

# T S S 漢 字 エ デ イ タ の 設 計

荻野綱男 (東京大学文学部)

## 1. はじめに

最近 は 漢 字 かな まじり 文 の 入 力 が 手 軽 に で き る よ う に な り、各 研 究 機 関 等 で も 磁 気 デ ィ ス ク や 磁 気 テ ー プ の 形 で そ れ ら の デ ー タ を 保 管 し 必 要 に 応 じ て 処 理 を 行 な う よ う に な っ て き た。し か し な が ら、ど の よ う な 入 力 方 法 に よ っ て も、計 算 機 に 入 力 さ れ た デ ー タ に 誤 り は つ き も の で あ る。そ こ で、今 後 の 漢 字 かな まじり 文 の デ ー タ 処 理 に は 校 正・修 正 の 作 業 が 重 大 な 問 題 に な っ て く る。各 メ ー カ か ら は 漢 字 デ ィ ス プ レ イ が 発 売 さ れ た り そ れ な り の 編 集 用 ソ フ ト ウ ェ ア が 提 供 さ れ る よ う に な っ て き た。

今 回 は そ う い う も の を 全 く 使 れ な い で、汎 用 の グ ラ フ ィ ッ ク・デ ィ ス プ レ イ を 用 い た、デ ー タ の 修 正 の た め の 漢 字 エ デ イ タ に つ い て 報 告 す る。こ れ に よ り 漢 字 デ ー タ の 修 正 作 業 の 特 徴 を 明 ら か に し た い。

## 2. 対象となったデータ

電 子 技 術 総 合 研 究 所 の デ ィ ス ク・フ ァ イ ル に は、新 明 解 国 語 辞 典 (三 省 堂) が 一 冊 全 部 入 っ て い る<sup>(3)</sup>。し か し 調 べ て み る と デ ー タ 作 成 時 の 誤 り が 可 成 り 多 く、そ れ を 修 正 す る 必 要 が あ っ た。

デ ー タ は 5 個 の ラ ン ダ ム・フ ァ イ ル に 入 っ て い る。行 数 は

$$23, 175 + 28, 787 + 30, 285 + 32, 430 + 6, 037 = 120, 714 \text{ 行}$$

で あ る。1 行 は 64 字 な の で 全 体 字 数 は

$$120, 714 \times 64 = 7, 725, 696 \text{ 字}$$

に な る。た だ し 後 の 例 を 見 れ ば わ か る よ う に 実 際 に は 空 白 が た く さ ん 入 っ て い る の で、実 質 は 4, 278, 072 字 で あ る。さ ら に 行 の 先 頭 2 カ ラ ム 目 か ら 8 カ ラ ム 目 ま で は 行 番 号 が 入 っ て お り、本 文 は 10 カ ラ ム 目 か ら は じ ま る の で 行 番 号 の 字 数 を 除 け ば、

$$4, 278, 072 - 120, 714 \times 7 = 3, 433, 074 \text{ 字 になる。}$$

デ ー タ の 字 数 を 後 述 す る 印 字 速 度 と の 関 連 で 字 種 別 に 集 計 す る と 次 の よ う に な る。

	}	空 白 3, 447, 624		
全 体 7, 725, 696		}	アラビア数字 845, 267 (うち 844, 998 は 行 番 号)	
	ひらがな 1, 314, 370			
	非空白 4, 278, 072		カタカナ 169, 980	
	漢 字 1, 526, 998		}	ベクトル表示 1, 518, 086
	記 号 421, 457			ドット表示 8, 912
			}	ベクトル表示 396, 197
				ドット表示 25, 260

### 3. ハードウェア環境

○ コンピュータ 電子技術総合研究所 T O S B A C - 5 6 0 0  
○ 端末 ソニーテクトロニクス製グラフィック・ディスプレイ 4 0 1 4

画面の大きさ 縦 29 cm × 横 38 cm

標準文字サイズ 縦 10 mm × 横 8 mm

一画面の文字数 縦 19 行 × 横 32 文字 ( 6 0 8 字 )

ただしコマンドによって文字サイズが変えられ、実用上標準の半分くらいでも充分読める。この場合縦 36 行 × 横 64 字 = 2304 字が一画面に表示される。

文字表示速度 ( 9600 ボーの場合 )

