

ミニコンピュータのオペレーティングシステム OS/8について

OS/8: Operating System for the Mini Computer

渡 辺 勝*

Masaru WATANABE

1. はじめに

コンピュータのオペレーティングシステムはユーザの側から見ると、コンピュータの使用を容易にし、プログラム開発の時間を短縮する効果をもたらす。オペレーティングシステムは第3世代のコンピュータにいたって急速な発達をとげ、現代の大型コンピュータには必須のものとなっている。初期の頃のオペレーティングシステムの設計や製作は力づく (brute force) で作り上げられたものであるが、この数年、その基本理論や設計の手法に関する研究が活発に進み、体系が確立されつつある。

ミニコンピュータについても、はじめの頃は高速の紙テープ装置しかついておらず、紙テープの手操作のわずらわしさをともなったので、大型計算機なみにオペレーティングシステムが望まれていた。最近になって小型で低価格の磁気ディスクや磁気テープ装置が開発され、それなりに相当の容量を備えているので、ミニコンピュータにも、それにふさわしいオペレーティングシステムが開発できる状況になってきた。

オペレーティングシステムの中核となる部分は、管理プログラム (executive)、モニタないし制御プログラムなどと呼ばれていて、その一部は主記憶に常駐している。

アセンブラやFORTRANなどの言語処理プログラム、あるいはテキストの編集や虫取りプログラムは一括してシステムプログラムといわれているが、これらすべてをまとめてオペレーティングシステムということも多い。

ミニコンのオペレーティングシステムの場合にも、基本的な構成は同じであるが、メモリをはじめとする資源が限られたきびしい条件のもとで出来るだけのサービスを提供しなければならない。本報告はミニコンのオペレーティングシステムの代表的な一例として、PDP-8用に開発されたOS/8についての解説を試み、低価格のDECテープ装置を用いて具体化したシステムとその使用例について述べることにする。

2. システム装置としてのDECテープ

オペレーティングシステムの構成には、システムの非常駐部分や各種システムプログラムを格納するシステム装置が必要であり、小型の磁気ディスクまたは磁気テープなどが利用できることは既にのべた。OS/8で使用できる磁気テープ装置として、ミニコンピュータ用に特別に設計された小型で低価格のDECテープと称するものがある。DECテープの性能を他の磁気テープ装置と比較したものを第1表に示す。

第1表 磁気テープ装置性能比較表 (DEC社製品)

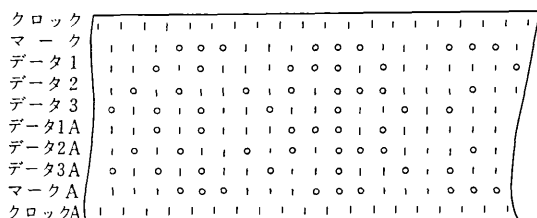
	DECテープ	カセット	汎用磁気テープ
テープ幅 インチ	0.75	0.15	0.5
テープ長 フィート	260	150	2400
記録方式	Manchester phase recording	phase encoded blocked	NRZI industry compatible
トラック数	10	1	9
記録密度 B P I	350±55	350~700	800
テープ速度 インチ/秒	97±14	9	45
転送速度 毎秒	33.3K×3bit	487byte	36K byte
容 量	188,672×12bit	87,000byte	
チェック 方 式	二重	16bitサイクリック リダダンシイ (CRC)	CRC, LRC
アドレス	固定長ブロッ クアドレス付	可変長 アドレスなし	なし
価 格	\$5000(2台)	\$3900(2台)	\$9950(1台)

