

CHAPTER 21. PRINTING UTILITY

The code discussed in this chapter is in UTILITY.BLK, screens 43 to 48.

The printing utility in F83 is designed for an EPSON printer. Using the compressed character size in the EPSON printer, 6 screens can be squeezed on a single 8.5" by 11" sheet of paper. You can print 6 consecutive screens to a page, or 3 screens of source code with 3 corresponding shadow screens to a page which is nice to show source and comments side by side. To use these high density printing format, it requires that your printer can print 128 characters per line. If your printer cannot handle 128 characters per line, the old faithful TRIAD should be used to print 3 screens to a page.

21.1. VARIABLES AND SETUP

: EPSON	(---)	Set up the EPSON MX-80 printer to print 132 columns per line.
CONTROL O EMIT		Send control O to printer, initialize compressed mode.
;		
DEFER INIT-PR		Printer initialization. Default is EPSON printer.
' NOOP IS INIT-PR		
DEFER FOOTING		Print message at the bottom of a page.
66 CONSTANT L/PAGE		Lines per page.
0 CONSTANT LOGO		The screen number of the logo screen where copyright notice can be stored and displayed.
VARIABLE #PAGE		Current page number during printing.
: PAGE	(---)	Do a form-feed and start a new page. It also increments the page number and resets line and column numbers. DOES>
	Vectored word.	
PERFORM		Do the form-feed in place of NOOP.
1 #PAGE +!		Increment page number.
#LINE OFF		Reset line number.
#OUT OFF		Reset column number.
;		
PAGE		Initialize itself.
: FORM-FEED	(---)	
CONTROL L EMIT		EPSON form feed control character.
;		
: (PAGE)	(---)	Print enough linefeeds to get to the next page. L/PAGE
	66 lines.	
#LINE @		Current line number.
OVER MIN ?DO		Use the lesser of the two.
CR		Out put that many linefeeds.

LOOP;

' (PAGE) IS PAGE

: (SEMIT)	(char ---)	Send a character to either printer or the console, but not both.
PRINTING @ IF		If printing flag is true,
(PRINT)		Send to printer.
ELSE (CONSOLE)		Otherwise, send to console.
THEN ;		
HIDDEN DEFINITIONS		HIDDEN is a vocabulary collecting words for internal usage to avoid cluttering up FORTH vocabulary with all kinds of junk words.
CREATE SCR#S 14 ALLOT		An array to hold a screen count and up to six screen numbers to be printed.
: PR-START	(---)	Initialize all printing functions.
PRINTING ON		Start the printer.
#LINE OFF		Top of page.
['] (SEMIT) IS EMIT		Re-vector EMIT to send characters to the printer.
SCR#S OFF		Reset screen numbers.
1 #PAGE !		Page number starts from 1, not 0.
INIT-PR		Initialize the printer.
;		
: PR-STOP	(---)	Stop the printer as the character output device.
['] (EMIT) IS EMIT		Vector EMIT to (EMIT) to send characters to the CRT terminal.
PRINTING OFF		Turn off printing flag.
;		

Vectoring the output word EMIT allows us to change the function of EMIT dynamically. The power of vectored execution is quite vividly demonstrated here. The output character string can be directed to any output device by defining individual device output words and store appropriate execution address in the parameter field of EMIT. EMIT was defined as a deferred word, which takes an address in the parameter field and executes it. PR-START and PR-STOP simply change the address in the parameter field of EMIT and the output can be directed at will.

21.2. PRINT TWO SCREENS SIDE BY SIDE

: TEXT?	(scr# --- f)	Given a screen number, return true if the first character in the screen is printable and the screen is not blank.	BLOCK DUP
C@		Get the first character in screen.	
BL ASCII ~ BETWEEN		Is this character printable?	
IF		Yes.	
B/BUF -TRAILING		Count of non-blank characters.	
NIP		Drop the buffer address.	
0<>		Return true if not a blank screen.	
ELSE		First character nonprintable.	

4

FALSE Push the false flag.
THEN ;

: PR (scr# ---) Add a screen to the SCR#S array and also increment the
 screen count at the beginning of SCR#S.
DUP CAPACITY >= Is scr# out of range?

IF DROP LOGO THEN	Yes. Substitute with logo screen.
1 SCR#S +!	Increment the screen count.
SCR#S DUP @	Fetch the screen count.
2* + !	Store scr into the appropriate cell in the SCR#S array.
;	

: 2PR (scr1 scr2 line# ---) Print the specified line from two screens given on the stack.
First the line in scr1, followed by the line in scr2.