#### A. 1000 字を表示する時間

数 字 ( ベクトル表示 ) 47 秒

ひらがな ( ベクトル表示 ) 60 秒

カタカナ ( ベクトル表示 ) 50 秒

漢 字 ( ベクトル表示 ) 100 秒

漢 字 ( ドット表示 ) 960 秒

#### B. 辞書データの一部を実際に表示する時間

サイズ 5 で一画面 36 行 × 64 字とした場合

1 ページ ( 1145 字 ) ( 1 行平均 35.4 字入っている )

N O C O L U M N 状態 66 秒

P C O L U M N 状態 82 秒

C O L U M N 状態 150 秒

○ 漢字入力装置 日本電気漢字システム製漢字タブレット C 5 1 2 1 J

### 4. ソフトウェア環境

TSSで動くFORTRANでコーディングしてある。ソース・プログラムの大きさは約 2300 行で、実行時の主記憶は約 30Kワードを要する。プログラムは EDIT 2 という名前のファイルに入っており、端末でログオン手続きをしたあと、計算機が SYSTEM? と聞いてきた時、 / O G I N O / EDIT 2 と答えることにより呼び出して使用できる。以下、この漢字エディタを EDIT 2 と呼ぶことにする。

### 5. 端末使用時間

EDIT 2 にはいくつかの計測機能が組み込んであるが、最近 100 回の使用の平均接続時間は 61.3 分で、その平均 CPU 時間消費量は 93.4 秒であった。つまり、端末操作者は、一度 EDIT 2 を使いはじめると約一時間連続使用するわけである。その間 CPU は 1 分半しか使っていないので、画面をじっと見る時間がかなりあることをうかがわせる。ちなみに、最高速度でデータを表示すると CPU 全体の 30% ほどを使うようである。

### 6. EDIT 2 のコマンド体系とその動作

エディタの使いやすさはコマンド体系に大きく左右される。EDIT 2 の場合は、その主な利用者が TOSBAC-5600 のテキスト・エディタ<sup>(4)</sup>に親しんでいたことや上記のような環境で使用されることにより、コマンド体系は TOSBAC-5600 のテキスト・エディタのコマンド体系に合わせるようにした。コマンド形式が似ているだけでなく、英字の小文字が全面的に使えること、1 行中に空白で区切った

複数のコマンドが入れられることなども、TOSBAC-5600のエディタを見習った結果である。

TSS端末による校正作業が始まってしばらくしてから、筆者はEDIT2のコマンドの各々の使用頻度をカウントする機能をつけて出現数を記録した。1年目の作業の終了時点でその結果を報告した<sup>(1)</sup>が、2年目の作業にもこのカウントは行なわれ、また小さな追加コマンドもあったので、それを合わせて次にコマンド一覧を掲げる。ここでnは自然数、cは7で述べる文字列表現である。

