FORTH EDITOR

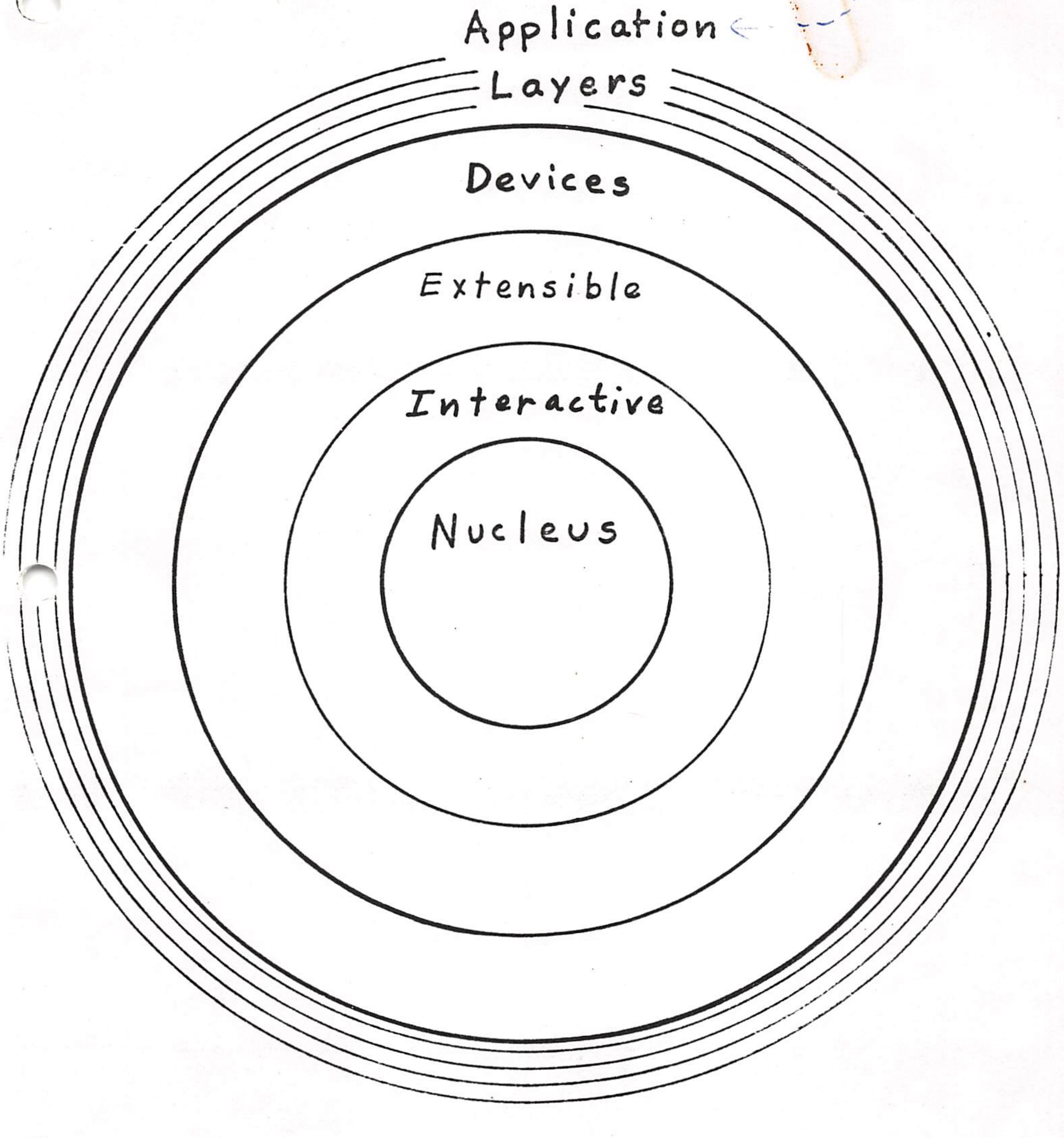


fig FORTH Editor: Bootstrap & Extensions

```
HEX
:  TEXT HERE C/L 1+ BLANKS WORD HERE PAD C/L 1+ CMOVE  ;
:  LINE DUP FFF0 AND 17 ?ERROR SCR @ (LINE) DROP ;
:  -MOVE LINE C/L CMOVE UPDATE ;
:  P 1 TEXT PAD 1+ SWAP -MOVE ;
DECIMAL
```

