Table of contents

```
                                        .TITLE  TSSPOL  T/S SPOOLING ROUTINES
                                ;
                                ;  COPYRIGHT (C) 1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,
                                ;                1986,1987,1988.
                                ;
                                ;  S&H COMPUTER SYSTEMS,  INC.
                                ;  NASHVILLE,  TENNESSEE
                                ;
                                ;  THE ROUTINES IN THIS MODULE IMPLEMENT AUTOMATIC
                                ;  SPOOLING TO OUTPUT DEVICES SUCH AS LINE PRINTERS.
                                ;
000000                                  .CSECT  TSSPOL
000000  074514                  TSSPOL: .RAD50  /SPL/            ;System overlay ID word
                                ;
                                        .MCALL  .READW,.WRITW,.PRINT
                                ;
                                        .ENABL  LC
                                        .DSABL  GBL
                                ;-----------------------------------------------------------------
                                ;  MACROS TO ENABLE AND DISABLE INTERRUPTS.
                                ;
                                        .MACRO  DISABL
                                        BIS     #340,@#PSW
                                        .ENDM   DISABL

                                        .MACRO  ENABL
                                        BIC     INTPRI,@#PSW
                                        .ENDM   ENABL
                                ;
                                ;  Macro definition to call global routines residing in mapped system regions.
                                ;
                                     .MACRO  OCALL   ENTADD
                                     .IF     B,ENTADD
                                             .ERROR  ;OCALL SPECIFIED WITH NO ENTRY ADDRESS
                                             .MEXIT
                                     .ENDC
                                     CALL    OVRHC           ;CALL THE OVERLAY HANDLER
                                     .WORD   ENTADD          ;SPECIFY THE ENTRY POINT
                                     .ENDM

                                ;
                                ;  GLOBAL DEFINITIONS
                                ;
                                     .GLOBL  CHKSD,SDCLOS,TSSPOL,EMSPHL,EMSPNH
                                     .GLOBL  SPOLGO,SPLSHF,SPLINI,SFLGPG
                                     .GLOBL  SDMOVE,SPLTOP,SPLDEL
                                ;
                                ;  GLOBAL REFERENCES
                                ;
                                     .GLOBL  DF$CLS,DS$NRD,DVSTAT,Q.FUNC,DF$ENT,EMTXIT
                                     .GLOBL  EMTBLK,SDCB,SDCBND,NUMDEV,PNAME,DX$NRD,DVFLAG
                                     .GLOBL  CORUSR,SPLND,SPLNB,SF$BN1,CSIBUF,CSIBND
                                     .GLOBL  SDFRBL,PVSPBL,UFORM,SPDLBF
                                     .GLOBL  NFRESB,SB$TXT,QMSG,GTSYMB,LUNAME
                                     .GLOBL  SFCBSZ,SF$1ST,SF$DEL,SDNAME,DMYDEV
                                     .GLOBL  SD$SMS,SD$SNG,SFFORM,NSPLFL,NSPLBL
                                     .GLOBL  SDFORM,SDANAM,SD$FLK,SD$WFM,SD$FLG
```

```
                                        .GLOBL  $DILUP,LSW,SFID,SPOLID
                                        .GLOBL  NESB,SD$BWT,FMMSG,SPLERR,FLGPAG
                                        .GLOBL  SD$HLD,SF$HLD,SNBUX,LPROG,LPROJ
                                        .GLOBL  SXBPNT,SXSFCB,SDDVU
                                        .GLOBL  FILSPC,CS$SPL,SPUBUF,SPLCHN,SPLARG
                                        .GLOBL  INTPRI,PSW,CHNADR,C.CSW
                                        .GLOBL  S$SPCB,S$SPDB,USPLCH
                                        .GLOBL  SPLBHD,SFCB,SFCBND,SYBFAD
                                        .GLOBL  UREGO,QNSPND,CHKABT
                                        .GLOBL  SFUSER,SFFLAG,SFCHAN,SFFILE
                                        .GLOBL  SFSDCB,SFNMBL,SFSTRT,SFFLNK
                                        .GLOBL  SFQLNK,SF$BSY,SDCHAN,SDFLAG
                                        .GLOBL  SDSFCB,SDFLNK,SDBUF1,SDBUF2
                                        .GLOBL  CTRLTT,SDWLST,FREUSR
                                        .GLOBL  SDBLK,SDFHD,SD$INR
                                        .GLOBL  SD$DEL,SDCBSZ,SBUFWD
                                        .GLOBL  SFCBFH,SD$CLR,SD$BAK,SD$RSV
                                        .GLOBL  SDBU,SDBULS,SDSKIP,LPRG1,LPRG2
                                        .GLOBL  CHNNUM,URO,GETUCH,SPUBND
                                        .GLOBL  GETSYQ,Q.BLKN
                                        .GLOBL  Q.WCNT,Q.COMP,Q.CHAN,SYQIO
                                ;  Global for mapped system handler
                                        .GLOBL  OVRHC

                                ;
                                ;  LOCAL DATA CELLS
                                ;
000002  000000                  GRNTBL: .WORD   0               ;Addr of start of spool file allocation table
000004  000000                  GRNEND: .WORD   0               ;Addr of end of spool file allocation table
000006  000000                  SPBWHD: .WORD   0               ;SPOOL DEVICES WAITING FOR BUFFERS
000010     015      012          CRTXT:  .BYTE   CR,LF
                                ;
                                ;  MISC. PARAMETERS.
