EDITOR USER MANUAL

Bill Stoddart
of FIG, United Kingdom

1 Introduction

FORTH organizes its mass storage into "screens" of 1024 characters. If, for example, a diskette of 250k byte capacity is used entirely for storing text, it will appear to the user as 250 screens numbered 0 to 249.

Each screen is organized as 16 lines with 64 characters per line. The FORTH screens are merely an arrangement of virtual memory and need not correspond exactly with the screen format of a particular terminal.

2 Selecting a Screen and Input of Text

To start an editing session the user types `EDITOR` to invoke the appropriate vocabulary.

The screen to be edited is then selected, using either:

`n LIST` list screen n and select it for editing, *or*

`n CLEAR` clear screen n and select for editing.

To input new text to screen n after `LIST` or `CLEAR` the `P` (put) command is used.

Example:

```
0 P THIS IS HOW
1 P TO INPUT TEXT
2 P TO LINES 0, 1, AND 2 OF THE SELECTED SCREEN.
```

3 Line Editing

During this description of the editor, reference is made to `PAD`. This is a text buffer which may hold a line of text used by or saved with a line editing command, or a text string to be found or deleted by a string editing command.

`PAD` can be used to transfer a line from one screen to another, as well as to perform edit operations within a single screen.

3.1 Line Editor Commands

`n H` Hold line n at `PAD` (used by system more often than by user).

`n D` Delete line n but hold it in `PAD`. Line 15 becomes blank as lines n+1 to 15 move up 1 line.

`n T` Type line n and save it in `PAD`.

`n R` Replace line n with the text in `PAD`.

`n I` Insert the text from `PAD` at line n, moving the old line n and following lines down. Line 15 is lost.

`n E` Erase line n with blanks.

`n S` Spread at line n. n and subsequent lines move down 1 line. Line n becomes blank. Line 15 is lost.

4 Cursor Control and String Editing

The screen of text being edited resides in a buffer area of storage. The editing cursor is a variable holding an offset into this buffer area. Commands are provided for the user to position the cursor, either directly or by searching for a string of buffer text, and to insert or delete text at the cursor position.

4.1 Commands to Position the Cursor

`TOP` Position the cursor at the start of the screen.

`n M` Move the cursor by a signed amount n and print the cursor line. The position of the cursor on its line is shown by a `_` (underline).

4.2 String Editing Commands

`F text` Search forward from the current cursor position until string '*text*' is found. The cursor is left at the end of the text string, and the cursor line is printed. If the string is not found an error message is given and the cursor is repositioned at the top of screen.

`B` Used after `F` to back up the cursor by the length of the most recent text.

`N` Find the next occurrence of the string found by an `F` command.

`X text` Find and delete the string '*text*'.

`C text` Copy in text to the cursor line at the cursor position

`TILL text` Delete on the cursor line from the cursor till the end of the text string '*text*'.

NOTE: Typing `C` with no text will copy a `NULL` character into the text at the cursor position. This will abruptly stop subsequent compilation! To delete this error type `TOP X` 'return'.

4.3 Screen Editing Commands

`n LIST` List screen *n* and select it for editing

`n CLEAR` Clear screen *n* with blanks and select it for editing

`n1 n2 COPY` Copy screen *n1* to screen *n2*.

`L` List the current screen. The cursor line is relisted after the screen listing, to show the cursor position.

`FLUSH` Used at the end of an editing session to ensure that all entries and updates of text have been transferred to disk.

5 Editor Glossary

`TEXT c ---`
Accept following text to pad. *c* is text delimiter.

`LINE n --- addr`
Leave address of line *n* of current screen. This address will be in the disc buffer area.

`WHERE n1 n2 ---`
n2 is the block no., *n1* is offset into block. If an error is found in the source when loading from disc, the recovery routine `ERROR` leaves these values on the stack to help the user locate the error. `WHERE` uses these to print the screen and line nos. and a picture of where the error occurred.

`R#  --- addr`
A user variable which contains the offset of the editing cursor from the start of the screen.

`#LOCATE --- n1 n2`
From the cursor position determine the line-no *n2* and the offset into the line *n1*.

`#LEAD --- line-address offset-to-cursor`

`#LAG --- cursor-address count-after-cursor-till-EOL`

`-MOVE addr line-no ---`
Move a line of text from *addr* to *line-no* of current screen.

`H  n ---`
Hold numbered line at `PAD`.

`E  n ---`
Erase line *n* with blanks.

`S  n ---`
Spread. Lines *n* and following move down. *n* becomes blank.

`D  n ---`
Delete line *n*, but hold in `PAD`.

`M  n ---`
Move cursor by a signed amount and print its line.

`T  n ---`
Type line *n* and save in `PAD`.

`L  ---`
List the current screen.

`R  n ---`
Replace line *n* with the text in `PAD`.

`P  n ---`
Put the following text on line *n*.

`I  n ---`
Spread at line *n* and insert text from `PAD`.

`TOP  ---`
Position editing cursor at top of screen.

`CLEAR n ---`
Clear screen *n*, can be used to select screen *n* for editing.

`FLUSH ---`
Write all updated buffers to disk. This has been modified to cope with an error in the Micropolis CP/M disk drivers.

`COPY n1 n2 ---`
Copy screen *n1* to screen *n2*.

`-TEXT addr1 count addr2 -- boolean`
True if strings exactly match.

`MATCH cursor-addr bytes-left-till-EOL str-addr str-count`
`--- tf cursor-advance-till-end-of-matching-text`
`--- ff bytes-left-till-EOL`
Match the string at *str-addr* with all strings on the

cursor line forward from the cursor. The arguments left allow the cursor `R\#` to be updated either to the end of the matching text or to the start of the next line.

`1LINE --- f`
: Scan the cursor line for a match to `PAD` text. Return flag and update the cursor `R\#` to the end of matching text, or to the start of the next line if no match is found.

`FIND ---`
: Search for a match to the string at `PAD`, from the cursor position till the end of screen. If no match found issue an error message and reposition the cursor at the top of screen.

`DELETE n ---`
: Delete n characters prior to the cursor.

`N ---`
: Find next occurrence of `PAD` text.

`F ---`
: Input following text to `PAD` and search for match from cursor position till end of screen.

`B ---`
: Backup cursor by text in `PAD`.

`X ---`
: Delete next occurrence of following text.

`TILL ---`
: Delete on cursor line from cursor to end of the following text.

`C ---`
: Spread at cursor and copy the following text into the cursor line.