

編集子SPE

立教大学・理学部

島内 剛一

SIMAUTI TAKAKAZU

0. 総論

0.1 SPEの履歴 SPEは、1973~1974年の「NHKコンピューター講座」用として作られた編集子に始まる。(島内, 見捷彦)^[1] この編集子は、着組進行中に、ゲストの生徒さんが算語や算料を編集し、その内容を視聴者へその場で理解してもらおうことを目的とした。したがって、わかりやすさと敏捷性を重視して作られ、ユニモード、ワンタッチコマンド、スリム、リエントラントなどの特徴を持つものとなった。この編集子は、1974~1975年の同講座用としてやや拡張された。(見, 加賀美鉄雄)^[2] その後、立教大学理学部教員教育等の計算機に積極的改版された。(見, 石井博; 島内)^[3] ここに報告するのは、パソコンMB16000用に作られた最新版であり、ワープロとして使えるように機能が拡張されている。

0.2 特徴

(1) ランゲージフリーなユニヴァーサルエディタ SPEは、特定の言語等に依存しない汎用編集子である。バイトの列を対象としたテキスト・エディタであるが、バイトの内容をビット単位で読取りできる[†]ので、どんな算帳(ファイル)でも(バイナリ・ファイルであっても)、編集することが出来る[†]。

(2) ファイルの分割、結合が容易なオンコア・エディット 編集は、DOS管理下の外部算帳を指定してディスク内で行なうのではなく、編集子の管理下にある内部算帳で行ない、外部算帳とのやり取りは入出力指令による。任意個数の内部算帳を扱うことが出来る[†]、外部算帳の作成、消去、分割、結合、複製、並行編集[†]などが容易に行なえる。

(3) スーパー・スクリーン・エディタ 編集作業は、行やページなどの切れ目から行なわれることなく、字(キャラクタ)を単位として行なう。編集位置の位置決めは際して、行番号などを必要としない。遊標(カーソル)の移動は、画面上、上下左右に動くが、巻上げ巻下げ(スクロールアップ/ダウン)は、自動的に行なわれるので、画面という単位は存在しない。

(4) ユニモード、ワンタッチコマンド テキスト・モード、コマンド・モードなどのモード切替えが無い。すべての指令(コマンド)は、鍵盤の鍵を1つ打つだけで実行される。また、おもな指令については、この鍵の割当てを自由に変更することが出来る[†]。しばしば使われる定形句の挿入や、指令の連続などをまとめて1つの指令とする指令語の機能もある[†]。

(5) スリープ機能 編集作業中いつでも、ワンタッチで作業を休止し、後日、なるべく同じ状況から作業を続けることが出来る。

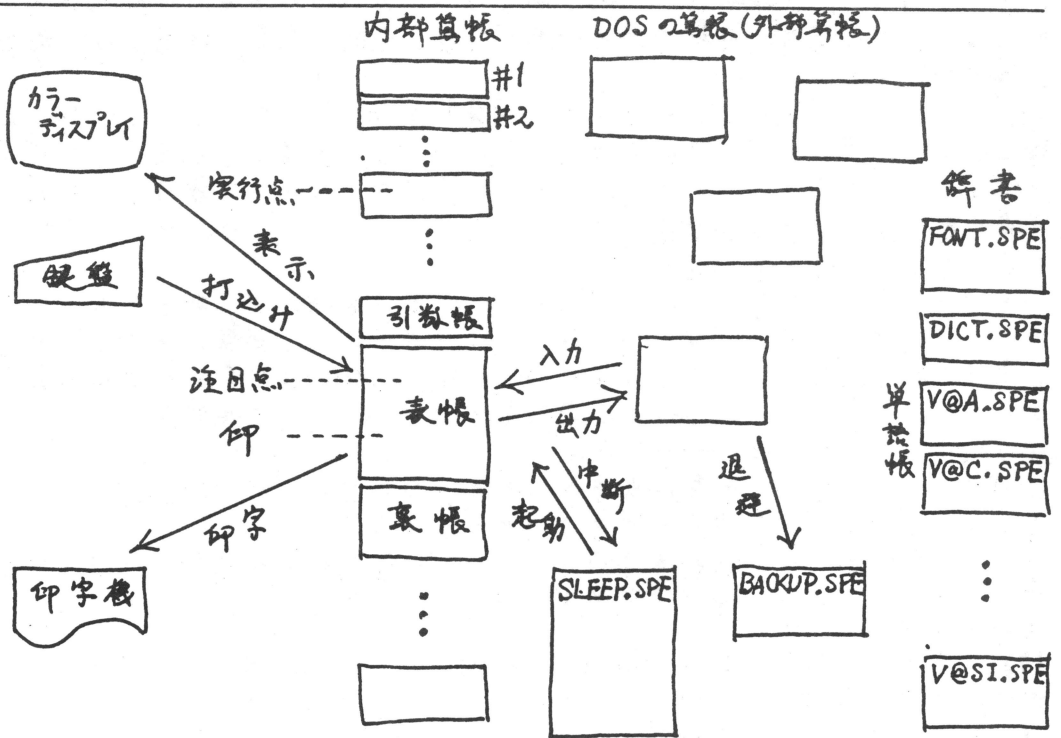
(6) ワープロ機能 英文/漢字両用のワープロ機能が追加できる。ローマ字を基本としたワープロで、ローマ字文が、ほとんどそのまま、漢字をまじり文に変換できる。

