

ラインエディタEDITの追加機能について

平野, 広幸
九州大学大型計算機センター

岩永, 美穂子
九州大学大型計算機センター

二村, 祥一
九州大学大型計算機センター

<https://doi.org/10.15017/1468171>

出版情報 : 九州大学大型計算機センター広報. 21 (1), pp.23-34, 1988-01-25. 九州大学大型計算機センター

バージョン :

権利関係 :

ラインエディタEDITの追加機能について

平野広幸*, 岩永美穂子*, 二村祥一*

1. はじめに

今回、TSSのラインエディタEDIT[1]に三つのサブコマンドCORRECT, MATCH, UNDOが追加された。これらは、ラインエディット環境で、テキストの編集や検索を能率よく行うためのものである。CORRECTは行単位に文字列を変更するために、MATCHは指定した文字列を含む行を検索するために、UNDOは更新を取り消して、更新前の状態に戻すために使用するサブコマンドである。

現在、九州大学大型計算機センターのOS IV/F4 MSPのもとで使用可能なエディタとしてはEDITのほかに、画面単位の編集が行えるEDIT/FSO(以後、FSOと呼ぶ。FSOはFull Screen Optionの略)[2]とPFDのエディタ(以後、単にPFDと呼ぶ。PFDはProgramming Facility for Display usersの略)[3]がある。FSOやPFDなどのフルスクリーンエディタは、F9526やF6650系の端末、あるいはそのエミュレータをもつパソコン端末で使用できる。PFDは、さらに、タイプライタ型端末に対する支援機能をもっており、これを利用することにより、ほとんどの端末で画面単位の編集が可能になっている。しかし、タイプライタ型端末の中には、この機能が使えない端末もある(例えば、ソニーテクトロニクス社のT4010シリーズ端末など)。それらの端末ではラインエディタしか使用できず、ここで述べるサブコマンドがとくに有効である。

2節で追加サブコマンドの簡単な使用例を示し、3節でそれらの詳細を説明する。付録には、エラーメッセージの意味とエラーの対処法をまとめる。

2. 追加サブコマンドの使用例

2.1 CORRECTサブコマンド

CORRECTサブコマンドでは、修正するテキストを1行ずつ表示し、次行に特定の文字をキーとして入力することにより、文字の削除、置き換え、挿入などを行うことができる。表示なしの修正、複数行にわたる修正も可能である。

次に使用例に添って説明していく。下線を付した部分が利用者の入力である。_は空白を、✓は改行を示す。サブコマンドの詳細については3節を参照されたい。

CORRECTサブコマンドの使用例

```
READY
EDIT CORR.FORT77✓
EDIT
```

昭和62年11月26日 受理

* 九州大学大型計算機センター

```

LIST✓
00010 REAL A(30000),ER(10),EV(256,10),VW(2000)
00020 CALL INIT(M,N,A)
00030 CALL SEIG((A,256,M,ER,EV,256,VW,ICON)
00040 IF(ICON.LT.20000) THEN
00050 * WRITE(6,*) 'ERR OCCUR'
00060 STOP
00070 END IF
00080 WRITE(6,*) (ER(J),J=1,M)
00090 DO 20 I=1,256
00100 WRITE(6,*) I,(EV(I,J),J=1,M)
00110 20 CONTINUE
00120 END
EDIT
(1) CORRECT 30 50✓
      CALL SEIG((A,256,M,ER,EV,256,VW,ICON)
(2)       D✓
      CALL SEIG(A,256,M,ER,EV,256,VW,ICON)
(3)       I2✓
      CALL SEIG2(A,256,M,ER,EV,256,VW,ICON)
(4) ✓
      IF(ICON.LT.20000) THEN
(5)       RGE✓
      IF(ICON.GE.20000) THEN
(6) ✓
      * WRITE(6,*) 'ERR OCCUR'
(7) B      D      IOR      IRED✓
      WRITE(6,*) 'ERROR OCCURRED'
      ✓
EDIT
(8) CORRECT 90 110 'IC'✓
EDIT
LIST✓
00010 REAL A(30000),ER(10),EV(256,10),VW(2000)
00020 CALL INIT(M,N,A)
00030 CALL SEIG2(A,256,M,ER,EV,256,VW,ICON)
00040 IF(ICON.GE.20000) THEN
00050 * WRITE(6,*) 'ERROR OCCURRED'
00060 STOP
00070 END IF
00080 WRITE(6,*) (ER(J),J=1,M)
00090 C DO 20 I=1,256
00100 C WRITE(6,*) I,(EV(I,J),J=1,M)
00110 C 20 CONTINUE
00120 END
EDIT

