



DECUS

PROGRAM LIBRARY

DECUS NO.	12-95
TITLE	PDP-12 PS/8 UTILITY PROGRAMS
AUTHOR	Charles M. Moore, III
COMPANY	1700 Sunset Blvd. Apt. 26 Houston, Texas 77005
DATE	June 27, 1972
SOURCE LANGUAGE	SABR

PDP-12 PS/8 Utility Programs

Abstract:

This package contains the following utility programs, along with their sources and write-ups:

- 1) MAGSPY displays the contents of any block on any PS/8 device, using any of several modes selected by the sense switches. Included is a mode which displays PS/8 source files as text listings.
- 2) INDEX displays PS/8 device indices, deletes files with rubouts, transfers groups of one or more files from one device to another.
- 3) COPY duplicates entire PS/8 linctapes.
- 4) MARK12 marks linctapes.
- 5) DIALPS copies files from DIAL linctape 1 to PS/8 device SYS, automatically converting core-image file headers.
- 6) PSDIAL copies files from PS/8 to DIAL.

MAGSPY, INDEX, and COPY use SABR routine IHELP to display operating instructions. Other SABR or FORTRAN programs can also use IHELP.

CHARLES M. MOORE III
1700 SUNSET BLVD., APT. 26
HOUSTON, TEXAS 77005

PDP-12 PS/8 UTILITY PROGRAMS.

.R PIP
*SYS:/E

6/23/72

PIP .SV	9	7/1/71
EDIT .SV	9	12/2/70
INDEX .SV	24	6/23/72
FORT .SV	25	7/1/71
MAGSPY.SV	27	5/19/72
LOADER.SV	11	8/24/71
SABR .SV	23	11/7/71
LIB8 .RL	29	12/2/70
LIB12 .RL	25	10/7/71
COPY .SV	10	6/10/72
MARK12.SV	7	10/22/71
ABSLDR.SV	5	10/31/70
PAL8 .SV	14	8/6/71
NAMES .RL	2	8/12/71
INFILE.RL	2	8/12/71
INSYS .RL	2	8/12/71
IOFILE.RL	3	8/13/71
IOSYS .RL	2	8/13/71
PLOT6 .SB	12	9/16/71
PLOT7 .SB	11	9/16/71
XPLAIN.PL	18	10/7/71
HELP .SB	15	9/20/71
DIALPS.FT	10	10/19/71
PSDIAL.FT	3	10/19/71
TESTMS.FT	3	5/19/72
MAGSPY.SB	54	5/19/72
MSPY2 .SB	25	5/19/72
UTILITY.SB	18	5/31/72
COPY .SB	24	6/10/72
INDEX .SB	68	6/23/72
INDEX2.SB	68	6/23/72
WRITE2.UP	9	6/16/72
<EMPTY>	114	
114 FREE	BLOCKS	

*!C

.

PDP-12 PS/8 UTILITY PROGRAMS.

THIS FILE SUMMARIZES THE CONTENTS OF THIS TAPE, WHICH CAN BE CONSIDERED A SUPPLEMENT TO A PREVIOUS FORTRAN LIBRARY TAPE (DECUS 12-48). THE MAIN PURPOSE OF THIS TAPE IS TO PROVIDE THE FOLLOWING PDP-12 PS/8 PROGRAMS, WHICH CAN BE USED WITHOUT ANY FAMILIARITY WITH DECUS 12-48:

- 1) MAGSPY.SV DISPLAYS THE CONTENTS OF ANY BLOCK ON ANY PS/8 DEVICE (DISK, LINCTAPE, DECTAPE), USING ANY OF SEVERAL MODES (POINTS, NUMBERS, CHARACTERS) SELECTED WITH THE SENSE SWITCHES. INCLUDED IS A MODE WHICH DISPLAYS PS/8 SOURCE FILES AS TEXT LISTINGS.
- 2) INDEX.SV DISPLAYS THE FILE INDEX OF ANY FILE-ORIENTED DEVICE, USING A FORMAT SIMILAR TO THAT OF THE PIP /E OPTION. FILES MAY BE DELETED FROM THE INDEX BY TYPING RUBOUTS. GROUPS OF FILES MAY BE COPIED (IN IMAGE MODE) FROM ONE DEVICE TO ANOTHER.
- 3) COPY.SV DUPLICATES ALL OF PS/8 LINCTAPE 0 ONTO LINCTAPE 1, INCLUDING PS/8 SYSTEM BLOCKS (WHICH ARE NOT COPIED BY THE PIP /S OPTION).
- 4) MARK12.SV IS USED TO MARK LINCTAPES. THIS PROGRAM IS DIAL MARK12 ADAPTED TO RUN UNDER PS/8.
- 5) DIALPS.FT IS A FORTRAN PROGRAM WHICH WILL COPY FILES FROM DIAL LINCTAPE 1 ONTO PS/8 DEVICE SYS. CORE IMAGE FILES ARE CONVERTED FROM DIAL INTO PS/8 FORMAT, AND SHOULD BE SELF-STARTING IN 8-MODE SO THAT THEY CAN BE STARTED UNDER PS/8. THIS PROGRAM WAS USED TO COPY MARK12.SV FROM DIAL TO PS/8.
- 6) PSDIAL.FT COPIES FILES FROM PS/8 DEVICE SYS TO DIAL LINCTAPE 1. NO CONVERSION OF CORE IMAGE FILES IS PROVIDED.