コマンド形式	出現数			意味
	合計	旧使用	新使用	
P	25928	15163	10765	ポインタの指している行(カレント行)を印字する。
P:n	2887	2504	383	第n行にポインタを移動させ、その行を印字する。
P;n	251	196	55	カレント行からn行印字する。
PS:c	562	553	9	cという文字列を含む行を探して、その行を印字する。
PS:c;n	58	54	4	cをn回探して、それらを印字する。
PS#n	0	0	0	第nカラムの文字を印字する。
PS#n <sub>1</sub> , #n <sub>2</sub>	1	0	1	n <sub>1</sub> カラムからn <sub>2</sub> カラムまでの文字を印字する。
F	5671	4074	1597	ポインタを次の行に進める。
F:n	8074	2209	5865	ポインタを第n行に進める。
F;n	5321	4215	1106	ポインタをn行先に進める。
FS:c	3	0	3	cを含む行にポインタを進める。
FS:c;n	0	0	0	n回目のcを含む行にポインタを進める。
B	878	670	208	ポインタを1行後に戻す。
B:n	52	14	38	ポインタを第n行に戻す。
B;n	501	420	81	ポインタをn行後に戻す。
I	29	21	8	カレント行の次に行を挿入する。
I:n	6	6	0	第n行の次に行を挿入する。
IS:c <sub>1</sub> :c <sub>2</sub>	3784	2192	1592	c <sub>1</sub> の次にc <sub>2</sub> を挿入する。
IS#n:c	1310	928	382	第nカラムの次にcを挿入する。
D	171	124	47	カレント行を削除する。
D:n	3	3	0	第n行を削除する。
DS:c	1003	701	302	cを削除する。
DS:c <sub>1</sub> , c <sub>2</sub>	163	154	9	c <sub>1</sub> からc <sub>2</sub> までを削除する。
DS#n	502	355	147	第nカラムの文字を削除する。
DS#n <sub>1</sub> , #n <sub>2</sub>	322	264	58	第n <sub>1</sub> カラムから第n <sub>2</sub> カラムまでを削除する。
DS#n <sub>1</sub> , n <sub>2</sub>	6	6	0	第n <sub>1</sub> カラムからn <sub>2</sub> 文字を削除する。
R	24	20	4	カレント行を置き換える。
R:n	41	29	12	第n行を置き換える。
RS:c <sub>1</sub> :c <sub>2</sub>	15754	7741	8013	c <sub>1</sub> をc <sub>2</sub> に置き換える。
RS:c <sub>1</sub> , c <sub>2</sub> :c <sub>3</sub>	36	33	3	c <sub>1</sub> からc <sub>2</sub> までをc <sub>3</sub> に置き換える。
RS:c <sub>1</sub> ;n:c <sub>2</sub>	1567	1456	111	c <sub>1</sub> をn回c <sub>2</sub> に置き換える。
RS#n:c	1727	1262	465	第nカラムの文字をcに置き換える。
RS#n <sub>1</sub> , #n <sub>2</sub> :c	201	150	51	第n <sub>1</sub> カラムから第n <sub>2</sub> カラムまでをcに置き換える。
RS#n <sub>1</sub> , n <sub>2</sub> :c	14	13	1	第n <sub>1</sub> カラムからn <sub>2</sub> 文字をcに置き換える。

CUT	1 3 8	9 5	4 3	カレント行をメモリに移し、削除する。
CUT : n	2 4	2 4	0	第 n 行をメモリに移し、削除する。
CUT ; n	2 2	7	1 5	カレント行から n 行分をメモリに移し削除する。
CUTS : c	8	7	1	c をメモリに移し削除する。
CUTS : c <sub>1</sub> , c <sub>2</sub>	3 8	3 1	7	c <sub>1</sub> から c <sub>2</sub> までをメモリに移し削除する。
CUTS # n	2 6	1 3	1 3	第 n カラムの文字をメモリに移し、削除する。
CUTS # n <sub>1</sub> , # n <sub>2</sub>	1 6 9	4 3	1 2 6	n <sub>1</sub> から n <sub>2</sub> カラムまでをメモリに移し、削除する。
CUTS # n <sub>1</sub> , n <sub>2</sub>	1	0	1	第 n <sub>1</sub> カラムから n <sub>2</sub> 文字分をメモリに移し、削除する。
(注) メモリに入れるのみで削除を行わない COPY コマンドも 8 種類あるが、全く使われなかった。				
PASTE	1 1 6	2 0	9 6	メモリの内容をカレント行の次に挿入する。
PASTE : n	1 0	1 0	0	メモリの内容を第 n 行の次に挿入する。
PASTES : c	7 2	5 2	2 0	メモリの内容を c の次に挿入する。
PASTES # n	1 6 8	9 7	7 1	メモリの内容を第 n カラムの次に挿入する。
O # n	4 0 7	3 3 9	6 8	第 n カラムの文字のコード番号を印字する。
O ; n	1	1	0	カレント行の 1 ~ n カラムまでのコード番号を印字する。
OLD	5 7 7	3 2 7	2 5 0	カレント行を修正以前の内容に戻す。
OLD : n <sub>1</sub> ; n <sub>2</sub>	2 6	1 4	1 2	第 n <sub>1</sub> 行の内容を n <sub>2</sub> 回目の修正の以前の内容に戻す。
SHIFT n	1 3	0	1 3	画面上で第 n カラムよりも右に漢字の印字を行なう。
SIZE n	5 2	3 5	1 7	漢字の大きさを n にする。(標準値は 1 0)
PAGE	9 0	5 6	3 4	画面をクリアして最上段の行から書く。
TEST	1 6	7	9	見かけの修正だけ行ないファイルは修正しない。
NOTEST	3	3	0	TEST コマンドを取り消す。(標準)
VERIFY	1 6 5	8 8	7 7	以後のコマンドの実行結果を表示する。(標準)
NOVERIFY	4 5 4	2 6 6	1 8 8	以後のコマンドの実行結果は確認表示しない。
COLUMN	2 4 4	2 3 1	1 3	各漢字の下にカラム番号を印字するようにする。
PCOLUMN	1 1 3	新コマンド	1 1 3	漢字の下に 5 文字置きにカラム番号を印字するようにする。
NOCOLUMN	2 5 7	2 3 4	2 3	漢字の下にカラム番号を印字しない。(標準)
HELP	2	1	1	コマンド形式一覧表を印字する。
HISTORY	1 3	7	6	過去のコマンドのうち最新のもの 1 0 0 個を印字する。
EDIT	1 8 0	1 3 5	4 5	別のファイルの編集に切りかえる。
DONE	3 2 1	2 0 5	1 1 6	EDIT 2 を終了させる。
合 計	8 0 5 7 6	4 7 8 7 8	3 2 6 9 8	

