

複数画面プログラミング環境における 画面エディタ MDED

真野 芳久 大崎 和仁 鳥居 宏次
(電子技術総合研究所)

1. はじめに

複数画面プログラミング環境MDPS [1,2]は、複数個のCRTを持つ端末上に種々の支援ツールを備えることで、ユーザと計算機との間のパイプを広くし、より進んだ会話型のプログラミング環境を構成しようとするものである。複数個のCRTの存在によって、机上での作業と同様に、様々な対比・参照・利用が容易に行なえることが期待できる。これらの動作は基本的なものであるが、MDPSの開発はこれらがプログラミング時に重要な役割を果たしているという認識に基づいている。また、プログラミングという困難な作業を支援するためには、ハードウェア、ソフトウェア両面からの総合的な支援環境を作ることが重要であるとの認識にも基づいている。

会話型のプログラミング環境下では、計算機使用の多くはエディタの使用であり、エディタの機能・使い易さは、プログラミング言語やファイルシステムとともに、重要な要素である。

tty向きのエディタ(QED, TECO, SOS等)では、低い通信速度の克服のために、コマンドの簡潔表現、正規表現記述による強力な文字列探索、プログラミング言語としての機能、等を持つこととなった。

CRT端末、高速の通信回線が使える状況では画面エディタが可能である(EMACS, vi, KED等)。パーソナル計算機やインテリジェンスを持つ端末内でのローカルなエディタでは、表示速度が通信速度で抑えられないので、画面エディタ向きと言える(PERQエディタ等)。編集地味もす早く示すための

指示装置(専用キー、ライトペン、マウス、ジョイスティック等)も考案されてきた。

画面の概念の発達として、画面の分割、ウィンドウの概念が生じ、種々の独立した内容を持つウィンドウの重ね合せによる2.5次元的画面環境が作成された[3]。ただし bitmap型の高解像度ディスプレイと高速のスクリーン機構を持つことが必要である。

編集対象を特にプログラムに絞って、(構造的に)正しいプログラムの容易な作成や、言語構造に即した編集手段を与える言語指向型エディタ(MENTOR, Cornell program synthesizer等)の開発も盛んに行なわれている。

本稿で述べる複数画面向きエディタMDEDは、複数画面を持ちマイクロ計算機でもある特殊な装置・システムを前提として、広い対話中、迅速な応答を活用した効率良い編集作業を目的として設計開発されたエディタである。

なお、MDPS下では、MDPSインタプリタ言語であるμPL[4]用の言語指向型エディタも、言語指向型エディタジェネレータ[5]を利用して作成されている。

2. MDEDにおける基本概念

MDEDはMDPSの1つのサブシステムである。本稿は、MDEDの立場から種々の概念、コマンド、操作法を述べるが、複数画面への表示方法等に関しての記述はその多くがMDPSモニタ下のものと共通である。

MDEDが編集の対象とするものはMDPSで言っているところのテキストである。テキストは同時に複数個存在してよい。実際、MDED下では後に述べるように、

1つ以上のユーザ定義テキスト、履歴テキスト、バックアップテキストが同時に存在する。

各テキストは、図形文字と改行を示す仮想的な記号(eol)とから成る列であると考えられる。ただし列の最後には必ずeolが存在する。eolはテキストの画面表示時には特別の意味を持つが、編集時には図形文字とeolは全く同等に扱われる(eolの入力はCRキーを用いる)。

各テキストは3種のポインタany, bef, curを持つ。各ポインタはテキスト中のある文字の直前を指す。ポインタpの指す位置をp位置と呼び、p位置直後の文字をp文字と呼ぶ。bef位置とcur位置の間にある文字列をbef-cur文字列と呼ぶ。

編集は主としてcurポインタを用いてなされるが、対象が文字列である場合にはbef-cur文字列として指示し或いは表示される。

どの時点でも直接編集の対象となるのは1つのテキストであり、これをカレントテキストと呼ぶ。カレントテキスト以外のテキストに対して、表示方法の変更やポインタの移動は可能であるが、混乱を避けるためテキスト内容の変更はできないようになっている。