(7) サブセット スリムで強力なサブセット(基本SPE)がある。^[5]

1., 2., 3., 4. が編集子SPE 5., 6. がワープロ機能 †は基本SPEでは除く(部分)

1. 編集域 (editing area)

- 1.1 字 (character) JIS 8 bit code の拡張 (うらがなを追加) 256種 16進2ケタを [] で括って表わす。 例: [5C]
 復帰改行 [OD] [OA] または, 改行 [OA] (直前の字は, 復帰ではない) を 2 で表わす。
- 1.2 算帳 (file) 字が並んだもの。空行こともある。
 2 により, 行 (line) に区切られる。
- 1.3 点 (point) 算帳の中で並んでいる2つの字の間, 先頭の字の前 (始点), 最後の字の後 (終点) を点と言う。
 行の直前の点を, 次の行の左端, 直後の点を右端という。
- 1.4 算帳系 編集域の本体は, 算帳が並んだものであり, 各算帳には, #1, #2, ..., #n と番号が付けられている。これらの算帳を内部算帳と呼ぶ。n かよび各算帳の内容は, 編集作業の進行に伴って, 変化する。ただし, つね $n \geq 3$ {基本SPEでは, $n=3$ }
- 1.5 表帳 (front file) 編集作業中, i ($2 \leq i \leq n-1$) が指定され, 算帳 # i を表帳という。
- 1.6 裏帳 (hidden file) 算帳 # $i+1$ 。
- 1.7 引数帳 (parameter file) 算帳 # $i-1$ 。
- 1.8 注目点 (scanned point) それぞれの内部算帳で, 1点が指定され, 注目点と呼ばれる。とくにここからなければ, 表帳の注目点のこと。
- 1.9 印 (mark) 各算帳の第2の注目点。ふつうは, 表帳の印。
- 1.10 実行点 (execution point) 算帳系の中の1つの算帳内の1つの点が, 実行点として指定されることがある。

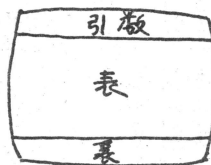


2.† 画面 (25行×80字; 400ドット×640ドット; 15色)

2.1 編集画面 SPEの標準的な画面。図形字は緑で表示。非図形字は、JISコードで[40]だけずらした図形字を白で表示。

引数窓 (1行; 空色地) 引数帳の先頭を表示。

表窓 (23行; 黒地) 表帳の、注目点の附近を表示。注目点は、直後の字に遊標(カーソル)を置いて示す。即ち、赤縦線を示す。窓の右端と、表帳内の「J」の直後で改行する。注目点は、(それが表帳の最初の5行以内にあるときを除いて)表窓の6行目~20行目の範囲内に表示する。注目点が表窓の上端5行または下端3行内にはいっらときは、1行ずつの巻上げ/巻下げ(スクロール・アップ/ダウン)が自動的に1行をわれ、調整する。



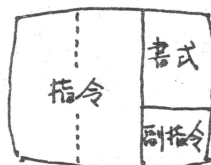
表窓 (1行; 空色地) 表帳の先頭を表示。

2.2 HELP画面 HELP指令の直後の画面(黒地, 黄/白-字)

指令窓 (26ヶ9 25行2列) 指令の一覧表

書式窓 (26ヶ9 10行) 書式記号の一覧表

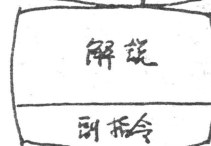
副指令窓 (26ヶ9 4行) ここでの副指令の一覧表



2.3 HELP-HELP画面 HELP画面で副指令HELPを打ったとき。

解説窓 指令等のくわしい説明

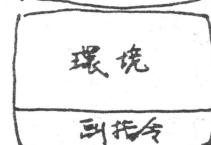
副指令窓 ここでの副指令



2.4 SHELP画面 SHELP指令の直後(黒地, 黄/白-字)

環境窓 編集環境の表示

副指令窓 ここでの副指令

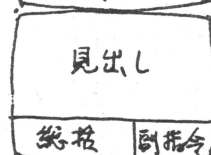


2.5 CHELP画面 CHELP指令の直後(黒地, 黄/白-字)

見出し窓 内部算帳, 表帳の行の見出しを表示

状態窓 編集域の使用状況を表示。

副指令窓 ここでの副指令



3. 鍵

3.1 シフト鍵 SHIFT CTRL ALT† GRAPH†
略記号 s c a g

3.2 図形字鍵 A~Z a~z 0~9 ア~ン あ~ん ! " # ...