                                ;
000015                          CR      =       15              ;CARRIAGE RETURN
000012                          LF      =       12              ;LINE FEED
000007                          BELL    =       7               ;BELL
000135                          RSQBR   =       135             ;RIGHT SQUARE BRACKET
```

```
                                        .SBTTL  SPLINI -- Initialization for spooling system
                                ;-------------------------------------------------------------------------
                                ;  SPLINI is called during system initialization to perform spooler related
                                ;  system initialization.
                                ;
                                ;
000012  010246                  SPLINI: MOV     R2,-(SP)
000014  010346                          MOV     R3,-(SP)
000016  010446                          MOV     R4,-(SP)
                                ;
                                ;  Allocate spool file table at end of this segment
                                ;
000020  012703  005252'                 MOV     #SPLTOP,R3      ;Get address of start of buffer area
000024  010367  177752                  MOV     R3,GRNTBL       ;Set address of start of table
000030  016702  000000G                 MOV     NSPLBL,R2       ;Get # blocks for spool file
000034  062702  000017                  ADD     #15.,R2         ;Bound up to word boundary
000040  072227  177774                  ASH     #-4,R2          ;Divide by 16 to get # words to allocate
000044  012700  177777                  MOV     #-1,R0          ;Get word with all bits set
000050  010023                  10$:    MOV     R0,(R3)+        ;Set all bits on in table
000052  077202                          SOB     R2,10$          ;Fill entire table with ones
000054  010367  177724                  MOV     R3,GRNEND       ;Set address of end of table
                                ;
                                ;  Allocate spool buffers within this system overlay region just beyond
                                ;  the end of the code.
                                ;
000060  012702  000000G                 MOV     #SPLNB,R2       ;Get number of spool buffers wanted
000064  001414                          BEQ     3$              ;Br if none wanted
000066  010367  000000G                 MOV     R3,SPLBHD       ;Set pointer to start of buffer list
000072  000407                          BR      2$
000074  020327  137000          1$:     CMP     R3,#137000      ;Would another buffer overflow segment?
000100  103005                          BHIS    4$              ;Br if yes
000102  010300                          MOV     R3,R0           ;Get address of current buffer
000104  062703  001000                  ADD     #512.,R3        ;Calculate address of next buffer
000110  010310                          MOV     R3,(R0)         ;Make current buffer point to next buffer
000112  077210                  2$:     SOB     R2,1$           ;Loop for all buffers except last
000114  005013                  4$:     CLR     (R3)            ;Make link in last buffer be zero
                                ;
                                ;  Initialize spool file control block linked list
                                ;
000116  016702  000000G         3$:     MOV     SFCB,R2         ;Point to start of SFCB area
000122  010267  000000G                 MOV     R2,SFCBFH       ;Set this as the 1st free SFCB
000126  016703  000000G                 MOV     NSPLFL,R3       ;Get # of SFCB's
000132  010204                          MOV     R2,R4           ;Point to next SFCB
000134  012700  000000C         6$:     MOV     #SFCBSZ/2,R0    ;Get # words in SFCB
000140  005024                  8$:     CLR     (R4)+           ;Zero the SFCB
000142  077002                          SOB     R0,8$
000144  020327  000001                  CMP     R3,#1           ;Is this the last SFCB?
