2) データの入出力に関しては、出力の場合に問題を残している。すなわち、現在は出力留保用の特定ファイル DEF の内容を無条件に全部 CRT へ表示する。これは必要に応じて選択表示できるようにすれば画面を更に有効に利用できることになる。
- (3) ファイルの編集に関しては、ソースデータに限ればほぼ十分な作業ができる。しかし、ソースプログラムの編集の場合、その結果を更にオペレーティングシステムの言語処理プログラムに渡すための準備作業も必要となる。それはメーカー提供のディスクエディターが管理するファイルラベル⁴⁾を参照しその一部を編集の結果に応じて書き換えてやることである。今のところこの部分のプログラムを作っていないので、ファイル編集についてはデータファイルだけが対象となりうる。
- (4) コンソールオペレーションに関しては、今のところ、タイプライター又はカードリーダーの補助としての役割しか M345 は果していない。それはプログラムの起動がかけられないからである。われわれとしては M345 でシステム操作のすべてを行わせるつもりはないが、プログラムの起動とそれに伴うシステム定数の設定や変更などの作業はもう少し自由にできるようにしたいと考えている。

6. む す び

M345文字表示装置支援ソフトウェアが作られたこと
Vol. 30 No. 1 (1979)

とにより M345 の使い易さは飛躍的に増大した。すなわち、会話型処理を支援する24本のオブジェクトモジュールプログラムの任意のものが、それぞれに決められた5個以内の引数を指定するだけで FORTRAN 言語のプログラムから CALL 文によって自動呼出しできるようになった。但し、この場合 M345 を通じて入出力されるデータは一つの変数または配列の中のものだけである。これだけでもかなり多様なプログラミングが可能である。しかし、同時に多種類のデータの入出力ができるならば更にプログラミングの多様性は増すことになる。われわれはこれをディスク上のファイル経由で行うこととし、M345 とファイルとのデータの受渡しを支援するロードモジュールのプログラムも作成した。これにより、実質的に FORTRAN 言語の READ 文と WRITE 文で M345 からのデータ入出力ができることとなった。このように、2.で述べた基本ソフトウェアだけのときに比べ、M345 ははるかに使い易くなったと言える。しかし、5.で指摘したようにいくつかの改善の余地も残っている。今後はこれらの点を改善し更には M345 を有効に利用した実用システムの開発も試みたいと考えている。

参 考 文 献

- 1) 三菱電機：M345 回線制御プログラム(LCP345) 説明書、技術ノート、DCSNF-No. 1001 (1974)
- 2) 三菱電機：MEL.COM9100 TSOS バッチオペレーティングシステム説明書、FM-SR00-07A <471D0> (1974)
- 3) 石原、田町：日本語終端表現生成システムの作成と実験、信学論、J61-D, No. 9 (1978)
- 4) 三菱電機：MEL.COM9100 ディスクエディター説明書、FM-SK00-33A <47A0> (1974)

付 録

ここには、3.で述べた考え方に従って実際に作成した支援プログラムの一覧表を、3.で行った機能分類に準拠して示す。なお、Table 3~9 の各プログラムはサブルーチン形式のオブジェクトモジュールとして外部記憶装置に登録されている。そして FORTRAN 言語のプログラムからは CALL 文でプログラム名と引数を指定するだけで自動呼出しされるようになっている。これに対して Table 10 はロードモジュールのプログラムであって、制御文 // EXEC、又は、// CALL、で呼出される。

Table 3 Code conversion

プログラム番号	プログラム名	機能
1-1	ETACTW	EBCDIK コードを ISO コードに変換する。
1-2	ATECTW	ISO コードを EBCDIK コードに変換する。

Table 4 Display operation in conversational processing

プログラム番号	プログラム名	機能
2-1	CRTOUX (N1, N2, N3, N4, N5)	1行分80文字までカナ文字も含め表示可能。
2-2	WOCRTX (N1, N2, N3, N4, N5)	4行分320文字までカナ文字も含め表示可能。
2-3	WOCRT (N1, N2, N3, N4, N5)	8行分640文字まで表示可能。ただし、カナ文字不可。

