パーソナル・データ処理システム（PDPS）の試作

阿江　忠，照井善雄†，菅原　淳††，輝平盛重
（広島大学　工学部）

1．まえがき

マイクロプロセッサをはじめとするコンピュータの各機能ブロックがLSI化されるようになってから，小形でかつ安価に一応のメモリ容量をもつシステムが実現できるようになった。そこで，我々は，マイクロコンピュータの一実用化例として，個人的なデータ処理を目的としたシステムを試作したので御報告する次第である。

システムは，内部記憶容量ROM最大8Kバイト，RAM 12Kバイトをもつ CPU（およそ汎用マイクロコンピュータと同規模）と外部記憶（カセットレコーダ），それに入出力装置として，キーボード付CRTディスプレイ装置およびプリンタ（現在，TTYのプリンタ機構を流用）からなる。

本システムを計画するもとになった個人的なデータ処理の例を思いつくままに2，3挙げてみよう。

i）成績処理

学籍番号（数値）	氏名（記号）	試験結果(教科1)（数値）	-----	試験結果(教科k)（数値）	総合点（数値）
———	———	———	-----	———	———
———	———	———	-----	———	———

ii）趣味のリスト

背番号（記号）	氏名（記号）	打数（数値）	安打数（数値）	四死球数（数値）	打率（数値）
———	———	———	———	———	———
———	———	———	———	———	———

iii）金銭出納帳

日付（数値）	項目（記号）	収入（数値）	支出（数値）	残高（数値）
———	———	———	———	———
———	———	———	———	———

これらは割に普遍的な例であり，“パーソナル”なものとなれば限りがなくなる。このような個人的なデータ処理を行なうシステムとして，パーソナル・データ処理システム（Personal Data Processing System ― PDPS ―）の仕様に留意したのは次のような点である。

（1）ユーザは“しろうと”であるとして，電卓的な気軽さで表を作成（ファイルの作成）ができること。

---

†）現在，電々公社武蔵野通研　　††）現在，ファコム・ハイタックKK

(2) 必要な処理(計算やリーティングなど)も同じく容易にでき,結果をファイルできること。

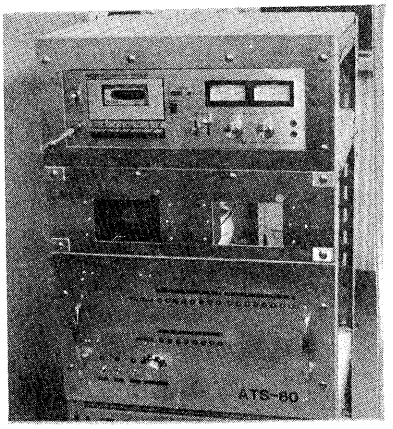
PDPSにインプリメントするインタプリタは当初,独自につくる予定であったが,Tiny BASIC のリスト公開以来[1][2],Tiny BASIC を軸にして上の要求を満たす言語(PDL)をつくる方針に変えた[4][5]。本稿ではPDLのVERSION 1.0を走らせた結果を述べる。

2. システムの概要

2-1. ハードウェア

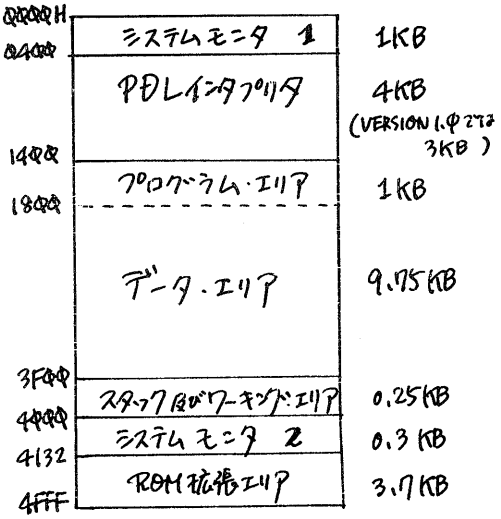
PDPSのハードウェアを我々はATS-8Φ(Augmented Tiny System using 8Φ8Φ)と呼ぶ。CPUはIntel 8Φ8ΦA,周辺回路もそのファミリーからなる。(写真1参照)