MAGSPY.SV, INDEX.SV, AND COPY.SV CONTAIN EXPLANATION DISPLAYS WHICH APPEAR WHEN THE PROGRAMS ARE RUN. SOURCE FILE HELP.SB CONTAINS THE EXPLANATION DISPLAY SUBROUTINE, ALONG WITH INFORMATION ON HOW ANY FORTRAN OR SABR PROGRAM CAN USE IT.

SOURCE FILES ARE PROVIDED FOR ALL THE ABOVE PROGRAMS EXCEPT MARK12. THESE SOURCE FILES REQUIRE SEVERAL OF THE SUBROUTINES IN LIBRARY FILES LIB8.RL AND LIB12.RL. BOTH LIBRARIES ARE SEARCHED AUTOMATICALLY BY THE LOADER.SV ON THIS TAPE. LIB8 IS THE STANDARD PS/8 LIBRARY FILE. LIB12 IS THE PDP12 LIBRARY FILE PROVIDED ON THE DECUS 12-48 TAPE, EXPANDED TO INCLUDE 2 ADDITIONAL BINARY FILES, WHOSE SOURCE FILES ARE PLOT6.SB AND PLOT7.SB. DECUS 12-48 SOURCE FILE XPLAIN.PL, EXPANDED TO INCLUDE EXPLANATIONS OF PLOT6 AND PLOT7, IS ALSO INCLUDED ON THIS TAPE. THE DECUS 12-48 WRITE-UP DESCRIBES THE OTHER ROUTINES IN LIB12.RL.

THE REMAINING CORE IMAGE AND BINARY FILES ON THIS TAPE ARE IDENTICAL TO THOSE ON DECUS 12-48, AND ARE INCLUDED TO COMPLETE THE PS/8 OPERATING SYSTEM, CONSISTING OF FILES PIP.SV TO IOSYS.RL AT THE BEGINNING OF THIS TAPE. ALL OTHER FILES ON THIS OR THE DECUS 12-48 TAPE ARE EXPLANATIONS OR SOURCES.

THE FOLLOWING SEQUENCE OF PROGRAMS MAY BE RUN TO CREATE A PS/8 SYSTEM ON AN UNMARKED LINCTAPE:

- 1) START PS/8 WITH NEW TAPE ON UNIT 1, THIS TAPE WRITE-LOCKED ON UNIT 0.
- 2) R MARK12 TO MARK NEW TAPE WITH 128-WORD BLOCKS (TAKES 2.5 MINUTES)
- 3) R COPY TO DUPLICATE THIS TAPE ONTO NEW TAPE (TAKES 3 MINUTES)
- 4) R INDEX TO DELETE ANY FILES NOT NEEDED ON NEW TAPE, OR TO COPY ANY ADDITIONAL FILES FROM OTHER TAPES OR PS/8 DEVICES.

#L

/MAGSPY.SB, PAGE 1 OF 26.

(5/19/72 - C.M.MOORE, RICE U., HOUSTON)

/THE FOLLOWING EXPLANATION IS DISPLAYED WHEN MAGSPY IS STARTED:

/TYPE H FOR HELP AT ANY TIME DURING BLOCK DISPLAY.

/(THIS EXPLANATION DISPLAY WILL REAPPEAR.)

/TYPE M FOR MORE HELP DURING EXPLANATION.

/MAGSPY CAN DISPLAY ANY BLOCK ON ANY PS/8 I/O DEVICE.

/THE BLOCK IS SPECIFIED IN ONE OF THREE WAYS:

/ 1)DEVICE:FILE (STARTS DISPLAY WITH 1ST BLOCK OF FILE)

/ 2)DEVICE: AND BLOCK NUMBER

/ 3)LINCTAPE UNIT NUMBER AND BLOCK NUMBER

/METHODS 1 & 2 USE 256-WORD PS/8 BLOCKS, EACH OF WHICH CONSISTS OF

/2 128-WORD PHYSICAL DEVICE BLOCKS.

/METHOD 3 USES LINCTAPE BLOCKS OF ANY SIZE (UP TO 1000 OCTAL WORDS).

/TO DISPLAY A PS/8 BLOCK, TYPE D AND WAIT FOR THE COMMAND DECODER TO

/ECHO A STAR (*). AFTER * APPEARS, SPECIFY THE DEVICE OR DEVICE:FILE,

/FOLLOWED BY CARRIAGE-RETURN. DSK IS THE ASSUMED DEVICE NAME. IF A FILE

/IS NOT SPECIFIED, MAGSPY WILL TYPE B* TO ASK FOR AN OCTAL BLOCK NUMBER

/TO DISPLAY A LINCTAPE BLOCK, TYPE L, U, OR B. AFTER * APPEARS, TYPE

/THE APPROPRIATE OCTAL NUMBER, FOLLOWED BY A CARRIAGE-RETURN.

/ TYPE L TO SPECIFY LINCTAPE BLOCK LENGTH.

/ TYPE U TO SPECIFY LINCTAPE UNIT NUMBER.

/ TYPE B TO SPECIFY BLOCK NUMBER.

/TYPE THE FOLLOWING CHARACTERS TO INCREMENT OR DECREMENT THE NUMBER

/OF THE BLOCK CURRENTLY BEING DISPLAYED:

/ 1 INCREMENTS BY 1

/ 2 INCREMENTS BY 8

/ Q DECREMENTS BY 1

/ W DECREMENTS BY 8

/OR TYPE B TO SPECIFY A NEW OCTAL BLOCK NUMBER.