これらのうちキーワードの長いものは先頭 4 文字だけでもよい。たとえば、NOCOLUMN は NOCO でも NOCOL でもよい。PASTES は PASTS とはできるが、PAST とはできない。

## 7. 文字列の表現

6 のコマンド形式一覧中の C についてここにまとめておく。汎用のグラフィック・ディスプレイ端末からコマンドを入れるため、英数字特殊記号のみの普通のキーボードで、いろいろな文字を表現するように考えた。

- 文字列の両端は、文字列中に含まれない任意の一字(ただし“ \ ”は除く)でくくる。
- 文字列はいくつかの副文字列の並びから成る。副文字列の間は“ \ ”で区切る。

副文字列はそれぞれがひとつの字種をもっている。字種は副文字列の先頭一字だけを見て判定される。字種には、ひらがな、カタカナ、ASCII、漢字コードの4種類がある。

○ひらがな副文字列は、Hまたはhで始まり、そのうしろにローマ字つづりが続くものである。ローマ字つづりの方式は、ヘボン式、訓令式、新日本式（服部式）のいずれでもよいし、それらの混合したものでよい。また、小さい“っ”は“Q”で表わしてもよいし、小さい“ゃゅょ”は“.A.U.O”と表記してもよい。“FAFIFU”は“ふあふいふ”と見なされる。

○カタカナ副文字列は、Kまたはkで始まり、そのうしろにローマ字つづりが続くものである。以下ひらがな副文字列の規則に準ずる。

○ASCII副文字列は、Aまたはaで始まり、そのうしろに端末が持っている文字が続くものである。ただし“\”は入力できない。

○漢字コード副文字列は、十進数字列である。4ケタより長い数字列は4ケタごとにひとつの漢字コードを表わすものと見なす。4ケタより短いコードを使用する時は、カンマ“\”で区切る。Oコマンドの出力がそのまま入力に使える。

○H・h・K・k・A・a・数字以外で始まる副文字列は、EDIT2によってASCII文字列と見なされる。

○数字列中に数字以外が出てきた時、ひらがな副文字列やカタカナ副文字列中にローマ字として使われない英数字記号が出てきた時、EDIT2によって自動的にそこに“\”があると見なされて解釈される。

コマンドの例

RS:/HSHA/:/KSYA /	“しゃ”を“シャ”に置き換える。
rs:/ksa\239/:/hsa\197/	“サ・”を“さ】”に置き換える。
RS:/KSA239/:/HSA197/	同上
RS::@:::##:	“@”を“#”に置き換える。
IS#9:/*\HATCHI 239/	9カラムの次に“* あっち .”を挿入する。
ds: hciq	“ちっ”を削除する。

次に、I・R・IS#24（挿入する文字の指定がない）・RS#32, #38などのコマンドを入れた場合であるが、EDIT2は

enter  
\*  
と打ってくるので、ここに上に述べた文字列の表現の、両端にデリミタを入れない形を入れることによって、ひとつのコマンドとして認められる。またこの方法では\*の次に端末のキーボードから“B”を入力することで漢字鍵盤からの入りに切りかえることができる。

