

漢字端末機としてのパーソナルコンピュータ

東京大学大型計算機センター 石田 晴久, 牛丸 守

1. 調歩同期無手順通信方式の漢字端末機の必要性

わが国では、1979年から80年にかけて、大型コンピュータ・メーカーから、日本語処理システムが相次いで発表され、最近では大学の大型計算機センターなどでも、日本語処理をTSSで行うユーザがじょじょに増え始めている。しかし、日本語処理をもっと広く利用可能にする立場から考えると、現在のTSSによる日本語処理には次のような問題がある。

- (1) 各社の漢字（日本語）端末機は、各社独自の伝送手順方式をもつものであって、当メーカーの日本語処理システムにしかつながらない。
- (2) そうした端末機の価格は、ソフトウェア開発費が含まれていることもあって、きわめて高価である。大学の研究室で買える値段ではない。
- (3) IBM系（Mシリーズなど）の大型機では、内部コードがEBCDICである。漢字コードが内部的にJISコードに近いものであっても、OSが無手順端末にたいしては、EBCDIC対JIS（ASCII）のコード変換を行うため、JISに準じた漢字の内部コードは、プログラムであらかじめEBCDICに直しておく必要のあることがある。
- (4) 漢字用の制御コードの使い方が必ずしも標準化されていない。
- (5) 電話線を通して無手順方式で使える安い端末機としては、漢字受信専用のプリンタ方式のものは出始めているが、CRT方式のものは、まだほとんどない。
- (6) 各社から、日本語ワードプロセッサが多数発売されているが、汎用性のあるデータ通信機能をもつものはほとんどなく、またユーザ・プログラマブルにもなっていない。

一方、最近のパーソナル・コンピュータ（パソコン）の上位機種には、16×16点程度の粗いパターンではあるが、漢字の表示ができるものも現れ始めた。こうしたパソコンは、RS232Cインターフェースと呼ばれる調歩同期無手順のデータ通信インターフェースをもっているから、必要なソフトウェアを開発すれば、一応は漢字端末として構成することが可能である。われわれは、2種類のパソコンに対し、BASIC言語やPascal言語でプログラムを書くことによって、次の方針のもとにそれらを漢字（日本語）処理用の端末機として使うことを試みている。

- (a) TSS端末として使うときに、ホスト（大型機）から送られてきた漢字まじり文を画面に表示したり、フロッピー・ディスクに格納したりできるようにする。
- (b) データ通信方式は調歩同期無手順で、データ通信線としては、電話を使うことを前提に、当面は300ビット/秒（bps）ないし1200bpsとする。
- (c) 漢字の入力は、原則として、大型機の機能（ローマ字漢字変換や2文字コード入力法など）を使って行う。ただし、パソコンに漢字タブレットをつけて、漢字を直接入力してフロッピー・ディスクに格納しておき、あとでそれを大型機に転送することは可能にする。〔パソコン自体でローマ字漢字変換を行わせることは、われわれとしては今後の課題である。〕
- (d) 漢字まじり文の清書出力は、大型機のレーザ・プリンタで行う。
- (e) パソコン用に必要なソフトウェアは、できるだけ他機種に移植できるようにするため、マイクロソフトBASICおよびUCSD Pascalで記述する。こうしたソフトウェアは、多少の手直しでどこのシステムに対しても使えるよう、単純で汎用にする。

(f) パソコン漢字端末をサポートするために、大型機の上にも、ローマ字漢字変換や2文字コード入力機能をもつ和文エディタや和文清書プログラムを開発する。

本稿では、以下にこうした試みを行った際にとった方法や現在のパソコンの問題点などについて述べる。

2. パーソナル・コンピュータの漢字入出力機能

現在市販されているパソコンの上位機種には、漢字を表示する機能をもつものが始まっている。主な機種は表1の通りである。プロセッサ(CPU)は、C-180KとMulti 16が16ビット型なのを除くと、他はすべて8ビット型(主にZ80)である。これらの漢字入出力機能に関する傾向を整理してみると次のようになる。

