解　説

汎用キャラクタディスプレイのための画面エディタ†

田端哲彦††　杉山紀子†††

0. まえがき

キャラクタディスプレイ装置がコンピュータとの対話に使われるようになったのは，1960年代後半である．当時のディスプレイは，単にタイプライタ型端末の紙の部分が画面になった感じで，データの中に改行記号やタブ記号を挿入することで，文字の表示位置を制御していた．1971年に，画面を複数のフィールドに分割し，それぞれに輝度や入力可否といった属性を与えることのできる新しい概念のディスプレイ（IBM 3270）が発表されたが，漢字やカラーを含めて現在商用大型システムの端末として使われているキャラクタディスプレイには，この概念を引き継いだものがかなり見受けられる．この種の端末はクラスタコントローラと呼ばれるマイクロプロセッサとバッファを備えた制御装置に接続され，さらに入出力チャネルや通信回線を介してホストプロセッサと交信する．このディスプレイの大きな特徴は，従来はエディタの仕事であった文字の追加/削除が，ホストシステムの助けを借りることなく端末側で行えることである（ローカルエディット）．当初のエディタは，いわゆるラインモードのみであったが，その後，全画面モードのエディタが開発され，使用者の使いやすさを大幅に改善した．

現在，この種のエディタはごく一般的になり，かつ高級なものも現われているが，本稿では10年前に開発され現在もDPプログラム開発に使われているSPFのエディタについてとりあげる．

1. SPF

1.1 SPFとSPFエディタ

'SPF'という言葉からエディタを連想する人が多いが，厳密には'SPF'は大型タイムシェアリングシステムを使って効率良くプログラム開発を行うことを目的とした開発支援パッケージの総称で，(1) 3270系端末を活用してCOBOLやFORTRANといった汎用プログラムを作成/編集するためのエディタ，(2)その過程でよく使われる種々のユーティリティプログラム，(3)それらを最少のOSまわりの知識で動かせるようにするための対話管理機能の3つで構成されている．SPFは1975年にTSO-3270 Structured Programming Facility (SPF) として発表された．SPFの持っている大きな特徴のひとつは，IBM 3270のハードウェア機能を最大限に利用するように設計されている点である．その中には全画面サポート機能やプログラム機能キー（PFキー）の使用などがある．もうひとつの大きな特徴はメニュ方式を採用した点である．今でこそメニュ方式はディスプレイを使用したアプリケーションでは，ごく一般的なやり方となっているが，当時（1975年）の環境の中でこの方式を取り入れたことは，より良い使いやすさを追求したSPFの大きな特徴と言えるだろう．以下にSPFの特徴として一般的にあげられる項目を列挙しておく．

(1) 使用上の容易性を充分考慮した機能の提供（メニュ方式，PFキー，全画面，画面分割，豊富なユーティリティ機能，スクローリングなど）．
(2) 使用者の作業時間の短縮
(3) プロジェクトコントロールのための機能提供
(4) オンラインでの個別指導（テュートリアル）
(5) 構造化プログラミング技法の採用（エディタ）
(6) 言語処理プログラムとのインタフェース提供

SPFはこれらの特徴をもちながら，基本オプションとして図-1に示すメニュによって10種の機能を提供している．

したがって，以下本稿で単に'SPF'と言う場合は上記10種類の機能及び使用者との対話を司どる対話管理機能，ならびにそれらすべてを制御する機能を含めた全体を指し，その中のエディタ部分について論ず

† Screen Editor for Character Display Terminals by Tetsuhiko TABATA (Technical Relations-Software, IBM Japan Ltd.) and Noriko SUGIYAMA (Systems Engineering, IBM Japan Ltd.).

†† 日本アイ・ビー・エム(株)ソフトウェア技術渉外
††† 日本アイ・ビー・エム(株)システムズ・エンジニアリング

```
---------------------- ISPF/PDF PRIMARY OPTION MENU ------------------------
OPTION  ===> _
                                                          USERID   - FLAG3
  0  ISPF PARMS  - Specify terminal and user parameters    TIME     - 12:47
  1  BROWSE      - Display source data or output listings  TERMINAL - 3278
  2  EDIT        - Create or change source data            PF KEYS  - 12
  3  UTILITIES   - Perform utility functions
  4  FOREGROUND  - Invoke language processors in foreground
  5  BATCH       - Submit job for language processing
  6  COMMAND     - Enter TSO command or CLIST
  7  DIALOG TEST - Perform dialog testing
  C  CHANGES     - Display summary of changes for this release
  T  TUTORIAL    - Display information about ISPF/PDF
  X  EXIT        - Terminate ISPF using list/log defaults

Enter END command to terminate ISPF.
```

図-1 SPF のメニュ画面

る時は 'SPF エディタ' あるいは 'エディタ' として区別している.