/KNOBS 3 & 7 DETERMINE HOW MUCH OF THE BLOCK WILL BE DISPLAYED.

/ KNOB 3 IS THE 1ST LOCATION DISPLAYED.

/ KNOB 7 IS NUMBER OF LOCATIONS DISPLAYED.

/SENSE SWITCHES 0 - 5 DETERMINE THE DISPLAY MODE:

/ SW0: OFF FOR POINTS, ON FOR OCTAL NUMBERS.

/(POINTS ARE SCALED TO AVOID WRAP-AROUND.)

/(SWITCH TO CHANNEL 1 TO DELETE POINT DISPLAY HEADING.)

/ SW1: OFF FOR UNSIGNED, ON FOR SIGNED NUMBERS & POINTS.

/ SW2: ON TO ADJUST POINT SCALING WITH KNOBS 2 & 6.

/(KNOB 2 IS MIN, KNOB 6 IS MAX.)

/ SW2: ON FOR FULL-SIZED CHARACTERS.

/ SW3: ON TO CONNECT POINTS WITH STRAIGHT LINES.

/(THIS WILL OCCUR ONLY IF YOU HAVE EAE HARDWARE.)

/ SW4: ON FOR 6-BIT ASCII (2 CHARACTERS PER WORD).

/ SW5: ON FOR 8-BIT ASCII (3 CHARACTERS PER 2 WORDS).

/(USE THIS TO LOOK AT PS/8 SOURCE FILES.)

SUMMARY:

/THE FOLLOWING INPUTS ARE ACCEPTED DURING A BLOCK DISPLAY:

/ H D L U B 1 2 Q W

/THE FOLLOWING INPUTS ARE ACCEPTED DURING THIS EXPLANATION DISPLAY:

/ TYPE D OR L TO SPECIFY A DEVICE:FILE OR LINCTAPE.

/ TYPE B TO SPECIFY A NEW BLOCK.

/ TYPE CARRIAGE-RETURN TO RETURN TO THE SAME BLOCK.

/ TYPE M TO REPEAT EXPLANATION.

/DISPLAYS INDICES OF PDP-12 PS/8 DEVICES.
/DELETES FILES FROM ANY DEVICE.
/PIPS GROUPS OF FILES FROM ONE DEVICE TO ANOTHER.
/
/THE FOLLOWING EXPLANATION IS DISPLAYED WHEN INDEX IS STARTED:

/TYPE H FOR HELP AT ANY TIME DURING INDEX DISPLAY.
/(THIS EXPLANATION DISPLAY WILL REAPPEAR.)
/TYPE M FOR MORE HELP DURING EXPLANATION.
/TYPE M NOW!

/INDEX DISPLAYS THE FILE INDEX OF ANY PS/8 DEVICE.
/THE DEVICE IS SPECIFIED BY TYPING D AND WAITING
/FOR THE COMMAND DECODER TO ECHO A STAR (*).
/WHEN * APPEARS, TYPE IN THE DEVICE NAME, FOLLOWED
/BY A COLON AND A CARRIAGE-RETURN. (IF THE : IS
/NOT TYPED, DSK: WILL BE THE DEVICE SELECTED.)

/WHEN THE DEVICE INDEX APPEARS ON THE SCOPE, YOU MAY
/TYPE THE FOLLOWING CHARACTERS TO INCREMENT OR
/DECREMENT THE FILE NAMES BEING DISPLAYED:

/ 1 INCREMENTS BY 1 FILE
/ 2 INCREMENTS BY 1 FRAME
/ Q DECREMENTS BY 1 FILE
/ W DECREMENTS BY 1 FRAME
/ANALOG KNOB 7 DETERMINES THE FRAME SIZE.
/(I.E., THE NUMBER OF FILES DISPLAYED.)

/TYPE RUBOUT TO DELETE THE LAST FILE CURRENTLY
/BEING DISPLAYED ON THE SCOPE.
/TYPE : TO MAKE PERMANENT ALL FILE DELETIONS MADE
/SINCE THE LAST D, P, R, OR : WAS TYPED.
/TYPE R TO RESTORE ALL DELETIONS NOT YET PERMANENT.
/IF CTRL/C, D, OR P IS TYPED BEFORE MAKING THE DELETIONS
/PERMANENT, THEY ARE AUTOMATICALLY RESTORED.

/TYPE P (AND WAIT FOR * TO APPEAR) TO PIP ONTO AN
/OUTPUT DEVICE ALL FILES CURRENTLY BEING DISPLAYED.
/WHEN * APPEARS, SPECIFY THE OUTPUT DEVICE.
/FILES ARE TRANSFERRED IN IMAGE MODE (LIKE PIP /I).
/FILE DELETIONS ON THE DISPLAYED DEVICE WHICH HAVE NOT
/BEEN MADE PERMANENT WILL BE RESTORED WHEN P IS TYPED.
/THUS RUBOUTS CAN BE USED TO TEMPORARILY DELETE THOSE
/FILES WHICH ARE NOT TO BE TRANSFERRED.
/ANY OUTPUT DEVICE FILE WITH THE SAME NAME AS ONE OF
/THE DISPLAYED FILES WILL BE REPLACED BY THAT FILE.
/ALL OTHER OUTPUT DEVICE FILES WILL REMAIN UNCHANGED.
/THE DISPLAYED INDEX WILL REAPPEAR AFTER THE FILES
/HAVE BEEN PIPPED.