表1. 漢字の表示できる主なパソコン(*印は本稿で試用したもの)

メーカー	機種	漢字ROM	ドット数	漢字字種	その他の字	漢字入力法	BASIC
* 沖	IF800-30	○(ワンチップ)	16×16	2965	453	カナ読み	漢字列向
* 日本電気	PC-8801	○(8チップ)	16×16	2965	700	(16進数)	PUT
* 富士通	Micro 8	○(16チップ)	16×16	2965	453	(16進数)	Print
ソード	M243	○(16チップ)	16×16	2965	453	カナ読み	漢字列向
パナファコム	C-180K	○	16×15	7184	296	カナ読み	(可)
信州機器	QC-20		24×24	6000		カナ読み	(可)
東芝	パソピア	×	16×16	2965	453	カナ読み	(可)
日立	BM-3	×	16×16	1900	400	カナ読み	(可)
三菱	Multi 16	×	16×16	4000	200	(16進数)	(可)

(1) 漢字のパターンは16×16点のものが多い。これらのパターンはROM化しているものと、フロッピー・ディスク上にもっているものがあるが、ROMの方がもちろんな本格的である。漢字の字種は、JIS第1水準の2965種が中心である。特筆すべきは、IF800/30の漢字ROMで、電々公社開発のものと同仕様のワンチップ1メガビットの大型素子が使われている。

(2) 漢字を表示するやり方には、文字として表示する機種と、図形として表示する機種とがある。文字として表示する前者の例はIF800/30で、これだと(半角の)英数字の混合表示、画面のスクロール、文字としてのプリンタへの打出などが可能である。これに対し、図形として表示する方の例はPC8801やMicro 8で、これだと、プログラム上で図形座標を意識しなければならない。このため、(半角の)英数字(サイズがタテ横半分)との混合表示は困難であり、画面のスクロールや文字としてのプリントも不可能である。これでは本格的な漢字処理ができるとはいいがたい。

(3) パソコンの標準言語はBASICであり、8ビット型の標準OSはCP/Mであるから、BASICやCP/Mで漢字データがどう扱えるかも問題である。上記(2)の図形的表示のパソコンでは、BASICでは16進数コードで漢字を次のように扱えるだけの機種もある。

PUT(x,y),KANJI(16進数コード),PSET [1文で1字,PC8801]

PRINT@ 16進数コード,16進数コード,... [Micro-8]

これに対して、文字として漢字が扱えるパソコンのBASICでは、次のような文が可能である。

PRINT "東京":DATA 大阪 (IF800/30)

X\$\$=!電算機!+!業界!

INPUT !名前は!,N\$\$ (M243)

PRINT STRING\$\$ (10%,HEX("2121"));!電!

ただ、問題は、BASICにおける漢字機能の実現法が各社のパソコンでバラバラにこなっていて、統一がとれていないことである。

(4) 漢字を扱うとなると、ディスプレイの分解能は高くなければならないが、画面に表示できる点の数は、640×200点あるいは640×400点に向上している。各点あたり8色(3ビット)のカラー表示も可能で、このため48KB(キロバイト)あるいは96KBの表示記憶が主記憶とは別に実装されている。

(5) 漢字処理がからむと、応用プログラムも一般に大きくなるから、主記憶は64KB程度では不足で、8ビット型パソコンでも128KBあるいは256KBに拡大されている。しかし、8ビット機では、アドレス空間は64KBしかないから、IF800/30を例にとると、図1のようにプログラムでまとめて使えるメモリ領域は32KBのバンク単位ということになり、プログラミング上かなりの制約となる。これの解決には、仮想記憶が欲しいところである。96~128KBの主記憶でマイクロソフトBASICを使う場合。ユーザ・プログラムでとれるデータ領域は実測してみると次の通りである。