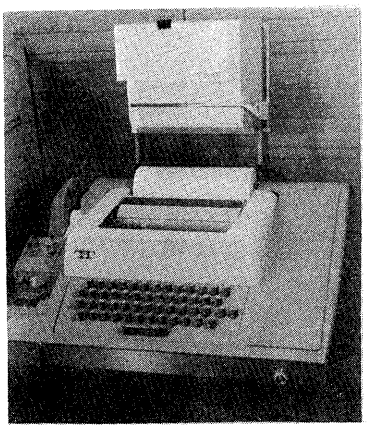
ROM:(最大) 8Kバイト
RAM: 12Kバイト
外部記憶:(現在) オーディオ・カセット
(近々,デジタル・カセットに変更予定)
I/O: キーボード付CRTディスプレイ
(110~2400 Baud 可変)
(写真3参照)
プリンタ(TTY流用)
(写真2参照)



[写真1] (上)オーディオ・カセット(現用)
(中)ディジタル・カセット(予定)
(下)ATS-8Φ(CPU)

ATS-8Φの内部メモリマップは次の通り。





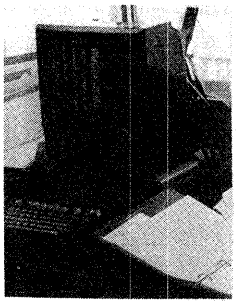
[写真2] プリンタ(TTY流用)

汎用マイクロコンピュータと比べて,データ・エリアを相対的に大きくとり,専用機化した使い方になっている。

2-2. ソフトウェア

システム・プログラム（モニタ）は Intel 仕様にカセット・テープへのコマンド（L および T）を追加したものを用いている。PDL インタプリタの最終バージョンが完成した時点では，システムモニタ 1 は不要となるが，開発中ではデバッグ等のため必要である。

PDL インタプリタについては次節で詳述する。なお，インタプリタの開発ツールとしては，Intel FDOS MDS-800 マクロ・アセンブラを用い，このホスト機から試作 PDPS へは電流ループで直接プログラムが転送される。さらに，PDPS 上ではモニタのデバッグ機能に加えて，別途開発したシミュレータ（ブレークポイントを適宜設けてレジスタの内容を表示するプログラム）やリロケータなどサポート・ソフトウェアもカセット・テープの形で用意した。



［写真3］キーボード & CRT ディスプレイ

3. PDL インタプリタ

PDL は Personal Data-processing Language の略で，1. で述べた目的のためにつくられた会話形言語である。Tiny BASIC（Palo Alto 版 (1)(2) 詳しい解説は (3) 参照）を拡張した形でつくったので，Tiny BASIC の機能はそっくりそのまま含む。以下では，Tiny BASIC に関する説明は省略し，PDL の特徴のみを紹介する。

コマンド名	形式	機能
Display	**D** n_1, n_2 (CR)	n_1 番地から n_2 番地までのメモリ内容を表示
Go	**G** n (CR)	n 番地からプログラムを実行
Insert	**I** n (CR)	n 番地からメモリの書込み
Load	**L**	テープからのロード
Move	**M** n_1, n_2, n_3 (CR)	n_1 番地から n_2 番地までのメモリの内容を n_3 番地以降に移動
Substitute	**S** n (SP)	n 番地のメモリ内容の表示及び変更
sTore	**T** n_1, n_2 (CR)	n_1 番地から n_2 番地までの内容をテープにストア
eXamine	**X** (CR)	すべてのレジスタの内容を表示
	X m	m で示すレジスタの内容の表示及び変更

注. 1) n, n_1, n_2, n_3 はそれぞれアドレスを示す 16 進数．
2) m は A, B, C, D, E, F, H, L, M, P, S のいずれか．
3) (CR) キャリッジ・リターン・キー，(SP) スペース・キー（これは , でも可）

［表1］モニタ・コマンド一覧表

PDLの特徴は

①2次元配列の要素として，数値（整数 ±32767 以内）のほか任意長の記号列（文字列）を許すこと。

②その宣言は容易で，データの入出力が簡単なこと。とくに，配列をダイレクトモードで読込み，消去または書換え，出力ができること。

である。②のダイレクトモードとは，文番号なしで，いきなり文（ステートメント）を書き，実行させる"電卓的"使用法である。（この場合，：で区切れば文は複数個でもよい。）

Palo Alto 版 Tiny BASIC は2Kバイトというコンパクトさにもかかわらず，文法的には立派である[3]。しかし，数値が2バイト整数に限られるという点と配列は1次元で1個だけという制約がある。PDLでは，まず配列の強化を目的としているが4K BASIC 程度で用意される配列とはかなり異なる。①で述べたように，数値と記号列の混在した行列の定義を許すのと，その読込みを表の形そのままでできるようにしたことである。（この点は，APLの入力形式と似ている。）ただし，配列は1つのプログラム中で1個しか許されず，それは$という形で，次のように宣言される。