3.3 非図形字鍵等 c@ cA~cZ a@aA~aZ a0~a9

3.4† 特殊鍵 ESC TAB BACKSPACE ⇐ ⇨ ⇩ ⇪ ⇫ ⇬
SESC STAB SMODE SCOPY CMODE COPY ⇐ ⇨ ⇩ ⇪ ⇫ ⇬

F1~F10 CANCEL DEL INS HOME ← → ↑ ↓ HELP
SF1~SF10 SINS CLS SHELP
CF1~CF10 CDEL CINS HOME ← → ⇩ ⇪ HELP
AF1~AF10

4. 指令 (command)

指令は、鍵盤から打ち込まれて実行される場合と、算帳内の字が指令語を構成する指令として実行される場合[†]とがある。

	鍵盤から打たれる鍵	指令語中の字 [†]
固定指令	図形字を表わす鍵等	図形字 [20 ~ FF], [1B ~ 1F] F
浮動指令	Q, A ~ Z	[00] ~ [1A]
特殊指令 [†]	特殊鍵	

浮動指令は、鍵や字と指令動作との対応を変えることができる[†]

4.1 挿入 (insert) 注目点の直前に、字が挿入される。仰と注目点とが一致しているときは、仰は挿入された字の直前に残る。実行点と注目点とが一致しているときは、実行点は挿入された字の直後に移る。編集域が一杯で挿入できないときは、そのまま* (*は、警告音を意味する。以下同様)

- <指令> 図形字 [20 ~ FF] 図形字がそのまま挿入される。
- <指令> ◀ (M) M [0D], J [0A] の2字が挿入される。
- <指令> (J) 上と同じ。ただし、上の指令の直後では、何もしない。
- <指令>[†] aA ~ aZ aA [01] ~ aZ [1A] が挿入される。
- <指令>[†] a0 または a@ [00] が挿入される。
- <指令>[†] a1 ~ a5 [1B] ~ [1F] が挿入される。
- <指令>[†] a6 [60] が挿入される。
- <指令>[†] a7 [7F] が挿入される。
- <指令>[†] a8, a9 [FE], [FF] が挿入される。

4.2 抹消 (delete)

- <指令> BACKSPACE (H) 注目点の直前の字を消す。始点のときは、そのまま*
- <指令> DEL (K) 注目点の直後の字を消す。終点のときは、そのまま*
- <指令>[†] CDEL (一) 注目点の直後の字を、□ [20] に書き替える。注目点はその直後に移る。終点のときは、そのまま*

4.3 注目点の移動

- <指令> ← (L) 注目点を1字前に移す。始点のときは、そのまま*
- <指令> → (R) 注目点を1字後に移す。終点のときは、そのまま*
- <指令> ↑ (U) 注目点を1行上に移す。最上行のときは、そのまま*
- <指令> ↓ (D) 注目点を1行下に移す。最下行のときは、そのまま*

指令 ↑ や ↓ が連続したときには、↑ や ↓ が始まる前のケタ位置 (行内で注目点より左にあるバイトの個数) を憶えていて、移動して行った先の行内のそのケタ位置に移る[†]。行が短くてそのケタ位置が無いときには、右端に移るが、さらに ↑ や ↓ が続いて、十分な長さの行に移れば、最初のケタ位置を回復する[†]。行末の M と J の間から ↑ や ↓ が連続したときは、J の直前にたどる[†]

- <指令> ⇐ (A) 注目点を左端に移す。
- <指令> ⇒ (V) 注目点を右端に移す。
- <指令> ⇑ (T) 注目点を始点に移す。
- <指令> ⇓ (E) 注目点を終点に移す。

4.4↑ TAB TAB位置は、引巻帳の最上行に設定される：「以外の字が連続しているところの先頭の字の直前の点が、TAB位置のケタ数を示す。

例：

引巻帳	A B C	┌┌	D	┌┌	E F	┌┌┌┌	G H I	┌┌	J	┌	┌	...
TAB位置		↓		↓	↓		↓		↓			
表帳	...	┌	P Q	┌┌	R S T U V	┌┌	W	┌	X Y Z	┌	┌	...