IF800/30	17KB(128KB)
PC8801	24KB(96KB)
Micro 8	30KB

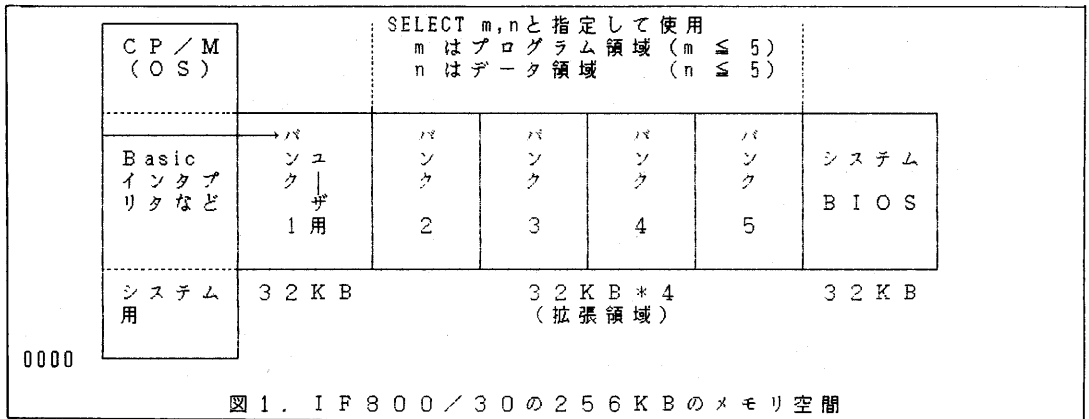


図1. IF800/30の256KBのメモリ空間

(6) 漢字用制御コードの使用はまだ必ずしも標準化されていない。われわれはJICST(科学技術情報センター)用ポータブル漢字端末(サンヨー、日立、沖)で使われているのと同じ次のような制御コードを用いている(後の図4も参照)。

制御コード	JISコード	同左用EBCDICコード	KEISコード
EBCDICコード開始	1B2848	274DC8	0A41
漢字コード開始	1B2461	275B7C	0A42
全角スペース(文字扱い)		4040	A1A1

ここでKEISコードは日立の大型OS(VOS3)で使われるホスト内のコードでこれを端末とやりとりするときは、EBCDICコードに直さなければならないので、参考までにここにあげておいた。これについては後述する。

(7) 漢字文字列のパソコン内部での表現も必ずしも統一されていない。2バイトの漢字コード自体はJISコードであるが、そうしたコード列を指定するのに、上記のような漢字コード開始というモード指定をする方式と、IF800/30のように、各漢字コードの先頭に1バイト(16進で01)を付加する方式とがある。後者は、1バイ

ト・コードと2バイト・コードを混ぜるときに便利であるが、各漢字コードが3バイトと長くなる欠点をもっている。しかしこの場合も、ファイルの中では2バイト・コードにする手はある。

3. BASICによるパソコン端末用ソフトウェア

最近のパソコン用のマイクロソフトBASICは大幅に拡張されているから、データ通信速度が300ビット/秒であれば、端末用ソフトウェアはBASICで書くこともできる。またこれでスピードが間に合わないときは、ホストがVOS3の場合、VOS3に対して

```
terminal filler (nT)      [ n はある数 ]
```

というコマンドを出せば、ホストが1行出力する毎に、nミリ秒待つことになるので、伝送速度を実質的に下げることが可能である。

パソコンを単に端末として使うための基本的なプログラムは、IF800/30の場合図2の通りである。このプログラムで、一般のBASICに余りみられない表現は次のものである。

open	COM1:3E71は通信ポートのファイル名
loc (1)	受信バッファ中の文字数
input \$(loc (1),1)	受信バッファ1の内容を入力
print #1,....	送信
while ~ wend	ループ
out n,data	ポートnにdataを出力する
on key gosub	機能キー1が押されたらサブルーチンへ
on error goto	エラーが発生したら飛べ