たとえば，　＞10 DIM $(A)=(S15,I,I,I,S5)

とすれば，行数A，列数5の記号列まじりの行列が定義される。　列の要素は①記号列のときSi（iは長さで省くとエラー），②整数（±32767以内）のときIとすれば定義される。プログラムの書き方はBASICと同じで，ダイレクト・モード以外は文番号をつける。Aは整数のほか（大小比較式を含まない）式でもよい。

配列要素の読込みは，単に，

＞20　READ

とすればよい。プログラムをRUNさせると，コンピュータから

1：（15文字の記号列），（整数），（整数），（整数），（5文字の記号列）

と行番号とコロンでデータを要求してくるので，定義通りに要素をキーインし，コンマ（，）で区切りながら入力する。（文字列は入力しないと空白とみなすが数値は0でも入れておかないとエラー。要素の型違いや数値オーバーはエラー。要素数（列数）のオーバーは無視。）

1行の終りは↵（キャリッジリターン）で区切るが，コンピュータはただちに次の行

2：

の入力を要求し，宣言した行数Aに達するまで繰返す。

```
LIST
  5 REM "SEISEKI HYO NO REI"
 10 DIM $(10)=(S15,I,I,I)
 20 READ
 25 PRINT "SEISEKI HYO"
 27 PRINT "       NAMAE                  SANSU   KOKUGO   GOKEI"
 30 FOR J=1 TO 10
 40 PUT(J,4)=PICK(J,2)+PICK(J,3)
 50 WRITE J,J
 60 NEXT J
 70 PRINT "80 IJO NO HITO"
 80 FOR K=1 TO 10
 90 IF PICK(K,4)<80 GOTO 110
100 WRITE K,K
110 NEXT K
120 STOP
```

図1．プログラム例（簡単な成績処理プログラム）

途中でデータ入力を打切るときはETB(Cont.W)をキーインする。読んだデータをそのまま表示するには

> 3Φ WRITE Φ

とすればよく，表の形で出力する。(写真4参照) カセットへはデータはSTORE(プログラムはSAVE)でストアされ，逆にロードはLOAD(プログラムも同じ)コマンドで実行される。

以上のことを"電卓的使用法"ダイレクトモードで行なうと，

> DIM $(1ΦΦ)=(S15,I,I,I,S5);READ
[データの読込み(行数は1ΦΦとした)]
> WRITE Φ
[データの再チェック] [注]WRITE I,J とすればI行からJ行まで出力
> STORE
[テープに保存が必要なとき]

となる。WRITE Φ(省略形 W.Φ)でデータ表示している例が写真4である。(定義はしたが未定のデータはΦとして入れてある。)

詳しいPDLの仕様および文法は付録にゆずり，配列の要素を出入れする関数PICKと文PUTを使うデータ処理プログラム例を図1に示す。ここで，PICK(J,K)はJ行K列の要素(整数)をとり出す関数，PUT(J,K)=〈〈大小比較式を含む〉式 > は式であらわされる数値をJ行K列に入れる文である。なお，VERSION 1.Φではいずれも数値に対してのみ有効である。図1のプログラム実行結果(ただし，READの部分は除く)を図2に示す。試作システムの場合，データエリアを10KB弱にとってあるので，この成績表のような場合，10教科とすると200人位のファイルの作成および処理が行なえる。

[写真4] ダイレクト・モードによるデータの出力例

ダイレクトモードによるデータ処理

配列(データ)の行単位での削除はコマンドBLANK n(nは該当するデータの行番号)でなされ，n+1行以下の行はすべて1行づつくり上る。行単位でのデータの追加はREADにより一番最後につけ加える。

データを列単位で削除するコマンドはBLANK,n(nはデータの列番号)で，該当列が記号列なら空白に，数値ならΦになる。列単位のデータの追加は数値については

> FOR I=1 TO A(行番号の最後) ; INPUT C ; PUT(I,k)=C ; NEXT I (k:挿入する列番号)

とすればよい。

直接のデータ修正は > PUT(J,K)=〈式〉 とすれば，やはり，ダイレクトモードで行なえる。

4. むすび

PDLインタプリタのVERSION 1.0†をしばらく使ってみて，マイナーな改良以外に次の点をバージョン・アップの際に考慮したいと考えている。

(1) プログラムをさらに簡単化するため主変数（配列）に対する関数や文を増やす。（付録は現在開発中のVERSION 2.0を示す。）
(2) ディジタル・カセットに変更する。
(3) 配列の仕数をふやし，マージできるようにする。
(4) 整数の大きさを拡張する。