000150  001404                          BEQ     5$              ;Br if yes
000152  010462  000000G                 MOV     R4,SFQLNK(R2)   ;Set forward link to next SFCB
000156  010402                          MOV     R4,R2           ;Point to next SFCB
000160  077313                          SOB     R3,6$           ;Loop of more to initialize
                                ;
                                ;  Initialize word that contains number of free public spool file blocks
                                ;
000162  016702  000000G         5$:     MOV     NSPLBL,R2       ;Get total number of spool file blocks
000166  162702  000000C                 SUB     #<SPLND*<SNBUX+10.>>,R2 ;Subtract number of private blocks
000172  010267  000000G                 MOV     R2,NFRESB       ;Set number of free public blocks
```

```
     58                                 ;
     59                                 ;  Finished
     60                                 ;
     61 000176  012604                  9$:     MOV     (SP)+,R4
     62 000200  012603                          MOV     (SP)+,R3
     63 000202  012602                          MOV     (SP)+,R2
     64 000204  000207                          RETURN
```

```
                                        .SBTTL  CHKSD  -- Check for spooled on .ENTER
                                ;-----------------------------------------------------------------------
                                ;  CHKSD IS CALLED FOR EACH .ENTER TO SEE IF THE DEVICE
                                ;  TO BE OPENED IS A SPOOLED DEVICE.   IF THE .ENTER IS FOR
                                ;  A SPOOLED DEVICE, A SPOOL FILE CONTROL BLOCK (SFCB) IS
                                ;  SET UP AND LINKED INTO THE WAITING QUEUE FOR THE REQUESTED
                                ;  DEVICE.   SEE TSDEFS FOR THE FORMAT OF A SFCB AND SDCB.
                                ;  WHEN CALLED THE ARGUMENT LIST FOR THE .ENTER MUST BE IN
                                ;  EMTBLK.
                                ;  ALL REGISTERS ARE PRESERVED.
                                ;
                                ;  Inputs:
                                ;   FILSPC = File specification
                                ;   CHNADR = Address of channel block.
                                ;   CHNNUM = Current channel number
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> This is a spooled device.
                                ;   C-flag set     ==> This is not a spooled device.
                                ;
000206  010046                  CHKSD:  MOV     R0,-(SP)
000210  010146                          MOV     R1,-(SP)
000212  010346                          MOV     R3,-(SP)
000214  016701  000000G                 MOV     FILSPC,R1       ;GET DEVICE NAME
000220  004767  004546                  CALL    CVTDVU          ;Convert to dev index # and unit #
000224  012700  000000G                 MOV     #SDCB,R0        ;POINT TO FIRST SDCB
000230  020027  000000G         2$:     CMP     R0,#SDCBND      ;CHECKED ALL SDCB'S?
000234  103172                          BHIS    5$              ;BRANCH IF YES
000236  026027  000000G 000000G         CMP     SDNAME(R0),#DMYDEV;Is this an entry for dummy device?
000244  001403                          BEQ     13$             ;Br if yes
000246  020160  000000G                 CMP     R1,SDDVU(R0)    ;DO NAMES MATCH?
000252  001403                          BEQ     3$              ;BRANCH IF YES
000254  062700  000000G         13$:    ADD     #SDCBSZ,R0      ;GO CHECK NEXT SDCB
000260  000763                          BR      2$
                                ;
                                ;  Channel is being opened to a spooled device.
                                ;  Save spool file name on stack then free USR module.
                                ;
000262  016746  000002G         3$:     MOV     FILSPC+2,-(SP)  ;SAVE 1ST 3 CHARS OF FILE NAME
000266  016746  000004G                 MOV     FILSPC+4,-(SP)  ;SAVE LAST 3 CHARS OF FILE NAME
000272  010046                          MOV     R0,-(SP)        ;PROTECT SDCB ADDRESS
000274                                  OCALL   FREUSR          ;RELEASE OUR LOCK ON USR MODULE
000302  012600                          MOV     (SP)+,R0        ;RECOVER SDCB ADDRESS
000304  000410                          BR      1$
                                ;  THERE ARE NO FREE SFCB'S.
                                ;  SUSPEND USER UNTIL ONE IS RELEASED.