この場合、機能キー1はBREAK信号(端末からの割込み)の送出的のために使っている次に、図3はすべての漢字を画面に表示するためのプログラムである。これでは、すべての漢字を16進数で指定すればよい。

さて、以上のような基本機能を使うことにすると、パソコンを漢字端末にするためのプログラムは容易に書けることになる。われわれのプログラムでは、パソコンに備わっている10個の機能キーを活用することにし、ホストのTSSを使っている最中に機能キーを押すと、次のようなことが出来るようにした。

キー1.....	あるファイルの内容を1行ずつホストへ送信するモードにはいる。 このときファイル名はユーザに問い合わせる。1行送信のきっかけは ベル・コードを受信したとき、あるいはキーボードからベル・コード を投入したときとする。このモードにするときは、ホスト側では エディタの入力モードにしておくのが普通である
キー2.....	上記ファイルからの送信を中止する
キー3.....	メモリに格納した受信情報を画面に表示する
キー4.....	メモリ内容をファイルに格納する〔メモリが一杯になったら、自動 的にファイルに入れるようにしてもよい〕
キー5.....	上記のファイルを閉じて、ファイルへの受信を止める
キー6.....	受信ファイルの名前を入れる
キー7.....	BREAK信号を送出する
キー8.....	機能キーの使い方を表示する(helpメニュー)
キー10....	このプログラムの実行を止める

```

100 REM-----term1-----
110 LF$=CHR$(10): BELL$=CHR$(7)
120 OPEN "COM1:3E71" AS 1
130 PRINT "...Entering terminal mode.."
140 PRINT "    300 bps, even parity"
150 PRINT "-----"
160 KEY ON
170 ON KEY GOSUB 380
180 ON ERROR GOTO 430
190 WHILE 1
200 C$=INKEY$
210 WHILE C$<>" "
220 PRINT C$;
230 PRINT #1,C$;
240 C$=""
250 WEND
260 WHILE LOC(1)<>0
270 S$=INPUT$(LOC(1),#1)
280 C$=MID$(S$,LEN(S$),1)
290 IF C$<>LF$ THEN PRINT S$;
300 WHILE C$=BELL$
310 LINE INPUT A$
320 PRINT #1,A$
330 C$=""
340 WEND
350 WEND
360 WEND
370 REM-----send break----
380 OUT 33,&H28
390 FOR I=0 TO 255: NEXT I
400 OUT 33,&H27
410 RETURN
420 REM-----ignore error--
430 RESUME 200

```

図2. パソコンを普通端末にするためのプログラム (IF800-30)

```

100 REM-----term-----
110 FS$=CHR$(28): KS$=CHR$(1)
120 PRINT "...Entering terminal mode..."
130 OPEN "COM1:3E71" AS 1
140 WHILE 1
150 C$=INKEY$
160 WHILE C$<>" "
170 PRINT C$;
180 PRINT #1,C$;
190 C$=""
200 WEND
210 WHILE LOC(1)<>0
220 S$=INPUT$(LOC(1),#1)
230 IF S$=FS$ THEN S$=KS$
240 PRINT S$;
250 WEND
260 WEND

```

K=0A42A3B1A1A5A1A1A3B1BFCD5C5F6A4EAA4CEA5D5A5A1A5A4A5EBC1EDCDC6CECCA4CFA1A2
E=1C007BF11C004F6C1C004F4F1C007BF11C006FD41C00C59C1C005B701C005BD51C006CE4
1. 1人当りのファイル総容量は、1MB以内にしてください。
K=0A42A4B3A4ECA4F2B1DBA4A8A4C6A4A4A4EBA4C8A4ADA4CFA1A2A5BBA5F3A5BFA1BCC2A6
E=1C005BF31C005B721C005B781C00F14A1C005B4D1C005BC61C005B5B1C005B711C005BC8
これを越えているときは、センター側でファイルを消します。
K=0A42A1A5A3D3A3D0A1A1A3B140
E=1C004F6C1C007BE21C007BD71C004F4F1C007BF11C004F4F1C004F4F1C004F4F1C004F4F