メモリの集積度の上昇の結果，試作システム程度を現在つくれば，キーボードとディスプレイを一体化したスタンドアロン形でコンパクトに収められよう。そして，個人用の安価なファイルシステムとして十分普及させられると考えている。

```
>RUN
SEISEKI HYO
      NAMAE                ,  SANSU   KOKUGO  GOKEI
   1 AKIMOTO               ,  35,     47,     82
   2 ITO                   ,  46,     34,     80
   3 UEDA                  ,  44,     28,     72
   4 EMOTO                 ,  50,     32,     82
   5 OSAKA                 ,  33,     50,     83
   6 KANDA                 ,  44,     34,     78
   7 KONO                  ,  36,     21,     57
   8 SAGAWA                ,  43,     34,     77
   9 SIMIZU                ,  39,     41,     80
  10 SUMITANI              ,  35,     41,     76

80 IJO JO HITO
   1 AKIMOTO               ,  35,     47,     82
   2 ITO                   ,  46,     34,     80
   4 EMOTO                 ,  50,     32,     82
   5 OSAKA                 ,  33,     50,     83
   9 SIMIZU                ,  39,     41,     80
```

図2. 図1のプログラムの実行結果
（合計はほめのゆと入れておく）

最後に，ホスト機使用の便宜をはかっていただいた本学工学部吉田典可教授に深謝する次第である。

参考文献

(1) Li-Chen Wang：“Palo Alto Tiny BASIC”，D.D.J.，Vol.1，No.5，pp.12-25（May 1976）
(2) 今井：“2K BASIC”，ASCII，No.2（1977-08）
(3) 小野：“Tiny BASIC インタプリタ”，（石田編）bit 臨時増刊「マイクロコンピュータのプログラミング」（1978-02）
(4) 阿江，輝平，照井，菅原：“パーソナル・データファイル・システムPDFSの開発(1) 基本構想”，電気四学会中国支部連合大会 22311（1977-11）
(5) 阿江，大崎，照井，菅原：“同上 (2) 言語仕様”，同上 22312（1977-11）
(6) 阿江，照井：“パーソナル・データ処理言語PDL”，インターフェース，No.16（1978- ）

†) 全ソースリストは文献(6)で公開されている。サイズは3Kバイト。

付録 PDL文法一覧（VERSION 2.0）

（［］内はない場合もあることを示す）

（英字の小文字は正整数を表わす）

1. コマンド

キーワード	省略形	形　式	機　能
NEW	N.	NEW	プログラムをすべて消す
LIST	L.	LIST [n]	プログラムを印刷する
RUN	R.	RUN	プログラムを実行する
SAVE	S.	SAVE	プログラムをテープにストアする
STORE	ST.	STORE	データをテープにストアする
LOAD	LO	LOAD	プログラムまたはデータをテープからロード
BLANK	B.	BLANK	データをすべて消す
		BLANK n	n行目のデータを消す
		BLANK,n	n列目のデータを消す

2. 文

（主変数；$で定義される配列，補助変数；TinyBASICでいう変数）

〈文の並び〉　〈文〉［;〈文の並び〉］

キーワード	省略形	形　式	機　能
LET	L.	[LET]〈補助変数代入リスト〉	代入
PRINT	P.	PRINT [〈印刷リスト〉[,]]	印刷
INPUT	I.	INPUT〈入力リスト〉	補助変数入力
GOTO	G.	GOTO〈式1〉	飛越
GOSUB	GOS.	GOSUB〈式1〉	呼出
RETURN	R.	RETURN	帰還
IF	IF	IF〈式1〉〈文の並び〉	判定
FOR	F.	FOR〈代入〉TO〈式1〉[STEP〈式1〉]	ループ
NEXT	N.	NEXT〈補助変数〉	
STOP	S.	STOP	終了
REM	REM	REM〈任意の記号列〉	コメント
READ	REA.	READ	データ入力
WRITE	W.	WRITE〈主変数印刷リスト〉	データ印刷
DIM	D.	DIM〈主変数定義リスト〉	配列の宣言
MSORT	M.	MSORT〈指定リスト〉	マキシマムソーティング
NSORT	NS.	NSORT〈指定リスト〉	ミニマムソーティング