000306  010046                  4$:     MOV     R0,-(SP)        ;SAVE SDCB ADDRESS
000310  012700  000000G                 MOV     #S$SPCB,R0      ;QUEUE FOR CONTROL BLOCK
000314  004767  000000G                 CALL    QNSPND          ;ENQ AND SUSPEND EXECUTION
000320  004767  000000G                 CALL    CHKABT          ;SEE IF WE WERE ABORTED WHILE ASLEEP
000324  012600                          MOV     (SP)+,R0        ;RECOVER SDCB ADDRESS
                                ;  FOUND SPOOLED DEVICE -- SET UP SFCB.
                                ;  FIND FREE SFCB.
000326                          1$:     DISABL                  ;** DISABLE **
000334  016701  000000G                 MOV     SFCBFH,R1       ;HEAD OF FREE SFCB CHAIN
000340  001762                          BEQ     4$              ;BRANCH IF NONE FREE
000342  016167  000000G 000000G         MOV     SFQLNK(R1),SFCBFH;REMOVE SFCB FROM LIST
```

```
     58 000350                                          ENABL                   ;** ENABLE **
     59                                 ;  GOT FREE SFCB (R1) -- SET IT UP
     60 000356  116761  000000G 000000G                MOVB    CORUSR,SFUSER(R1);SET USER # IN SFCB
     61 000364  105061  000000G                        CLRB    SFFLAG(R1)      ;CLEAR CONTROL FLAGS
     62                                 ;  Assign unique ID number to spool file
     63 000370  026727  000000G 023417                 CMP     SPOLID,#9999.   ;Is spool file ID number about to overflow?
     64 000376  103402                                 BLO     11$             ;Br if not
     65 000400  005067  000000G                        CLR     SPOLID          ;Reset spool ID number
     66 000404  005267  000000G              11$:      INC     SPOLID          ;Increment spool file ID number
     67 000410  016761  000000G 000000G               MOV     SPOLID,SFID(R1) ;Set ID number for this file
     68                                 ;  Save file name in SFCB
     69 000416  012661  000002G                        MOV     (SP)+,SFFILE+2(R1);SET LAST 3 CHARS OF FILE NAME
     70 000422  012661  000000G                        MOV     (SP)+,SFFILE(R1);SET 1ST 3 CHARS OF FILE NAME
     71 000426  001010                                 BNE     10$             ;BR IF FILE NAME WAS SPECIFIED
     72 000430  116703  000000G              9$:       MOVB    CORUSR,R3       ;GET JOB INDEX NUMBER
     73 000434  016361  000000G 000000G               MOV     LPRG1(R3),SFFILE(R1);IF NO FILE NAME SPECIFIED, USE
     74 000442  016361  000000G 000002G               MOV     LPRG2(R3),SFFILE+2(R1);PROGRAM NAME
     75                                 ;  SET DEFAULT FORM NAME IN SFCB
     76                                 ;  (USE LAST FORM NAME SPECIFIED BY "FORM" COMMAND)
     77 000450  016761  000000G 000000G 10$:       MOV     UFORM,SFFORM(R1)
     78 000456  016761  000002G 000002G               MOV     UFORM+2,SFFORM+2(R1)
     79 000464  016761  000004G 000004G               MOV     UFORM+4,SFFORM+4(R1)
     80                                 ;  SEE IF OUTPUT SHOULD BE HELD UNTIL CHANNEL IS CLOSED.
     81 000472  032760  000000G 000000G               BIT     #SD$HLD,SDFLAG(R0);IS DEVICE IN HOLD MODE?
     82 000500  001403                                 BEQ     8$              ;BR IF NOT
     83 000502  152761  000000G 000000G               BISB    #SF$HLD,SFFLAG(R1);SET HOLD FLAG IN FILE
     84 000510  116761  000000G 000000G 8$:          MOVB    CHNNUM,SFCHAN(R1);SAVE CHANNEL #
     85 000516  010061  000000G                        MOV     R0,SFSDCB(R1)   ;ADDRESS OF SDCB
     86 000522  005061  000000G                        CLR     SFNMBL(R1)      ;# OF BLOCKS IN FILE
     87                                 ;  ADD SFCB TO TAIL OF LIST OF FILES WAITING FOR DEVICE
     88 000526                                         DISABL                  ;** DISABLE **
     89 000534  062700  000000G                        ADD     #SDFHD,R0       ;POINT TO LIST HEAD IN SDCB
     90 000540  005710                       7$:       TST     (R0)            ;AT END OF LIST?