CR DUP 2 .R SPACE	Print the line number.
C/L * >R	Save the character number.
PAD 129 BLANK	Clear the PAD buffer.
SWAP BLOCK	Buffer address of scr1.
R@ +	Address of first character in the specified line.
PAD C/L CMOVE	Copy one line from scr1 to PAD.
BLOCK R> +	Address of the first character of the line in scr2.
PAD C/L + 1+	Second half of PAD buffer with an additional space.
C/L CMOVE	Copy the line from scr2.
PAD 129 -TRAILING TYPE	Print the entire 129 characters.
;	

: 2SCR(scr1 scr2 ---) Print 2 screens across on a page. Call 2PR on a line by line basis.

CR CR 4 SPACE	Space between screens.
OVER 4 .R	Print header of scr1.
61 SPACES DUP 4 .R	Header of scr2.
16 0 DO	Scan down 16 lines.
I 2PR	Print.
LOOP	
2DROP	Discard the screen numbers.
;	

21.3. PRINT 6 SCREENS ON A PAGE

To print 6 screens on one page, one has to manage the screens and also put headings and footings on the page, making it look good and convenient to reference.

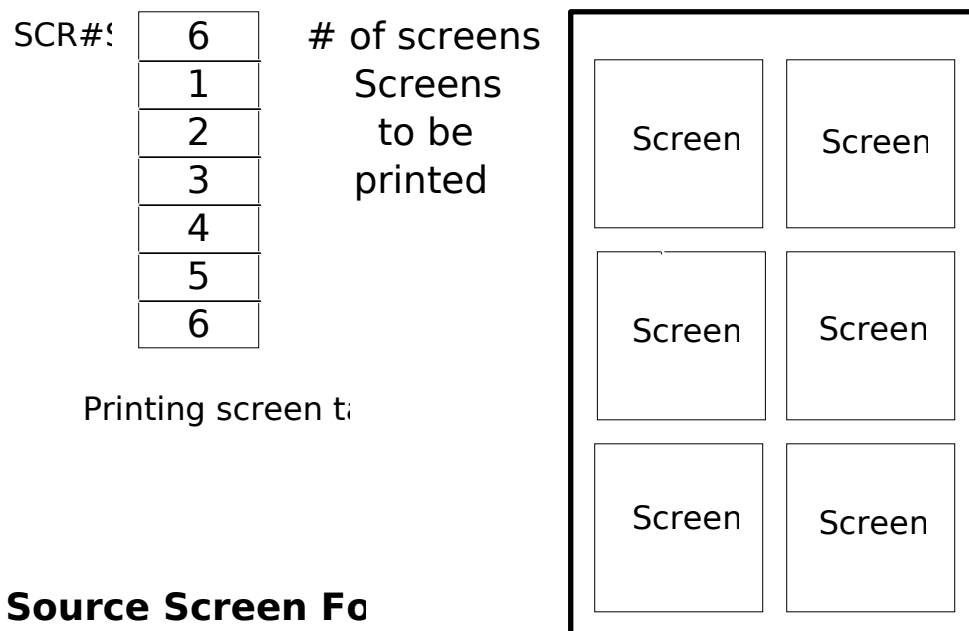
: P-HEADING(---)	Print a heading for each new page.
CR CR 5 SPACES	Top blank.
." page#" #PAGE ?	Print page number.
8 SPACES	
FILE? CR	Print the file name.
;	

: P-FOOTING (---)	Print the footing for each page and also do form feed.	CR	CR	58
SPACES	Some space.			
." Forth 83 Model"	Footing.			
PAGE	Form feed.			

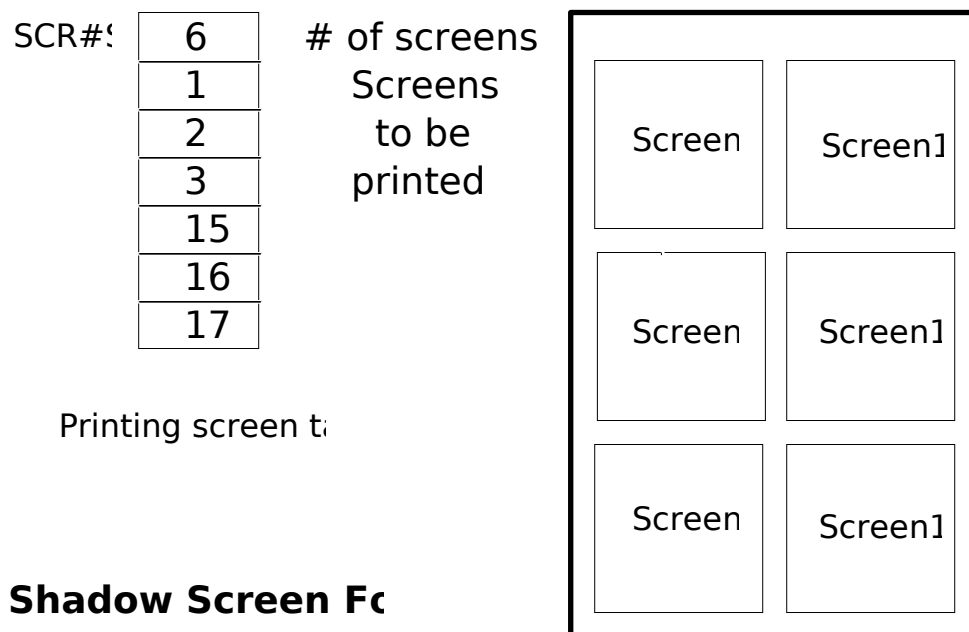
;

' P-FOOTING IS FOOTING

Figure 21.1 Two printing formats.