ユーザが定義し編集の対象となる1つ以上のテキストの他に、システムが管理し更新するテキストが2つある。1つはMDPSモニタ下でも存在する履歴テキストであり、ユーザがキーボードより入力したコマンドやデータがある一定量保存される。他の1つはバックアップテキストで、削除された文字列がやはりある一定量保存される。これらのテキストもカレントテキストとして文字的探索やポインタ移動等を行なうことはできるが、システムにより更新される場合を除きテキスト内容を変更することはできない。

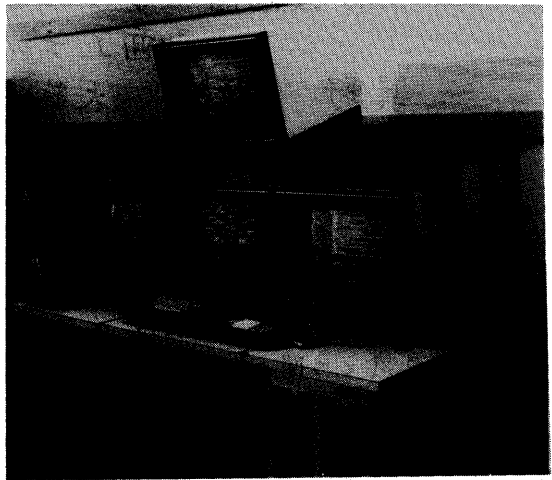
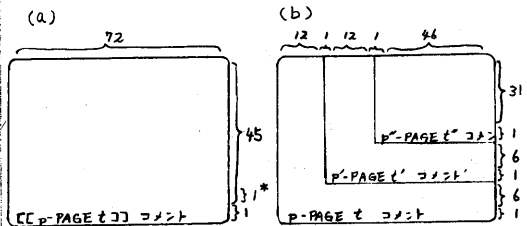


図1. MDPS システム外観

MDPS システムの外観を図1に示す。各CRTは72×46文字表示し、それらは論理的に同等の位置にあり、ユーザはそれらの使用法を自由に定めてよい。

画面上に表示されているテキスト内の部分をページと呼ぶ。ページは1つのポインタpと対応付けられp位置を含む。テキストも、ポインタpに対応するページをpページとも呼ぶ。

画面上でのページ表示方法は図2に示されるようになってきている。図2(a)は1画面1ページの場合で、最下段の1行にページ識別のための情報がある。図2(b)は、



- P: ポインタの種類 (any, bef, cur のいずれか)
- #: テキスト番号
- コメント: テキストに付けられたコメント
- [[]] : かがカレントテキストの場合 curページに対して付けられる
- *の1行: 通常はテキスト内の1行が表示されるが、ファイル名やサ行文字列のようにコメント中に現われる文字列がある場合にそれが表示される

図2. ページ表示方法

* MDED (MULTI-DISPLAY ORIENTED EDITOR) *	
CRT1 p-PAGE t	(イ)
CRT2 p-PAGE t > p-PAGE t > p-PAGE t	
CRT3	
CRT4	
UNDISPLAYED TEXT(S)	(ロ)
0 (HISTORY) 20	
FKB COMMAND: MOVE-PAGE p-t ABOVE x /	(ハ)
CURRENT TEXT: t ON CRT x	
MOVE POINTER: p	
CURRENT MODE: COMMAND MODE	(ニ)
MEMORY: aaaa	
ESC-	(ホ)

p: any, bef, cur のいずれか
t: テキスト番号
x: CRT番号
aaaa: メモリ内の番地

図3. モニタTV表示内容

数多くの情報を手近に表示し必要な情報を迅速に見つけるために導入されたページ重ね書き機能を使って表示した場合で、図に示されるサイズで3ページまで表示される。

4つのCRTへの表示状況やシステムの状態は、モニタTV上に常時表示されている。その内容は図3に示される。(イ)はCRTとページとの関係、(ロ)は表示されてはいないがシステム内に存在するテキスト、(ハ)は次節で述べるファンクションキーボード(FKB)による画面制御コマンドのモニク表現、(ニ)はカレントテキストや移動ポインタ等の情報、(ホ)は2~3文字から成るコマンドの場合の最初の1~2文字等の補助情報を表示する。