     91 000542  001404                                 BEQ     6$              ;BRANCH IF YES
     92 000544  011000                                 MOV     (R0),R0         ;CHAIN TO NEXT SFCB
     93 000546  062700  000000G                        ADD     #SFQLNK,R0      ;POINT TO ITS FLINK
     94 000552  000772                                 BR      7$
     95 000554  010110                       6$:       MOV     R1,(R0)         ;ADD US TO END OF CHAIN
     96 000556  005061  000000G                        CLR     SFQLNK(R1)      ;CLEAR OUR LIST FLINK
     97 000562                                         ENABL                   ;** ENABLE **
     98                                 ;  ALLOCATE FIRST DISK BLOCK FOR FILE
     99 000570  004767  003510                         CALL    GTSBLK          ;GET A DISK BLOCK
    100 000574  010061  000000G                        MOV     R0,SFSTRT(R1)   ;SET AS START ADDR OF FILE
    101 000600  010061  000000G                        MOV     R0,SFFLNK(R1)   ;SET AS NEXT BLOCK ADDR
    102                                 ;
    103                                 ;  Set up channel so it will do I/O to spool file.
    104                                 ;
    105 000604  016703  000000G                        MOV     CHNADR,R3       ;GET ADDRESS OF CHANNEL BLOCK
    106 000610  052763  000000G 000000G               BIS     #CS$SPL,C.CSW(R3);SET FLAG SAYING THIS IS A SPOOLED CHANNEL
    107 000616  000241                                 CLC                     ;Signal success on return
    108 000620  000401                                 BR      12$
    109                                 ;
    110                                 ;  This is not a spooled device
    111                                 ;
    112 000622  000261                       5$:       SEC                     ;Signal failure on return
    113                                 ;
    114                                 ;  Finished
```

```
    115                                         ;
    116 000624  012603                          12$:    MOV     (SP)+,R3
    117 000626  012601                                  MOV     (SP)+,R1
    118 000630  012600                                  MOV     (SP)+,R0
    119 000632  000207                                  RETURN
```

```
                                        .SBTTL  SDMOVE -- Write user's data to spool file
                                ;-------------------------------------------------------------------
                                ;  SDMOVE is called during .write emt processing to move the user's data
                                ;  from his buffer to the spool buffer and then write it to the spool file.
                                ;
                                ;  Inputs:
                                ;   CHNNUM = User's channel number.
                                ;   EMTBLK = EMT argument block.
                                ;
000634  010146                  SDMOVE: MOV     R1,-(SP)
000636  010246                          MOV     R2,-(SP)
000640  010346                          MOV     R3,-(SP)
000642  010446                          MOV     R4,-(SP)
000644  010546                          MOV     R5,-(SP)
                                ;
                                ;  Find SDCB associated with this channel
                                ;
000646  004767  004050                  CALL    FNDSD           ;FIND SDCB FOR THIS CHANNEL
000652  103534                          BCS     SDMVX           ;BR IF COULDN'T FIND IT
                                ;
                                ;  Make sure spool file buffer is assigned to our channel
                                ;
000654  004767  000352          8$:     CALL    SGETBF          ;GET SPOOL FILE BUFFER FOR US
                                ;
                                ;  Set up info about data to be moved
                                ;
000660  016703  000004G                 MOV     EMTBLK+4,R3     ;GET USER'S BUFFER ADDRESS
000664  016705  000006G                 MOV     EMTBLK+6,R5     ;GET # WORDS TO MOVE
000670  010567  000000G                 MOV     R5,UR0          ;RETURN WORD COUNT TO USER IN R0
000674  001523                          BEQ     SDMVX           ;BR IF NOTHING TO WRITE
000676  006305                          ASL     R5              ;CONVERT TO BYTE COUNT
000700  060305                          ADD     R3,R5           ;GET ADDRESS PAST END OF USER'S BUFFER
                                ;
                                ;  See if this is the first write to the spooled device.
                                ;
000702  132761  000000G 000000G         BITB    #SF$1ST,SFFLAG(R1);IS THIS 1ST WRITE?
000710  001073                          BNE     SDMV1           ;BR IF NOT 1ST WRITE
                                ;
                                ;  This is first write to this spooled device.