1.2 プログラム開発環境でのエディタの役割

ここで少しプログラム開発に目をむけてみよう. (SPF エディタはプログラム開発時の編集作業に充分対応できる必要がある.) プログラム開発は設計, コーディング, 単体テスト, 統合テスト, システムテストなどいくつかの段階に分ける事ができ各段階ごとに様々な作業項目がある.

一般的にも言われていることであるが, プログラム開発で最も多くの作業量を必要とするのがソースプログラムの作成, 修正, コンパイルの繰返し部分である. 逆に言えばこの部分—ほとんどがエディタにかかわる部分—に着目してオペレーションなどの効率化をはかれば, 大きな効果が得られることになる. SPF エディタは, これらの作業時にあがってくる要求をみたし, しかも効率良く処理しなければならない.

ソースの作成, 修正, コンパイル, 実行等の作業であがってくる要求項目で, とくにエディタにかかわりそうなものは次のようなものである.

- 一度に多くのデータを見ながら修正したい
- 効率よく大量データを扱いたい
- ソース全体のどの部分でも簡単に表示したい
- 一度に2つのファイルを見ながら編集したい
- 打鍵の数を出来るだけ減らしたい
- ソース内のレコードを簡単に追加/削除/繰返し/移動/コピーできるように
- 別ファイルのデータを取り込みたい
- ソースデータの特定個所を簡単に探したい

このようにエディタに対する要求はきわめて多く, しかも, これらは使い易い形で満たされなければならない. SPF エディタがこれらの要求をどのような形で実現しているかは, 次章で主なものを個々の機能ごとに説明する.

2. SPF エディタの機能と操作

2.1 機 能

SPF エディタは1～255バイトまでの長さの固定長/可変長レコードを取り扱うことができる.

3270系の端末は端末側に画面バッファを持ち, さらにそのバッファ上で一部の編集機能を実行するためのキー (挿入, 削除など) を持ったキーボードを備えている.

SPF エディタの画面は図-2 のようになっており, ファイル情報表示域, コマンドを入力するためのコマンド入力行, スクロール量設定域, メッセージ域, 行番号表示域およびデータ表示域より構成される.

ディスク上のデータを編集する場合, まず SPF はデータを全部仮想記憶装置上の作業域に読みこむ. これは勿論パフォーマンスを考慮したものであるが逆に編集できるデータの最大量は仮想記憶域の大きさにより決定される. (1メガバイトあたり80桁カードにして約1万枚となり実用上は支障はない.)

使用者は最初に必要な情報を画面に表示した後, カーソルを動かしてローカルエディットの機能により画面上での変更を行う. 画面上での修正結果はこの時点では端末側の画面バッファにのみ反映されていて, ENTER キーを押すことではじめてホストシステムに転送される. これによって変更されるのは仮想記憶装置上にある作業域だけで, ディスクへの書込は使用