3. MDEDコマンド概要

MDEDのコマンドは大きく

- 画面表示方法の変更やテキストの生成・削除を行なう画面制御コマンド、
 - テキスト編集のための編集コマンド、
- に分けられる。

a)はFKB(0~9, A~Fの16キー)から入力され、MDPSモニタ下の画面制御コマンドとはほぼ同じである*。その要約を図4にまとめておく。

FKB入力形式	機能
1 x	新しいリストの生成
2 t E	テキストの削除
3 { A B } t { A B } x [C] E	指定ページの移動 (非表示化も含む)
4 t E	カレントテキストの指定

t: テキスト番号
x: CRT番号
[C]: 省略可能
{}: 区括
{ A B } : any/bef/curページ
{ A B } : above/below

図4. 主要な画面制御コマンド

以下ではMDED固有のコマンドであるb)について述べる。これらは更にb)-1. ファイルとテキスト間の入出力
b)-2. ポインタ移動
b)-3. テキスト内容の変更
b)-4. その他に分けられる。

これらのコマンドはキーボードから与える。その一般形式は
 \hat{N} コマンド \hat{N}
である。

\hat{N} は引数としての数であり、コマンドにより、テキストを指定するためのテキスト番号か、繰返し回数を指定するための数かのいずれかとなる。引数としての数が要求されるコマンドで、それが省略されたときは、一般に標準的な値(カレントテキスト番号又は繰返し数1)が仮定される。 \hat{N} は $\wedge N$ 10進数の形で入力される**。

コマンドは、一般に機能文字である1文字ないし3文字から成る。2~3文字から成る場合、最初の1~2文字はコマンドのタイプを表わす。

* ページの指定にポインタが必要(省略時 cur)であること、等の違いがある。

** $\wedge N$ はコントロールキーとNとを同時に押して得られる機能文字を表わす。 \wedge が前置される他の文字についても同様。

点は引数としての文字列で、一般にESCで終了する。点の入力時には、編集対象ページの下部に入力ずみの点部分が表示される。

コマンド体系を設計する際、EMACS^[6]を参考にした。特に、エディタは通常入力文字を cur 位置に挿入し直ちに表示する点、コマンドが機能文字で始まる点と同じである。挿入モードを別に設けると、編集時に大きな比率を占める挿入モード時では、入力した文字列の修正に、挿入モード時以外で使用できない一般に強力な修正方法が使用できなくなる、という理由による。コマンドが機能文字で始まることはその当然の結果である。

3.1 ファイル入出力コマンド

ファイルとしては、ホスト計算機(DECシステム20)上のファイル、MDPSのフロッピディスク(FD)上のファイルの両方を扱う。コマンドは

^T ^x ファイル名 (x=V,W,F,G)
で、これによりファイルとメモリ内のテキストとの間で入出力が起る。

3.2 ポインタ移動コマンド

ポインタを移動させることは、編集地の変更、見たい部分の変更という2つの役割がある。ポインタ移動のためのコマンドとしては、文字単位・行単位・ページサイズ単位での前進及び後退、テキスト先頭・最後尾への移動、2つのポインタ位置の交換・同一化、及び文字列探索による移動がある。

3種類のポインタ、複数のテキストが存在するので、次の原則の下でコマンドを設けた。ポインタの指定は^A, ^B, ^Cを前置すること示す。ただし移動ポインタ(初期値 cur)の場合は省略できる。移動ポインタはコマンドにより変更される。カレントテキスト以外のテキストに対しては、見たい

部分の変更が主目的であるので、文字単位及び行単位による移動は除いた。即ちこの2種のコマンドでは引数としての数は繰返し回数であり、他ではテキスト番号である。

文字列探索は、不足文字を含んでいてもよい指定された文字列を捜す。不足文字は'任意の'、'任意個の'、'次の文字以外の'及びそれらの組合せである。発見文字列の直前を cur 位置とする探索と、発見文字列を bef-cur文字列とする探索とがある。