<指令> TAB (°I) 注目点を、それより右の最初のTAB位置へ移す。ただし、注目点を含む行が短くてそのような点が行内に取れないときは、行の右端に必要ならだけの「を追加挿入してから、TAB位置(右端)に移す。注目点の右にTAB位置が無いときや、行の右端への「の追加が編集域が足りなくてできないときは、そのまま*

<指令> STAB (°-) 注目点を、それより左の最初のTAB位置(無ければ左端)へ移す。

4.5 検索(search)と、置換(substitute)

<指令> HOME (°F) 注目点の1字後から始めて表帳の終わりまでの部分で、引巻帳全体と同じ字の並びを探し、最初に見付かったものの先頭に注目点を移す。即ち、4との注目点の位置に移る。見付からなかったときは、そのまま*

<指令> INS (°S) 注目点から始めて表帳の終わりまでの部分で、引巻帳全体と同じ字の並びを探し、最初に見付かったところを、裏帳の全体と同じ字の並びに、置き替える。注目点は、置き替えられた部分の直後に移り、即ち、もとの注目点の位置に移る。裏帳は変わらない。置き替えられてしまった部分の中には実行点があったときは、注目点と同じ位置に移る。編集域が不足して置替えができないときや、見付からなかったときは、そのまま*

例：引巻帳が空のとき、°S を実行すれば、表帳の注目点の直前に、裏帳の全体を複写することになる。即ち、直前に移る。

4.6 印(mark)

<指令> ← (°G) 印の位置に、注目点を移す。

<指令> → (°Q) 注目点の位置に、印を移す。

4.7 内部算帳の分割(cut)と、結合(join)

<指令> ↑ (°B) 注目点の直前に、裏帳の全体を挿入する。即ち、挿入された部分の直前に移る。算帳系の裏帳以降は1つずつ降りより、最後に空な算帳が補充される。実行点が裏帳にあれば、そのまゝ表帳に移る。

<指令> ↓ (°C) 注目点と印の間の部分が、表帳から切り落とされ、裏帳の内容となる。算帳系の裏帳以降は1つずつ押し下げられ、最後の算帳が消える。新しい裏帳の注目点、即ち、もとの始点を指す。実行点が切り落とされた部分に含まれているときは、そのまゝ裏帳に移る。はじめ、注目点と印とは、どちらが前であってもよい。

例：↑と↓とは、(注目点、印の位置、算帳系の最後の算帳の内容を除いて)互いに逆の指令とな、211。(繰り返して実行したとき、回復できる。)

例：↓の後、注目点を無効してから↑を実行すれば、表帳の中の隣り合った2つの部分の入替えができる。

4.8 算帳系

<指令> CLS (°N) 算帳系の表帳以降を1つずつ押し下げ、最後の算帳が消える。新しい表帳としては、空算帳が補充される。それぞれの算帳の注目点、仰および実行点は、算帳に乗るまで移動する。

<指令> ↑ (°Y) それまでの引数帳が表帳となり、表帳が裏帳に、引数帳の1つ前にあった算帳が引数帳となる。引数帳が最初の算帳だったときは、新しい引数帳として、空算帳が補充される。(内部算帳の個数が1つ減る) 編集域が不足して、補充できないときは、そのまま*

<指令> ↓ (°Z) それまでの裏帳が表帳となり、表帳が引数帳となり、裏帳の1つ後にあった算帳が裏帳となる。それまでの引数帳が最初の算帳であり、しかも空だったときは、その算帳は消し去られる。裏帳が最後の算帳だったときは、新しい裏帳として空算帳が補充される。補充できないときは、そのまま*

<指令> ⇄ (°X) 表帳と引数帳の内容を、(注目点、仰、実行点を乗せたまま) 入れ替える。

<指令> ↑ ↓ (°) 表帳と裏帳の内容を、(注目点、仰、実行点を乗せたまま) 入れ替える。

4.9 入出力 (input/output) 入出力指令の実行前、引数帳の先頭に、入出力指定子を設定しておく。指令実行後、この部分が、入出力報告子に変わる。

[入出力指定子] 様番 算帳指定子 バイト位置↑ バイト長↑ 終止符

例: A:BCDE.FGH, 12345/6789;

省略のとき 既定様番: , 0 /算帳の終わりのバイト数

様番Aの、算帳BCDE.FGH の、第12345バイトから、長さ6789バイトの部分。読み込む(へ、書き出す)

[入出力報告子]↑ 様番 算帳指定子 バイト位置 バイト長 残りバイト数 バイト長
例: A:BCDE.FGH, 12345/6789; 19134/10866;

様番Aの、算帳BCDE.FGH の、第12345バイトから、長さ6789バイトの部分が、入力(出力)され、第19134バイトから、長さ10866バイトが残っている。

