

複数画面をもつプログラミング環境 MDPS†

真野芳久^{††} 大蒔和仁^{††} 鳥居宏次^{††}

筆者らは、さまざまな情報の参照・比較・利用が、プログラミング作業の中で重要な役割を果たしていること、および計算機の通常の利用形態の下ではそれらを効率よく行うことが困難であることに注目し、それらを容易に行えるような一つの支援環境、すなわち複数画面をもつプログラミング環境 MDPS を設計・開発した。MDPS は、4 個のディスプレイ装置をもつ高度に知的な端末として、あるいは一つのマイクロ計算機システムとして使用されるものであり、諸情報の表示・積み重ね表示・消去・削除等の管理が、MDPS のもつテキスト、ページという自然な概念を用いた簡潔なコマンド群により実現されている。本論文では、複数画面をもつことの利点、MDPS の基本概念を実現するためのハードウェア、ソフトウェア両面からのシステム構成、MDPS の実際の使用形態の例について述べる。

1. はじめに

作業能率を高めてくれるプログラミング環境は、ハードウェア、ソフトウェア両面からの、人間の作業形態によく適合した支援内容をもつことが必要である。

プログラマーズ・ワークベンチ (PWB)¹⁾ は、目的計算機に依存しないプログラミング環境を設けることで作業の能率化を図ろうとしたものであり、計算機として PDP-11、OS として PWB-UNIX という組合せをプログラム開発の作業台とした。そこでは、一定の作業環境を維持できることと、UNIX というプログラム開発環境としてすぐれた OS の存在とが成功の要因である。

また、Alto²⁾ や PERQ³⁾ に代表されるような、ビットマップディスプレイ装置を中心に設計された個人向きプログラミング環境も注目に値する。ビットマップディスプレイ装置を利用したの図表を含む編集機能や、Teitelman⁴⁾ によって INTERLISP 上に実現されたようなウィンドウ機能を活用した種々の情報の参照機能は、今後ますますその利用が期待されているところである。システム全体の低廉化と使いやすいソフトウェアの存在が望まれる。

最近のハードウェア、とくにマイクロプロセッサやメモリ、の技術進歩と低廉化は、利用者の用途に応じた高度なシステムが容易かつ安価に構成されることを可能としている。プログラミングという高度に知的な作業を必要とする場においても、この可能性は大いに

追求されなければならない。

われわれは、さまざまな情報の参照・比較・利用が、プログラミング作業で重要な役割を果たしていること、および計算機の通常の利用形態の下ではそれらを効率よく行うことが困難であることに注目した。そして、さまざまな情報の参照・比較・利用を容易に行えるよう支援することによって、プログラミング作業の能率化、および作成されるプログラムの品質向上が期待されると考えた。

この種の支援のための方法としては、ウィンドウ機能の実現、迅速な画面切換えやハードコピー機能と支援ソフトウェアとの組合せ、等が考えられる。われわれは、参照・比較・利用のための作業空間を真に拡張、安価に構成することが可能な複数画面環境を検討し、その有効性を確信し、実際の場合で使用し評価することを目的として、複数画面をもつ知能端末システム MDPS (Multi-Display Programming System)^{5),6)} を設計・開発した。

MDPS は、4 個のディスプレイ装置をもつ高度に知的な端末として、また一つのマイクロ計算機システムとして使用されるもので、諸情報の表示・積み重ね表示・消去・削除等の管理が、MDPS のもつテキスト、ページという概念を用いた簡潔なコマンド群により実現されている。

複数のディスプレイ装置を活用することは、画像処理の場ではしばしば見かけられるものである。また、データベース管理システムにおいても、階層的な、あるいは詳細さの度合いに応じた参照を支援する目的で、複数ディスプレイ装置をもつシステムが実現された例⁷⁾ が報告されている。われわれの主張は、プログラミング環境としても複数画面の活用が有効なもの

† MDPS: A Programming Environment with Multi-Displays by YOSHIHISA MANO, KAZUHIRO OHMAKI and KOJI TORII (Language Processing Section, Computer Science Division, Electrotechnical Laboratory).