8. 漢字データ修正の特徴

旧使用 (コマンド総数 47878)			新使用 (コマンド総数 32698)		
順位	コマンド形式	出現数 (出現率%)	順位	コマンド形式	出現数 (出現率%)
1	P	15163 (31.7)	1	P	10765 (32.9)
2	RS:c <sub>1</sub> :c <sub>2</sub>	7741 (16.2)	2	RS:c <sub>1</sub> :c <sub>2</sub>	8013 (24.5)
3	F;n	4215 (8.8)	3	F:n	5865 (17.9)
4	F	4074 (8.5)	4	F	1597 (4.9)
5	P:n	2504 (5.2)	5	IS:c <sub>1</sub> :c <sub>2</sub>	1592 (4.9)
6	F:n	2209 (4.6)	6	F;n	1106 (3.4)
7	IS:c <sub>1</sub> :c <sub>2</sub>	2192 (4.6)	7	RS#n:c	465 (1.4)
8	RS:c <sub>1</sub> ;n:c <sub>2</sub>	1456 (3.0)	8	P:n	383 (1.2)
9	RS#n:c	1262 (2.6)	9	IS#n:c	382 (1.2)
10	IS#n:c	928 (1.9)	10	DS:c	302 (0.9)

コマンド出現数をカウントしてみた結果、便利なコマンド・必要なコマンドはそれなりに多く使われていることがわかった。また新旧ふたつの表の比較から端末操作者の好み・傾向の変化などもうかがい知ることができた。

#### ① 漢字表示速度がおそい。

グラフィック・ディスプレイでは9600ボーでも漢字の表示に時間がかかる。汎用のハードウェアの限界かもしれない。

8000字種ほどの文字セットを使用しているが、そのうちよく使われる約4000字はベクトル表示を行ない、使用頻度の少ない特殊な漢字・記号はドット表示を行なうようになっている。3にあげたように、グラフィック・ディスプレイではベクトル表示の方が速いのであるが、それでもかなり時間がかかる。

コマンドの出現頻度にもそれがはっきり表われている。EDIT2では修正コマンドの実行後それを確認するためデータ一行分を表示する。これをベリファイと呼ぶ。一方ユーザはNOVERIFYコマンドによってベリファイをやめたり、またVERIFYコマンドによってベリファイをするようにしたりできる。EDIT2の使用回数はDONEコマンドの回数321にほぼ等しい。この中で、NOVERIFYコマンドは454回、VERIFYコマンドは165回使われているので、 $454 - 165 = 289$ 回はNOVERIFYの状態で行われていることになる。 $289 / 321 = 90\%$ はベリファイを使っていない。これを旧使用と新使用に分けると、旧では $178 / 205 = 87\%$ 、新では $111 / 116 = 96\%$ となり、ユーザがEDIT2に慣れるにつれてベリファイしなくなっていることがわかる。

ポインタを移動させるコマンドの場合も、P:nでは必ず第n行を表示するが、F:nならNOVERIFYの状態の場合行の表示を行なわない。上の表に見られるように、旧から新への変化では、P:nが使われなくなり、F:nが使われるようになってきた。またカレント行を表示するPコマンドの出現率は、旧より新の方が高い。実際、エディタのユーザはまず最初にNOVERIFYコマンドを入れ、次に、  
-F:144      RS:/HKA/:/HHAI/      P  
というようなコマンドを入れていくことが多かったようで、最近ますますその傾向が強くなっている。結論として、毎回ベリファイする必要はなく、1行に対していくつかの修正をして最後に1度だけ表示すればよいのである。

カラム番号の表示も、表示速度によってはしない方がいいわけである。旧使用では、COLUMNコマンドが231回、NOCOLUMNコマンドが234回使われた。これは、カラム番号を各漢字の下に入れると表示速度がおそくなるので、(実測値は3であげた)それをきらったものと思われる。実際、エディタのユーザは一画面のはじめの一行にだけカラム番号を入れ、他の行には入れない方針だったようだ。新使用時には、5カラムごとにカラム番号を表示するようにするPCOLUMNコマンドを用意した。これは113回使われ、DONEの回数116に比べるとほぼ必ずといっていいほどPCOLUMNコマンドを使っていたことがわかる。

#### ② カラム位置指定コマンドは場合によって便利である。

上の表のように、よく使われるコマンドというのはポインタを移動するコマンドと文字列の置換コマンドである。すでにあるデータの修正という性格からもこれはうなずける。さて、文字列<sub>1</sub>を文字列<sub>2</sub>に置き換えるコマンドを考えてみよう。ここで文字列<sub>1</sub>をそのまま指定することは漢字を入力することと同じ手間な

ので、カラム位置指定の方が楽なわけである。文字列<sub>1</sub>を指定するのは、それがひらがな、カタカナ、ASCII文字の連続で端末から入力しやすいものが多い。