<指令> SINS (°O) 入出力指定子で指定されている算帳(または、その部分)を、表帳の注目点の直前に読み込む。仰ほ、読み込まれた部分の直前に移る。読み込みは、実行前の編集域の余剰(算帳系で占められたい部分)の半分まで打ち切られる。結果が、入出力報告子に表される。様番、算帳指定子によ、指定された算帳が無ければ、そのまま*

<指令> CINS (°W) 入出力指定子で指定されている算帳(または、その部分)の代わりに、表帳全体の内容と同じ字の列が書き込まれる。表帳の内容は変わらないが、注目点は終点へ、仰ほはじめの注目点の位置に移る。もとの外部算帳(様番、算帳指定子で指定されたもの; 入出力指定子が算帳の一部を指定しているときでも、算帳の全体)は、BACKUP算帳に逃避される。結果が、入出力報告子に表される。外部の様番の容量などのために、実行できないときは、そのまま*

4.10† 指令簿 (command program) 指定された算帳の始点に、実行点を設定し、実行点を1字ずつ進めながら、実行点が通過した字を指令として実行する。算帳の終点に達したり、指令の実行が不存の理由により行なわれなくなったりしたときは、この指令簿の実行を終え、実行点は消える。指令簿中の指令として、指令簿 指令を実行したり、鍵盤から指令が打ち込まれたときは、はじめの指令簿の実行点は消え、新しい指令が実行される。

<指令> F1 (-) 算帳 #1 を実行する。

<指令> F2 " #2 "

⋮ ⋮ ⋮

<指令> F10 " #10 "

<指令> SF1 " #11 "

⋮ ⋮ ⋮

<指令> SF10 " #20 "

<指令> CF1 " #21 "

⋮ ⋮ ⋮

<指令> CF10 " #30 "

<指令> AF1 " #31 "

⋮ ⋮ ⋮

<指令> AF10 " #40 "

<指令> CHOME (CP) 引寄せ帳を実行する。

指定された算帳が存在しないときは、そのみず*

例：算帳の内容が、図形字や、CM, CJ だけから成っているときは、指令簿の実行は、その算帳の内容を表帳の注目点の直前に挿入することを意味する。

例：CUUUUUUUUUUU 10行上へ挿入。

例：引寄せ帳の内容を α 、裏帳の内容を β とし、引寄せ帳の1つ前の算帳の内容を、CZCSYCP とするとき、鍵盤から、CYCP を打ち込めば、表帳の注目点以降の α の部分をすべて β に置き替えることになる。

4.11 表示 (display), 印字 (print)

<指令> COPY (-) 注目点を1行ずつ下げ、最終行まで行く。編集画面の表窓の巻上げは、1行ずつではなく、注目点が下から3行目に入ろうとする時に、いっぺんに15行巻き上げる! (注目点は、上から6行目になる) 初めは、4行/秒で注目点が下りて行くが、COPY 指令を打つ度に、速度が2倍になる! さら、SPACE を打つ度に、半分になる! (どちらか、限度がある) 図形字、COPY 以外の鍵を打てば、COPY 指令の実行を止め、打ち込まれた指令を実行する!†

<指令> ^COPY (-) 注目点を1字ずつ下げながら、終点まで行く。このとき、注目点が通り過ぎた字を、印字機に出力する。図形字以外の鍵を打てば、^COPY 指令の実行を止め、打ち込まれた指令を実行する!†

(指令)† ^SCOPY (-) 画面を印字する。(DOSの機能)

4.12† HELP

<指令> HELP (-) 画面が HELP画面に変わり、指令の一覧表、書式記号の一覧表等が表示される。

指令の一覧表は、各指令を、指令鍵と動きの簡潔な説明とを対比させ、並べてある。最初遊標は先頭の指令の所にいる。スペース鍵を打つたびに、遊標は、1つずつ次の指令へと移り、最後の指令まで行ったら、また最初にもどる。

HELP 鍵を打てば、遊標の示す指令の、もっとくわしい説明が表示される。

(HELP-HELP画面) ここで、 \square 鍵を打てば、もとの HELP画面にもどる。

HELP画面にいるときに、@, A~Z のいずれかを打てば、その文字に C (CTRL+) を付けた浮動指令と、遊標が示す場所に表示されている浮動指令とが、入れ替わり、画面の相当箇所も書き替えられ、赤字で表示される。遊標が示す箇所に浮動指令が割り当てられているときは、そこは、打ち込まれた文字に相当する浮動指令が割り当てられ、前にその浮動指令があったほうには、浮動指令の割り当てはなくなる。