KEY	K.	KEY（任意の文字列）[,n]	ｷｰﾜｰﾄﾞ選択
MOVE	MO.	MOVE i, j, k	行移動
CMOVE	C.	CMOVE i, j	列移動
PUT	PU.	PUT <挿入リスト>	挿入
MD	MD	MD <主変数代入リスト>	代入

3. 印刷リスト

<印刷リスト>	<印刷要素>[,<印刷リスト>]	
<印刷要素>	<式1>	値を印字
	<ストリング'>	文字列を印字
	#<式1>	桁数を指定
	←	同じ行の先頭から印字

4. 主変数印刷リスト

<主変数印刷リスト>	φ	全データ印刷
	<式1>, <式1>	ある範囲のデータ印刷

5. 入力リスト

<入力リスト>	<入力要素>[,<入力リスト>]	
<入力要素>	<ストリング>[<補助変数>]	<ストリング>をプロンプト
	←[<補助変数>]	㊐のみをプロンプト
	<補助変数>	補助変数名をプロンプト

6. 補助変数代入リスト

<補助変数代入リスト>	<代入>[,<補助変数代入リスト>]
<代入>	<補助変数> = <式1>

7. 主変数代入リスト

<主変数代入リスト>	<主変数代入>[,<主変数代入リスト>]
<主変数代入>	<主変数> = <式2>

8. 主変数定義リスト

<主変数定義リスト>	<主変数> = <主変数定義要素>	
<主変数定義要素>	(<配列要素>[,<配列要素>])	
<配列要素>	Sn	記号列
	I	数値

9. 指定リスト

〈指定リスト〉	〈指定要素〉	
〈指定要素〉	〈式1〉	行指定
	,〈式1〉[〈制限要素〉]	列指定
〈制限要素〉	(〈式1〉,〈式1〉)	行の制限

10. 挿入リスト

〈挿入リスト〉	〈挿入〉
〈挿入〉	〈主変数要素〉=〈式1〉
〈主変数要素〉	(〈式1〉,〈式1〉)

11. ストリング

〈ストリング〉	"〈"↘以外の文字列〉"
	'〈'↘以外の文字列〉'

12. 式1

〈式1〉	〈大小関係式〉
	〈算術式1〉
〈大小関係式〉	〈算術式1〉〈比較演算子〉〈算術式1〉
〈算術式1〉	[〈符号〉]〈乗除算1〉[〈加減演算子〉〈算術式1〉]
〈乗除算1〉	〈算術因子1〉[〈乗除演算子〉〈乗除算1〉]
〈算術因子1〉	〈関数1〉
	〈補助変数〉
	〈整定数〉
	(〈式1〉)

13. 式Z

〈式Z〉	〈算術式Z〉
〈算術式Z〉	〈乗除算Z〉[〈加減演算子〉〈算術式Z〉]
〈乗除算Z〉	〈算術因子Z〉[〈乗除演算子〉〈乗除算Z〉]
〈算術因子Z〉	〈関数Z〉
	〈主変数〉

14. 演算子, 符号

〈比較演算子〉	=	等しい	〈加減演算子〉	+	
	≠	等しくない		−	

>	大きい	〈符号〉	+
>=	大きいか等しい		−
<	小さい	〈乗除演算子〉	*
<=	小さいか等しい		/

15. 関数 1

キーワード	省略形	形式	機能
RND	R.	RND (〈式1〉)	乱数 (1~〈式1〉)
ABS	A.	ABS (〈式1〉)	絶対値
SIZE	S.	SIZE	プログラム・エリアの残り(バイト)
DSIZE	D.	DSIZE	データ・エリアの残り(バイト)
PICK	P.	PICK〈主変数要素〉	要素の抽出

16. 関数 Z†)

SUM	SU.	SUM[〈指定リスト〉]	合計
AVERAGE	AV.	AVERAGE[〈指定リスト〉]	平均
MAX	M.	MAX〈指定リスト〉	最大
MIN	MI.	MIN〈指定リスト〉	最小

17. 補助変数

〈補助変数〉	〈単純変数〉	A ~ Z	
	〈配列〉	@(〈式1〉)	0≦〈式1〉≦SIZE/2

18. 主変数

〈主変数〉	〈配列〉	$(〈式1〉)	(可変長)

* 特殊文字 (省略 — 文献(6)参照 —)

†) 関数Zのうちで、補助変数に代入されるものは関数1と同等にとりあつかわれる。