なお、注目すべきは、ASCII文字セットにない特殊記号（・、「」など）はくり返し出現するためにユーザがコードを自然に覚えてしまい、カラム位置より文字コードの指定の方が楽だと感じるようになったことである。これはユーザの習熟にも大いに関係がある。結果として、カラム位置指定形式のコマンドは予想ほどには多く使われていない。

### ③ キーインミスのダメージはかなり大きい。

コマンドの誤入力によってデータが破壊されると、その回復作業は大仕事になる。ファイルのバックアップはもちろん必要であるが、うっかり文字列の置換の指定をまちがえただけでもちよつとした作業になってしまうことがある。これはエディタ一般についても言えるように思う。エディタのコマンドはプログラムのテストランに当たるものがなく、いつでも即実行されるものなのだから。

EDIT2ではキーインミスからの回復のために、OLDコマンドがある。カレント行を修正している時にこのコマンドを入れると、修正を始める前の状態に戻るのである。さらに、その行の修正を終えて先に進んでいってあとで誤りに気づいた時も、OLD:nで回復できる。回復したデータのさらに以前の状態がほしい時は、OLD:n;2やOLD:n;3などを使えばよい。このOLDコマンドは、EDIT2から一目抜け出た後またEDIT2に入った場合でも有効なので、ファイル回復の強力な手段となる。

あるコマンドが本当にうまくいくかどうか自信のない時はTESTコマンドを入れておいてからそのコマンドを入れれば、実際のEDIT2の動作をシミュレートし、かつファイルは変更されないので、ファイル破壊の危険を防ぐことができる。特にEDIT2を使いはじめたばかりの人に便利である。

### ④ エディタの虫は早く確実に取り除かなければならない。

エディタの虫によるデータ破壊はもっとも恐いものである。特に重要なことは、何がエディタの虫によるもので、何がユーザのキーインミスかをはっきり区別することである。このため、EDIT2ではHISTORYコマンドを用意し、最近のコマンド100個のイメージがそのままファイルに保存されるようにした。これは突然の異常終了の場合も保存されているので、EDIT2の虫取りに大変役立った。

## 9. TSS漢字エディタによる修正の利点と問題点

### ① TSS一般の利点（パッチによる修正に比べて）

- 修正状況がその場で確認できる。
- 誤修正がすぐ再修正できる。
- 画面やゲラ刷りを見ていることによって校正もれに気づくことがあるが、それもすぐ修正できる。
- 思い立った時すぐ修正できる。

### ② EDIT2の利点（普通のテキスト・エディタに比べて）

- 修正部分の指定に、カラム位置も使えるし、前後文脈付きの文字列も指定できる。

- 誤修正に対する配慮 (OLDコマンドとテスト・モード)
- デバッグのしやすさ (FORTRANによるコーディングとHISTORYコマンド)
- 表示する文字の大きさが変えられる。(SIZEコマンド)これにより一画面にたくさんの文字を詰めこむこともできる。

### ③ EDIT 2 の問題点

- 表示速度がおそいので、修正速度が頭打ちになる。
- 蓄像管端末なので目が疲れる。印字の質もあまりよくない。
- コマンドの種類が TOSBAC-5600 のテキスト・エディタほど多くないので、機能がやや劣る。
- 辞書データの修正専用エディタなので、行の追加等の機能が弱い。

## 謝 辞

EDIT 2 の設計には、主としてそれを使った荻野孝野の意見がかなり取り込まれている。このようなエディタの開発の機会を与えて下さった電子技術総合研究所推論機構研究室長淵一博氏、漢字印字部分のプログラムを提供して下さい下さった同研究室の横山晶一氏、漢字のドット方式の表示の速度を改善して下さい下さった当時東京大学工学部の清松哲郎氏の各氏に感謝する。

## 参 考 文 献

- [1] 荻野綱男 (1978) 「漢字エディタの作成と使用」日本語情報処理シンポジウム
- [2] 荻野孝野 (1978) 「大量漢字データの TSS による校正」日本語情報処理シンポジウム
- [3] 横山晶一 (1977) 「国語辞典データベース化の準備」電総研彙報 Vol. 41, No. 11
- [4] 東芝電気 (1978) 『TOSBAC-5600 プログラミング説明書 TSS テキスト編集』56 APR 17 B