3.3 テキスト内容変更コマンド

前述のように、MDEDは文字挿入が標準状態であるので、図形文字・CRの入力は cur 位置への図形文字・eolの挿入となる。

削除は、文字・行・文字列単位のコマンドがあり、文字単位以外の削除では削除文字列がバックアップテキストに追加されそこでの bef-cur文字列となる。

他のテキストの bef-cur文字列をカレントテキストの cur 位置に複写するコマンドがある。特にバックアップテキストからの複写は^U(undo)として用意されており、誤削除の回復や、同一テキスト内での移動・複写にも用いる。通常のエディタではシステム内に隠されているバッファの内容も、こゝでは直視しながら作業を進めることができる。

文字列の置換えコマンドでは、すべての出現を置き換えてしまうものと、毎日置換えの良否をたずねるものがある。

3.4 その他のコマンド

任意のテキストの bef-cur文字列をエディタのコマンド列と解釈して実行させる機能がある。履歴テキストにはユーザの打ったコマンドが自動的に(機能文字はある種の変換を受けて)挿入されていくので、ユーザは以前に使用したコマンド列を bef-cur文字列とすることでそれを1つ

のマクロとして利用することができる。

4. インプリメント

テキストはすべてメモリ(64KB2セット)内に置かれ、表示行のレコード単位で保持される。これはMDPSモニタ下と共通である。MDPSモニタ内のルーチンの多くは他から使用可能な形に使われている。MDEDは画面制御やテキスト管理のために、MDPS内のルーチンを利用している。

コマンド入力や編集結果の画面への表示というユーザとの対話部分と、実際の編集部分とは、機能的に独立であり、それぞれできる限り閉じていることが望ましい。MDEDではそれらをほぼ完全に独立なユーザインタフェース部分とテキスト編集部分として分離し、編集中間言語で記述された命令列を介して連結させた。編集中間言語は基本的な編集命令と簡単な制御構造を持つものである。ユーザインタフェース部分は、MDEDコマンドを編集中間言語命令列に変換し、テキスト編集部分(編集中間言語インタプリタ)を起動し、必要ならばモニタ内の画面制御ルーチンを呼び出して画面表示内容を更新する。この独立性はそれぞれの機能変更時に非常に役立った。

MDEDの内部構造、モニタとの関係の概要を図5に示す。なお、MDEDはすべてμPLで記述されている。

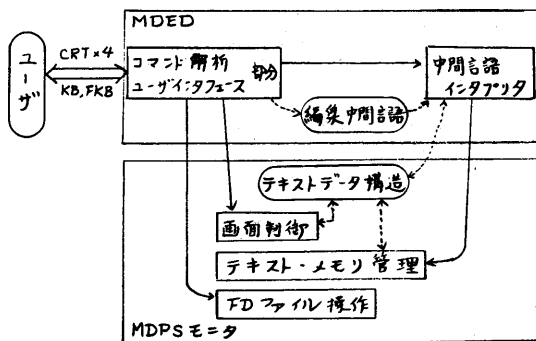


図5. MDED の内部構造

5. MDED 使用経験

MDPS は1981年初夏より、MDEDは同年晩夏より動き始めた。設計時と稼働時とのホスト計算機の違いによる問題等の解決や、初期使用経験から判明した望ましい機能の追加等を行ない、徐々に拡張・固定化されてきた。

まだ十分な使用経験はないが、現在、MDPSを使用しているプログラミング作業と、通常のCRT端末を使用しているプログラミング作業とも交互に行なっている作業能率の比較を計画している。より詳細な使用経験と評価は、その結果に基づいて別の機会に報告したいが、ここではこれまでのいくつかの使用経験からの感想を列挙する。

- i) 4画面をすべて使うことは稀であるが、2-3画面を有効に使うようになる。
- ii) ハードウェア上の欠点は使い易さにとって極めて重要である。MDPSでは、KB入力時のリポート機能欠如、FDファイルI/Oの遅さ、メモリの少なさ、等が不満の種である。
- iii) ポインタ3個をうまく使いこなすのは意外難しい。
- iv) いくつかのbef-cur文字列がスタックによって記憶されていると便利である。
- v) CRT間の物理的な間隔がやや長すぎる。目線の移動だけで隣のCRTも見れるとよい。
- vi) 全二重の対話方式は、多少の実行結果表示の遅さはカバーできる。
- vii) 位置指示装置があるとよい(ただしKBから手を離したくない)。
- viii) 語単位の編集機能は必要。