/TYPE CARRIAGE-RETURN TO RETURN TO THE
/CURRENT DEVICE INDEX DISPLAY.
/TYPE D (AND WAIT FOR * TO APPEAR) TO
/SPECIFY A NEW DEVICE INDEX DISPLAY.
/TYPE CTRL/C TO EXIT TO THE KEYBOARD MONITOR.
/TYPE M TO REPEAT EXPLANATION.

```

#L
/COPY.SB, PAGE 1 OF 9. (6/10/72 - C.M.MOORE, RICE U., HOUSTON)
/
/COPIES LINCTAPE 0 ONTO LINCTAPE 1.
/
/TAPES ARE ASSUMED TO BE PS/8 LINCTAPES.
/(3000 OCTAL BLOCKS OF LENGTH 128 OR 129 WORDS.)
/
/THE ENTIRE TAPE IS DUPLICATED IN 3 MINUTES, INCLUDING THE PS/8
/SYSTEM BLOCKS, WHICH ARE NOT COPIED WHEN YOU USE THE PIP /S OPTION.
/ALSO, ALL TAPE READS & WRITES ARE CHECKED. (PIP DOESN'T CHECK WRITES.)
/
/TYPE RETURN TO INITIATE COPYING AFTER EXPLANATION DISPLAY APPEARS.
/TYPE CTRL/C TO RETURN TO PS/8 KEYBOARD MONITOR.
/YOU MAY LEAVE DEVICE SYS WRITE-LOCKED WHILE RUNNING COPY.
/(TO PERMIT THIS, COPY USES THE MODIFIED VERSION OF UTILITY.SB ON THIS
/TAPE, RATHER THAN THE LIB8 LIBRARY VERSION.)
/
/TOTAL BUFFER SPACE HOLDS 53 OCTAL 128-WORD OR 129-WORD BLOCKS.
IBUFF, COMMN 7200 /36 OCTAL BLOCKS IN FIELD 1, STARTING AT LOC 0000.
/15 OCTAL BLOCKS IN FIELD 0, STARTING AT LOC 4200.

```

```

OPDEF TADI 1400
OPDEF DCAI 3400
ABSYM DCONT 100
ABSYM ICONT 101
ABSYM JCONT 102
ABSYM KCONT 103
ABSYM NBLK1 104
ABSYM NBLK0 105
ABSYM LOC 106
ABSYM ERRNUM 107
/

```

```

ENTRY MAIN /START OF PROGRAM
MAIN, NOP
CALL 0,OPEN
/ADJUST JOB STATUS WORD TO PREVENT RESTARTING COPY.
6201 /CDF 0
TADI (7746
AND (6777
TAD (1000
DCAI (7746
HELP, CALL 1,IHELP /CALL HELP DISPLAY ROUTINE
ARG (1
CLA /IGNORE INDEX RETURNED BY IHELP
/LOOP TO COPY TAPE 0 TO TAPE 1
TAD (-44 /- # OF BUFFER LOADS REQUIRED
DCA ICONT
TAD ICONT
DCA JCONT
TAD (-36 /- # OF BLOCKS STORED IN FIELD 1 BUFFER
DCA NBLK1
TAD (-15 /- # OF BLOCKS STORED IN FIELD 0 BUFFER
DCA NBLK0
DCA IBLK0 /TAPE 0 BLOCK # FOR FIELD 0 BUFFER
DCA JBLK0 /TAPE 1 BLOCK # FOR FIELD 0 BUFFER
/CHECK FOR FINAL BUFFER LOAD
CPLOOP, ISZ JCONT
JMP RDTAPE
CLA CMA
DCA NBLK0 /ON THE FINAL LOAD, ONLY 1 BLOCK IS STORED IN FIELD 0

```


COPIES FILES FROM DIAL LINCTAPE 1 TO PS/8 DEVICE SYS:

DIAL FILES ARE SPECIFIED BY A STARTING BLOCK NUMBER AND LENGTH IN BLOCKS, AVAILAble FROM THE DIAL INDEX.

PS/8 FILES ARE SPECIFIED BY A FILE NAME AND EXTENSION.

HEADER BLOCKS OF CORE IMAGE FILES (I.E., DIAL BINARY TO PS/8 .SV FILES) ARE AUTOMATICALLY CONVERTED FROM DIAL TO PS/8 FORMAT. THESE DIAL BINARY FILES MUST BE SELF-STARTING IN 8-MODE, IN ORDER TO START CORRECTLY UNDER THE PS/8 SYSTEM.