These words define the elementary editing command "P" which places a line of text on a screen. Blanks are significant. FORTH should respond "OK" after each line is entered. The syntax for its use is:

```
line-number P text-to-be-entered-on-the-line
```

For example, to enter line one of screen 87 type:

```
1 P FORTH DEFINITIONS HEX
```

and type return. FORTH should respond "OK". If you then type:

```
screen-number LIST
```

you should see that text at line number 1.

```
16 LIST
SCR # 16
  0  ( Screen Editor... CLEAR  COPY                                  )
  1  : CLEAR             ( CLEAR screen by number-1 *)
  2      SCR ! 10 0 DO FORTH I EDITOR E LOOP ;
  3
  4  : COPY          ( duplicate screen-2 onto screen-1 *)
  5      B/SCR * OFFSET @ + SWAP B/SCR * B/SCR OVER + SWAP
  6      DO DUP FORTH I BLOCK 2 - ! 1+ UPDATE LOOP
  7      DROP FLUSH ;
  8    EDITOR
  9 : WIPE    ( 1stScr# lastScr# ---      blanks range of screens )
 10    1+ SWAP DO    FORTH I    EDITOR CLEAR    LOOP ;
 11
 12 : RIGHT   ( 1stScr# lastScr# --- )
 13    ( copies range of screens from DR0 to DR1 )
 14    1+ SWAP DO    FORTH I    I FA +    EDITOR COPY    LOOP ;
                                 ↑
                                 Hex [illegible]
 15
OK
17 LIST
SCR # 17
  0 (    EDITOR:    NEW )
  1 DECIMAL
  2 : NEW     ( line# ---    replaces text from line# until null line)
  3    FORTH    16 0 DO    CR I 3 .R SPACE    I OVER =
  4       IF    QUERY    1 TEXT    PAD 1+    C@
  5          IF ( not null line )    I    EDITOR R    FORTH 1+
  6          ELSE    08 EMIT ( BS )    I    SCR @    .LINE
  7          THEN
  8       ELSE    I    SCR @    .LINE
  9     THEN    LOOP    DROP    ;
 10
 11
```

```
SCR # 148
  (  double number support                          WFR-80APR24 )
  (  operates on 32 bit double numbers   or two 16-bit integers )

  : 2DROP    DROP  DROP  ;  ( drop double number )

  : 2DUP    OVER  OVER  ;   ( duplicate a double number )

  : 2SWAP    ROT  >R  ROT  R>  ;
        ( bring second double to top of stack )
  ;S


  XXXXX



SCR # 149
  (  String MATCH for editor                     PM-WFR-80APR25 )
  : (MATCH)                     ( address-3, address-2, count-1 --- )
                          ( leave boolean matched=non-zero, nope=zero )
    -DUP  IF  OVER  +  SWAP    ( neither address may be zero ! )
              DO  DUP  C@  FORTH  I  C@  -
                 IF  0=  LEAVE  ELSE  1+  THEN    LOOP
             ELSE  DROP  0=  THEN  ;
  : MATCH   ( cursor address-4, bytes left-3, string address-2, )
  ( string count-1, ---  boolean-2, cursor movement-1 )
    >R  >R  2DUP  R>  R>  2SWAP  OVER  + SWAP
    ( caddr-6, bleft-5, $addr-4, $len-3, caddr+bleft-2, caddr-1 )
    DO  2DUP  FORTH  I  SWAP  (MATCH)
     IF R 2DROP  [illegible]  FORTH  I  SWAP  -  0  SWAP  0  0  LEAVE
     ( caddr  bleft  $addr  $len  or else  0  offset  0  0  )
       THEN  LOOP    2DROP  ( caddr-2, bleft-1, or 0-2, offset-1 )
     SWAP  0=  SWAP  ;
 OK
```

MATCH finds ok but cursor advancement must step <u>over</u> the found string. Parameter return must be incremented by string length!