Table 5 Input operation in conversational processing

プログラム番号	プログラム名	機能
3-1	CRTIN (N1, N2, N3, N4, N5)	文字データの入力。77文字以内。ただし、カナ文字の入力不可。
3-2	CRTINX (N1, N2, N3, N4, N5)	文字データの入力。77文字以内。ただし、カナ文字の入力も可能。
3-3	INSUJI (N1, N2, N3, N4, N5)	数字打ち込み用。76文字以内の数字、小数点、+、-符号の入力ができる。
3-4	NSDATA (N1, N2, N3, N4, N5)	10ケタの数値が倍精度整数変数 N4、倍精度実変数 N5 に入力される。
3-5	NLDATA (N1, N2, N3, N4, N5)	14ケタの数値が倍精度整数変数 N4、倍精度実変数 N5 に入力される。

Table 6 Light-pen manipulation in conversational processing

プログラム番号	プログラム名	機能
4-1	PENFRM (IBUF) IBUF: 40ワード	キー操作でライトペンディテクタブルフィールド作成用の生データを IBUF に入力し、必要なコード変換を行った後改めて IBUF にもどす。
4-2	PENCR (IBUF) IBUF: 40ワード	データカードからフィールド作成用の生データを IBUF に入力し、必要なコード変換を行った後改めて IBUF にもどす。
4-3	PENASC (IBUF) IBUF: 40ワード	ENCODE 文でフィールド作成用の生データを IBUF に入力し、必要なコード変換を行った後改めて IBUF にもどす。
4-4	PENCON (N1, N2, IBUF, M)	IBUF の内容を使ってライトペンディテクタブルフィールドを表示する。又、オペレータが選択したメニューを左からの順序数で M にセット。
4-5	CRTIF2 (N1, N2, B1, B2)	B1, B2 は任意の文番号に "S" をつけたもの。2分岐のライトペンディテクタブルフィールドが表示される。
4-6	CRTIF6 (N1, N2, B1~B6)	B1~B6 は任意の文番号に "S" をつけたもの。6分岐のライトペンディテクタブルフィールドが表示される。

Table 7 CRT-frame edition in conversational processing

プログラム番号	プログラム名	機能
5-1	CLEAR0	CRT の全画面を消去し、初期状態にもどす。
5-2	CLEAR1 (N1, N2, N3, N4, N5)	CRT 画面の部分 N5 (消去コードで4種類ある) に従って消去する。
5-3	UPDOWN (N1, N2, N3)	CRT 画面の巻上げ、巻下げを行う。 N3=2HUP, 又は, 2HDW.
5-4	LINECN	ライトペン及びキー操作によって、画面の巻上げ巻下げを行う。

Table 8 CRT-frame holding in conversational processing

プログラム番号	プログラム名	機能
6-1	WAITIO (N1)	N1 秒間文字表示装置の画面は保持される (N1 は $2^{15}-1$ 秒以内)
6-2	TVWAIT	トランスミットキーが押下されるまで画面は保持される。ただし約9分をこえる時は、保障されない。

Table 9 Miscellaneous in conversational processing

プログラム番号	プログラム名	機能
7-1	DEVOUT(IBUF,N) IBUF: 40ワード	IBUF の内容を N で指定するデバイスに表示する。 N=2HTV, 2HLP 又は 2HTW
7-2	TABSET(N1, N2)	N1(行), N2(列)で指定する位置にタブコードを入れる。
7-3	DELETE(N1, N2, N3)	N1(行), N2(列)で指定される位置からブロックの終り又は、行 N1 の終りまでの間で N3 文字分削除する。
7-4	INSERT(N1, N2, N3)	N1(行), N2(列)で指定される位置からブロックの終り又は、画面の終りまでの範囲で空白を挿入する。

Table 10 Load-Module programs for data-input, data-output, File-Editing and Console-Operation

プログラム番号	プログラム名	機能
8-1	CRTDSK	キー操作によって、DATAFAに1レコード当り80バイトの順編成ファイルを作成する。
8-2	DEF CRT	ソースリスト及び結果を CRT 画面に表示する。画面の切替、終了はライトペンで指示する。
8-3	HISDEF	ソースリスト及び結果を CRT 画面に表示する。DEF CRT より高速度で表示し、自動的に巻上げていく。
8-4	EDITDT	任意のファイルの表示、編集、他のファイルへの転送を行う。
8-5	DEF DSP	システムファイル DEFSF の内容を CRT 画面上に表示する。
8-6	PACRED	ディスク・ダンプの操作を CRT 画面上で行う。
8-7	TVCONS	一部システム定数の表示及び変更を行う。LP 枚数、ジョブ名チェック桁、処理時間制限。

(昭和 54 年 4 月 16 日 受理)