SPLINT

by

Alex Petruncola

PURPOSE

The purpose of SPLINT is to produce a magnetic tape in IBM 37Ø format containg parameters for the Fortran plotting routines from parameter lists in PDP-11 format. This is accomplished through a series of entry points - one for each subroutine, one for initialization, and one for closing the file.

Each record shall contain one subroutine call in the format shown on Pages 9 and 10. Each file will contain all of the information needed for one plot. Special error handling features have been incorporated in case of an underflow or overflow in the floating point conversion from PDP-11 IBM 37Ø formats.

NOTE: Entry points also exist for using certain common blocks in the subroutines.

INITIAL CALL

This entry point is used to initialize the tape unit number and error processing flags. The format is:

```
JSR        R5, SPLINT
```

```
        .WORD      KEY, FILE, SCALE, MAXB

        .BYTE      HALT, SUPPRESS, UNIT, DUMMY
```

WHERE: SPLINT =   Starting address of SPLINT

        KEY = $\emptyset$ (for initialization)

      FILE =   Initial tape file number

    SCALE = $\emptyset$ Large arrays are in PDP-11 floating point

          format.

          $\neq \emptyset$ Scale is a scaling factor for the arrays,

          which are in PDP-11 integer format.

     MAXB =   Maximum number of BYTES in buffer and tape record

          (should be a multiple of 4)

     HALT = $\emptyset$ Continue if over or under-flow occurs in

          PDP-11 to IBM 37$\emptyset$ conversion.

        =   1 Halt after record containing the errors is

          written.

 SUPPRESS = $\emptyset$ Print an error message if over-or-under flow

          occurs.

        =1 Suppress error message.

     UNIT =   Tape unit number.

   DUMMY =   Dummy variable

PLOT ROUTINE

ENTRY POINTS

A different entry point exists for each of the 18 plotting

routines. The entry point is specified by one of the arguments in the call.
The format is:

    JSR   R5, SPLINT

    .WORD   Key   {   Key is a numerical value, not an address.

    .WORD

      :   Contain addresses of subroutine arguments

    .WORD   or common block elements.

Key is an integer which corresponds to the value of key on Page 8
for each of the subroutines.

    KEY = 0   is used for initiauzation

        = 19   is used by Splint. A call with KEY = 19 is a

               dummy call and does nothing

        = 20   is used to finish a file

The addresses of the arguments must occur in the order shown in
the fortran calling sequence for the subroutine. The arguments are assumed
to have the following properties:

(1)   Integer values are single word

(2)   Floating point values are normalized

(3)   All character strings are contained in a buffer whose first
      word contains a byte count

(4)   The string for the variable DD is 12 Bytes long - no byte
      count is needed

(5)   Subroutine VLABEL -variable FMT will be a 12 character string,
      no byte count.

NOTE: Some of the subroutines have an argument for the length of a character string, however the byte count will be used to determine the number of characters to be transmitted. In order to avoid confusion, use the byte count as that argument or at least make certain that the two agree.

EXAMPLES

(1) To call SYSPLT (X,Y,IPN) :

```
        JSR     R5, SPLINT      ; Key from Page 9 for subroutine SYSPLT
        .WORD   4.
        .WORD   X
        .WORD   Y
        .WORD   IPN

          .
          .
          .

X:      .WORD   VALUE           ;FLOATING PT
Y:      .WORD   VALUE           ;FLOATING PT
IPN:    .WORD   VALUE           ;INTEGER
```

(2) To call SYSSYM (X,Y,SIZE, BCD,N,THETA)

```
        JSR     R5, Splint
        .WORD   5.              ;Key from Page 9 for subroutine SYSSYM
        .WORD   X
        .WORD   Y
        .WORD   SIZE
```

```
        .WORD    BCD

        .WORD    N

        .WORD    THETA

           .
           .
           .

X:      FLOATING POINT VALUE

Y:      FLOATING POINT VALUE

SIZE:   FLOATING POINT VALUE

BCD:    BYTE COUNT

        CHARACTERS

N:      INTEGER VALUE

THETA:  FLOATING POINT VALUE
```

NOTE: N and Byte count should agree. To insure this we can use the following calling sequence:

```
        JSR      R5, Splint

        .WORD    5

        .WORD    X

        .WORD    Y

        .WORD    SIZE

        .WORD    BCD

        .WORD    BCD        ;USES BYTE COUNT AS VALUE OF N

        .WORD    THETA
```

(3) To use common block LBLCOM of subroutine LABEL

COMMON/ LBLCOM/ ITEST, SLBL, STTL, SSCL.

```
        JSR     R5, Splint
        .WORD   30.                 ;KEY from Page 10 for common block LBLCOM
        .WORD   ITEST
        .WORD   SLBL
        .WORD   STTL
        .WORD   SSCL

          .
          .
          .

ITEST:  .WORD   VALUE               ;INTEGER
SLBL:   .WORD   VALUE               ;FLOATING POINT
STTL:   .WORD   VALUE               ;FLOATING POINT
SSCL:   .WORD   VALUE               ;FLOATING POINT
```

ERRORS

OVERFLOW AND UNDERFLOW

Over-and under-flows can result in conversion from PDP-11 to IBM 370 floating point formats. Unless the suppress flag is set, the following message will be printed:

OVER OR UNDERFLOW FILE XXXXXX RECORD XXXXXX POSITION XXXXXX

In addition, the V-Flag will be set upon return from Splint. If

the halt flag is set, the programme will halt after the record is written. To proceed, press continue.