```

使用例の説明

- (1) CORRECTサブコマンドにより、行番号30から行番号50までの3行のテキストの修正を行う。まず、行番号30の行が表示される。
- (2) 修正文字列‘__...D✓’により、左括弧‘(’を削除している。
修正文字列には、処理内容を指示する次の文字を含めることができる。
_(空白入力) : 対応する文字はそのままとする。
B(Blank) : 対応する文字を空白文字に修正する。
D>Delete) : 対応する文字を削除する。
I(Insert) : 対応する文字の前に文字列を挿入する。
R(Replace) : 対応する文字から後ろを指定した文字列で置き換える。
- (3) 修正文字列‘__...I2✓’により、数字‘2’を挿入している。
- (4) 改行のみを行っている。行番号40の行が次に表示される。
- (5) 修正文字列‘__...RGE✓’により、文字列‘LT’を文字列‘GE’に変更した。
- (6) 改行のみを行っている。行番号50の行が次に表示される。
- (7) 修正文字列‘B__...D__...IOR__...IRED✓’により、先頭文字‘*’を空白にし、‘)’を削除し、‘ERR’を‘ERROR’に、‘OCCUR’を‘OCCURRED’に変更している。このように、一行に複数の処理を指示することができる。
- (8) CORRECTサブコマンドにより、行番号90から行番号110までの各行の先頭に、‘C’を挿入している。この場合、修正文字列をサブコマンド入力時に指定している。続いて、LISTサブコマンドで結果を表示している。

2.2 MATCHサブコマンド

MATCHは、検索文字列を含むテキストを探すサブコマンドである。MATCHは、FINDと違って、検索文字列の中に、変数文字を指定することができる。変数文字としては、&(英数字)、¥(英字)、#(数字)、%(特殊記号、空白)の4個がある。

MATCHサブコマンドの使用例

```

READY
EDIT MATCH.FORT77✓
EDIT
LIST✓
00010 C    TEST PROGRAM
00020      DIMENSION A(1000,1000),B(1000,1000)
00030      K0=1
00040      K1=1000
00050      DO 10 I=1,1000
00060      DO 10 J=1,1000
00070      A(I,J)=SQRT(I*J*0.5)
00080      B(K0,J)=SQRT(I*K1*0.5)
00090      B(I,K1)=SQRT(J*K0*0.5)
00100      10 CONTINUE
00110      END
EDIT

```

```

TOP
EDIT
(1) VERIFY ON
EDIT
(2) MATCH 'K#'
00030      K0=1
EDIT
(3) MA
00040      K1=1000
EDIT
(4) MA
00080      B(K0,J)=SQRT(I*K1*0.5)
EDIT
(5) MA
00090      B(I,K1)=SQRT(J*K0*0.5)
EDIT
(6) MA
KEQ52506I  TEXT NOT FOUND
EDIT

```

使用例の説明

- (1) **VERIFY**機能をオンにする。これにより検索した行の内容が表示されるようになる。
- (2) **MATCH**サブコマンドにより、英字'K'に数字が続く文字列を検索している。行番号30の行が表示されている。
- (3)~(6)
 同じ検索文字列で次々に検索を行っている。'MA'は'MATCH'の省略形である。行番号90以降には、該当する行がないので'KEQ52506I TEXT NOT FOUND'のメッセージが表示されている。

2.3 UNDOサブコマンド

UNDOサブコマンドは、**EDIT**の修正用サブコマンド(**APPEND,CHANGE,CORRECT,DELETE,RENUM**など)による修正を元に戻す。この機能を動作させるためには、エディタ起動後に'**UNDO ON**'と入力しておく必要がある。反対に、'**UNDO OFF**'と入力すると**UNDO**機能は停止する。**UNDO**機能が動作しているときに、'**UNDO**'と入力すれば、直前の修正が元に戻る。続けて、'**UNDO**'を入力すると、さらにその前の修正が元に戻る。