†† 電子技術総合研究所ソフトウェア部言語処理研究室

なろうということにある。

本論文では、MDPS の設計の基本的考え方を述べ、さらに MDPS のシステム構成、および利用形態の例を述べる。

2. 複数画面プログラミング環境でのプログラミング

本節では、複数個の表示装置をもつプログラミング環境において効率よく行うことができるいくつかのプログラミング作業形態を考えてみる。これらのいずれも単一画面プログラミング環境下で実現することは、ハードウェア、ソフトウェアの両面で困難を伴うであろう。

i) 複数の情報を参照しながらの作業

作成しつつあるプログラムの仕様書、プログラム中の宣言部分、作るべき機能と類似する機能を含む他のプログラム、等は、プログラムを作成していくうえで参照したいものである。エラーメッセージを含むコンパイルリストはプログラムを修正する際に役立つ。これらの中の参照したい部分は、機械のもつ探索機能によりすばやく見いだすことも可能であり、机の上にリストや文書などの資料を置いて作業を行う場合よりはるかにすぐれた状況を作ることができる。

ii) 複数の情報を組立て加工しながらの作業

新たに開発されるプログラムといえども、既存プログラムを利用して、機能を追加したり変更したり組み合わせたりして作ることが多い。参考となるプログラムから機械の助けを借りながら必要な部分を抜き出し組み立てる場合、複数個の情報それぞれについてある程度の量を表示できる複数画面環境はきわめて好都合である。

iii) 比較、対比を行いながらの作業

各時点の処理結果に対して比較、対比を行うことにより、作業の確認ができ、全体の見通しが明確になることがある。たとえば、テキストの修正前と修正後の様子、プログラムテキストとその実行経過を示す出力は、“過去”と“現在”，あるいは“原因”と“結果”を対比させることで作業状況・結果を明確化させる例である。また、プログラムの全体的な入れ子の様子を概要図とそのなかの一部であるテキストとの対比は、“全体”と“部分”との対比により把握しやすくなる例である。

iv) メモ帳としての画面活用

その時点での作業にただちには使わないが、いずれ役立つであろう‘ひらめき’等を、作業の中断からの影響を最小限として、別の画面上に書き加え残しておくことができる。

v) システムの状況や作業履歴を確認できる環境での作業

正常に動作している間は重要でないが、何かトラブルが生じたときには、種々の資源の使用状況等のシステムの状態やそれまでの作業履歴は、トラブルの回復や原因究明に不可欠な情報となる。わずかなタイプミスによるトラブル等はときおり経験されるところであり、この種の情報とトラブル状況との両者が表示されることは円滑なトラブル回復に役立つ。

以上、複数画面のある環境下でのプログラミング作業の形態例を見てきた。人間が机上で知的作業をする場合、種々の資料を広げ、種々の道具を活用して進められるが、複数画面環境ではそのような状況を模擬できるようにシステムを組み立てることができる。さらに、資料の整理や必要箇所の探索、作表や計算能力を機械の中にもたせることで、たんなる模擬ではなく、人間-機械系としてのすぐれた作業環境を構成することが可能である。

3. MDPS ハードウェア構成

複数画面をもつプログラミング環境下で効率よく行うことができるプログラミング作業形態について前節で述べたが、われわれはこのような環境を構成し使用・評価することを目的として、複数画面をもつプロ

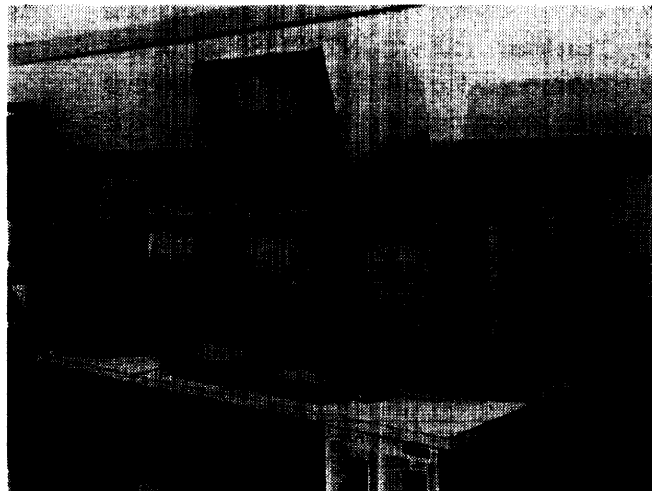


図 1 MDPS ハードウェアの外観

Fig. 1 The external appearance of MDPS hardware.

