DECUS

PROGRAM LIBRARY

DECUS NO.	8-445
TITLE	FYLHLP - PS/8 FILE UTILITY PROGRAM
AUTHOR	David M. Kristol
COMPANY	2401 Pennsylvania Avenue Wilmington, Delaware
DATE	May 10, 1971
SOURCE LANGUAGE	PAL-8

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

FYLHLP - PS/8 FILE UTILITY PROGRAM

DECUS Program Library Write-up DECUS NO. 8-445

ABSTRACT

FYLHLP is a utility program designed to help the PS/8 systems programmer maintain the file system and debug file-handling programs. It allows the user to list specific directory entries plus all "empty" entries on a file-structured device and to examine, modify, and search blocks on the same device. The program is divided into two logical sections, referred to below as "HELP" and "FYLLUK."

HELP Section

HELP receives its input from the system Command Decoder (CD) and modifies its operation depending on the CD options specified.

E Option

HELP prints out the Empty file entries on the directories of each device in the CD input string.

N Option

"No List." HELP just prints the location of the directory entry and the first data block number for each entry printed.

S Option

After printing out each file in the CD input string, control is transferred to the FYLLUK portion of the program.

Normal printout includes the location of the directory entry, the expanded entry, and the initial data block number.

Requests for non file-structured devices will result in error messages specifying the request number in the input string. After the message, HELP will proceed.

FYLLUK Section

FYLLUK may be reached only by going through HELP first, using the S option. The device that FYLLUK will "look at" is the last device which HELP has examined. Entrance into this section of the program is signified by '>' being typed.

OPERATION

FYLLUK maintains a one block (256 word) buffer in memory. Modifications to blocks are made in this buffer, which is later written back onto the device being examined. Blocks are not rewritten if no change has been made to them or if writing is inhibited by closing the block. The block in core will be rewritten, subject to the above conditions, when the user requests to examine a block other than the one in memory.

DEFINITIONS

Numbers

All numbers are octal. If more than four digits are typed, only the last four are considered.

Addresses

Addresses consist of two numbers separated by '.'. The first number is the block number, the second, the displacement within the block (Ø-377). When an address is called for, a single number typed without the '.' will be taken as that displacement within the current block ('CURBLK').

Examples:

Ø.3	location 3 in block Ø
1Ø.173ØØ51	location 51 in block 1Ø
177	location 177 of CURBLK

CURBLK is always displayed in the console accumulator lights.

Open Location

A location is considered "open" if its contents have been just typed out by some part of FYLLUK. A location must be open before it may be modified. Only one location may be open at a time.

Closed Location

A location which is not currently being examined and therefore cannot be modified is considered "closed".

New Line State

The program is considered to be at a "new line" if no location is open and either nothing has been typed on a line, or the number on the line has been cancelled.

Examples:

```
(carriage return just typed)
100?
     >cancelled with RUBOUT
1.45?
```

OPERATIONS

Open a Location

In a new line condition, typing an address, followed by '/', opens the location specified. Typing a number will result in that number being placed in the location when it is closed. The number may be cancelled by typing RUBOUT. If nothing is typed, or if the last number typed by the user is then cancelled, the location will not be modified.

Open Next Location

Typing line feed (LF) will close a location if it is open, then open the next location. If the last location open was N.377, the new location opened will be N+1.Ø.

Close Location

If a location is open, it can be closed with a carriage return (CR). CR may also be used at any time in FYLLUK to generate blank lines.

Set Search Limits

The lower or upper limit for searching may be setby typing an address followed by 'L' or 'U', respectively.

Set Search Mask

The mask for searching is set by typing a number, followed by 'M'. 'M' may also be typed when a location is open, in which case the mask is set to the contents of that location, and the location is closed. Initially the mask is set to 7777.

Searching

A search is initiated by typing a number followed by '=' (for matches) or '<' (non-matches). All locations within the limits are tested under the mask, and matches or non-matches are opened. At each success the current address is set to the location where the success occurred. The search may be terminated by striking any keyboard key. The current address will be the last location opened. (If no successes occur, the current address will remain the last location opened before the search.)

Example:

Find all IOT's between 3.147 and 1Ø.377:

```
3.147L
1Ø.377U
7ØØØM
6ØØØ=
```

(Addresses of successful locations and their full contents are printed.)

Type Limits or Mask

Typing 'L', 'U', or 'M' in a new line state will cause the corresponding item to be typed out.

Open Current Location

Typing '*' in a new line state causes the current address to be typed, opened, and its contents typed.

Close a Block

If it should be necessary to cancel all changes to a block in memory, or if it is desired that the block be reread, 'C' is typed while in a new line state. This will inhibit rewriting the block presently in memory and force a read the next time a location is opened.

Return to HELP

When all examination and modification of a device is complete, the user has two means of returning to HELP. If there are more files on the CD input string to be printed, ↑ P (CTRL P) will allow HELP to Proceed. On the other hand, if the user wishes to go back to the Command Decoder to enter a new input string, he should type ↑ R to Restart.