UNDOサブコマンドの使用例

```

READY
EDIT UNDO.FORT77
EDIT
LIST
00010 C   TEST PROGRAM
00020     DIMENSION A(1000,1000),B(1000,1000)
00030     K0=1
00040     K1=1000
00050     DO 10 I=1,1000

```

```

00060      DO 10 J=1,1000
00070      A(I,J)=SQRT(I*J*0.5)
00080      B(K0,J)=SQRT(I*K1*0.5)
00090      B(I,K1)=SQRT(J*K0*0.5)
00100      10 CONTINUE
00110      END
EDIT
(1)  UNDO ON
EDIT
(2)  DEL 30 40
EDIT
(3)  C 70 90 /0.5/0.8/ALL
EDIT
LIST
00010 C   TEST PROGRAM
00020     DIMENSION A(1000,1000),B(1000,1000)
00050     DO 10 I=1,1000
00060     DO 10 J=1,1000
00070     A(I,J)=SQRT(I*J*0.8)
00080     B(K0,J)=SQRT(I*K1*0.8)
00090     B(I,K1)=SQRT(J*K0*0.8)
00100     10 CONTINUE
00110     END
EDIT
(4)  UNDO
EDIT
LIST
00010 C   TEST PROGRAM
00020     DIMENSION A(1000,1000),B(1000,1000)
00050     DO 10 I=1,1000
00060     DO 10 J=1,1000
00070     A(I,J)=SQRT(I*J*0.5)
00080     B(K0,J)=SQRT(I*K1*0.5)
00090     B(I,K1)=SQRT(J*K0*0.5)
00100     10 CONTINUE
00110     END
EDIT
(5)  UNDO
EDIT
LIST
00010 C   TEST PROGRAM
00020     DIMENSION A(1000,1000),B(1000,1000)
00030     K0=1
00040     K1=1000
00050     DO 10 I=1,1000
00060     DO 10 J=1,1000
00070     A(I,J)=SQRT(I*J*0.5)
00080     B(K0,J)=SQRT(I*K1*0.5)
00090     B(I,K1)=SQRT(J*K0*0.5)

```

```
00100    10 CONTINUE
00110    END
EDIT
```

使用例の説明

- (1) UNDO機能を有効にする。
- (2) 行番号30, 40の行を削除する。
- (3) 行番号70から行番号90までの3行について、文字列'0.5'をすべて'0.8'に変更する。
- (4) UNDOサブコマンドにより、今の修正を取り消す。LISTサブコマンドで結果を表示している。
- (5) UNDOサブコマンドにより、その前の修正を取り消す。LISTサブコマンドで結果を表示している。

3. サブコマンド詳説

3.1 CORRECTサブコマンド

(1)入力形式

サブコマンド	オペランド
CORRECT	$\left[\left[\begin{array}{l} * \quad [\text{整数1}] \\ \text{行番号1} [\text{行番号2}] \end{array} \right] \left[\begin{array}{l} [\text{修正文字列}] \\ [\text{COLUMN} (\text{カラム1} [\text{カラム2}])] \end{array} \right] \right]$

(2)機能

CORRECTサブコマンドは、指定した修正文字列にしたがい、行単位にカラム範囲内で文字列を修正するのに用いる。文字列の修正を行う範囲は、行区間やカラム範囲で指定することができる。

CORRECTサブコマンドを用いて以下のようなことが行える。

- ・ 特定の文字列を空白に修正する。
- ・ 特定の文字列を削除する。
- ・ 特定の文字列の前に文字列を挿入する。
- ・ 特定の文字列の後に修正カラム範囲を指定した文字列で置き換える。

修正文字列オペランドを指定した場合で、VERIFY機能が起動されていれば、CORRECTサブコマンドで修正された行が端末に出力され、修正結果を確認することができる。CORRECTサブコマンドの実行後、行指標は修正された最終行を指す。

(3)オペランドの説明

* [整数1]

修正対象範囲を、行指標(*)を利用して指定する。行指標が現在指している行から整数1で指定した行数分の一連の行区間が修正文字列オペランドで指定されたキーにより修正され