Source Screen Fo



Shadow Screen Fo

```

: PR-PAGE      ( --- )      Print a page worth of screens, 6 to a page without shadows. P-
HEADING        Print the heading.
  SCR#S OFF      Reset the screen count.
  SCR#S 2+       Address of first screen number to be printed.
  3 0 DO         3 screens per column.
    DUP @        Screen number for 1st column.
    OVER 6+ @    Screen number for 2nd column.
    2SCR         Print two screens side by side.
    2 +         Next cell in SCR#S array.
  LOOP DROP     Discard the SCR#S pointer.
  FOOTING       Print footing.
;

: PR-S-PAGE    ( --- )      Print a page worth of screens with shadows. Source screen
                           on the left and shadow screens on the right.
  P-HEADING
  SCR#S OFF
  SCR#S 2+
  3 0 DO
    DUP @        Screen number of source.
    OVER 2+ @    Screen number of shadow.
    2SCR         Print.
    4 +         Next pair of screens.
  LOOP DROP FOOTING
;

: PR-FLUSH     ( --- f )    Fill the SCR#S array with LOGO screen if a page is partially
                           filled. Return true flag if there is more to print.
  SCR#S @        Screen number.
  DUP IF        Yes, more screens to print.
    BEGIN
      SCR#S @    Screen number again.
      5 <        If screen number is less than 5, the SCR#S must be filled.
    WHILE 0 PR   Fill the array with 0's.
      REPEAT
        LOGO PR   Put the LOGO screen as the last.
      THEN
        0<>       Return the flag.
    ;

```

21.4. SHOW

There are two versions of SHOW defined in the F83 system to print screen files. One version prints consecutive screens and the other prints 3 screens of source with their respective shadows. The first version is defined in the FORTH vocabulary and the second one in the SHADOW vocabulary so that they are both accessible to the user.

1 20 SHADOW SHOW Print source with shadows.

1 20 FORTH SHOW Print source without shadow.

FORTH DEFINITIONS Define the SHOW without shadow screens in the FORTH
vocabulary.

```

: SHOW      ( first last --- ) Print 6 consecutive screens on a page. Blank screens are not
              printed.
  [ HIDDEN ] PR-START    Call PR-START in the HIDDEN vocabulary to turn on the
                        printer.
  1+ SWAP DO      Run through the range of screens.
    I TEXT?       Is this screen printable?
    IF I PR THEN Yes. Include it in the SCR#S array to be printed.
@              Get the number of screens in SCR#S array.
    6 =           Full?
    IF PR-PAGE THEN Yes. Print one page.
  LOOP PR-FLUSH    Fill the last page.
  IF PR-PAGE THEN  Print it if necessary.
  PR-STOP         Turn off the printer.
  ;

```

SHADOW DEFINITIONS Now get the SHADOW vocabulary to define the second version of SHOW with shadow screens.

```

: SHOW      ( first last --- ) Print 3 source screens with their shadow screens.
  [ HIDDEN ALSO ]      Push HIDDEN into the resident vocabulary array so that
                        other vocabulary can be invoked while HIDDEN is still
                        available for searching.
  PR-START           Turn on printer.
  1+ SWAP DO
    I TEXT?          A valid source screen?
    IF               Yes.
    I PR             Put it in SCR#S array.
    [ SHADOW ]       We need some words in the SHADOW vocabulary.
  >SHADOW           Get the number of shadow screen.
    PR              Put it in SCR#S also.
    THEN
    SCR#S @ 6 = End of SCR#S?
    IF PR-S-PAGE THEN Yes. Print page with shadows.
  LOOP
  PR-FLUSH
  IF PR-S-PAGE THEN
  PR-STOP      ;

```

ONLY FORTH ALSO DEFINITIONS Reset the vocabulary order and make FORTH a resident vocabulary as well as the context (transient) and current vocabulary.

```

: LISTING      ( --- )      Print the entire current file with shadow screens.
  0            First source screen.
  CAPACITY     Last screen in current file.
  2/ 1-        Last source screen in file.
  [ SHADOW ]    We want the SHOW with shadow, which is in SHADOW
                  vocabulary.

```

SHOW

Print the entire file in the shadow screen format.

;

Source screens printed with their corresponding shadow screens side by side serve very well as program reference and documentation.