This patch is untested!

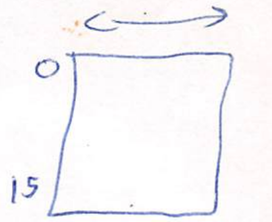
R> - ← Add ~~Same~~ as in NAUTILUS ~~Editor~~ Fig source

Scr 71

Scr# LOAD results in executions & compilations at end of screen (or block, if sooner)

E1
fig

64 char's



figFORTH EDITOR GLOSSARY

#LAG	--- addr n	88	Leave address of start of current line in a disk buffer. Also leave n, the # characters following the current cursor position.
#LEAD	--- addr offset	88	Leave the address of the start of the current line and the offset to the current cursor position.
#LOCATE	--- offset line#	88	Leave the current cursor offset relative to start of line and current line#. Uses contents of R# .
-MOVE	addr line# ---	88	Move C/L characters from addr to line# of the current screen on the disk.
CLEAR	screen# ---		Erase the designated screen with blanks.
COPY	source# dest# ---		Copy contents of screen from source# to dest# .
D	line# ---	89	Copy line# of current screen to PAD. Delete it by copying lower lines up one line and erase line 15.
E	line# ---	89	Erase line# of current screen with blanks.
H	line# ---	89	Copy line# of current screen to PAD. (non destructive)
I	line# ---	91	Insert the contents of PAD after line# of current screen. Lines below line# are moved down one line; the contents on line 15 is lost.
L	---	90	Relist the current screen then the current line followed by the current line number. Uses the contents of SCR .
LINE	line# --- addr	87	Leave the address of line# of the current screen.
M	n ---	90	Move the cursor by the signed number or characters, n . Print the current line followed by its line number.
NEW	line# ---		Print the current screen down to line# . Replace lines with entered text until a null line is entered (ie, (CR) only) then print the remainder of the screen.
P	line# ---	91	Put text following P in line# of current screen. Previous contents of this line is lost.

Use LOAD (as & text word) to stuff a lot of

- throws away any excess over 64 char's

Note: any overhang is lost

C Copy text following the space after C at the cursor pos'n, pushing the remaining line contents to the right ← "insert" (same as MVP FORTH's I)

~~TRICK:~~

X Extract text following the space after X, sliding the remainder of the line to the left (same as MVP FORTH's D)

~~357 CONSTANT APPLICATION~~

Note: If right end of line has a char, and left end of next line has a char, the "word" overlapping the line boundary will get "sucked" into the current line and the ~~next~~ line will be slid to the left accordingly

~~then~~ ~~APPLICATION LOAD~~

52

R	line# ---	91	Copy line at PAD to current screen at line# .
S	line# ---	89	Spread lines of the current screen. line# becomes blank, the previous contents of this line is moved down one line as are lower lines. Line 15 is lost.
T	line# ---	90	Make line# the current line of the current screen. Move the cursor to the beginning of this line, copy it to PAD , then print the line followed by its line number.
TEXT	delim ---	87	Move text from TIB to PAD until delim character is encountered.
TOP	---	91	Move cursor to the beginning of the current screen.
WHERE	in blk# ---	88	List the screen corresponding to blk# and type the line where in is pointing (the contents of IN) inside the screen. Used after a compilation error from mass storage (ie, a LOAD).

;S forces immediate termination of loading of the screen

on line 0, add a screen description & date, author I.D.