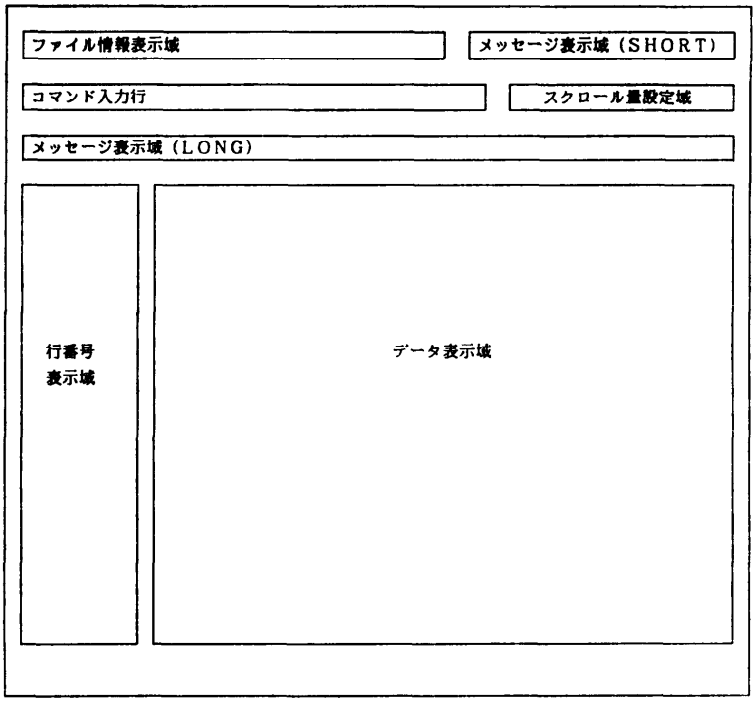

図-2　エディット画面の構成

者が SAVE コマンドを実行するか，編集作業を正常に終了させる (END) まで行われない．

したがって使用者が CANCEL コマンドにより編集作業を中止すれば，ディスク上のデータは作業開始前の状態で保存される．この方式では編集作業中にホストシステムがダウンすると，それまでの結果を失ってしまうので，エディタとのやりとりをディスクに記録しておいて，エディタ再始動後に自動的にダウン直前の状態へもどす機能がオプションとして提供されている．ただし，3270 系端末の構造上の制約からホストへ送信していなかった端末側画面バッファの中身については回復不可能である．以上が SPF エディタの基本的な動きであるが，次に，この動きを踏まえたうえで，SPF エディタのもつ主な機能について詳しく説明する．まず SPF 自身が持っていてエディタにとってもきわめて重要な機能を 3 つ紹介する．

（1）　コマンド入力

SPF，とくにエディタには様々なコマンドが用意されているが，このコマンドの実行には画面上のコマンド入力行にコマンドを打鍵して ENTER キーを押す方法と，コマンドが割当てられている PF キーを押す方法がある．PF キーを利用する方法は打鍵の数を減らすことによって時間の短縮をはかれると同時に打鍵のミスを減らせる利点がある．

PA1	PA2
TSO ATTENTION	RESHOW SCREEN

PF1	PF2	PF3
HELP	SPLIT SCREEN	END
PF4	PF5	PF6
PRINT OR RETURN	REPEAT FIND	REPEAT CHANGE
PF7 ↑ SCROLL	PF8 ↓ SCROLL	PF9 SWAP SCREEN
PF10 ← SCROLL	PF11 → SCROLL	PF12 CURSOR OR RETURN

図-3　標準的な PF キーの割当て

標準的な PF キーの割当ては図-3 のようになっている．

（2）　スクローリング

使用者が画面サイズを越えた情報を扱う場合には必ず使用する，エディタにとっては欠かせない機能で，

情報に対する画面上への'ウインドウ'を自由にずらすことができる．SPF では上下左右の4方向へ画面1ページ単位，行数・桁数単位などスクローリングの振幅を自由に設定して動かすことができる．

しかも，これらの機能は（1)で示したように PF キーにセットされているため，操作が非常に簡単である．

（3） 画 面 分 割

プログラム開発時には一人で同時に複数のファイルを見たり，結果を見ながらソースを修正したりすることが非常に多くなる．これを一台のディスプレイで実現したものが画面分割である．最近のディスプレイにはハードウェアにこの機能を持たせたものもあるが，SPF ではソフトウェアの機能として実現しており，物理的にひとつのスクリーンを上下に区切って2つの論理画面を持つことができる．どの大きさで分割するかは自由に決めることができ，1つ1つの論理画面はまったく別々に動かせるのでコンパイル結果のリストを見ながらソースプログラムを修正するようなことも自由にできる．ただし PF キーの操作はカーソルがある方の論理画面に対して有効となる．