DECテープはカセットと比較して、アクセス時間が1桁早いことが分る。この点はOS/8の応答時間をよくする点で性能向上に役立っている。

2.1 DECテープの記録方式 DECテープは大型計算機で使用されている汎用の磁気テープと異なり、各データブロックごとにアドレスがふってあり、任意のデ

* 東京大学生産技術研究部 第3部

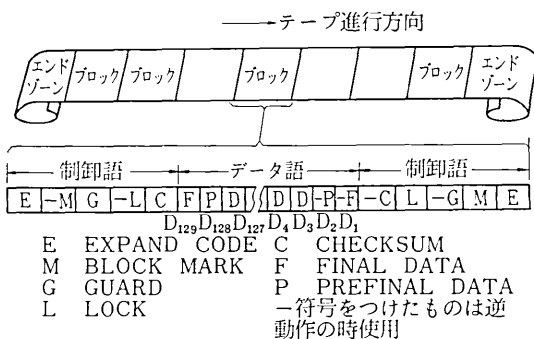
ータブロックをランダムに読み書き出来る。そのためテープには情報トラックの他に、アドレスなどの制御情報を書きこんだマークトラックおよびクロックパルスを記録したタイミングトラックが設けてある。データは3本のトラックに分けて記録され、結局5本のトラックがあることになるが、信頼性をよくするため全く同じトラックが二重になっていて合計10本のトラックを読み書きするようになっている。



第1図 DECテープのトラック配置

データは129語で1ブロックを構成し、各ブロックのアドレスはデータ部分の前後におかれている制御語の1つに書いてある。(データブロックの128語がコアメモリの1ページに対応して読み書きされ、残りの1語はリンクとして使用できる。ただしOS/8の場合はディスク装置との互換性を考慮してこの1語は使用していない。)

テープ1巻(直径3.9インチのリールに収まるポケットサイズ)に収容できるブロック数は2,702₈ = 1,474で、約188K語に相当する。



第2図 DECテープのフォーマット

2.2 DECテープの駆動方式 この方式の特徴は、テープの駆動がリールに直結した2相交流サーボモータの制御だけで行われている点であり、真空柱や張力制御などの複雑な機構を用いず、したがってピンチローラやキャプスタン等が省かれている。前進の際には、巻取りリールにフルトルクを、巻戻りリールに約半分のトルクを逆方向に加えることにより、適当な張力をテープに与えて、ヘッドに押しつけながらテープが前方に送られる。

2.3 アクキュムレータ経由のDECテープ 磁気テープや磁気ディスク装置とミニコンピュータとの接続は、通常data-break方式(一般にはDirect Memory Access, 略してDMAといわれている)により直接メモリとの間でデータ転送を行うようなインタフェイスが設けてある。この方式はそれなりの利点はあるが、インタフェイスのハードの価格が高くつく。そこでDECテープ装置のデータ転送をアクキュムレータ経由にして制御回路の簡略化をはかり、価格を大巾に安くしたものがTD8E型の磁気テープ装置である。ただしその代償としてメモリとの間のデータ転送はソフトウェアで行う必要がある。タイミング等の問題を考慮して作成されたTD8E専用の入出力サブルーチンが用意されている。

2.4 TD8E型磁気テープ装置を使用したOS/8 OS/8ははじめDMA方式の磁気テープまたはディスクを使用して開発されたものであり、主記憶は最小8K語から使用可能である。しかしOS/8では割込機能を使用していないこと、およびTD8E型磁気テープ装置でもテープ自体については他の装置と完全な互換性があるので、この装置をOS/8のシステム装置として使用することができる。TD8Eをシステム装置として使う場合の問題はデータ転送の専用サブルーチン(256語)を主記憶に常駐させなければならない点である。その方法の1つは、主記憶を12K語にすることであり、今1つは専用サブルーチンを搭載したリードオンリメモリ(ROM)を付加することである。当然ながら後者の方が価格はずっと安い。

3. OS/8におけるファイル

第3世代のオペレーティングシステムの一つの特徴はファイルの処理が中心になっていることである。OS/8においてもファイルは情報処理の基本的な単位であり、その構造を理解しておくことが必要である。一般にファイルはプログラムまたはデータの集りで、一つのまとまった処理単位を構成する。