このプログラムで、漢字データを扱うときに、多少厄介なのが、漢字コードの扱いである。これは次のようになる。

(a) 漢字コードの受信

データの先頭に漢字開始コードがあったら、以後の表示は漢字コードで行い、漢字終了コードがきたら、英数モードに戻す。行の先頭では英数モードを仮定する。ホストがVOS3の場合、送られてくる漢字データは普通には図5の一番下の形をしている。漢字をIF800/30に表示するときは、図4で見えるように3バイト・コードにしなければならない。この点は、ホストから出力する漢字コードを3バイト・コードにしておくともモード切換えの必要がなく楽である。

図3(a)はホストから3バイト・コードを出すときのパソコン用プログラム、同(b)はそれによる受信例である。

(b) 漢字コードの送信

データの先頭に、漢字開始コードをつける。ホストの漢字エディタでは、行番号なし(nonum)の扱いで入力することにすれば、準備はこれだけでよい。

図3(a)

図3(b)



```
list
100 REM-----kanji-----
110 K1$=CHR$(I)
120 FOR I=48 TO 255
130 FOR J=32 TO 127
140 PRINT K1$+CHR$(I)+CHR$(J);
144 N=N+1
145 IF N=32 THEN N=0: PRINT I;J
150 NEXT J
160 NEXT I
Ok
run
```

■ 亜	啞	娃	阿	哀	挨	始	逢	葵	茜	種	憑	握	渥	旭	葦	麟	粹	庄	幹	級	宛	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	妯	48	63		
■ 粟	裕	衣	謂	遠	隱	數	宮	延	院	荏	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	48	95	
■ 雲	院	荏	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	48	127
■ 園	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	49	63
■ 壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	49	95
■ 壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	49	127

図4. すべての漢字を表示するBASICプログラム (IF800-30)
 [各漢字は01コードを先頭とする3バイト・コードになっている]

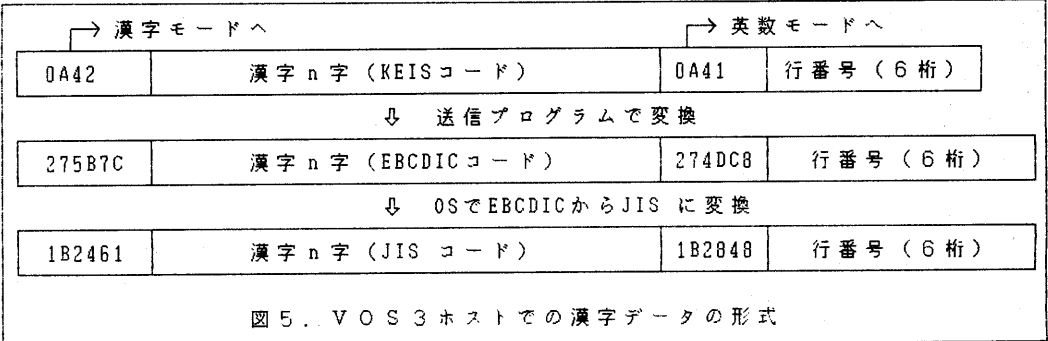


図5. VOS3ホストでの漢字データの形式

こうしたオンライン用プログラムとは別に、われわれのパソコン端末には、オフラインの漢字データ作成プログラムも用意した。このためには、パソコンには漢字入力装置（関東電子Logitec K-505）を図6のようにつないで使っている。これはタブレット型の装置で、各漢字はペンのようなものでタブレット上のマス目の位置を押すことで入力する。JIS第一水準の漢字の入力が可能である。このタブレットから出るコードはJISコードではなく、その各バイトの最上位を1にしたJEF（富士通）およびKEIS（日立）コードである。このため、このタブレットから漢字データを入力してフロッピー・ディスクへ格納するプログラムでは次のようにして、このビットを落す細工がある。

```
k$ = INPUT(2, #1)
k1$ = LEFT$(k$, 1)      : k2$ = RIGHT$(k$, 1)
k1  = ASC(k1$) AND &H7F : k2  = ASC(k2$) AND &H7F
kcode = k1 * 256 + k2
```