ALL OTHER TYPE FILES (E.G., DATA OR SOURCE FILES) ARE COPIED WITHOUT ALTERATION.

```
COMMON NSEGS, ICDIF, IADDR, JOBW, KNTRL,
1 IW1, IW2, IW3, IW4, KDUMY, NWDS, IWD0, IWD1, IBUFF
DIMENSION KNTRL(2, 126), KDUMY(219), IWD0(16), IWD1(16)
DIMENSION IBUFF(256), NAME(4), IDEV(2)
```

```
C
100 CALL CRLF
C READ NAME OF NEW FILE TO BE CREATED ON SYS:
110 WRITE(1, 120)
120 FORMAT('WHEN * APPEARS, TYPE NAME OF NEW PS/8 FILE')
CALL NAMES(IDEV, NAME, IDEV, NAME)
IF(NAME) 130, 110, 130
C READ OCTAL STARTING BLOCK NUMBER OF FILE TO COPY FROM LINCTAPE 1
130 WRITE(1, 140)
140 FORMAT('NOW SELECT DIAL FILE:')
CALL ALPHA('START')
CALL ALPHA('ING B')
CALL ALPHA('LOCK ')
CALL ALPHA('=')
IELK1=INTIN(8)
C READ OCTAL LENGTH OF FILE ON LINCTAPE 1.
CALL ALPHA('LENGT')
CALL ALPHA('H IN ')
CALL ALPHA('ELOCK')
CALL ALPHA('S =')
NELKS=INTIN(8)
C MAKE A NEW TENTATIVE FILE ON SYS AND CHECK AVAILAble SPACE
NB1=MSYS(NAME, LMAX)
IF(LMAX-NELKS) 150, 190, 190
C FILE WON'T FIT ON SYS
150 WRITE(1, 160)
160 FORMAT('FILE WILL NOT FIT ON SYS:')
GO TO 100
C CHECK FOR .SV FILE
190 IF(NAME(4)-1238) 195, 300, 195
C COPY FILE BLOCKS
195 DO 200 I=1, NELKS
CALL FLINC(1, IELK1, 1, IBUFF, 256)
CALL WSYS(2, NB1, IBUFF)
NB1=NB1+1
200 IELK1=IELK1+1
C CLOSE (MAKE PERMANENT) NEW FILE ON SYS
220 CALL CSYS(NELKS)
GO TO 100
C
```

```

#L
C      DIALPS.FT, PAGE 2 OF 3.
C
C      CONVERT .SV FILE HEADER BLOCK FROM DIAL TO PS/8 FORMAT
300    CALL RLINC(1, IELK1, 1, IW1, 256)
C      CHECK FOR STARTING ADDRESS
      IF(IW1)304, 302, 304
302    WRITE(1, 303)
303    FORMAT('NO STARTING ADDRESS FOR .SV FILE')
      GO TO 308
C      CHECK THAT DIAL .SV FILE STARTS IN 8-MODE.
304    IF(IW1-2)306, 310, 306
306    WRITE(1, 307)
307    FORMAT('.SV FILE MUST START IN 8-MODE')
308    ICDIF=0
      IADDR=0
      GO TO 312
C      CONVERT HEADER POINTERS
310    ICDIF=IW2+1
      IADDR=IW4
312    JOEW=512
      NSEGS=-NWDS
      DO 320 I=1, 126
      DO 320 J=1, 2
320    KNTRL(J, I)=0
C      CONVERT FIELD 1 CORE IMAGE POINTERS
      IPS8=1
      DO 400 I=1, 15
      IF(IWD1(I))340, 400, 340
340    KNTRL(1, IPS8)=(I-1)*256
      KNTRL(2, IPS8)=136
      IPS8=IPS8+1
400    CONTINUE
      IF(IWD1(16))420, 440, 420
420    KNTRL(1, IPS8)=15*256
      KNTRL(2, IPS8)=72
      IPS8=IPS8+1
440    IFLD1=IPS8-1
C      CONVERT FIELD 0 CORE IMAGE POINTERS
      DO 500 I=1, 15
      IF(IWD0(I))460, 500, 460
460    KNTRL(1, IPS8)=(I-1)*256
      KNTRL(2, IPS8)=128
      IPS8=IPS8+1
500    CONTINUE
      IF(IWD0(16))520, 600, 520
520    KNTRL(1, IPS8)=15*256
      KNTRL(2, IPS8)=64
      IPS8=IPS8+1
600    IFLD0=IPS8-1-IFLD1
      IF(IPS8-1-NWDS)620, 700, 620
620    WRITE(1, 630)
630    FORMAT('BAD .SV FILE HEADER BLOCK')
      GO TO 100
C      WRITE CONVERTED HEADER BLOCK
700    CALL WSYS(2, NB1, NSEGS)
C

```

#Y

AL

C

DIALPS.FT, PAGE 3 OF 3.

C

C

COPY FIELD 1 CORE IMAGE BLOCKS

IF(IFLD1)1000,1000,800

800

DO 900 I=1,IFLD1

JELK1=IELK1+IFLD0+I

JB1=NB1+I

CALL RLINC(1,JELK1,1,IBUFF,256)

900

CALL WSYS(2,JB1,IBUFF)

C

COPY FIELD 0 CORE IMAGE BLOCKS

1000

IF(IFLD0)220,220,1100

1100

DO 1200 I=1,IFLD0

JELK1=IELK1+I

JB1=NB1+IFLD1+I

CALL RLINC(1,JELK1,1,IBUFF,256)

1200

CALL WSYS(2,JB1,IBUFF)