る。整数1を省略すると1を指定したとみなされ、行指標が現在指している行のみが修正行となる。行範囲が指定されていないときは、行指標が現在指している行が修正対象となる。整数1の指定できる範囲は次のとおりである。

$$1 \leq \text{整数1} \leq 99999999$$

行番号1 [行番号2]

修正対象行区間を、行番号で指定する。行番号付きで編集しているときのみ指定できる。行番号1で示した行から行番号2で示した行までの一連の行が、修正文字列オペランドで指定されたキーにより修正される。行番号1と行番号2が等しいとき、又は行番号2を省略したときは、行番号1で指定した一行のみが修正対象となる。行番号1と行番号2の関係とその指定できる範囲は次のとおりである。

$$0 \leq \text{行番号1} \leq \text{行番号2} \leq 99999999$$

修正文字列

以下に示す修正文字がキーとなって対応する文字位置の文字が修正される。

修正文字列に小文字を指定しても無効である。

- _(空白入力) : 対応する文字はそのままとする。
- B(Blank) : 対応する文字を空白文字に修正する。
- D>Delete) : 対応する文字を削除する。
- I(Insert) : 対応する文字の前に文字列を挿入する。
- R(Replace) : 対応する文字から後の修正カラム範囲を指定した文字列で置き換える。

IとRで指定する挿入文字列あるいは置換文字列は、修正文字列のうち、IあるいはRの後にある部分である。

修正文字列オペランドが省略されると、指定された行範囲の指定されたカラム範囲の行が一行ずつ順に端末に出力される。このとき、端末ユーザは修正文字を入力し、対応する文字を修正する。修正された行が再度端末に出力され、まだ修正の必要があるなら修正文字を入力する。修正の必要がなければ空行を入力すればよい。行範囲が指定されているなら、次の行が端末に出力され、順次修正することができる。

COLUMN (カラム1 [カラム2])

修正対象カラム区間を指定する。修正対象の行や行区間のうち、カラム1からカラム2までのカラム区間が修正対象となる。カラム1のみ指定してカラム2を省略すると、カラム1以降の全カラムが修正対象となる。カラム2はカラム1と等しいかより大きくなければならない。このオペランドを省略すると、行全体が修正対象となる。カラム1とカラム2の関係とその指定できる範囲は次のとおりである。

$$1 \leq \text{カラム1} \leq \text{カラム2} \leq \text{レコード長}$$

(4)使用上の注意

- a. 修正文字列オペランドは引用符法と特殊句切り符号法のどちらでも指定可能である。

3.2 MATCHサブコマンド

(1)入力形式

サブコマンド	オペランド
MATCH	$\left[\left[* \quad [\text{整数1}] \right] \left[\text{検索文字列} \right] \right]$ $\left[\left[\text{行番号1} \quad [\text{行番号2}] \right] \left[\text{COLUMN}(\text{カラム1} \quad [\text{カラム2}]) \right] \right]$

(2)機能

MATCHサブコマンドは、指定した検索文字列を含む行を探し、そこに行指標を移動する。検索文字列を探す範囲は、行区間やカラム範囲で指定することができる。検索文字列が見つからなければ行指標は変化しない。

VERIFY機能が起動されていれば、検索文字列が見つかりその行に行指標が移動されると同時にその行が端末に出力される。

オペランドをすべて省略したMATCHサブコマンドを入力すると、最後に入力したオペランド付きのMATCHサブコマンドで指定した検索文字列及び検索対象範囲が有効となり、その検索文字列が、行指標が現在指している行の次の行から探し始められる。したがって、オペランドをすべて省略したMATCHサブコマンドをその後、次々に入力すれば、検索文字列を含む行を順に見つけることができる。

(3)オペランドの説明

* [整数1]

検索対象範囲を、行指標(*)を利用して指定する。行指標が現在指している行から整数1で指定した行数分の一連の行区間が検索対象区間となる。整数1を省略すると1を指定したとみなされ、行指標が現在指している行のみが検索対象となる。行範囲が指定されていないときは、行指標が現在指している行から最終行までが検索対象区間となる。整数1の指定できる範囲は次のとおりである。

$$1 \leq \text{整数1} \leq 99999999$$

行番号1 [行番号2]

