EDIT 81 Glossary
9/21/81

This is a description of the EDIT voabulary, which inludes the new file management system. The words are presented in ASCII order. In addition, there is also a Control Character Mode in the editor. The explanation of these characters is presented at the end of the EDIT 81 Glossary. The symbols used here are the same as those used in the TERSE 81 Glossary.

NOTE: A * in an entry denotes that this is an actual editting verb.
All characters in Control Character Mode are editting verbs.

<ADRIVE

Temporarily switch directories to the A disk. This can be used to load files off the A drive from a file on the B drive (See <BDRIVE and DRIVE>).

<BDRIVE

Temporaily switch directories to the B disk. This can be used to load files off the B drive from a file on the A drive (See <ADRIVE and DRIVE>).

?]] *

Terminate the "repeat" command (See [[ ).

[[* [[command ?]]

Repeat the R or S command delimited by ?]] until a NOT FOUND condition is found.

A! n ---

Physical block n is stored as the allocation block in the directory. Used when a file is first CREATEd.

A!INIT n ---

Initializes a new file by making physical block n the allocation block for the file, and setting up any other variables.

A-FINSERT A-FINSERT nnnn

Does a FINSERT nnnn where nnnn specifically resides on the A drive (See B-FINSERT).

ABLOCK --- n

Brings the allocation block into memory and returns its address.

ADRIVE

Select the A disk for all file commands. This is the default.

ALLOC-BLK --- n

Returns the address of a variable containing the current allocation block number.

B *

Reset Line 0 to the beginning of the block and display the block.

B-FINSERT B-FINSERT nnnn

Does a FINSERT nnnn where nnnn specifically resides on the B drive (See A-FINSERT).

BDRIVE

Select the B disk for all file commands (See ADRIVE).

CREATE CREATE nnnn

Create a new file named nnnn. The file is originally empty so a INSERT or FINSERT must be done before editting can be performed.

CUR-BLK --- n

Returns the address of a variable containing the current relative block number.

CUR-DRIVE --- p

Return the address of a variable containing the current drive that the directory is from (0 = A drive).

CUR-PNT --- n

Returns the address in the allocation block of pointer to current block.

D *

Move Current Line down one line with the first character on the line being the next Copy Character.

DBLK --- n

Returns the physical block number of the directory for either drive.

DBLOCK --- p

Brings the current directory in from the disk, and returns the address of where it is in RAM.

DELETE

Delete the Current Block from the file. This block is now free for further use.

DESTROY DESTROY nnnn

Delete file nnnn from the directory on the currently selected disk. All blocks that were used by the file are now available for re-use.

DIR-BLOCK --- p

Return the address of a variable containing the physical block number of the disk directory.

DIR-SIZE --- n

Returns an address of a word containing the directory size.

DMAP! n m ---

Set the bit in the free block bit map for block m to n.

DMAP-OFFSET --- n

Returns the offset into the directory block of the bit map.

DMAP@ n --- m

Returns a 0 if block n is free or non-zero if block n is inuse.

DMAP --- p

Returns the address of where the bit map of free blocks is in RAM.

DRIVE+ m --- n

Returns the block number for physical block m whether it is on the A or B drive (ie. Adds 308 for blocks on the B drive).

DRIVE>

Re-select the disk that was selected before <ADRIVE or <BDRIVE was executed.

DRIVETOBE --- n

Returns the address of a variable containing the drive for file operations.

ED * b ---

Display block b and begin editting it. This works for physical blocks only (See FED).

ENDPRT

Executes the TERSE verbs PAGE PAGE .NLIST

FBLOAD m n p --- q

Loads memory beginning at address m with the binary data beginning at block p and ending with block n. Returns the last address loaded with binary data.

FBSAVE p n m ---

Saves memory beginning at address p into blocks beginning at m and ending at n.

FED * FED nnnn

Point to file nnnn, and begin editting relative block 0. Block 0 is listed.

FILE FILE nnnn

Point to file nnnn for further file operations.

FILE-EMPTY-CHECK

Checks to see if the file was empty, and ABORTs if it is.

FILE-PHYS n ---

Copy the entire contents of the currently selected file to physical block n and each consecutive block afterward.

FILE-SIZE --- n

Returns the number of blocks used by the current file.

FINSERT FINSERT nnnn

Insert file nnnn after the Current Block. If you have a blank file, this can be used so that an uninitialized block will not be inserted by INSERT. This is also a technique for copying files. First, CREATE a new file, then FINSERT the source (See A-FINSERT and B-FINSERT).

FREE n ---

Mark physical block n as free in the bit map for free blocks.

FSFIG m n p --- q r s t u

Given a starting block number (m), the beginning (n) and ending (p) addresses in memory, FSFIG will determine how many blocks are needed to hold the data between the 2 addresses. It returns next block number (q), beginning address (r), ending block number + 1 (s), starting block number (t), and the ending block number + 1 (u).