画面分割の操作は SPLIT コマンドで行うが，これは PF2 に割当てられており，PF2 を押すと現在カーソルがある位置で画面が2つに区切られる．

論理画面間でのカーソルの移動は通常のカーソル移動キーでも行えるが，一動作ですむように PF9 にカーソルの交換機能（SWAP）がセットされている(図-4)．

これら SPF のもつ基本的な機能に加えてエディタ自身がそのオペレーションの効率化をはかるため次に述べるようないくつかの機能を持っている．

（4） エディタのコマンド

エディタにおいては使用者が打鍵するソースデータ以外の部分はすべてコマンドで制御している．コマンドには2つのタイプがあり，ひとつはプライマリコマンドと呼ばれる．これらはエディット画面上のコマンド入力行で実行され，現在編集の対象となっているファイル全体に対して影響を及ぼすような処理のために使われる．その例としてはシークエンス番号の付加(NUMBER)，大文字への自動変換 (CAPS)，データの十六進表示 (HEX)，文字列の検索 (FIND) や変更 (CHANGE)，ソースデータの保存 (SAVE)，ジョブの OS への投入 (SUBMIT) などがあげられる．もうひとつのタイプはラインコマンドと呼ばれ画面上の行（レコード単位＝SPF のエディタでは画面上の1行が1レコードとなっている）単位に制御したい場合に使われ，画面上の行番号の表示部分にオーバタイプすることで実行することができる．行の追加/削除/繰返

```
BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------ LINE 000000 COL 001 080
COMMAND ===>                                              SCROLL ===> PAGE
********************************** TOP OF DATA ************************-CAPS ON-**
COINS:                                                                  00010001
  PROCEDURE OPTIONS (MAIN);                                             00020000
    DECLARE                                                             00030000
      COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),                    00040000
. . . . . .
--------------------- ISPF/PDF PRIMARY OPTION MENU --------------------------
OPTION  ===> _
                                                        USERID    - BECKT
   0  ISPF PARMS  - Specify terminal and user parameters  TIME      - 12:47
   1  BROWSE      - Display source data or output listings TERMINAL - 3278
   2  EDIT        - Create or change source data          PF KEYS   - 12
   3  UTILITIES   - Perform utility functions
   4  FOREGROUND  - Invoke language processors in foreground
   5  BATCH       - Submit job for language processing
   6  COMMAND     - Enter TSO command or CLIST
   7  DIALOG TEST - Perform dialog testing
   C  CHANGES     - Display summary of changes for this release
   T  TUTORIAL    - Display information about ISPF/PDF
   X  EXIT        - Terminate ISPF using list/log defaults

Enter END command to terminate ISPF.
```

図-4 画面分割

し (I, D, R), 行 (あるいは複数行) の移動/複写 (M, MM, C, CC) などがよく使われるものである.

コマンドの詳細についてはここでは割愛するが, 豊富なコマンド群が SPF エディタを使いやすくするうえで重要な役割を果たしている.

(5) エディタのモード

エディタを使って作成したり修正したりするデータにはソースプログラムやテスト用データ, ドキュメントといった具合にいろいろな種類があり, それぞれシークエンス番号がついていたり, 二進データであったりあるいは大文字と小文字が混在しているといったように特徴を持っている. 同じシークエンス番号をつける場合でも COBOL のプログラムとアセンブラのプログラムではその場所が異なる. このようにデータの持っている特徴をエディタでうまく取り扱えるようにしたのがエディット・モードである.

現在の SPF では, NUMBER, CAPS, RECOVERY, TABS といった 9 種類のモードがありデータのタイプによって自動的に ON/OFF がセットできるほか, コマンドで変更することもできる.

(6) エディット・プロファイル

サブシステムやアプリケーションである種の情報プールとしてプロファイルをもつことは今や常識ともなっているが, SPF エディタはこの考え方を初期の段階から採り入れている. 前述のモードや使用者が打鍵したりセットした様々な情報をプロファイルに持つことによって異なったセッション間での同期がとれ, 操作をより簡単にしている.

プロファイルにはエディタ全体に対する情報やデータタイプごとのモードの情報などが細かく記録されており, 使用者の手をわずらわせることなくスムーズなモードの切り換えが可能となっている.

2.2 編集作業の具体例

SPF のエディタには今まで述べてきたように様々な機能が用意されているが, これらを実際のプログラム作成, データ作成などの編集処理の場合にあてはめて具体的にみてみよう.

使用者がなにかデータを編集したい場合, まず SPF の主メニュで 2 を打鍵してエディット機能を選択する (図-1 参照). そして次の画面でファイルの名前など必要な情報を指定して自分が編集したいソースプログラムやデータを呼び出す. 画面上には呼び出されたデータの先頭の部分から一画面分が表示されている. (新しいファイルを作成する場合には SPF は空のデータを表示する.) 画面の左端にはエディタが使用する行番号 (6 桁) が表示される (図-5 参照). この状態でデータのある部分を修正したい場合には, その部分にカーソルを持っていってオーバタイプすればよい.

次のページに移りたいときや, 左右に隠れているデータを見たい時にはスクロールの機能を使う. スクロールの操作は PF キーを押すだけで簡単にでき, また 1 回の移動量も自由に指定できる. (画面右上のスクロール量に 15 と指定すれば, 1 回の操作で上下/左右いずれかの方向に 15 行または 15 桁画面を動かすこ

```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ---------------- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> HALF
000700           IF  ERROR-FLAG  THEN
CC                 DO;
000900               PUT FILE(SYSPRINT)
001000                   EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001100                       (COLUMN(21),A(34));
001200               PUT FILE(SYSPRINT) SKIP(2)
001300                   DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001400               ERROR-FLAG = '0'B;
CC                 END;
001600           ELSE
001700             ;
001800         END;
A_           ELSE
002000         ;
002100     END;
002200   END TESTDIR;
****** ***************************** BOTTOM OF DATA ******************************
```