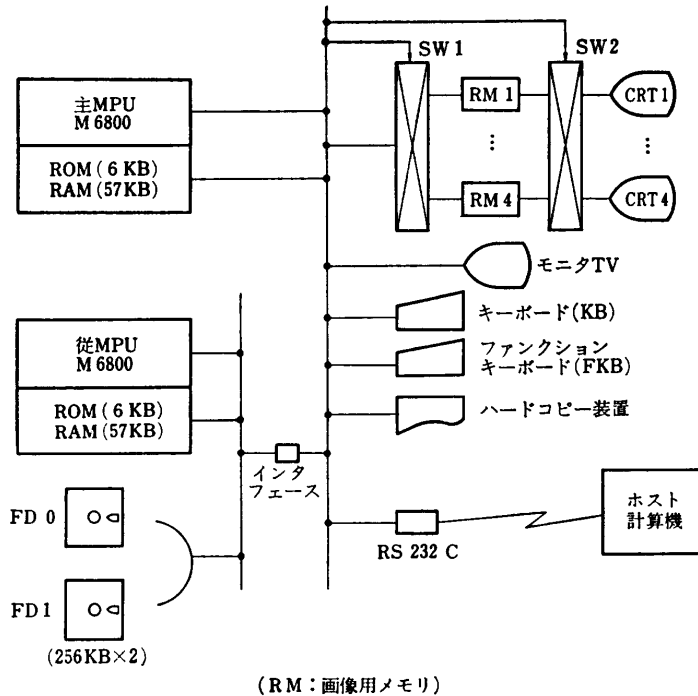


図 2 MDPS ハードウェア構成
Fig. 2 MDPS hardware block diagram.

プログラミング環境 MDPS を設計・開発した。本節では MDPS のハードウェア構成について概観する。

MDPS のハードウェアの外観は図 1 のようであり、4 台のディスプレイ装置* (中央, 右側, 左側, 上部の順に CRT 1~CRT 4 と呼ぶ) を中心に 1 台のモニター TV, キーボード (KB), ファンクションキーボード (FKB)**, 2 台のフロッピーディスクドライブ (FD), ハードコピー装置を, 利用者の使用するものとして配している。

MDPS ハードウェアの構成図を図 2 に示す。

利用者との対話, 画面制御等を行う主マイクロプロセッサ (主 MPU) と, 主して FD ファイルの I/O を行う従マイクロプロセッサ (従 MPU) の二つのモトローラ M 6800 マイクロプロセッサをもつ。4 台のディスプレイ装置は利用者のプログラミング作業の場として, モニタ TV はシステムの状態等を常時表示しておくものとして, FKB は 5 章で述べる画面制御コマンドの入力用として, 用いられる。ホスト計算機とは RS 232 C インタフェースの 9600 ボー (または 2400 ボー) で接続される。現在, DEC システム 20,

* ハードウェア的には 8 台のディスプレイ装置を接続可能であるが, 人間の管理能力の面を考慮して 4 台となっている。

** FKB は 0, ..., 9, A, ..., F と名付けられた 16 個のキースイッチから成る。

Z 8000 を CPU とする UNIX システム等をホスト計算機として使用できる。なお, SW 1 はバスラインから画面メモリ (RM) 群への, また SW 2 は RM 群から CRT 群への対応を与えるものである*。

4. 利用者から見た MDPS

MDPS は, 知的端末装置として, あるいは複数画面向きエディタ MDED 等を利用できるマイクロ計算機システムとして使用される。前者の使用形態を端末モード, 後者を MDPS コマンドモードと呼び, 任意の時点で二つのモードを自由に切り換えることができる。

利用者にとって上記両モードは, 対話の対象がホスト計算機であるか MDPS の基本ソフトウェアである MDPS モニタであるかの違いであり, 四つのディスプレイ装置を中心とした操作方法に大きな違いはない。以下の記述で単にシステムとあるのは, モードの違いによりホスト計算機または MDPS モニタを指す。

利用者は複数個の作業空間をもち, 利用者との対話内容がそのうちの一つに記録される。これらの作業空間をテキストと呼ぶ。机上作業との対比で言えば, テキストは参照したい資料, 過去の作業内容メモ, 作業結果, 等に相当する。システムとの対話を記録するために選択されたテキストをカレントテキストと呼ぶ。利用者は十分な数のテキストをもつことができ, そのうちの任意の一つをカレントテキストとして指定する。