GO TO 220

END

```

C      PSDIAL.FT, PAGE 1 OF 1. (10/19/71 - C.M.MOORE, RICE U., HOUSTON)
C
C      COPIES FILES FROM PS/8 DEVICE SYS TO DIAL LINCTAPE 1.
C
C      DIMENSION IBUFF(256),NAME(4),IDEV(2)
C
100    CALL CRLF
C      READ NAME OF FILE TO COPY FROM SYS:
      WRITE(1,110)
110    FORMAT('WHEN * APPEARS, TYPE NAME OF PS/8 FILE')
      CALL NAMES(IDEV,NAME,IDEV,NAME)
C      LOOK UP FILE TO GET STARTING BLOCK NUMBER AND LENGTH
      NB1=LSYS(NAME,NELKS)
C      CHECK IF FILE EXISTS ON SYS:
      IF(NB1)120,120,150
120    WRITE(1,130)
130    FORMAT('FILE DOES NOT EXIST ON SYS')
      GO TO 100
C      READ STARTING OCTAL BLOCK NUMBER FOR FILE ON DIAL LINCTAPE 1
C      (A FILE AREA OF APPROPRIATE LENGTH SHOULD ALREADY EXIST)
150    WRITE(1,160)
160    FORMAT('NOW SELECT DIAL FILE:')
      CALL ALPHA('START')
      CALL ALPHA('ING B')
      CALL ALPHA('LOCK ')
      CALL ALPHA('=')
      IELK1=INTIN(8)
C      COPY FILE BLOCKS
      DO 200 I=1,NELKS
      CALL RSYS(2,NB1,IBUFF)
      CALL WLINC(1,IELK1,1,IBUFF,256)
      NB1=NB1+1
200    IELK1=IELK1+1
      GO TO 100
      END

```

```

#L
C      TESTMS.FT, PAGE 1 OF 1. (5/19/72 - C.M.MOORE, RICE U., HOUSTON)
C
C      CREATES DATA FILE DUMMY.DA WHICH CAN THEN BE VIEWED BY
C      MAGSPY TO DEMONSTRATE THE POINT AND OCTAL NUMBER DISPLAY MODES.
C
C      USES SUBROUTINES CONTAINED IN LIBRARY FILE LIB12.RL
C      SEE WRITE-UP OF DECUS 12-48 FOR THEIR EXPLANATION.
C
C      DIMENSION XFILE(2),IDAT(256)
C
C      XFILE='DUMMY '
C      XFILE(2)='DA'
C
C      NB=MSYS(XFILE,LMAX)
C      IF(NB)100,100,140
C      SYS NOT FOUND???
100    CALL EXIT
C      CHECK SPACE AVAILABLE ON SYS
140    IF(LMAX-2)160,200,200
C      NO ROOM ON SYS
160    CALL EXIT
C      CREATE 2 256-WORD DATA BLOCKS ON SYS
C      1ST BLOCK CONTAINS INTEGERS FROM 0017 TO 7777
200    DO 300 I=1,256
300    IDAT(I)=I*16-1
C      CALL WSYS(2,NB,IDAT)
C      2ND BLOCK CONTAINS SINE WAVES
C      DO 400 I=1,256
C      X=I-1
C      X=1600.0*SIN(X*6.2831853/255.0)+400.0*SIN(X*6.2831853/12.75)
400    IDAT(I)=X
C      NB=NB+1
C      CALL WSYS(2,NB,IDAT)
C      CALL CSYS(2)
C      END

```

#L

/HELP.SB, PAGE 1

(9/20/71 - C.M.MOORE, RICE U., HOUSTON)

/PDP-12 PS/8 FORTRAN/SABR HELP DISPLAY ROUTINE

/FORTRAN INTEGER FUNCTION "IHELP(IFLAG)"

/DISPLAYS TEXTUAL INFORMATION ON SCOPE.

/THE TEXT IS SEGMENTED INTO FRAMES.

/THE FRAMES ARE DISPLAYED ONE AT A TIME.

/THE TEXT IS ADVANCED 1 FRAME EACH TIME M IS TYPED.

/IHELP IS CALLED AS AN INTEGER FUNCTION IN FORTRAN.

/IT RETURNS THE INDEX VALUE OF THE CHARACTER TYPED

/TO TERMINATE THE HELP DISPLAY.

/THE INDEX STARTS AT 1, AND CAN BE USED IN A

/COMPUTED GO TO STATEMENT.

/CHARACTER #1 IS ALWAYS CARRIAGE-RETURN, WHICH IS

/IGNORED UNLESS ARGUMENT IFLAG IS NONZERO.

/THE REMAINING CHARACTERS ARE USER DEFINED.

/CHARACTERS NOT DEFINED BY THE USER AS VALID

/TERMINATION INPUTS ARE IGNORED BY IHELP.

/USES ROUTINES KEYBD AND DISP9.

/THESE ARE CONTAINED IN LIBRARY FILE LIB12.RL.

/*****WARNING*****/

/YOU MAY FIND THAT ONE OF THE TEXT LINES DISPLAYS

/IN THE WRONG POSITION, SUCH AS ON TOP OF ANOTHER

/LINE. THIS CAN OCCUR IF ITS Y-COORD OCCUPIES THE

/LINC-MODE INSTRUCTION FIELD LOC 1 USED BY DISP9.

/THIS CAN BE AVOIDED BY ALTERING THE RELATIVE CORE

/LOCATIONS OF IHELP AND DISP9. (E.G., LOAD IHELP

/FIRST INTO FIELD 1, CALL LIB12 TO LOAD DISP9 INTO

/FIELD 0, THEN LOAD YOUR REMAINING BINARY FILES.)

/IT CAN ALSO BE AVOIDED BY INSERTING 2 DUMMY LOCS

/IN FRONT OF THE X-Y COORDS OF THE OUT-OF-PLACE LINE.