図-5 COPY コマンドの使用例(1)　

```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ---------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> HALF
000700          IF  ERROR-FLAG  THEN
000800            DO;
000900              PUT FILE(SYSPRINT)
001000                  EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001100                      (COLUMN(21),A(34));
001200              PUT FILE(SYSPRINT) SKIP(2)
001300                  DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001400              ERROR-FLAG = '0'B;
001500            END;
001600          ELSE
001700            ;
001800        END;
001900      ELSE
001910            DO;
001920              PUT FILE(SYSPRINT)
001930                  EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001940                      (COLUMN(21),A(34));
001950              PUT FILE(SYSPRINT) SKIP(2)
001960                  DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001970              ERROR-FLAG = '0'B;
001980            END;
****** ***************************** BOTTOM OF DATA ********************************
```

図-6 COPY コマンドの使用例(2) ©IBM Corporation, 1982

とができる．図-5 右上のHALF は画面サイズの半分を意味する．)

画面上の途中にレコードを追加したい場合にはラインコマンドの I (INSERT) を使う．例えば図-5 で行番号 002100 のつぎに 2 行追加する時はその行番号のところに I2 とオーバタイプして ENTER キーを押せば，002100 の直後に 2 行の空白行が表示され，そこにデータを自由に入力することができる．さらに，レコード(行)単位の移動/複写/繰返し/削除などもラインコマンドを使って自由に行える．例えば図-5 において 000800 から 001500 までの 8 行をまとめて 001900 の後ろへ複写したい場合には 000800 と 001500 の上に CC をオーバタイプし，001900 のところに A (AFTER を示す) を打鍵して実行すれば CC で囲まれた部分が A の後ろに複写される (**図-6**)．移動/複写/繰返し/削除のラインコマンド (M/C/R/D) は，それぞれ 1 行を対象にしても実行できるが，上の例で示したように複数行をまとめて処理するブロック操作も可能である．これは特にプログラムを編集する場合，使用者の作業量を軽減するのにきわめて有効な手段となっている．

同じように複数行をまとめて扱う場合でも，D コマンドのように画面上とびとびに削除したい行がある時には各行に対して D をオーバタイプしておいて一度にまとめて実行することもできる．

ソースデータを編集する場合に前述のコマンドとともによく使われるのが，FIND と CHANGE コマンドである．一般に編集作業は全データが一画面におさまることは少なく，大きなファイルのごく一部分を画面に表示して作業している場合がほとんどである．このような時に，自分が見つけたいデータストリングを最初から一画面ずつ探していかなければならないとしたら大変な話である．

これを簡単に出来るようにしたのが FIND コマンドで，画面上のコマンド入力行に探したいデータストリング (文字データでも十六進データでも可能) を指定してこのコマンドを実行すると，エディタがファイルの中の該当個所を探して画面に表示してくれる．

また，この作業を繰返して実施できるよう PF キーがセットされているので，何度も同じコマンドを打鍵することなくファイル全体にわたって，データストリングを探すことが可能である．CHANGE コマンドで特定のデータストリングの内容を変更する場合にも同様なことがあてはまる．勿論その部分にカーソルをもっていってオーバタイプすることもできるが，同じ修正をファイル全体にわたって行う場合には，CHANGE コマンドと PF キーを活用する方がはるかに簡単である．

このほかにも操作上有効なコマンドがいろいろあるが紙面の都合で割愛させていただく.

3. むすび

SPF は1975年以来，サポート機能の拡張，ハードウェアサポート範囲の拡大，新しい機能の追加等を重ね，現在では ISPF (Interactive System Productivity Facility) と ISPF/PDF (ISPF/Program Development Facility) という形で，名前も内容も単なるプログラム開発ツールから，システムの生産性向上ツールへと大きく変化してきている．特に対話式アプリケーションの作成，実行のためのサービス機能の提供 (Dialog Manager) はきわめて注目すべき点である.

今回は SPF のエディタの部分について解説してきたが，10年前に作られたひとつの古典的(?)エディタとして理解していただければ幸いである.

ここ数年間にエディタの世界は急速な進歩をとげてきた．最近のいろいろなエディタをみるにつけ，SPF エディタにも新しい時代に対応した大幅な飛躍を期待してやまない．SPF をより詳細に知りたい方は文献1)を参照されたい.

参考文献

1) Interactive System Productivity Facility/Program Development Facility for MVS: Program Reference (SC34-2089), IBM Corporation, N. Y. (1982).

(昭和59年4月10日受付)