HELP画面にいるときに、浮動指令または特殊指令を打てば、画面はもとの編集画面にもどり、打ち込まれた指令が実行される。

<指令> SHELP (-) 画面が SHELP画面に変わり、環境の一覧表が表示される。

環境の各項目には、英字が割り当てられている。その英字と、それに続く数字の列を打ち込むことになって、環境の項目の内容を覚えることができる。

浮動指令や特殊指令を打てば、もとの編集画面にもどり、打ち込まれた指令が実行される。

<指令> CHELP (-) 画面が CHELP画面に変わり、各内部算帳や表帳の各行の先頭約15字が表示され、また、編集域全体の使用状況が表示される。

内部算帳や行の見出しは、1画面に合わせて80個しか表示できないので、表示しきれないときは、スペース鍵を打てば、残りが表示される。最後の行の表示まで終われば、最初の算帳にもどる。

F f L l B b \square (f, l, b は、10進数字の列) を打てば、内部算帳#f を表帳とし、その第l行第bバイトに注目点を移す。(編集画面にもどり)

ライトペンで、見出しの部分に描いても、その位置への注目点の移動ができる。浮動指令や特殊指令を打てば、もとの編集画面にもどり、打ち込まれた指令が実行される。

4.13† 中断

<指令> ESC (-) SLEEP算帳に、編集状況(編集域の内容、浮動指令の割当て、環境)をしまい、SPEの実行を終える。

しもう余地の無いときは、一旦待つ* さらに ESC を打てば、(SLEEP算帳は以前のまゝにして) 実行を終える。ESC以外の鍵を打てば、それを指令として実行する。

SPEの実行開始のとき、既定機着に SLEEP、SPE という算帳があれば、その内容を編集状況として回復してから、編集作業実行に換る。

<指令> \square ESC (-) SPEの実行を終える。(編集状況の追跡は、しない)

5.† 漢字

5.1 漢字相

<指令> MODE (-) 英字相 ⇔ 漢字相 の移行

漢字相は、英字相(前節までに述べた普通の相)に比べて、つぎの点に異なる:

- (1) 全角字(漢字を含む JIS 第 1 水準)が使える。
- (2) 編集画面 外わくが黒から紫色になり、漢字相であることを示す。全角字は 2 バイト分の場所に表示される。引数窓が、今迄通りに引数帳の矢頭を表示する場合と、その状態での打鍵の案内を表示する場合とがある。
- (3) ↑, ↓, TAB, STAB 指令の実行結果で、注目点(全角字)の 2 バイトの中間を指すこととなる場合には、自動的にその字の直後に修正される。
- (4) 図形字指令 挿入指令から、つぎのようを表換挿入指令に変わる。
 - (4.1) a~z の列 ローマ字 → ひらがな変換により、ひらがな(ひらがな相)またはカタカナ(カタカナ相)に変換し、挿入する。
 - (4.2) [に続く a~z の列 ローマ字 → 単独変換により、単独(全角字や半角字の列)に変換し、挿入する。
 - (4.3) ¥に続く 16 進 4 桁字 全角字基本入力により、全角字 1 字に変換し、挿入する。
 - (4.4) ^に続く 1 字 その字が半角字として挿入される。(半角字入力)
 - (4.5) []に括られた、a~z の列 見出し語としてそのまま挿入
 - (4.6) A~Z [と a~z の列に等値
 - (4.7) あ~ん ローマ字(訓令式) a~z の列に等値
 - (4.8) ア~ン [と ローマ字 に等値
 - (4.9) 0 ~ 9 引数窓の案内に従って、表換挿入をする。
 - (4.10) +, - 引数窓の案内を、進める/戻す。
 - (4.11) \ ひらがな相 ⇔ カタカナ相
 - (4.12) □, - 半角字を挿入
 - (4.13) その他の記号 全角字の記号を挿入

5.2 全角字 英字相では、英字中のバイトは、すべて字に対応し、したがって、字の数は 256 である。それに対し、漢字相では、内容が [81~98] (2 行の最後ではない) のバイトと直線のバイトを対応して、2 バイトで 1 字を表わす。これらの字を、2 バイト字(または全角字)と言い、いままでの 1 バイト字(半角字)と区別する。2 バイト字は、JIS 第 1 水準の漢字コードと順序を保存して 1 対 1 に対応する MSKDC コードにより、漢字その他の文字を表わす。(対応する MSKDC コードの無い 2 バイト字は、非正規のスペースを表わす)

5.3 全角字基本入力 ¥に続く、16 進 4 桁字(0~9, A~F)または a~f) 4 文字により、漢字 JIS コードを表わし、全角字を挿入し、打ち込まれた 5 文字(半角字で)意味にはいる。引数窓の案内に従って打ち込めばよい。