                                ;  See if 1st write is to block # 0 or non-zero.
                                ;
000712  005767  000002G                 TST     EMTBLK+2        ;Is 1st block # 0?
000716  001403                          BEQ     2$              ;Br if yes
000720  152761  000000G 000000G         BISB    #SF$BN1,SFFLAG(R1);Remember 1st block is not 0
                                ;
                                ;  See if user specified a form name.
                                ;
000726  152761  000000G 000000G 2$:     BISB    #SF$1ST,SFFLAG(R1);REMEMBER 1ST WRITE HAS BEEN DONE
000734  004767  000000G                 CALL    GETUCH          ;GET CHAR FROM USER'S BUFFER
000740  005303                          DEC     R3              ;POINT BACK TO CHAR JUST GOTTEN
000742  120027  000135                  CMPB    R0,#RSQBR       ;DID USER SPECIFY A FORM NAME?
000746  001037                          BNE     16$             ;BR IF NOT
                                ;
                                ;  User specified a form name with this file.
                                ;  Move the name to buffer in SFCB.
                                ;
000750  010102                          MOV     R1,R2           ;GET ADDRESS OF SFCB
```

```
     58 000752  062702  000000G                 ADD     #SFFORM,R2      ;POINT TO FORM NAME BUFFER IN SFCB
     59 000756  005203                          INC     R3              ;SKIP OVER RIGHT-SQUARE-BRACKET CHAR
     60 000760  012704  000006                  MOV     #6,R4           ;FORM NAMES MAY BE UP TO 6 CHARS LONG
     61 000764  004767  000000G          13$:   CALL    GETUCH          ;GET CHAR FROM USER'S BUFFER
     62 000770  120027  000015                  CMPB    R0,#CR          ;CARRIAGE-RETURN?
     63 000774  001412                          BEQ     14$             ;BR IF HIT END OF NAME
     64 000776  120027  000141                  CMPB    R0,#141         ;Is this a lower case character?
     65 001002  103405                          BLO     1$              ;Br if not
     66 001004  120027  000172                  CMPB    R0,#172
     67 001010  101002                          BHI     1$
     68 001012  162700  000040                  SUB     #40,R0          ;Convert lower case char to upper case
     69 001016  110022                  1$:     MOVB    R0,(R2)+        ;MOVE NAME TO SFCB BUFFER
     70 001020  077417                          SOB     R4,13$          ;MOVE UP TO 6 CHARS
     71 001022  004767  000000G          14$:   CALL    GETUCH          ;NOW SKIP UP LINE-FEED
     72 001026  120027  000012                  CMPB    R0,#LF
     73 001032  001373                          BNE     14$
     74 001034  005704                          TST     R4              ;DID WE GET ALL 6 CHARACTERS?
     75 001036  001420                          BEQ     SDMV1           ;BR IF YES
     76 001040  112722  000040          15$:    MOVB    #' ,(R2)+       ;PAD WITH BLANKS
     77 001044  077403                          SOB     R4,15$
     78 001046  132761  000000G 000000G 16$:    BITB    #SF$BN1,SFFLAG(R1);Is 1st block is not 0?
     79 001054  001011                          BNE     SDMV1           ;NO FLAGPAGE IF NOT
     80 001056  016100  000000G                 MOV     SFSDCB(R1),R0   ;GET SDCB FOR FILE
     81 001062  032760  000000G 000000G         BIT     #SD$FLG,SDFLAG(R0) ;FLAG PAGE FOR DEVICE?
     82 001070  001403                          BEQ     SDMV1           ;BR IF NOT
     83 001072                                  OCALL   FLGPAG          ;ELSE DO FLAG PAGE
     84                                 ;
     85                                 ;  Move data from user's buffer to spool buffer in job context area.
     86                                 ;
     87 001100  016702  000000G         SDMV1:  MOV     SXBPNT,R2       ;GET CURRENT POINTER INTO SPOOL BUFFER
     88 001104  020227  000000G         2$:     CMP     R2,#SPUBND      ;HAVE WE REACHED END OF SPOOL BUFFER?