3.1 ファイル名とファイル種別 個々のファイルはファイル名により識別される。ファイル名はOS/8の場合6文字以内であり、必要に応じて種別を指定する2文字が付加される。ファイルの種別にはASCII, バイナリまたはコアイメージ等があって、それぞれ.PA, .BN, .SVのように指定する。アセンブラでいえば、ソース、オブジェクトおよびロードされたプログラムに対応する。

3.2 ファイル構造を持つ装置 磁気テープや磁気ディスクのように多数のファイルを一つの装置に収容できるものをファイル構造を持つ装置という。これに対し紙テープ装置やタイプライタはファイル構造を持たない。

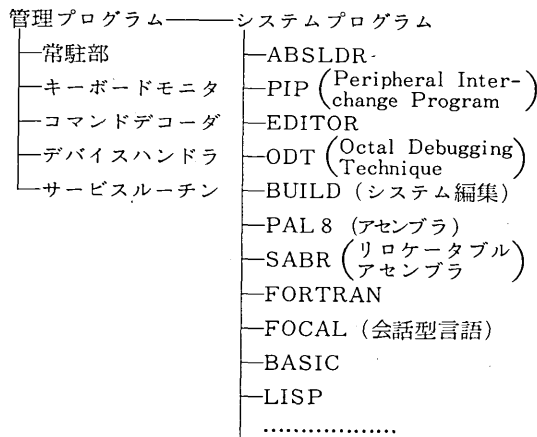
OS/8ではファイル構造をもつ装置は256語単位の

論理ブロックに分割されていて、各ブロックに順次に番号をふってある。一つのファイルは最小一つの論理ブロックから、最大は装置の全ブロックを占めることも可能である。

3.3 ディレクトリ 多数のファイルが一つの装置に収容されているとき、各ファイルを使用するにはそのファイルの存在場所および大きさについての情報が必要であるが、それを与えるものがディレクトリ（登録簿）である。OS/8の場合ディレクトリには、各装置のブロック番号の1から6迄が割りあてられ、最大240個迄のファイルを登録することができる。（次ページ第4図参照）

4. OS/8の構造

OS/8を構成する各プログラムはつぎのようになっている。



この中、管理プログラムの常駐部のみが主記憶を占有するが、その大きさは256語できわめて小さく、主記憶容量が小さい（たとえば8K語）システムでもユーザ領域は十分にとれる。常駐部以外のすべての管理プログラムおよびシステムプログラムは二次記憶に収容されている。

4.1 キーボードモニタ コンソルタイプライタからのコマンドの指示によりシステムあるいはユーザのプログラムを起動したり（RまたはRUNコマンド）、プログラムを二次記憶に保存したり（SAVEコマンド）、ふたたびロードしたり（GETコマンド）することができる。また入出力装置に標準以外の名称を与えたり（ASSIGN）、日付の設定をする（DATE）コマンドも用意されている。これによってユーザと管理プログラム間の連絡の役目を果たしている。

4.2 コマンドデコーダ システムプログラムたとえばアセンブラやエディタが起動されると、それらの入力および出力として使用される装置ないしファイル名をユーザに問いかけてくる。一般にシステムプログラム内での入出力は、いわゆる“装置と無関係”（device

independent）に作られていて、実行時に任意に指定できるようにになっている。指定された入出力装置ないしファイル名をシステムプログラムに通知する役割を果すのがコマンドデコーダである。同じコマンドという名称が使われていてキーボードモニタの処理するコマンドとまぎらわしいが、正確にはI/O指定とでもいうべきであろう。大型計算機のジョブ制御文で使うファイル定義文（FD文またはDD文）に相当するものと考えればよい。

4.3 デバイスハンドラ 周辺装置との間でデータ転送を実行する入出力サブルーチンのことをいう。OS/8においては最大15台までの周辺装置を組みこむことができる。システム構成に応じて必要なデバイスハンドラをOS/8に組みこむのは、編集プログラムBUILDによって行われる。同様にユーザが独自に作成したデバイスハンドラを追加することも可能である。