例: ¥を打つと、引数窓は、 2: 10 ... 9 3: 亜 ... 裾 4: 澄 ... 腕 と存。
 そので、4 " 0: 澄 1: 織 2: 蔵 3: 叩 4: 帖 5: 邸 6: 蓮 7: 如 ...
 " , 5 " 2: 邸 3: 徹 4: 点 5: 登 6: 凍 7: 盗
 " , 6 " 0: 凍 1: 刀 2: 磨 3: 塔 4: 糖 5: 套 6: 岩 7: 鳥 ...
 " , 7 を打つと、鳥 が挿入され、¥4567 が基帳に切り落とされる。

5.4 半角字入力 ^に続いて打つた字は、半角字として、挿入される。

5.5 ローマ字 → ひらがな変換

〈注意〉 \ ひらがな相 ↔ カタカナ相 の移行。読み込まれたローマ字が、全角ひらがなに変換されるか、全角カタカナに変換されるかの変更。

[変換手順] カタカナ相のとせば、対応するカタカナ(または括弧内)になる。

- (1) j → L'g f → ふ x v → ぶ x (ヴ x)
- (2) βy → βi'g
- (3) ts → つ x tc → っ x sh → し g ch → ち g
- kw → く l gw → ぐ l
- (4) nβ → ん β mm → ん m mp → ん p mb → ん b
- (5) ββ → っ β

β α	α						
	β	a	i	u	e	o	~
~		あ	い	う	え	お	
k		か	き	く	け	こ	く
s		さ	し	す	せ	そ	す
t		た	ち	つ	て	と	と
n		な	に	ぬ	ね	の	ん
h		は	ひ	ふ	へ	ほ	う
m		ま	み	む	め	も	ん
y		や	い	ゆ	え	よ	い
r		ら	り	る	れ	ろ	い
w		わ	わ	う	ゑ	を	う

β α	α						
	β	a	i	u	e	o	~
g		が	ぎ	ぐ	げ	ご	ぐ
x		ざ	じ	ず	ぜ	ぞ	ず
d		だ	ぢ	づ	で	ど	ど
b		ば	び	ぶ	べ	ぼ	ぶ
p		ぱ	ぴ	ぷ	ぺ	ぽ	ぷ

g	ゃ	~	ゅ	え	よ	~
x	あ	い	~	え	お	~
l	わ	い	う	え	お	う
c	(カ)	ち	っ	(ケ)	こ	っ

5.6 ローマ字 → 単語(漢字)変換

[注意] a~z の列が打ち込まれ、終止符が打たれるか; または、A~Z の列が打ち込まれ、終止符が打たれると; 辞書が引かれ、訳語が挿入される。A~Z は、[と a~z の集個である。a~z の列は、辞書の見出し語を表わす。

終止符が打たれるまで、引数窓には、そこで打つ終止符と訳語の対応を表わす案内が表示される。

終止符は、「」, ^, \, +, -, a~z, あ~ん」以外の字である。a~z や あ~ん も、それを追加したときに見出し語でなくるときは、終止符となる。

+ と - は、引数窓の案内の中の訳語の列を1つ前または後にする。

終止符 0~9 は、引数窓の訳語の列の何番目の訳語を選ぶかを表す。0が先頭の訳語で、1が次の訳語である。...

0~9 以外の終止符は、矢頭の訳語を選び、終止符と存った字は、変換後にその字が打ち込まれるべく、(終止符という役割を取り去って) そこに残る。

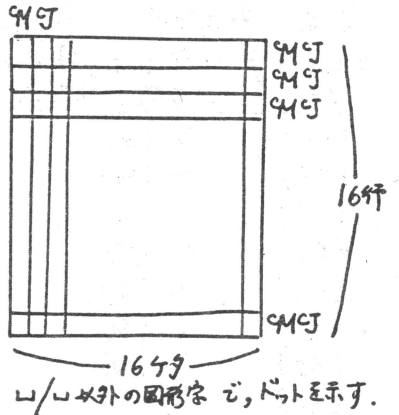
0~9 が終止符と存ったときは、辞書中の訳語の順は変わらないが、0~9 以外の字が終止符と存ったときは、採用された訳語が、訳語の列の矢頭に移される。

5.7 補助指令

<指令> $\langle \sigma \rangle$ (-) 辞書表示 注目点の直前に, $[\sigma]$ (σ は英小文字の列) があるとき, この指令を実行すると, σ を見出し語として含む辞書の巻の全体が, 注目点の直前に挿入される。即ち, その直前に移る。“ $[\sigma]$ ”の形式が整わないとき, 編集域に余白が無いときは, そのまま*