利用者により作成・更新・消去されるテキストのほか, MDPS により管理・更新される特別なテキストである履歴テキストがある。ここには利用者が KB より打ち込んだコマンドやデータ類が自動的に保存され, 任意の時点で利用者が参照できるようになっている。

テキストの内容は, MDPS のもつ四つの CRT の任意の一つに表示させることができる。ハードコピーが CRT 1 上の表示内容に対して行われることを除いて, 各 CRT のもつ機能は同じである。テキストを

* SW 1, SW 2 ともに, ハードウェア的には多対多の対応が可能であるが, MDPS ではシステムおよび利用者から見たときの簡明さを重視し, 1 対 1 の対応に制限している。後に述べる積み重ね表示はソフト的に実現している。

どの CRT 上に表示させるか、あるいは表示させないか* は、利用者によって自由に制御される。一つの CRT に複数（ただし、たかだか3）のテキストを表示させてもよい。この場合は紙を左右・上下少しずつずらして重ねた形式で表示される。これを積み重ね表示機能と呼ぶ。

CRT は限られた大きさもち、テキストの大きさは理想的には無制限である**。テキスト内のある時点で表示可能な部分をページと呼ぶ。利用者が直接目で見ることのできるものはページであるので、MDPS システムへのコマンドのいくつかはページの概念に基づいている。カレントテキストのその時点のページを

とくにカレントページと呼ぶ。

テキストの作成・消去、テキスト-CRT 間の対応づけ、テキスト内でのページ部分の移動は、すべて FKB からの簡潔なコマンド群（画面制御コマンドと呼ぶ）によってなされる。

こうして利用者から見たときの MDPS は、図3の説明図のように示すことができる。利用者は KB から直接的な作業指示を、FKB から作業空間の設定と表示方法に関する指示を行い、カレントページを介して、ホスト計算機あるいは MDPS モニタと対話し作業を進める。

表1に画面制御コマンドの一覧を示す。テキスト/

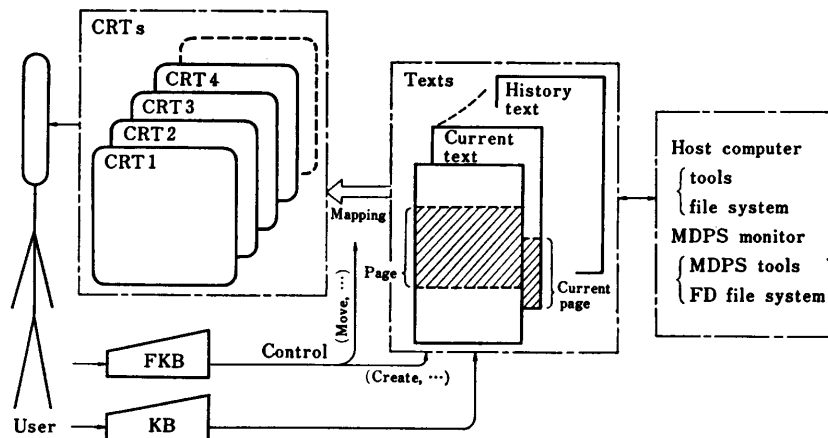


図3 利用者から見た MDPS の説明図

Fig. 3 Illustration of MDPS from the users' viewpoint.

表1 画面制御コマンド

Table 1 text/page manipulation commands.

ニモニック	コマンド形式	意味
move current page and create new current text	0 x	カレントページを CRT x 上に移し、新しいカレントテキストを作成し、それを元のカレントページのあった CRT 上に置く
create a text	1 x	新しいテキストを作成し、それを CRT x 上に置く
purge a text	2 n E	テキスト n を (完全に) 消去する
move a page	3 n $\begin{bmatrix} A \\ B \\ E \end{bmatrix} x$	ページ n を CRT x 上に移す ([A/B] x の場合)、またはページ n を非表示状態とする (E の場合)。CRT x 上にすでに他のページが表示されている場合は、'A' でそれらの上に積み重ねる形で、'B' でそれらの下にすべりこませる形で表示させることを示す。
set current page	4 n E	ページ n をカレントページとする (テキスト n はカレントテキストとなる)
set scroll page	5 n E	ページ n をスクロール (ページ枠の移動) の対象とする
scroll up	6 F 7	1行分 (6) または半ページ分 (F 6) だけページ枠を上に移す
scroll down	7 F 7	1行分 (7) または半ページ分 (F 7) だけページ枠を下に移す
send a line	8 n E	ページ n のカーソルの示す行をシステムに送る

* テキストと CRT との対応は、テキストの集合から、CRT 1~CRT 4 および表示されないテキストのための仮想的な CRT から成る集合への写像として見ることができる。

** しかし MDPS では、システムのもつメモリ容量の制約およびインプリメントの都合上、テキストの大きさを、MDED エディタを使用しているときを除き、2画面分以内と制限している。

表 2 MDPS ソフトウェア
Table 2 MDPS software.