検索対象行区間を行番号で指定する。行番号付きで編集しているときのみ指定できる。行番号1で示した行から行番号2で示した行までの一連の行区間で探され、その文字列が見つかった最初の行に行指標が移動する。行番号1と行番号2が等しいとき、又は行番号2を省略したとき、行番号1で指定した一行のみが検索対象となる。行番号1と行番号2の関係とその指定できる範囲は次のとおりである。

$$0 \leq \text{行番号1} \leq \text{行番号2} \leq 99999999$$

行区間を指定した場合、行番号1や行番号2で指定した行番号をもつ行がなければ、その範囲内の一連の行が検索対象となる。

検索文字列

検索文字列は、FINDサブコマンドで指定できる文字に加え、以下の変数文字を指定できる。

- & : 対応する文字は英数字である。(各国語文字を含む)
- ¥ : 対応する文字は英字である。(各国語文字を含む)
- # : 対応する文字は数字である
- % : 上記以外の文字列に対応する。

変数文字は、対応する文字であるなら、どんな文字でも探すことが可能である。

例えば、KEQEF01B, KEQEF02B, KEQEF03B, KEQEFAABを探すには、

MATCH 'KEQEF&&B'

と指定すればよい。

COLUMN (カラム1 [カラム2])

検索対象のカラム区間を指定する。検索対象の行や行区間のうち、カラム1からカラム2までのカラム区間が検索対象となる。カラム1のみ指定してカラム2を省略すると、カラム1以降全カラムが検索対象となる。カラム2は、カラム1と等しいかより大きくなければならない。このオペランドを省略すると、行全体が検索対象となる。カラム1とカラム2の関係とその指定できる範囲は次のとおりである。

$$1 \leq \text{カラム1} \leq \text{カラム2} \leq \text{レコード長}$$

(4)使用上の注意

- a. FSO処理中に実行すると、フルスクリーンモードからEDITモードの画面になる。
- b. 検索文字列オペランドは引用符法と特殊句切り符号法のどちらも指定可能である。

3.3 UNDOサブコマンド

(1)入力形式

サブコマンド	オペランド
UNDO	[OFF ON]

(2)機能

UNDOサブコマンドは、UNDO機能(更新結果無効機能)を起動又は停止させるのに用いる。UNDO機能とは次のような機能をいう。

- ・更新を伴うサブコマンドの実行結果を取り消して更新前の状態にする。更新されたレベルにより、順次に各サブコマンドの実行結果を取り消すことができる。

(3)オペランドの説明

OFF

UNDO機能を停止させるとき指定する。

ON

UNDO機能を起動させるとき指定する。

オペランドを指定しない場合

直前の修正用サブコマンドの更新結果を取り消す。

UNDOサブコマンドは、1回に1レベルの更新結果を取り消す。UNDOサブコマンドが入力されるたびに更新結果をさかのぼって取り消すことができる。

(4)使用上の注意

- a. UNDOサブコマンドを入力してもAUTOSAVEサブコマンドのカウンタは、更新結果を取り消した時点の値にならず、UNDOサブコマンドの入力前の値である。
- b. UNDOサブコマンドを入力しても現在の行指標は、更新結果を取り消した時点の値に戻らない。
- c. 1回のサブコマンド処理で多量の更新処理を実施すると(特に、MERGEやRENUM、UNNUMサブコマンドを実行した場合)、UNDOサブコマンドの制御テーブルに更新データを収納しきれなくなり、UNDOサブコマンドの機能が不可能となることがある。この場合は、メッセージ'KEQ52599I'が出力され、以後UNDOサブコマンドの制御テーブルに更新データが収納されなくなる。

なお、UNDOサブコマンドの制御テーブルの大きさは、データセットのレコード数に依存する。収納できるデータ数は、サブコマンドの更新処理に依存する。UNDOサブコマンドの制御テーブルの大きさ(一時データセットの大きさ)は、次の計算式で求められる。二次量は割り当てられない。

$$\begin{aligned} \text{一次量(ブロック数)} &= (\text{編集集中の総レコード数} / 80) \times 3 \\ \text{ブロック長} &= 8192 \text{バイト} \end{aligned}$$