6. おわりに

プログラミング環境やエディタも評価することは難しい。[7]に見られるようにエディタも評価する試みはいくつかあるが、主としてある編集操作の所要時間や基本動作の総量の立場からで、作成されるプログラムの質をも含めた総合的な

評価はなされていない。実際的には、長期にわたり多くの人に多くのかつ大きな仕事のために積極的な理由のもとで使われることが、最良の評価結果と考えられよう。

MDPS自身は、電統研で特別に開発された複数画面端末装置上のシステムであり、また8bitプロセッサ64KBx2のメモリという現時点で見てもかなり制限の強いハードウェアをベースとしている。従って、多人数かつ長期にわたる使用は望めないが、その上での使用経験を積み明確にしておくことは、今後のプログラミング環境のあり方にいくつかの示唆を与えることになると思える。

素直ながら本研究の機会を与えていただいた石井治ソフトウェア部長、熱心に討論-批評をしていただいた言語処理研究室の皆様、インプリメントを手伝ってくださった方々に感謝する。

参考文献

- [1] 真野, 大崎, 鳥居, 複数画面知能端末を用いたプログラミング環境, 信学技報 EC80-31 (1980.7).
- [2] K.Torii, Y.Mano and K.Omaki, A new programming environment with a multi-display terminal, COMPSAC 81 (Nov. 1981).
- [3] W.Teitelman, A display oriented programmer's assistant, Xerox PARC CSL-77-3 (Mar. 1977).
- [4] Y.Mano, A programming language μ PL for multi-display micro-computers, Bul. of ETL, 45, 1-2 (Jan-Feb. 1981).
- [5] 大崎, 真野, 鳥居, 言語指向型エディタ作成システムの設計と実現, 信学技報 AL81-59 (1981.10).
- [6] R.M.Stallman, EMACS, the extensible, customizable self-documenting display editor, SIGPLAN Notices 16, 6 (June 1981).
- [7] D.W.Embley and G.Nagy, Behavioral aspects of text editors, Computing Surveys 13, 1 (Mar. 1981).

[8] Papers in the Proc. of ACM SIGPLAN SIGOA Symp. on Text Manipulation June 1981, SIGPLAN Notices 16, 6 (June 1981).

付録 主要MDEDコマンド

- ```

^A -----> ANY subcommand
^B -----> BEF subcommand
^C -----> CUR subcommand
^D (R): delete next character of cur-character
^E (T): set execute text
^H (R): move backward
^I (R): move forward
^J (R): move down
^K (R): move up
^L -----> LINE subcommand
^M (R): insert a CR
^N : number parameter
^P (T): copy a string, set 'bef' and 'cur'
^R : replace string in query fashion
^S (T): search a string, set 'bef' and 'cur'
^T -----> TEXT subcommand
^U : undo a deletion (copy from back-up text)
^V (T): display the next screen
^W (T): display the previous screen
^X (T): exchange("X^X)/change("X) pointers
^Z : cancel a command
ESC-----> ESC subcommand
character (R): insert the character
DEL(R): delete cur-character

ANY/BEF/CUR subcommand
^B (T): move to text beginning
^E (T): move to text ending
^H (R): move backward
^I (R): move forward
^J (R): move down
^K (R): move up
^L -----> LINE subcommand
^V (T): display the next screen
^W (T): display the previous screen

LINE subcommand
^B : move to line beginning
^D (R): delete a line forward
^E : move to line ending
DEL(R): delete a line backward

TEXT subcommand
^F : load a FD file
^G : save to a FD file
^S (T): number of occurrences
^V : receive a host file
^W : send a host file

ESC subcommand
^A : move-pointer becomes 'any'
^B : move-pointer becomes 'bef'
^C : move-pointer becomes 'cur'
^D : delete bef-cur-string
^E : execute commands of execute-text
^M : change MDED mode
^Q : quit MDED
^P (T): copy a string, set 'cur'
^R : replace all occurrences
^S (T): search a string, set 'cur'

(T): 元のテキスト番号
(R): 元の線通し回数 であることを示す

```