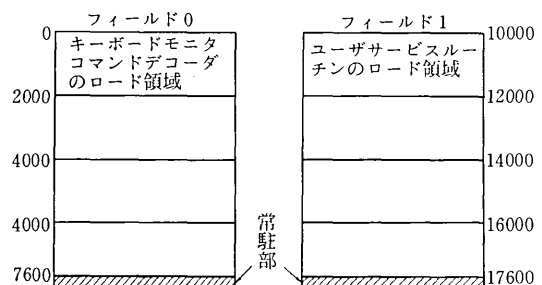
システム装置用のデバイスハンドラは主記憶に常駐しているが、それ以外のものは実行時に動的に主記憶にロードされる。また各デバイスハンドラは主記憶の割付を容易にするために最大256語以内に作成されている。

4.4 サービスルーチン 正確にはユーザサービスルーチン（USR）という。OS/8の運用のための各種のサービスを行うルーチンの集りであり、つぎのような機能を果たす。

- デバイスハンドラの主記憶へのロード
- 指定されたファイルをディレクトリにより検索
- ファイルのオープンとクローズ
- コマンドデコーダの呼出し
- プログラムのチェイニング

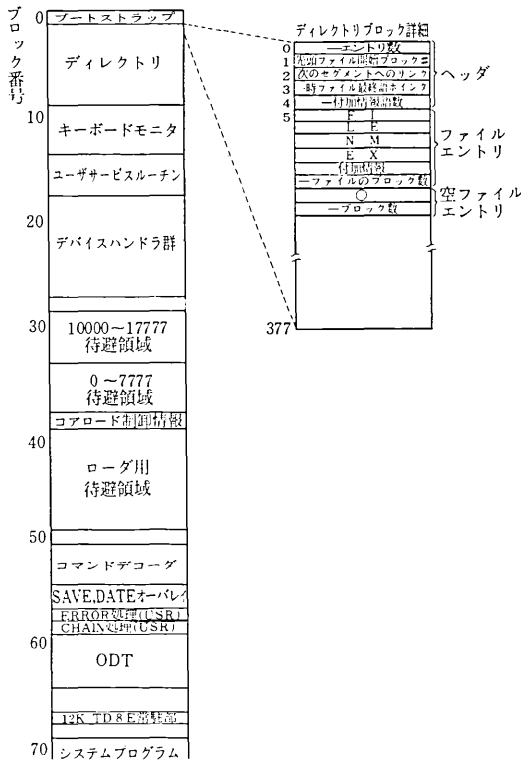
等である。サービスルーチンは普通はシステムプログラムから呼出されて使われるので、ユーザは個々の機能について知らなくても差支えないが、もちろんユーザプログラムから通常のサブルーチンと同様に呼出すことも可能である。

4.5 主記憶および2次記憶の割付 以上の各管理プログラムは既にのべたように通常はシステム装置（2



第3図 主記憶の割付

次記憶)に格納されていて、使用時に主記憶に呼出される。そのロードされる領域は第3図のようになる。この部分をユーザプログラムが使用している場合はシステム装置のスクラッチ領域に待避され後刻復元できる。



第4図 二次記憶 (システム装置) の割付

5. OS/8 の使用例

OS/8 を起動するには先ず77470番地よりスタートして、ROMの一部に入っているブートストラップを起動する。これによってシステム装置の先頭ブロックにある主記憶常駐部の写しがコアにロードされ、そこへ制御がうつされる。つづいてキーボードモニタが起動し、タイプライタに・を印字して、ユーザコマンドが入力可能なことを知らせOS/8の起動が完了する。