Return to Keyboard Monitor

↑ C will effect a return to the Keyboard Monitor

Note: For ↑ P, ↑ R, ↑ C, the program must be in a new line state. The user may notice activity on the device being examined, as the program may write one block, and will always read one before continuing as specified.

Also, if ALT MODE was used to terminate the CD input string, FYLHLP will return to the Keyboard Monitor when the last request has been typed out (no S option specified), or when the user types ↑ P to exit FYLLUK after the last request (S option specified). If, however, the user types ↑ R, the program will proceed to request another CD input string.

Cancel

RUBOUT cancels the number or address typed. Illegal characters will also cause cancellation. They will be echoed, followed by '?'.

ERROR CONDITIONS

If a device is specified in the CD input string which is not file-structured, a corresponding message is printed. FYLHLP HLT's at 2Ø42 if an error occurs while fetching an I/O handler. Messages are printed if I/O errors occur. If a read error occurs in the FYLLUK section, the block is closed. If a write error occurs, the block on which the error occurred remains in core as if no error occurred.

FYLHLP occupies locations 12ØØØ-14177. Buffer space extends from 142ØØ-14577, and the I/O handler resides in locations Ø46ØØ-Ø4777.

```
.R FYLHLP
*SYS:ABSLDR.SV

#  <———    Marks beginning of request
DIRECTORY ENTRY AT 1.5

    ———————Words as they appear in directory

Ø1Ø2  AB
2314  SL      Name
Ø422  DR
2326  .SV  extension
*1 EXTRA WORD *                  Extra words are followed by '*'
537Ø*   10/31/7Ø      Date
7773  FILE LENGTH = 5 BLOCKS

*               *
 FIRST DATA BLOCK = 7Ø

*SYS:ABSLDR.SV/E

                 ——————> Print "empty" entries
#
DIRECTORY ENTRY AT 1.5

Ø1Ø2  AB
2314  SL
Ø422  DR
2326  .SV
*1 EXTRA WORD  *
537Ø*  10/31/7Ø
7773  FILE LENGTH = 5 BLOCKS

FIRST DATA BLOCK = 7Ø

DIRECTORY ENTRY AT 1.2Ø3

ØØØØ  <EMPTY>
7216  FILE LENGTH = 37Ø BLOCKS              Empty entry print-out

FIRST DATA BLOCK = 557

*SYS:ABSLDR.SV (ENS)

                  ———————— "no list" of ABSLDR. V and "empty entries"; go to FYLLUK
```

```
#
DIRECTORY ENTRY AT 1.5
                                    ABSLDR.SV
FIRST DATA BLOCK = 7Ø
                                                   Compare above
DIRECTORY ENTRY AT 1.2Ø3
                                    <EMPTY>
FIRST DATA BLOCK = 557

>       <--------      FYLLUK entered
1.5/  Ø1Ø2
1.6/  2314
1.7/  Ø422             ABSLDR.SV directory entry
1.1Ø/  2326              examined by FYLLUK
1.11/  537Ø
1.12/   7773

7Ø./  7777
7Ø.1/ 6213
7Ø.2/ 2ØØØ             Loader control information
7Ø.3/ 6ØØ3               examined by FYLLUK
7Ø.4/ 2ØØØ
7Ø.5/ 1Ø1Ø

7Ø.L
7Ø.SU
ØM
Ø=                     Ø mask used to
7Ø.Ø/   7777             print successive locations
7Ø.1/   6213
7Ø.2/   2ØØØ
7Ø.3/   6ØØ3
7Ø.4/   2ØØØ
7Ø.5/   1Ø1Ø
*  <--------           Gives last open location
7Ø.5/   1Ø1Ø
↑ P
*DTA2:FYLHLP.SV,,TTY:/S

#
DIRECTORY ENTRY AT 1.35

Ø631   FY
141Ø   LH
142Ø   LP
2326  .SV
*1  EXTRA WORD  *
2Ø21*  4/2/71
7772    FILE LENGTH  =  6 BLOCKS

FIRST DATA BLOCK = 56
```

```
>
57./    4777
57.1/   ØØ1Ø
57.2/   4776    First few words of FYLHLP program
57.3/   ØØØ5
57.4/   ØØØØ

2ØØ.17L
3ØØ.47U
ØM
Ø<   (Pause, while search proceeds)

Ø.147L
1Ø.377U
7ØØØM
6ØØØ=
1.76/    64ØØ
1.1Ø4/   64ØØ
1.163/   6764
1.167/   6771
1.173/   6772
1.177/   6761
1.2Ø3/   6774
1.2Ø7/   6762
1.213/   6612
1.233/   66Ø5
1.237/   66Ø3
1.243/   6626
1.247/   6621
1.253/   6622
1.267/   6615
1.273/   6616
1.277/   66Ø1
1.3Ø3/   6611
1.313/   6642    ←———    Keyboard key struck here to end search
*
1.313/   6642
177/     6761
377/     72ØØ    ←———    LF typed here
2.Ø/     7777
↑ P

#>   ←————————     Device with no file name causes control to go directly
↑ P                   to FYLLUK, exception

#DEVICE IN REQUEST #3 IS NOT FILE-STRUCTURED
* ↑ C
```