ただし、この計算式によって求められたブロック数が20以下(レコード数が560未満)の場合は20となる。

- d. FSO処理中に実行すると、フルスクリーンモードからEDITモードの画面になる。

4. おわりに

ラインエディタEDITのサブコマンドCORRECT、MATCH、UNDOの利用法について述べた。フルスクリーンエディタFSOのもとではCORRECTは意味をもたないが、MATCHとUNDOは、FSOのもとでも非常に有効である。これらの3個のサブコマンドは、利用者要望により富士通が開発を行ったものである。エディタについての、このような要望があれば、お寄せいただきたい。

参考文献

1. 計算機マニュアル, FACOM OS IV/F4 MSP TSS コマンド文法書(78SP-1341-1), 富士通(株).
2. 計算機マニュアル, FACOM OS IV TSS テキスト編集使用手引書 Full Screen Option (FSO)編 (64sp-2080-1), 富士通(株).
3. 計算機マニュアル, FACOM OS IV/F4 MSP PFD 使用手引書 プログラム開発機能編 V10用(78SP-3101-4), 富士通(株).

付録. CORRECT, MATCH, UNDO サブコマンドのエラーメッセージ

KEQ52591I

UNDO ENVIRONMENT ALREADY DEFINED

[メッセージの意味] (UNDO サブコマンド)

UNDO 環境は既に定義されています。

[システムの処理]

サブコマンドの処理を中止し、次のサブコマンドの入力を待つ。

KEQ52592I

INVALID UPDATE STRING

[メッセージの意味] (CORRECT サブコマンド)

修正文字列が誤っています。

[システムの処理]

端末に修正範囲をプロンプティングし、次の修正文字列の入力を待つ。

[端末オペレータの処理]

修正文字列を確認して再度入力する。

KEQ52593I

LINE TRUNCATED

[メッセージの意味] (CORRECT サブコマンド)

修正された行が切り捨てられました。

[システムの処理]

データ長より長いレコード長をデータ長で切り捨て、処理を続行する。

KEQ52594I

UNDO ENVIRONMENT NOT DEFINED

[メッセージの意味] (UNDO サブコマンド)

UNDO 環境が定義されていません。

[システムの処理]

サブコマンドの処理を中止し、次のサブコマンドの入力を待つ。

KEQ52595I

INSUFFICIENT STORAGE TO EXECUTE, UNDO SUBCOMMAND IGNORED

[メッセージの意味] (UNDO サブコマンド)

UNDO サブコマンドを実行するための記憶域が足りません。UNDO サブコマンドは無視

されました。

[システムの処理]

サブコマンドの処理を中止し、次のサブコマンドの入力を待つ。

[端末オペレータの処理]

端末のリージョンサイズを拡張し、再度実行する。

KEQ52596I

UNDO SUBCOMMAND NOT AVAILABLE, EXCEED MAXIMUM LOGGING RECORD NUMBER

[メッセージの意味] (UNDOサブコマンド)

最大ログレコード番号を超えたので、UNDOサブコマンドは使用できません。

[システムの処理]

サブコマンドの処理を中止し、次のサブコマンドの入力を待つ。

[端末オペレータの処理]

いったんUNDO機能を停止し、再度UNDO機能を起動する。

KEQ52597I

NO LOGGING RECORD

[メッセージの意味] (UNDOサブコマンド)

ログレコードはありません。

[システムの処理]

サブコマンドの処理を中止し、次のサブコマンドの入力を待つ。

KEQ52598I

UNDO SUBCOMMAND NOT AVAILABLE, OCCURRED I/O ERROR

[メッセージの意味] (UNDOサブコマンド)

入出力エラーが発生したので、UNDOサブコマンドは使用できません。

[システムの処理]

サブコマンドの処理を中止し、次のサブコマンドの入力を待つ。

[端末オペレータの処理]

いったんUNDO機能を停止し、再度UNDO機能を起動する。

KEQ52599I

LOGGING FILE OVERFLOWED

[メッセージの意味] (UNDOサブコマンド)

ログファイルがオーバーしました。

[システムの処理]

更新結果のログレコードを中止し、サブコマンドの処理を続行する。

[端末オペレータの処理]

いったんUNDO機能を停止し、再度UNDO機能を起動する。または、オペランドなしのUNDOサブコマンドを入力してからレコードの更新を行う。