Underflows result in zero.

Overflows result in the largest positive or smallest negative number available in 37Ø format.

ERRORS OR MAGNETIC TAPE

The system routine GET/PUT is used for writing magnetic tapes. An uncorrectable parity error will result in the message:

FILE XXXXX   RECORD XXXXX

RETRY ATTEMPTS UNSUCCESSFUL

If a carriage return is typed, execution will continue. Any other character will cause the programme to halt and you will not be able to continue.

CLOSING THE FILE

To close a file, a call must be made as follows:

JSR   R5, Splint

.WORD 2Ø.

This causes an EOF to be written on the tape and also clears the file number, record number, and unit number flags.

A call must be made to the initialization routine in order to write a new file.

CONTINUATION BLOCKS

Subroutines without variable length arrays will be 56_{10} Bytes long.

The other subroutines will contain records with as many Bytes as are needed up to the maximum defined by the user. If additional room is needed, a record will be written with KEY=19 indicating a continuation of the previous record. The programme will produce as many continuation records as are needed.

KEY MAP

CAL COMP SUBROUTINES

SUBROUTINE	1 KEY	2	3	4	5	6	7	8	9	10	11	12 BYTE ARRAYS	MANY BYTE ARRAYS		
					4 BYTE ARRAYS										
SYSPLT	4	X	Y	IPN											
SYSPSZ	1			I											
SYSXMX	2			SMAX											
SYSOFF	3			XOFF	XFACT	YOFF	YFACT								
SYSSYM	5	X	Y			N	THETA	SIZE					BCD		
CPLOT	11					N	YMAX	YMIN	LAB		M	DD	Y		
CPLT	12					N	YMAX	YMIN	LAB	ISYM	M	DD	Y		
XYPLOT	13			XMAX	SMIN	N	YMAX	YMN	LAB			DD	Y	X	
XYPLT	14			XMAX	XMIN	N	YMAX	YMIN	LAB	ISYM		DD	Y	X	
PLOTY	15	IP	ISP			N	YMX	YMN	LAB	ISYM		DD	Y		
PLOTXY	16	IP	ISP	MXM	XMN	N	YMX	YMN	LAB	ISYM		DD	Y	X	
SYSEND	6								LAB	FLAG					
LABEL	7	X	Y	SMS	XMN	NCT		SIZE		ND	NI		TITLE		
PRTNUM	8	X	Y			NUM	THETA	SIZE					FMT		
INTPLT	9		NO	XMAXI	SMINI	N	YMAXI	YMINI	LAB	KTYPE	NI		YY	XX	
LOGAXS	10	X	Y	FL	ID	NCT	NUMDEC	ILBL		IHIGH	IROT		TITLE		
WAMPLT	17			FMAX	FMIN	MODE	THEDEG	PHIDEG	NBINX	NGINY			F		
SYMPLT	18			NTIT	NSHEET	N			ISYS				TITLE	Y	X
CONTINUE	19														
VLABEL	21	X	Y	MXM	XMN	NCT	LF	SIZE	ND	NI			TITLE	FMT	
PRTNUM	22	X	Y			NUM	THETA	SIZE					FMT		

COMMON BLOCKS

SUBROUTINE	COMMON NAME	1 KEY	2	3	4	5	6
LABEL	LBLCOM	30	ITEST	SLBL	STTL	SSCL	
LOGAXS	LOGCOM	31	ITEST	SLBL	STTL	SSCL	STICK
INTPLT	COMPLO	32	ITEST	XLNGTH	YLNGTH		
CPLOT	COMCPL	33	ITEST	XLNGTH	YLNGTH		
CPLT	COMCPT	34	ITEST	XLNGTH	YLNGTH		
XYPLOT	COMPLO	35	ITEST	SLNGTH	YLNGTH		
XYPLT	COMPLT	36	ITEST	XLNGTH	YLNGTH		
PLOTY	COMPLY	37	ITEST	XLNGTH	YLNGTH		
PLOTXY	COMPXY	38	ITEST	XLNGTH	YLNGTH		

NOTE: KEY 21   FMT is 12 words long 1 byte/word

PRTNUM   Key 8   NUM is integer.

Key 22   NUM is real