つぎの使用例では最初に日付けの設定を行っている。つぎにPIPプログラムを起動して、システム装置であるDECテープ0番のディレクトリをタイプライタにリストする。これにより登録されているシステムプログラムのファイル名、大きさおよび作成年月日を知ることができる。

実際にOS/8のもとでプログラムの作成を行うには

• DA 6/1/74

• R PIP
* TTY: ~DTA0:/E

6/1/74

```

AESLDR.SV 5 9/12/72
LIE8 .RL 30 9/20/72
PIP .SV 11 9/20/72
ECIT .SV 10 9/20/72
PAL8 .SV 16 9/20/72
FORT .SV 25 9/20/72
SAER .SV 24 9/20/72
LOADER.SV 12 9/20/72
CREF .SV 13 9/20/72
EITMAP.SV 5 9/20/72
EUILD .SV 29 9/20/72
SRCCOM.SV 5 9/20/72
DTCOPY.SV 5 9/20/72
EPIC .SV 17 9/20/72
<EMPTY> 5
FOCAL8.EN 19 10/1/72
NGNEAE.EN 8 10/1/72
EAEFPP.EN 8 10/1/72
27EFPP.EN 9 10/1/72
TLCCOPY.SV 7 9/20/72
<EMPTY> 418
423 FREE BLOCKS
    
```

先ずEDITORを用いて原プログラムの編集を行い(次頁),これをファイルとして登録する。EDITORのコマンドで指示されているファイル名DEM0となる。また入力の指定はないが、その場合はテレタイプを意味する。EDITORのサブコマンド#Aはテキストの挿入を、また#Lはリストをとること、そして#Eは作成したテキストを指定された出力ファイルに書き出すことを指示する。この場合DECテープ1番に記録され次の使用のため保持される。

こうして出来た原プログラムファイルを入力にして、アセンブルを行えば(PAL8),オブジェクトであるバイナリファイルが同様にDECテープ上に作成される。アセンブルリストを取ることもできるが下の例はその一部のみ掲げてある。

最後にローダ(ABS LDR)によりバイナリファイルを主記憶にロードし、指定した開始番地よりプログラムを実行させることが出来る。

続けてアセンブルの際のクロスリファレンスをとるためのコマンド例をあげてある。

以上の手順で分るようにプログラムの作成から実行まですべてキーボードの操作のみで処理され、紙テープは一切、扱わないですますことが出来る。もちろん必要ならば原プログラムやオブジェクトを紙テープに出力することも可能である。

6. システム編集の実例

ユーザのハードウェア構成に応じて容易にシステムを作成または変更できるのもOS/8の特徴である。以

```
.R EDIT
*ETA1: DEMO.PA~
```

```
#A
```

```
A:U
/AC LIGHT MEMO
/THIS PROGRAM WILL DISPLAY
/A LIGHT PATTERN IN THE AC LIGHTS
/
```

```
*200
LLOOP, TAD LOC
LOOP,  ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        JMP LOOP
        CML CMA RAR
        DCA LOC
        ISZ TEMP
        JMP LLOOP
        JMP I .+1  /RETURN TO THE OS/8 MONITOR
        7605
```

```
L0:U
LOC,   2525
TEMP,  5000
$
```

```
#L
```

```
/AC LIGHT MEMO
/THIS PROGRAM WILL DISPLAY
/A LIGHT PATTERN IN THE AC LIGHTS
/
```

```
*200
LLCOP, TAD LOC
LOOP,  ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        ISZ LOC
        JMP LOOP
        CML CMA RAR
        DCA LOC
        ISZ TEMP
        JMP LLOOP
        JMP I .+1  /RETURN TO THE OS/8 MONITOR
        7605
```

```
LOC,   2525
TEMP,  5000
$
```

```
#E
```