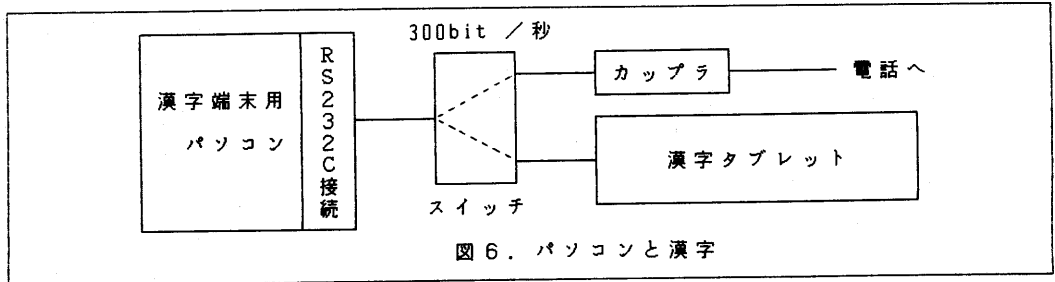


図6. パソコンと漢字

要するにここでやったのは、漢字タブレットから出てくる2バイト・コードを左右各バイトに分離し、各々の最上位1ビットを&H7F(16進数の7F)というマスクをかけて落し、再び2バイトの整数にまとめることである。ここでASC(c\$)は、文字cのASCIIコードを求める関数である。私見をいえば、こうした漢字タブレットの漢字コードはJISコードにすべきであり、その代りにJEFやKEISに合わせたのは、設計ミスといたいところである。

このオフライン・プログラムは本来なら、ローマ字漢字変換プログラムにしたいところであるが、現在それはまだ開発していない。

以上のべたBASICプログラムの開発からえた教訓は次の通りである。

- (1) 漢字を図形としてしか扱かわないパソコンは、漢字の表示に座標を意識しなければならぬので表示に時間がかかり、画面のスクロールもできないので、漢字端末には向かない。
- (2) 漢字を3バイト・コードで表すパソコンでは、ホストからの少なくとも出力のときは、3バイト・コードを送ることにすれば、パソコンでの受信はただそのまま行えばよく、モード切換えをしなくてすむので、楽である。この意味でパソコン用の入出力には3バイト・コードを使うのが意外によいかもしれない。

4. UCSD-Pascalによるパソコン端末用ソフトウェア

パソコン関係で最も互換性の高いPascalシステムはUCSD-Pascalシステムである。これにはUCSD独特のOS、ファイル・システム、pコードへのPascalコンパイラ、pコードのインタプリタなどが含まれている。この種のプログラムは、Pascal本来の機能のみではもちろん書けないので、次のような拡張機能(手続)を使っている。

unitstatus(n,st,l).....機器nの状態を取り込む(配列要素st(0)には、入力のバイト数が入る)

unitread(n,s,k,,).....機器nから配列sにkバイトを入力

unitwrite(n,s,k,,)....機器nへ配列sからkバイト出力

UCSD Pascalでは機器番号は次のように割当てられている。

1 = 画面 2 = キーボード 7 = リモート入力 8 = リモート出力

このPascalでどうしても出来ないのが、ブレーク信号送出や伝送速度・パリティなどの設定のようなハードウェア機能の直接制御である。そこでこの部分だけはやむなく、baud, breakon, breakoffといった名前のアセンブラ・プログラムを書いて、Pascalプログラムとリンクさせた。この場合、Pascalプログラム内では、次のような宣言が必要である。

```
procedure baud ; external ;
```