分類	名称		稼働場所	説明
基本ソフトウェア	MDPS モニタ		主 MPU	テキスト、ページ概念の実現、MDPS コマンドモード時のコマンド解析
	ファイルシステム		従 MPU	FD ファイルへの論理的アクセス実現
ツールプログラム	端末モード	LOE	ホスト計算機 (主 MPU)	Pascal, XPL, μ PL 等の言語用の言語指向型エディタ
	MDPS コマンドモード	ファイル転送プログラム	主 MPU ホスト計算機	ホストファイル、FD ファイル間でのファイル内容の転送
		MDED	主 MPU	複数画面向きスクリーンエディタ
		ファイルアクセスプログラム	主 MPU	利用者とファイルシステムとのインタフェース
μ PL		ホスト計算機	MDPS システム記述言語	
ライブラリ	テキスト、FD ファイル管理ルーチン群		主 MPU 従 MPU	MDPS モニタ、ファイルシステム中の多くのルーチンが利用可能

ページの識別は、テキストが新たに作成された時点で MDPS によって与えられる番号が使用される。履歴テキストは 0、利用者によって作成・管理されるテキストは 1 以上 20 未満の数、MDPS エディタ内で MDED によって管理されるバックアップテキストは 20、となっている。表 1 中の n はその識別番号であり、 x は CRT 番号 (1~4) である。

5. MDPS ソフトウェア

MDPS のもつソフトウェアは、表 2 に示されるように、基本ソフトウェアと種々のツールプログラムとから成る。基本ソフトウェアは MDPS の核となる部分で、テキスト/ページ概念やファイルシステムを実現するものである。ツールプログラムは MDPS の使用をより効果的とするためのもので、そのおもなものは本節で説明される。これらの多くは μ PL⁸⁾ で記述されている。

5.1 基本ソフトウェア

i) MDPS モニタ

MDPS モニタは主プロセッサ上で走り、テキストの内容をメモリ中に保持し、その作成・消去・表示等を管理する。前節で示した画面制御コマンドを受け付け必要な処理を行う。テキストはすべてメモリ中であり*、高速表示可能なディスプレイ装置**であることから、ページの移動・再表示等はきわめて迅速に実行される。

* 表示されているテキストおよび履歴テキストは主プロセッサ側の RAM、その他のテキストは従プロセッサ側の RAM 中に置かれる。しかし、テキスト保持用として使用できるメモリはなお計約 50KB で、この容量はテキストを多数もち MDPS 機能を活用しようとする際の大きなネックとなっている。
** 1画面約 2,000文字のたんなる表示は約 0.1秒で実行される。

MDPS モニタはまた、MDPS コマンドモード時において、コマンドの解析、実行形式ツールプログラムの起動を行う。MDPS コマンドの一般形式は、

コマンド名 引数 1 ; ... ; 引数 n

である。利用者が自身や他の利用者へ便利なプログラムをツールプログラムとして登録することを助長するため、コマンド名は FD ファイル名に一致させ、解析された引数列は一定のメモリ領域に定まった形式で格納されてツールプログラムに渡されるようになっている。

表 3 おもな MDPS コマンド
Table 3 Main MDPS commands.

カテゴリ	コマンド	意味
ファイル操作 関連	#FCRE	FD ファイルの作成
	#FDEL	FD ファイルの消去
	#FDIR	FD ファイルの属性表示
	#FMOD	FD ファイルの属性変更
	#FF	FD ファイルのコピー
	#FF*	一方のボリューム中のすべての FD ファイルの他方のボリュームへのコピー
	#VDIR	FD ファイル一覧表表示
	#VMOD	ボリューム属性変更
	#VOLINIT	ボリューム初期化
	COMPARE LIST	二つの FD ファイルの比較 FD ファイル内容の表示
テキスト/ ページ操作 関連	CATA	テキスト属性の一覧表表示
	NOTES	テキスト属性の追加/変更
	LOAD	FD ファイル内容のテキストへのコピー
	SAVE	テキスト内容の FD ファイルへの格納
	MDED	複数画面向きスクリーンエディタ
ホスト計算機 (DEC 20) 関連	FH	FD ファイルのホストファイルへの転送
	HF	ホストファイルの FD ファイルへの転送
	MPL	FD ファイル中の μ PL プログラムのコンパイル (ホスト計算機中のコンパイラ利用)
	PASCAL	FD ファイル中の PASCAL プログラムのコンパイル
その他	HELP	コマンド等の一覧表出力または説明