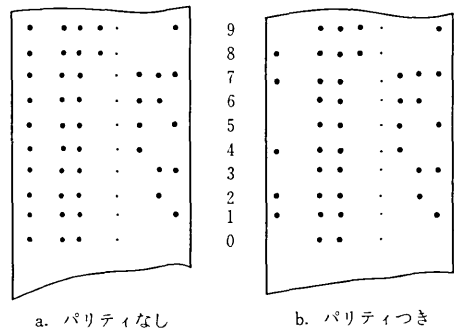
.R PAL8
 *DTA1: DEMO.BN, TTY: +DTA1: DEMO.PA

PAL8-V7 6/1/74 PAGE 1

/AC LIGHT MEMO
 /THIS PROGRAM WILL DISPLAY
 /A LIGHT PATTERN IN THE AC LIGHTS
 /

	0200		*200	
00200	1217	LLOOP,	TAD LOC	
00201	2217	LOOP,	ISZ LOC	
00202	2217		ISZ LOC	
00203	2217		ISZ LOC	
00204	2217		ISZ LOC	
00205	2217		ISZ LOC	
00206	2217		ISZ LOC	
00207	2217		ISZ LOC	
00210	5201		JMP LOOP	
00211	7070		CML CMA RAR	
00212	3217		ECA LOC	
00213	2220		ISZ TEMP	
00214	5000		JMP LLOOP	
00215	5616		JMP I .+1	/RETURN TO THE OS/S MONITOR
00216	7605		7605	
00217	2525	LOC,	2525	
00220	5000	TEMP,	5000	
			\$	

.R ABLEDF
 *DTA: DEMO.ENS
 DTA DOES NOT EXIST
 *DTA1: DEMO.ENS
 ILLEGAL SYNTAX 行の終りにALT
 'MODEを使用しなかったためのエラー
 *DTA1: DEMO.ENS
 .ST 200 プログラムの実行
 .R PAL8
 *, DTA1: DEMO.LS +DTA1: DEMO.PA
 .R CREF
 *TTY: +DTA1: DEMO.LS



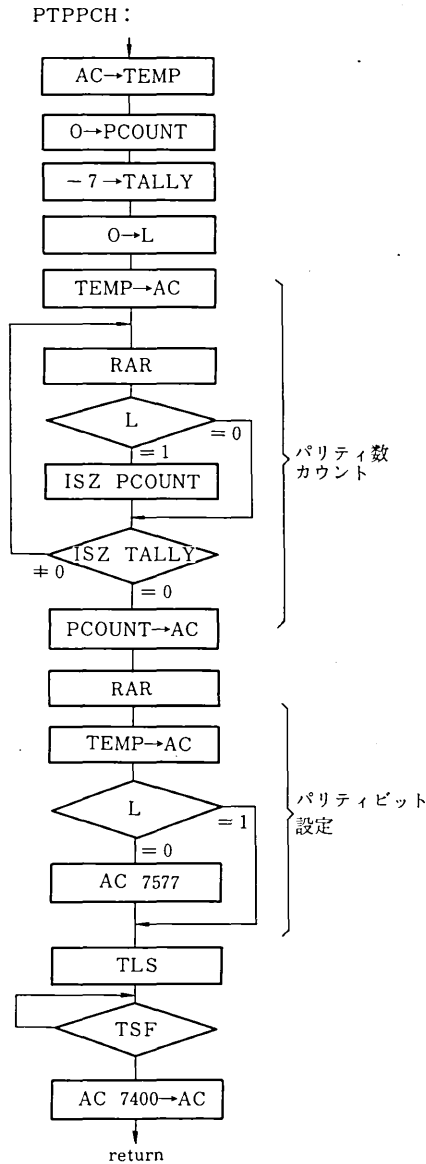
第5図 紙テープのASCIIコード

下にユーザの作成したデバイスハンドラをシステムに組込む場合を例にとり、システム編集の要領を説明しよう。

PDP-8の標準の紙テープではASCIIコードの8ビット目は常に1であり、パリティはとられていない。(第5図a) この紙テープをパリティつきの入力しか

受け付けないような他の計算機たとえばFACOM U-200の入力に使用する場合には、第5図bに示すパリティつきの紙テープを出力するよう、デバイスハンドラを修正しなければならない。