なお、伝送速度やパリティは、ユーティリティを使って、Pascalプログラムの外でセットすることは可能になっている。

次に図7にすべての漢字を表示するPascalプログラムの例を示す。このプログラムでは、Micro 8用に準備されている次の手続を使っている。

```
Kanji(x,y,kcode,color)
```

これは図形座標(x,y)の位置に16進コードkcodeの漢字を色colorで表示するための手続きである。

```

program pkanji;
var i,j,n,x,y,kcode,color: integer;
procedure kanji(x,y,kcode,color: integer);external;
begin x:=0; y:=0;
write(chr(27),chr(69));
for i:=48 to 255 do begin
for j:=32 to 127 do begin
kcode:=i*256+j; x:=n*17;
x:=n*17;
kanji(x,y,kcode,color);
n:=n+1;
if n=32 then begin n:=0;
x:=0; y:=y+17;
if y>183 then begin y:=0;
write(chr(27),chr(69)) end;
end;
end;
end;
end.

```

図7. すべての漢字を表示するPascalプログラム (Micro8)

5. 大型ホスト用の漢字処理ソフトウェア

東大センターの大型ホスト (HITAC M280H / M200H) のVOS3のもとには、大学で開発した次のような無手順端末向けの漢字処理ソフトウェアがある。

Klist..... 漢字ファイルの内容を、図5のようにKEIS→(最上位ビットを取り除く)→JIS→EBCDICの順にコード変換して端末へ送る (EBCDICコードはOSでJISコードに変換されて端末へ送られる)

KED (平賀)..... QED風の漢字エディタ。入力はタブレットからの漢字コードでもよいが、Tコードと呼ばれる2文字コードでもよい。

RK (前野)..... バッチ的なローマ字漢字変換機能をもつエディタと清書プログラム。Pascalで記述されている。

Kedit (野本)..... TSS用のローマ字漢字変換エディタ。開発中。PL/Iなどで記述されている。

okilist..... ファイル中の漢字を3バイト・コードで出力する。

これらのソフトウェアを開発する上で問題となった点を次にあげる。

- (1) 上述のように、VOS3の内部コードがKEISのため、KEIS↔JIS↔EBCDIC↔JISといったバカげたコード変換がある。われわれのプログラムでもKEISコードを使わざるを得ないのは、KEISにしておかないと、レーザプリンタへの清書出力とか、XYブロックへの漢字出力など、メーカーのサポートするソフトウェアが使えないからである。
- (2) EBCDICコード系では英小文字のコードが標準化されていない。東大システムではカナを割当てた後の空き場所を使っている。
- (3) 漢字用の制御コードも標準化されていない。VOS3では01(SOH)のように出力できない制御コードがある。(このため okilistコマンドでは、01の代わりに他の制御コードを使い、IF800-30で受信したときにそれを01に変えている)
- (4) KEISでは、空白はA1A1だが、行の余白では4040(EBCDICの空白)も使われているので、プログラム上は空白のコードが二つあるとする必要がある

6. お わ り に

パソコンを大型機の端末機に仕立てる試みをして痛感することは、現状では、大型機のパソコンとの相性が極めて悪いということである。まず、パソコンについては、漢字能力が未熟だということ以外に、パソコンでは次のような問題点がある。

- (1) 漢字コード(特にその制御コード)が大型機と合わない。
- (2) 3バイト・コードを使う機種がある。しかしパソコンのソフトウェアを単純にする上からいえば、3バイト・コードの採用はいいかもしれない。
- (3) 漢字タブレットには、最上位ビットを落す必要のある機種がある。
- (4) 漢字の扱いが(特にBASICやOSで)標準化されていない。

一方、大型機の方にも次のような問題がある。

- (1) パソコン端末への標準サポートがない。必要なプログラムはユーザ側で用意する必要がある。とくに3バイト・コードを使うとなるとそうである。
- (2) 漢字を扱うときは、実効的に2~3段のコード変換がいる。
- (3) 制御コードの使い方が標準化されていない。