<指令> $\langle \sigma \rangle \varphi$ (-) 単語登録 注目点の直前に, $[\sigma] \varphi$ (σ は英小文字の列, φ は漢角字や英小文字等の列) があるとき, この指令を実行すると, 辞書の (相当する巻の), 見出し語 σ の項目 (無ければ, 新設して) の, 最後に訳語 φ が追加される。“ $[\sigma] \varphi$ ”の形式が整わないときは, そのまま*

<指令> $\langle \sigma \rangle$ (-) 字形表示 注目点の直前に, 全角字または, 半に続く (JISコード) 16進4ケタ があるとき, この指令を実行すると, その字の字形が, 右図のような形で, 注目点の直前に挿入される。即ち, その直前に移る。注目点の直前の形式が整わないとき, 編集域に余白が無いときは, そのまま*



<指令> $\langle \sigma \rangle$ (-) 字形登録 注目点の直前に, 半に続く 16進4ケタ (半2820以降) のコードと, それに続いて右図のような形で字形があるとき, この指令を実行すると, このコードと字形の対が, 辞書 (の字形集帳) に, 追加される。コード, 字形の形式が整わないときは, そのまま*

5.8 辞書 SPE の辞書は, 字形集帳 (FONT.SPE), 目次集帳 (DICT.SPE), いくつかの巻に分かれた単語帳 ($V@ \sigma$.SPE ; σ は 1~6 文字) とから成る。

字形集帳 (FONT.SPE) つぎのような項目の列

2バイト字 字形 945J

(外字コード) L 32バイトに収められた 16x16 ドット

目次集帳 (DICT.SPE) 単語帳の巻のまとめ; つぎのような項目の列

$V@ \sigma$.SPE 巻の大きさ (詰め算) 945J

(巻名)

単語帳 ($V@ \sigma$.SPE) 見出し語が σ から, 次の巻の前までのものについて, 見出し語と訳語の対照表; つぎのような項目の列

$[\sigma] \varphi_1, \varphi_2, \dots, \varphi_n, 945J$

見出し語 訳語 訳語

単語帳の項目は, 見出し語の順に整列されている。

項目の中の訳語の順は, 辞書の使用によって入れ替わる。

単語帳の巻の大きさは, その上限が決められていて, 訳語の追加によってこの上限を越えるときは, その巻は, 又つな分割される。分割点は, 見出し語の上位の字が切り変わる所, はじめの巻の中央に近い所が選ばれる。

いくつかの巻を保持できる領域 (辞書領域) が用意され, 必要に応じて, 巻はその領域に流し込まれてから利用される。領域に余白が無いときは, もっとも長く使われている巻が, 領域から追い出される。

6.1 書式

6.1 書式相

<指令> SMODE (-) 字列相 ⇔ 書式相 の移行

字列相は、書式相(前節までに述べた普通の相)に比べて、つぎの点で異なる:

- (1) 印字 COPY 指令によって印字する際に、字列相では、字列をそのまま印字機に送るだけであるが、書式相では、環境として設定されている全域書式や、字列中に含まれる書式記号で示される局所書式などの書式を織り込んで、書式に則った書類として印字する。
- (2) 編集画面 編集画面の表窓に表示される表帳の部分についても、ある程度の書式を織り込んで、表示する。

6.2 書式記号 表帳内の下記の字は、書式相における印字(一印は、表示のとき)の際して、書式記号として働く。

字	字列相での表示	書式相で印字のときの働き
[00]	@ (白)	(印刷を)一旦停止する
[0B]	Et (白)	
[0C]	→ (白)	行を、右詰め印刷する
[0D]	← (白)	行を、左詰め印刷する
[0E]	↑ (白)	半行上げる
[0F]	↓ (白)	半行下げる
[60]	〒 (白)	
[7F]	\	
[FE]	繻	文字を拡大する
[FF]	? (白)	文字を縮小する

6.3 全域書式 つぎのような書式は、環境として設定されている(SHELP 指令で表示され、変更することが出来る)ものである。

- 印刷用紙の、横幅、縦幅(、ページの有無)
- 1行の字数、左端の空き(、右端の空き)
- 1ページの行数、上端の空き(、下端の空き)、行間隔
- 右端合わせをするかどうか
- アンダーラインの取扱い

文献

- [1] NHKコンピューター講座 '73-4 ~ -9 ('73-10 ~ '74-3) テキスト 日本放送出版協会
- [2] NHKコンピューター講座 '74-4 ~ -9 ('74-10 ~ '75-3) テキスト 日本放送出版協会
- [3] 第15回 プログラミング・シンポジウム 報告集
- [4] 島内: 作著類求記 bit (vol.11, No.3) 1979
- [5] 島内: エディタを作ろう bit (vol.16, No.11 ~) 1984

本 PDF ファイルは 1985 年発行の「第 26 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>