GET-FREE-BLOCK --- n

Return a physical block number for the next free block on the disk.

INCREASE n ---

Inserts n blocks after the current block; the blocks are unitialized.

INIT-DIR

Initializes a new directory on a disk and marks all blocks except the directory block number as free.

INIT-DMAP

Initialize the bit map for a disk. This marks blocks 1 - 44 as inuse and all the rest as free.

INSERT

Insert a free block after the Current Block. All relative block numbers after the current block are renumbered to 1 higher value. The information on the inserted block is not initialized so it may not be blank (See FINSERT).

INUSE n ---

Mark physical block n as "being used" in the bit map.

IS *

Insert all saved text between Current Line and the one before it (See SV).

K * n ---

Delete n+1 lines beginning at the Current Line.

L * n ---

Move the Current Line up or down n lines, and enter Control Character Mode.

LED *

Begin editting the last block that was listed by the List command, V.

LOG-PHYS n --- m

Given a relative block number of a block in the current file, it returns the physical block number.

LOGBLK --- n

Returns the current relative block number.

LOOK LOOK nnnn --- n f

Looks for word nnnn on the dictionary and returns its address and False if its found; otherwise, it returns a True.

M * n ---

Move the Current Line up or down n lines. Equivalent to L except that Control Character Mode is not entered.

MAX-SIZE --- n

Returns the maximum directory size.

N *
Edit the Next block. This works for both files and physical blocks.

NED * m NED nnnn
Begin editting relative block m in file nnnn. The block is listed first.

NL *
Edit the Next block, set the Current Line to the first line, and enter Control Character Mode.

O=
Executes the TERSE verbs OVER then =.

P *
Edit the Previous block. This works for both files and physical blocks.

P-MARK n ---
Insert physical block n after the current block.

PBLK --- n
Returns the current physical block number.

PHYS-INSERT m n ---
Insert physical blocks n thru m-1 after the Current Block. This is useful for transfering blocks from the non-file system to the file system. First, CREATE the file, then do a m n PHYS-INSERT.

PRINTFILE PRINTFILE nnnn
Lists out file nnnn on the currently selected output device.

R * R ssss1\ssss2\
Search for string ssss1 beginning at the Current Line and Character. If found, replace it with string ssss2 which may be null. If it's not found, Current Line is set to the beginning of the block.

RENAME RENAME nnnn1 nnnn2
Rename the file nnnn1 as nnnn2.

RMOVE p q n ---
Works like the TERSE BMOVE, but moves in the reverse order.

S * S ssss\
Search for string ssss beginning at the Current Line and character. If the string is not found, current is set to the beginning of the block.

SB * m n SB ssss\
Search for string ssss beginning with block n and ending with block m. This works for both files and physical blocks.

SN * SN ssss\
Search for string ssss beginning at the current block and sequentially through blocks (FILES or NOFILES).

SDIR

Displays the directory of files on the currently selected disk.

SV

* n ---

Save n+1 lines starting at the Current Line (See IS).

U

*

Move Current Line up one line, and the first character on the line will be the copy character.

V

*

Display the screen being editted. Current Line is numbered 0.

EDIT - CONTROL CHARACTER MODE

<CR>

Insert the Line Buffer between the Current Line and the next line. The insert line becomes the Current Line.

<ESC>

Put the Line Buffer on top of the Current Line, display the block, and return to non-Control Character Mode.

<LF>

Put the Line Buffer on top of the Current Line. The next line becomes the Current Line.

<RUB>

Delete the previous character in the Line Buffer.

^A

This inserts the rest of the characters on the line beginning at the current copy character into the Line Buffer.

^B

Insert the Line Buffer before the Current Line. The inserted line becomes the Current Line.

^C

Insert the next Copy Character into the Line Buffer.

^D

Same as D in non-Control Character Mode.

^E

Move the Current Edit Character up by one word.

^F

See FLUSH in the TERSE 81 Glossary.

^G

Toggle the ability of ^E and ^W to get the trailing space.

^I

Tab to the next Mod 8 position in the Edit Buffer.

^L

Advance the Current Edit Character by one character.

^N

Same as N in the non-Control Character Mode.

^O

Display the numbered overlay of character positions.

^P

Sames as P in non-Control Character Mode.

^Q

Return to non-Control Character Mode. No FLUSH is performed.

^R

Advance the Current Edit Character back to the beginning of the Edit buffer.

^T

Display the Line Buffer and reset the Next Copy Character to the first character on the line.

^U

Same as U in non-Control Character Mode.

^V

Same as V in non-Control Character Mode.

^W

Insert the next word starting at the next Copy Character into the Line Buffer.

^X

Kill the Line Buffer.

^Y

Delete the Current Line, and put it in the Line Buffer.

^Z

Delete the previous word in the Line Buffer (<DEL> also does this).

----------------End of EDIT 81 Glossary----------------