しかし、パソコン自体はプログラマブルであるから、プログラムさえ作れば、パソコンをインテリジェントな漢字端末に仕立てることは一応可能である。これで和文のワード・プロセッシングを行う場合、使い勝手は専用のワード・プロセッサほどよくはないが、レーザ・プリンタへの出力など大型機の機能の活用はできる。

最後に、パソコンの漢字機能については、次のような点の検討が必要である。

- (1) BASICやPascalやOSでの漢字の扱い方を標準化すること。
- (2) ローマ字漢字変換機能をOSの基本機能としてもたせ、BASICやPascalでそれが使えるようにすること。〔200KBないし1MBの辞書があるので大変だが、近い将来これの基本部分はROM化できよう〕
- (3) 漢字の表示や印字(特に印字)は16×16点でなく、24×24点以上にする。
- (4) 漢字の入出力に3バイト・コードを使うことの可否。

《 謝 辞 》

パソコンの漢字端末化にあたり、ご協力頂いた沖電気、コンピュータランド、内田マイコン・スクール、富士通、日本電気などの関係者に感謝したい。本稿はOASYS100Jで作成したことを付記する。

参考文献

1. 石田：漢字テキストのJIS漢字端末およびXYプロッタへの出力法，東大センターニュース，13巻9・10月号，pp.65～70(1981)
2. 石田：JIS漢字端末へのKEISファイルへの出力法，東大センターニュース，13巻11月号，pp.13～17(1981)
3. 平賀：日本語用テキストエディタKEDの紹介，東大センターニュース，13巻12月号，pp.39～44(1981)
4. 平賀：VOS3上での日本語処理——テキストエディタKEDを中心に，個人メモ(1982年4月)
5. 前野：漢字入出力の紹介，東工大総合情報処理センター，No.92，pp.4～6(1981.12)

(1) okilist コマンドによる出力例 (IF800-30)

>>okilist

*PROGRAM UNITS = 2, NO DIAGNOSTICS GENERATED

. PL 60

別紙

. CE

VAX/UNIX 利用上の注意

. SP 2

. HI 2

1. 1人当りのファイル総容量は、1MB以内にして下さい。
これを越えているときは、センター側でファイルを消します。

. SP 1

. HI 0

. HI 2

2. パスワードは、次のようにして、自分でセットして下さい。

. IN 7

\$#Upasswd 課題番号#U

New password: #U #U

(2) ソード・パソコンの漢字 BASIC プログラムの例

-- M243 KBASIC Ver.-00A Program list Page -- 01

```

01 100 註 ソード電算機・M243漢字システム
02 註 見積書発行のプログラム例です。
03 110 Integer 条件, 番号, 行数, 真, 1 -- 変数型宣言
04 Decimal 数量, 単価, 金額, 合計
05 Japan 品名, 一行, 返事
06 Dim 数量(20), 単価(20), 金額(20), 品名(20,20), 一行(127), 書式$(20)
07 120 /
08 Gosub 変数の初期設定を行なう。
09 While 条件 = 真 loop
10 Print clear, cursor(0,5),
11 Input line | 品名は? |, 品名(番号)
12 If len(品名(番号)) <> 0 then
13 Input | 数量は? |, 数量(番号)
14 Input | 単価は? |, 単価(番号)
15 Let 金額(番号) = 数量(番号)*単価(番号)
16 Let 合計 = 合計+金額(番号)
17 Let 番号 = 番号+1
18 Else
19 Input |入力を終わりますか? (はい/いいえ) |, 返事
20 If 返事 = |はい| then exit loop
21 End if
22 End if
23 End loop
24 Gosub 見積書の前文をプリンターに出力する。
25 Gosub 入力された表をプリンターに出力する。
26 End
27 130 /
28 $ 変数の初期設定を行なう。
29 Let 真 = 1
    
```