デバイスハンドラをシステムに組込む手順は、まずデバイスハンドラのルーチンを作成すること、つぎに



第6図のつづき

コードのパンチを実行するサブルーチンPTPPCHに分かれている。偶数パリティを付加する機能はPTPPCHの中に組みこんであり、パリティビットを除く7つのビット中の1の数を合計し、その奇偶によりパリティをセットしている。

6.2 デバイスハンドラ組込の情報 前項によって作成したハンドラを組込むには、システム編集プログラムBUILDによって行う。そのために上記のハンドラルーチンのあたりに、つぎのようなその入出力装置に固有な情報をあらわすヘッダブロックを付加しておく。

TYPE NAMEのPS33はこのグループに属するハンドラ群に共通のグループ名であり、今の場合は一つだけであるが、これに紙テープ入力を加える場合とか、磁気テープやディスクが複数台ある場合など、それぞれ共通のTYPE NAMEにまとめておく。

DEVICE NAMEは各装置に固有の名前であり、もし紙テープ入力追加される場合は別の装置名PPRのように指定する。これらの装置に関する情報はシステム編集の際にOS/8の制御プログラム内にある各テーブルに追加して登録される。

6.3 デバイスハンドラのシステムへの組込 つぎにシステム編集の方法を説明しよう。先ず上のように作られたハンドラをアSEMBルして出来たバイナリテープを用意する。編集プログラムBUILDはふつうのシステムプログラム同様キーボードコマンドで起動され、以下に説明するような編集用の各サブコマンドの処理を行う。

最初PRINTコマンドによりシステム編成の現状を確認しておく。KS33は標準の紙テープ入出力であり、*印のついた紙テープパンチPTPと紙テープリーダーPTRが組込まれていることを示す。つぎのLOADコマンドで今登録しようとするデバイスハンドラの紙テープを読みこませる装置を指定する。今の場合テープリーダーであり↑印が印字されてくるので、そこで紙テープを読みこませる。INSERTコマンドはTYPE名、DEVICE名で指定した装置を実際にシステムに組込むためのコマンドである。LOADしただけではBUILDプログラ

*0					
-1				ハンドラの数	} ヘッダブロック
DEVICE	PS33			TYPE NAME	
DEVICE	PIF			DEVICE NAME	
1020				DCB	
0				ENTRY POINT	
ZELGCK	2			0;0	
*200					} ハンドラ本体
PTP,					
	CLA	CLL	CLL		
	JLS		PSETUP		
PTPLF,	RSF				
	JLF		PTPCKT		

以下省略

.R EUILD

```

$PRINT
SYSTEM                      システムの現状の確認
TC08      TC8E      LINC      RF08      EF32      RK8      R08

```

```

NONSYS
PT0E:PTP      PTR
R33:PTP*      PTR*
TC08:DTA0      DTA1      DTA2      DTA3      DTA4      DTA5
LINC:LTA0      LTA1      LTA2      LTA3      LTA4      LTA5
TE0A:DTA0*     DTA1*
TE0E:DTA2      DTA3
A33:TTY*
LP0E:LPT
LS0E:LPT
L645:LPT
RK0:RKA0      RKA1      RKA2      RKA3
CR0E:CR

```

```

$LOAD PTR:                      ハンドラの登録
↑                               紙テープ(バイナリ)の読みこみ
$INSERT PS33,PPP                ハンドラのシステムへの組込み

$SYSTEM ROM=1                  システム装置指定

$E00T                           システム編集の実行

DSK=DTA1
SYS EUILD                       システム作成の完了
.SAVE SYS EUILD 0-7577,10000-17577=0;200