この方式は、MDPS コマンド処理部分の構造を簡略化し、MDPS ツールを段階的に作成していく上で大いに役立った。実際、MDPS ツールの多くはこの方式を利用することで容易に追加されてきた。こうして MDPS ではシステムで用意されるツールプログラムと、利用者によって作成されるツールプログラムとも区別していない。

表 3 に主要な MDPS コマンドの一覧を示す。

ii) MDPS ファイルシステム

FD ファイルは、名前、種類、作成日等の属性と、ファイル内容とから成る。ファイルの種類としてはテキスト、実行形式、バイナリの 3 種がある。ファイル内容は、セクタ (128 バイト) の双方向リスト構造によって表現されている。

MDPS ファイルシステムは従プロセッサ側にあり、常駐部分 (ファイルのオープン・クローズ、レコードの入出力) と非常駐部分 (ファイルの作成・削除等) とから成る。主プロセッサ側からの論理的なファイルアクセス要求に応じて所期の動作を行い、結果を主プロセッサ側に返すようになっている。こうすることで、MDPS モニタ、MDPS ファイルシステムとも比較的単純な構造で構成することができ、段階的なツール群作成も容易となった。ただし、割込み中断等の例外事項発生時の同期回復処理がいくらか複雑になる欠点はある。

MDPS モニタ、MDPS ファイルシステムの中の基本ルーチン群は、その仕様とともに MDPS 利用者に公開されており、利用者によるツールプログラムの作成等に利用できる。

5.2 ツールプログラム

i) 言語指向型エディタ LOE

言語指向型エディタ生成システム⁹⁾を用いて、Pascal, μ PL, XPL 等の言語指向型エディタが作成されている。これらのエディタはそれぞれ扱っているプログラミング言語の構文に関する情報を持ち、たとえばトークン、ブロック単位での移動、削除等が可能である。

これらの言語指向型エディタは DEC システム 20 上で動くが、MDPS 上のテキストを制御する機能をもっており、MDPS 下で使用する場合は三つの CRT をそれぞれ異なる用途で使用する。すなわち、CRT 1 をシステムとの会話用に、CRT 2 と CRT 3 をそれぞれ 1 編集作業の前と後のポインタ周辺のプログラムテキスト表示用に用いている。

こうして、必要ならば編集効果を確認し、プログラムテキストとコマンドやシステムメッセージとの混在のない見やすい表示内容の下で、構文的には常に正しいプログラムの状態のまま編集作業を進めることができる。

ii) ファイル転送プログラム

ホスト計算機上のファイルと FD ファイルとの間でデータをやりとりするためのホスト計算機上、MDPS 上のそれぞれにある 1 対のプログラムである。9,600 ボーの回線を用いているが、実際はファイルの I/O が

表 4 おもな MDED コマンド
Table 4 Main MDED commands.

^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 (T)	:	move to line beginning
^D (R)	:	delete a line forward
^E (T)	:	move to line ending
DEL(R)	:	delete a line backward
TEXT subcommand		
^F	:	load a PD file
^G	:	save to a PD 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 (R)	:	execute commands of execute-text
^Q	:	quit MDED
^P (T)	:	copy a string, set 'cur'
^R	:	replace all occurrences
^S (T)	:	search a string, set 'cur'

^A などは control-Aなどを表す

(T): 数値パラメータがテキスト番号であることを示す

(R): 数値パラメータが繰返し回数であることを示す

cur-character: cur ポインタ直前の文字

bef-cur-string: bef ポインタと cur ポインタで囲まれる文字列

可能な速度(200~500字/秒)で抑えられている。DEC システム 20 上で MDPS ソフトウェアを開発して MDPS 内で実行させる場合等で用いられる。

iii) 複数画面向きスクリーンエディタ MDED¹⁰⁾

MDED は複数画面環境向きのスクリーンエディタとして、MDPS コマンドモード下で動くよう設計・開発された。常時追加モードであることやコマンドの形式は、すぐれたスクリーンエディタである EMACS¹¹⁾を参照とした。