#Y

THIS FILE EXPLAINS THE FOLLOWING PDP-12 SCOPE DISPLAY ROUTINES. ROUTINES DISP2 & DISP4 REQUIRE THE USE OF THE EAE (EXTENDED ARITHMETIC ELEMENT) HARDWARE OPTION.

ALL 7 BINARY FILES ARE IN LIBRARY FILE LIB12.RL (10/7/71 VERSION).

SOURCE FILE NAME	BINARY FILE NAME	CORE PAGES REQUIRED	ENTRY POINTS CONTAINED IN FILE.
PLOT1.SB	PLOT1.RL	1	DISP1(N,IXY) DISP3(N,IY) DISP5(LY) DISP6(LX) KEYBD(0)
PLOT2.SB	PLOT2.RL	1	DISP2(N,IXY) (REQUIRES EAE)
PLOT3.SB	PLOT3.RL	1	DISP4(N,IY) (REQUIRES EAE)
PLOT4.SB	PLOT4.RL	1	DISP7(LX,LY,N) DISP8(LX,LY,STRNG)
PLOT5.SB	PLOT5.RL	2	DISP7(LX,LY,N) DISP9(LX,LY,STRNG)
PLOT6.SB	PLOT6.RL	2	DSP10(LX,LY,MODE,N) DSP11(LX,LY,LENTH,STRNG)
PLOT7.SB	PLOT7.RL	2	DSP12(LY,LENTH,STRNG)

M=KEYBD(0) SETS M = VALUE OF CHARACTER STRUCK AT KEYBOARD. IF NO CHARACTER HAS BEEN STRUCK, SETS M=0. (THIS ROUTINE CAN BE USED TO TERMINATE DISPLAY LOOPS.)

CALL DISP1(N,IXY) DISPLAYS ON THE SCOPE THE N POINTS STORED IN VECTOR IXY(2,N). POINTS ARE STORED AS X,Y COORDINATE PAIRS.

CALL DISP2(N,IXY) DISPLAYS STRAIGHT LINE SEGMENTS BETWEEN ADJACENT POINTS STORED IN VECTOR IXY(2,N).

CALL DISP3(N,IY) DISPLAYS THE N VERTICAL VALUES STORED IN VECTOR IY(N). POINTS ARE SPACED EVENLY ACROSS THE SCOPE.

CALL DISP4(N,IY) DISPLAYS STRAIGHT LINE SEGMENTS BETWEEN ADJACENT VERTICAL VALUES STORED IN VECTOR IY(N). THE SEGMENTS ARE SPACED EVENLY ACROSS THE SCOPE.

CALL DISP5(LY) DISPLAYS A HORIZONTAL LINE ACROSS THE SCOPE AT THE VERTICAL POSITION SPECIFIED BY LY.

CALL DISP6(LX) DISPLAYS A VERTICAL LINE DOWN THE SCOPE AT THE HORIZONTAL POSITION SPECIFIED BY LX.

CALL DISP7(LX,LY,N) DISPLAYS N AS A SIGNED 4-DIGIT OCTAL NUMBER. LY AND THE ABSOLUTE VALUE OF LX ARE THE VERTICAL AND HORIZONTAL COORDINATES OF THE LOWER LEFT HAND EDGE OF THE NUMBER DISPLAY. IF LX IS NEGATIVE, FULL-SIZED CHARACTERS ARE DISPLAYED. OTHERWISE, HALF-SIZED CHARACTERS ARE USED.

CALL DISP8(LX,LY,STRNG) DISPLAYS A STRING OF CHARACTERS, THE DISPLAY PATTERNS FOR WHICH ARE STORED IN STRNG. EACH CHARACTER REQUIRES A 24-BIT, 2-WORD PATTERN. STRNG MUST BE TERMINATED WITH A WORD CONTAINING -1 (7777 OCTAL), WHICH WILL NOT BE DISPLAYED. LX AND LY ARE THE SAME AS THEY WERE FOR DISP7.

CALL DISP9(LX,LY,STRNG) DISPLAYS A STRING OF CHARACTERS WHICH ARE STORED IN STRNG AS 6-BIT STRIPPED ASCII CHARACTERS. (I.E., FORTRAN HOLLERITH OR A-FORMAT CHARACTERS.) STRNG MUST BE TERMINATED WITH A ZERO-VALUED CHARACTER, WHICH WILL NOT BE DISPLAYED. LX AND LY ARE THE SAME AS THEY WERE FOR DISP7.

CALL DSP10(LX,LY,MODE,N) IF MODE=0, DISPLAYS N AS A SIGNED 4-DIGIT OCTAL NUMBER. IF MODE IS NONZERO, DISPLAYS N AS AN UNSIGNED 4-DIGIT OCTAL NUMBER. LX AND LY ARE THE SAME AS THEY WERE FOR DISP7.

CALL DSP11(LX,LY,LENTH,STRNG) DISPLAYS A STRING OF CHARACTERS WHICH ARE STORED IN STRNG AS 6-BIT ASCII CHARACTERS. LENTH IS THE LENGTH OF STRNG IN WORDS. (2*LENTH CHARACTERS WILL BE DISPLAYED.) ZERO-VALUED CHARACTERS ARE DISPLAYED AS AMPERSANDS (@). LX AND LY ARE THE SAME AS THEY WERE FOR DISP7.