```

ムに登録されるにとどまり、INSERTによってOS/8内に組込まれ使用可能となるのである。

SYSTEMコマンドはシステムを収容すべき二次記憶を指定する。今の場合主記憶8KにROMを付加したTD8E DECテープ装置を指定している。つぎのBOOTにより、システムの編集が実行され、システムテーブルやデバイスハンドラを更新し、指定された構成のシステムを作成する。(その際、暗黙指定のデバイスはDECテープ1番DTA1として) システムの作成が完了する。

今作成したシステムを再確認するためにはPRINTコマンドを与えればよい。PS33:PPPが新たに組みこみ装置として*印がついていることが分る。

7. おわりに

OS/8の特徴を要約すれば

- 低価格でオペレーティングシステムを実現
- 主記憶常駐部は256語ときわめて小さい
- 各種の入出力装置の組込みが容易
- システムプログラムの登録が容易
- システムプログラムにおける装置よりの独立性

• システムの応答時間が早い
などの点があげられ、これによってミニコンピュータの使用をいちぢるしく容易にしたものといえよう。たとえばいくつかのソフトウェア開発に使用した実験によると、OS/8の性能や使い易さは大型計算機PDP-10のタイムシェアリングシステムに匹敵しているという。

一方ミニコンという制約のもとで、ユーザの期待する機能のうち、何をとり何をすてるかが設計者のもつとも苦心するところであろう。OS/8の場合でもそのような配慮から、つぎの点は除外されている。

- ファイルの保護
- インタラプト処理

割込処理を採用しなかったのはシステムを簡単化して主記憶の占有部分を小さくするという要請を優先させたためであり、プログラム開発を目的としたOS/8においては、この決定を賢明なものであったといえる。しかし実時間のオンライン処理のためには割込機能は必須であり、そのような用途にこたえて、最近Real-time System-8が発表されている。このシステムでは割込機能をもった実時間のモニタがforegroundで動

.R EUILD

```

$PRINT
SYSTEM
TC08      TD8E      LINC      RF08      DF32      RK8      RCM

NCNSYS
PT8E:PTP      PTR
KS33:PTP*     PTR*
TC08:DTA0     DTA1      DTA2      LTA3      DTA4      DTA5
LINC:LTA0     LTA1      LTA2      LTA3      LTA4      LTA5
TD8A:DTA0*    DTA1*
TD8E:DTA2     DTA3
AS33:TTY*
LP08:LPT
LS8E:LPT
L645:LPT
FK8:FKA0      RKA1      RKA2      RKA3
CF8E:CDR
FS33:PPP*

```

作し、backgroundとしてOS/8が使えるような設計になっている。

本研究に対し、浜田助教授、藤田講師および渡辺研、浜田研各位の協力を得たことを感謝する。なお本研究の一部は本所の49年度選定研究費により行ったものである。

(1975年3月10日受理)

文 献

- 1) "PDP 8/e Small Computer Handbook", Digital Equipment Corporation, 1973.
- 2) "OS/8 Handbook", Digital Equipment Corporation, 1974.
- 3) "OS/8 Software Support Manual", Digital Equipment Corporation, 1973.
- 4) Berry, G. W. and R. F. Lary: "PS/8: A Stand Alone Operating System for a PDP-8 Minicomputer", Proc. IEEE Computer Society Conference, Boston, 1971, pp. 59-60.
- 5) Wakelin, D.: "An Even Parity Patch for FOCAL 1969", DECUSCOPE, 13, 3, June 1974, pp. 18-19.
- 6) Watanabe, M.: "Save Memory Field1", ibid., 13, 3, June 1974, p. 24.
- 7) Diegenbach, D. C.: "Photograph of a PDP-12 LINK-tape", ibid., 13, 3, June 1974, p. 28.
- 8) 渡辺・鈴木: "ミニコンにおけるLISP", 生産研究, 26巻, 1号, 1974年1月, pp. 21-23.
- 9) "Real-Time System-8", Digital Software News, PDP 8/12, September 1974.