MDED の特徴は、MDPS のテキストを編集の対象としそれらを同時に複数個扱うこと、一つのテキスト内でも同時に複数個の部分参照できることである。したがって、関連するドキュメントやプログラム、編集対象プログラムのなかの宣言部分や呼出し部分等を参照し、あるいは利用することができ、容易かつ確実に編集作業が進められている。

に編集作業が進められている。

編集対象テキストは同時に複数個存在し、参照やポインタ変更はすべてのテキストが対象となるが、利用者の混乱を避けるため、直接変更の対象となるのはカレントテキストに限っている。

各テキストは3種のポインタ any, bef, cur をもつ。各ポインタの周辺を一つのページとして表示させることができる。編集は、文字に対しては cur ポインタ、文字列に対しては bef, cur ポインタを用いてなされる。any は表示場所を示すためのみに用いられる。

MDED は履歴テキストのほかにバックアップテキストも管理更新していて、ここに削除された文字列を格納する。同一テキスト内、異なるテキスト間の文字列の移動やコピー等が、バックアップテキストを介することで、基本的なコマンドの組合せとして、作業を

操 作	(コ メ ント)	CRT 1	CRT 2	CRT 3	CRT 4
(MDPS 使用開始)		《Page 1》			
(FKB) F 0	(端末モード)				
^C LOG ...	(DEC システム 20 ログイン)				
(FKB) F 0	(MDPS コマンドモード)				
HELP MDED		MDED コマンド一覧			
(FKB) 0 4	(move & create current page)	《Page 2》			《Page 1》
#VDIR 1		FD ファイル一覧			MDED コマンド一覧
MDED ^X^F EX.PAS	(テキスト3に)	《cur-Page 3》		《cur-Page 2》	《cur-Page 1》
(FKB) 3 3 A 1, 3 2 A 3	(move)				
4 3 E	(set current page)				
(FKB) 3 A3 A 2	(move)				
:	(Page 1, Page 3 を必要に応じて参照しながらの編集作業)				
^X^F LIB.PAS	(テキスト4に)	(編集作業)	《any-Page 3》	FD ファイル一覧	
(FKB) 3 4 A 3	(move)		EX.PAS プログラム先頭部分(種々の名前の宣言)	《cur-Page 4》	
^N4^S getfile ESC	(LIB.PAS 中で 'get file' を探す)			LIB.PAS プログラム	
:	(LIB.PAS プログラムから利用できる部分をコピーする等)				
(FKB) 2 4 E	(purge)				
^X^G EX.PAS	(ファイルへの格納)	《Page 3》		《Page 2》	《Page 1》
ESC ^Q	(MDED 終了)				
PASCAL EX.PAS	(FD ファイルのコンパイル)	コンパイルメッセージ(いくつかのエラー含)			
(FKB) 3 2 E	(ページ2の消去)				
MDED ^X^F EX.PAS	(テキスト4に)	《cur-Page 4》	《cur-Page 3》		
(FKB) 3 3 A 2, 3 4 A 1	(move)	(編集作業)	コンパイルメッセージ		
4 4 E					
:	(コンパイルメッセージを参照しながらの編集作業)				
^X^G EX.PAS	(ファイルへの格納)	《Page 4》	《Page 3》		
ESC ^Q	(MDED 終了)	コンパイルメッセージ(エラーなし)			
PASCAL EX.PAS	(FD ファイルのコンパイル)	実行結果			
(FKB) 2 3 E	(テキスト3の削除)	《Page 3》	《Page 4》		
F 0	(端末モード)	入力データのリスト	実行結果		
EXECUTE EX	(リンク & ゴー)				
(FKB) 0 2	(move & create current page)				
TYPE INPUT.DAT	(入力データと実行結果との比較検討)				

《...》はカレントページを示す。

図 4 MDPS 使用例
Fig. 4 An example of MDPS use.