CALL DSP12(LY,LENTH,STRNG) DISPLAYS 8-BIT ASCII TEXT STORED IN STRNG. 3 CHARACTERS ARE PACKED INTO EVERY 2 WORDS, AS EXPLAINED IN APPENDIX A OF THE PS/8 SOFTWARE SUPPORT MANUAL. THIS IS HOW PS/8 SOURCE FILES ARE STORED. THE DISPLAY IS FORMATTED LIKE AN EDITOR TEXT LISTING. (DSP12 DISPLAYS CHARACTERS 240-377, RECOGNIZES TAB AND CARRIAGE-RETURN, AND IGNORES ALL OTHER CHARACTERS.) TEXT LINES CONTAINING MORE THAN 51 CHARACTERS ARE DISPLAYED AS 2 LINES ON THE SCOPE TO AVOID WRAP-AROUND. ONLY HALF-SIZED CHARACTERS ARE USED. EACH LINE OF TEXT IS DISPLAYED STARTING AT THE LEFT-HAND EDGE OF THE SCOPE. THE FIRST LINE IS DISPLAYED AT VERTICAL POSITION LY. LENTH IS THE LENGTH OF STRNG IN WORDS. IF THE BOTTOM OF THE SCOPE IS REACHED BEFORE LENTH WORDS ARE DISPLAYED, THE REMAINDER OF STRNG IS IGNORED.

DISP1, DISP2, DISP3, DISP4, DSP12 DISPLAY ON SCOPE CHANNEL 1.
DISP5, DISP6, DISP7, DISP8, DISP9, DSP10, DSP11 DISPLAY ON SCOPE CHANNEL 2.

ARGUMENTS IXY AND IY ARE INTEGER ARRAYS.
ARGUMENTS N, MODE, LENGTH, LY, AND LX ARE INTEGER SCALARS.
ARGUMENT STRING IS A STRING OF EITHER 24-BIT DISPLAY PATTERNS (WHEN USED WITH DISP8), 6-BIT ASCII (WHEN USED WITH DISP9 OR DSP11), OR 8-BIT ASCII (WHEN USED WITH DSP12).

THE ARGUMENTS ARE NOT ALTERED BY ANY OF THE DISPLAY ROUTINES.

EACH FULL-SIZED CHARACTER IS DISPLAYED ON THE SCOPE USING AN ARRAY THAT IS 18 PLOT POINTS WIDE BY 32 PLOT POINTS HIGH, INCLUDING SPACING BETWEEN ADJACENT LINES AND CHARACTERS. EACH HALF-SIZED CHARACTER IS DISPLAYED USING AN ARRAY THAT IS 10 WIDE BY 16 HIGH.

THESE ROUTINES DO NOT SCALE THE INPUT VECTOR ELEMENTS.
THE HORIZONTAL SCOPE DISPLAY COORDINATES RUN FROM 0 TO 511 (DECIMAL).
THE VERTICAL COORDINATES RUN FROM -255 TO +255 (DECIMAL).
COORDINATES OUTSIDE THESE RANGES WILL "WRAP AROUND" WHEN DISPLAYED.

EACH CALL TO ONE OF THESE ROUTINES WILL DISPLAY THE POINTS ONCE.
THEREFORE, THE ROUTINES MUST BE PLACED IN PROGRAM LOOPS TO MAINTAIN THE DISPLAYS. THE FOLLOWING IS A SIMPLE EXAMPLE OF THIS:

```
C      PLOT LOOP
100    CALL DISP1(N,IXY)
C      EXIT THE LOOP WHEN ANY KEY IS STRUCK ON THE TELETYPE
      IF(KEYBD(0))200,100,200
200    CONTINUE
```

CHARACTER STRING ARGUMENTS FOR DISP9 MAY BE GENERATED IN FORTRAN THROUGH THE USE OF HOLLERITH CONSTANTS. THE TERMINATING ZERO-VALUED CHARACTER IS AUTOMATICALLY GENERATED IF THE CONSTANT IS LESS THAN 6 CHARACTERS IN LENGTH. FOR EXAMPLE, THE FOLLOWING PROGRAM WILL DISPLAY THE WORD HELLO IN FULL-SIZED CHARACTERS:

```
100    CALL DISP9(-1,220,'HELLO')
      IF(KEYBD(0))200,100,200
200    CONTINUE
```

STRINGS OVER 5 CHARACTERS IN LENGTH MUST BE INSERTED INTO VECTORS BEFORE CALLING DISP9. THE LAST HOLLERITH CONSTANT INSERTED SHOULD CONTAIN 5 OR FEWER CHARACTERS TO ENSURE A TERMINATING ZERO-VALUED CHARACTER FOR THE STRING. THE FOLLOWING EXAMPLE WILL DISPLAY THE WORDS 'THIS IS A STRING' IN HALF-SIZED CHARACTERS:

```
      DIMENSION X(3)
      X='THIS I'
      X(2)='S A ST'
      X(3)='RING'
100    CALL DISP9(0,220,X)
      IF(KEYBD(0))200,100,200
200    CONTINUE
```

FORTRAN PROGRAM PLDEMO.FT IS A DEMO PROGRAM FOR SOME OF THESE ROUTINES. IT CALLS FORTRAN SUBROUTINE PLOT, WHICH SETS UP THE DISPLAY. IT REQUIRES BINARY FILE PLOT.PL, PLUS SEVERAL ROUTINES IN LIBRARY FILE LIB12.PL.
IT IS SELF-EXPLANATORY WHEN RUN.