READY 001320 ㊦ あい けい㊦ [愛敬] 「敬愛」の意の漢語的  
 -P.132 表現 「Pの念を増す。」  
 -SHIFT COLU ㊦ あい けい㊦ [愛敬] 「敬愛」の意  
 の漢語的表現  
 -NOCO DS#43 0001320 ㊦ あい けい㊦ [愛敬] 「敬愛」の意  
 の漢語的表現。「-の念を増す」  
 -P.134 0001340 1 (ペットとして) 飼っている犬。  
 -RS./KHETTO./KRETOY 0001340 1 (ペットとして) 飼っている犬。  
 -P.4 0001340 1 (ペットとして) 飼っている犬。  
 0001350 2 犬の性情をよく知り、かわいが  
 ること。「-家㊦」  
 0001360 ㊦ あい こ㊦ [アヒー] [相子] どちら  
 も優劣の無いこと。勝負無し。「これでーだ」  
 0001370 ㊦ あい こ㊦ [愛願] -する (「願」  
 は、目をかける意) 客にひいきにされること。ひきたて。

図1 エディタの使用開始。①132行を表示する。②表示位置を右に5コラムずらし、コラム番号をつけて、その行  
 を表示する。③コラム番号を取り、43コラム目の文字を削除する。EDIT2によるペリフレイ出力が出る。④134  
 行を表示する。⑤“ペット”を“ベット”に置き換える。(ペリフレイ出力。)⑥カレント行から4行表示する。

-P.136 F

0001360 ㊦あい こ㊦㊦㊦③アヒー [相子] どちら  
も優劣の無いこと。勝負無し。「これでーだ」

0001370 ㊦あい こ㊦㊦㊦① [愛願]ーする (「願」

は、目をかける意) 客にひいきにされること。ひきたて。

-CUT.136

0001370

-PAST.135

0001360

0001360 ㊦あい こ㊦㊦㊦① [愛願]ーする (「願」

は、目をかける意) 客にひいきにされること。ひきたて。

0001370 ㊦あい こ㊦㊦㊦③アヒー [相子] どちら

も優劣の無いこと。勝負無し。「これでーだ」

0001370 ㊦あい こ㊦㊦㊦③アヒー [相子] どちら

も優劣の無いこと。勝負無し。「これでーだ」

0001360 ㊦あい こ㊦㊦㊦① [愛願]ーする (「願」

は、目をかける意) 客にひいきにされること。ひきたて。

0000010 ㊦あ㊦ (亜・阿・亞・鵜) → [漢語の造

語成分]

図2 ① 136行を表示し、ポイントを1行進める。(ペリフライ出力。) ② カレント行(137行)をカットする。  
③ 次に136行をカットする。④ 135行の次にカットしたものを挿入する。(ペリフライ出力。) こうして136行と  
137行の内容が入れかわった。⑤ 137行を表示する。⑥ ポイントを1行戻す。(ペリフライ出力。) ⑦ ポイントを  
最初の行まで戻す。





```

-p:51      0000510      [-シャドー]③ (eye) まぶた
            に塗る、青・灰色などの化粧品。アイシャード③。
-1s:/eye// shadow 0510      [-シャドー]③ eye shadow
            (W) まぶたに塗る、青・灰色などの化粧品。アイシャード
            ③。
-b,x ps./line/ 0000540      [-ライニング]③ (eyeliner)
            (女性が)目を大きく見せるために、目のまわりをふちどった
            線。

```

図5 ①51行を表示する。②“eye”の次に“ shadow”を挿入し、表示する。③ポインタを最初に戻し、“line”という文字列を含む行を表示する。

```

READY
-HIST
OLD
P.122
0&42
D
PAST.143 P
DONE
P.132
SHIFTS COLU P
NOCO DS&43
P.134
RS./KHETTO../KPETTO/
P,4
PAGE
OLD.136
OLD.136,2
OLD.136,3

```

```

0&17
F
PAGE
F
NOVE DS&38. &44 IS&14./HU\0\97/ PCOL P
P.144
B.141 COPY&34. &38 F.144 IS&41./239/ P
PASTES&42
P
IS&47./(/ RS./HYONI../)\HYOUNI/ P
P
page
p.510
p.51
1s:/eye// shadow/ p
b,x ps./line/

```

図6 最近入れたコマンドを表示するコマンド HISTの結果