確認しながら行うことができる。

表 4 に主要な MDDED コマンドを示す。

iv) ファイルアクセスプログラム

従プロセッサ上で動くファイルシステムと利用者とのインタフェースをとるもので、表 3 中のファイル操作コマンドを実現しているものである。

v) μ PL

μ PL は MDPS 開発のために設計されたシステム記述言語である。MDPS モニタの大部分、MDDED 全体、ツールプログラムの多くが μ PL で記述されている。 μ PL は M 6800 の特徴を反映した Pascal 風のミニ言語であり、byte, address の二つのデータ型のみをもつ。DEC システム 20 上に Pascal で書かれたクロスコンパイラが用意されている。

6. MDPS 使用例

MDPS 環境下におけるプログラミング作業の例を図 4 に示す。ここでは FD ファイル中の Pascal プログラム (EX.PAS) を MDDED エディタを用いて完成させていく過程の概要が示されている。作成途中でできていた LIB.PAS 中の一部を参照し活用している。

中央の CRT 1 は常にシステムとの対話や編集作業の場であるカレントページとして、左右の CRT 2, CRT 3 は作業を進める上で比較・参照を必要とする情報 (宣言部分, コンパイルメッセージ, 他のプログラムなど) を表示しておく場として、上部の CRT 4 はときおり役立つであろう固定的な情報 (MDDED コマンド一覧表) を表示しておく場として使われている。このような CRT 使用法は、経験上最も典型的な使用法である。

7. おわりに

複雑なプログラミング作業を支援するための方法は種々考えられるが、ここでは人間が直接対象とする端末装置に焦点をあて、作業環境として適した場を構成することを目的として設計・開発してきた一つの実験システム MDPS について報告した。

MDPS は 1981 年春から夏にかけ第 1 版が完成した。その後初期の使用経験に基づいた改良がなされてきており、現在一般利用者の使用に供されている。

この種のシステムの有効性は、多くの使用経験と組織的な使用実験に基づいて判断されるべきものであるが、現在までの約 1 年半近くの使用経験といくつかの

使用実験から述べれば、複数画面の利点を活用した作業が行われており、作業能率の向上に役立っていると言える。しかしながら FD ファイル I/O の遅さ、テキスト内に含めうる情報量の上限、等いくつかの問題もあり、総合的にすぐれたプログラミング環境を作成することの困難さも感じている。

MDPS の詳細なシステム構成、使用経験・実験に基づく評価等は別の機会に述べることにしたい。

謝辞 末筆ながら、本研究の機会を与えられた棟上昭男ソフトウェア部長、石井治パターン情報部長 (前ソフトウェア部長)、熱心に討論していただいた言語処理研究室その他の諸氏、MDPS ツールのインプリメントを手伝っていただいた諸氏に深謝する。

参 考 文 献

- 1) Ivie, E. L.: The Programmer's Workbench — A Machine for Software Development, *Comm. ACM*, Vol. 20, No. 10 pp. 746-753 (1977).
- 2) Thacker, C. P., McCreight, E. M., Lampson, B. W., Sproull, R. F. and Boggs, D. R.: Alto: A Personal Computer Xerox PARC, CSL-79-11 (Aug. 1979).
- 3) PERQ Manuals, Three Rivers Computer (1981).
- 4) Teitelman, W.: A Display Oriented Programmer's Assistant, Xerox PARC, CSL-77-3 (Mar. 1977).
- 5) Torii, K., Mano, Y. and Ohmaki, K.: A New Programming Environment with a Multi-Display Terminal, COMPSAC 81 (1981).
- 6) Mano, Y., Ohmaki, K. and Torii, K.: Early Experiences with a Multi-Display Programming Environment, 6th ICSE (Sept. 1982).
- 7) Herot, C. F.: Spatial Management of Data, *ACM Trans. on Database Systems*, Vol. 5, No. 4, pp. 493-514 (1980).
- 8) Mano, Y.: A Programming Language μ PL for Multi-Display Micro-computers, *Bulletin of the Electrotechnical Laboratory*, Vol. 45, pp. 78-102 (1981).
- 9) 大蒔, 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, AL 81-59 (1981. 10).
- 10) 真野: 複数画面プログラミング環境における画面エディタ MDDED, 情処ソフトウェア工学研究会, 22-6 (1982. 2).
- 11) Stallman, R. M.: EMACS Manual for ITS Users, MIT AI Memo 554 (Jun. 1980).

(昭和 57 年 9 月 3 日受付)

(昭和 57 年 11 月 8 日採録)

* テキスト n のポインタ p の周辺のページを p -ページ n と呼ぶ。