     89 001110  103406                          BLO     1$              ;BR IF NOT
     90 001112  010267  000000G                 MOV     R2,SXBPNT       ;SAVE BUFFER POINTER
     91 001116  004767  000036                  CALL    SBUFOT          ;WRITE BUFFER TO SPOOL FILE
     92 001122  016702  000000G                 MOV     SXBPNT,R2       ;GET POINTER INTO SPOOL BUFFER
     93 001126  004767  000000G         1$:     CALL    GETUCH          ;GET CHARACTER FROM USER'S BUFFER
     94 001132  110022                          MOVB    R0,(R2)+        ;MOVE CHAR TO SPOOL BUFFER
     95 001134  020305                          CMP     R3,R5           ;HAVE WE MOVED ALL OF USER'S DATA?
     96 001136  103762                          BLO     2$              ;BR IF NOT
     97                                 ;
     98                                 ;  Finished moving user's data
     99                                 ;
    100 001140  010267  000000G                 MOV     R2,SXBPNT       ;SAVE SPOOL BUFFER POINTER
    101                                ;
    102                                ;  Finished
    103                                ;
    104 001144  012605                 SDMVX:  MOV     (SP)+,R5
    105 001146  012604                         MOV     (SP)+,R4
    106 001150  012603                         MOV     (SP)+,R3
    107 001152  012602                         MOV     (SP)+,R2
    108 001154  012601                         MOV     (SP)+,R1
    109 001156  000207                         RETURN
```

```
                                        .SBTTL  SBUFOT -- Write current buffer to file
                                ;-------------------------------------------------------
                                ;  SBUFOT IS CALLED TO WRITE THE CURRENT SPOOL BUFFER TO
                                ;  THE SPOOL FILE AND THEN RESET THE BUFFER POINTERS AND
                                ;  ALLOCATE A NEW DISK BLOCK FOR THE NEXT BUFFER.
                                ;  WHEN CALLED, SXSFCB MUST POINT TO THE SFCB FOR THE
                                ;  CURRENT FILE.   ALL REGISTERS ARE PRESERVED.
                                ;  Inputs:
                                ;   CHNNUM = Current channel number.
                                ;   SPUBUF = Spool buffer.
                                ;   SXBPNT = Pointer to free space in spool buffer.
                                ;
                                ;  Outputs:
                                ;   SXBPNT = Reset to point to first free byte in buffer.
                                ;
001160  010046                 SBUFOT: MOV     R0,-(SP)
001162  010146                         MOV     R1,-(SP)
001164  016701  000000G                MOV     SXSFCB,R1       ;POINT TO SFCB
001170  001415                         BEQ     9$              ;BRANCH IF NO FILE USING BUFFER
                                ;  CHECK NUMBER OF DATA WORDS IN BUFFER
001172  016700  000000G                MOV     SXBPNT,R0       ;GET CURRENT POINTER
001176  162700  000000G                SUB     #SPUBUF,R0      ;SUBTRACT BUFFER BASE
001202  162700  000004                 SUB     #4,R0           ;REMOVE CONTROL WORDS
001206  001406                         BEQ     9$              ;BRANCH IF BUFFER IS EMPTY
                                ;  GET A NEW DISK BLOCK AND SET AS FILE FLINK IN THIS BUFFER
001210  004767  003070                 CALL    GTSBLK          ;GET A DISK BLOCK
001214  010067  000000G                MOV     R0,SPUBUF       ;SET AS FLINK IN BUFFER
                                ;  WRITE OUT THE BUFFER.
001220  004767  000042                 CALL    SFLUSH          ;DO THE WRITE
                                ;
001224  012601                 9$:     MOV     (SP)+,R1
001226  012600                         MOV     (SP)+,R0
001230  000207                         RETURN
```

```
                                        .SBTTL  SGETBF -- Reserve buffer for file
                                ;-------------------------------------------------
                                ;  SGETBF IS CALLED TO MAKE SURE THAT THE SPOOL BUFFER IS
                                ;  IN USE BY THE SPOOL FILE WHOSE SFCB IS POINTED TO BY R1.
                                ;  IF THE BUFFER IS BEING USED BY ANOTHER FILE, THE BUFFER
                                ;  IS WRITTEN OUT WITH ITS CURRENT CONTENTS AND
                                ;  THE BUFFER IS CLAIMED FOR THE NEW FILE BY STORING
                                ;  THE SFCB ADDRESS IN SXSFCB.
                                ;  ALL REGISTERS ARE PRESERVED.
                                ;
001232  020167  000000G        SGETBF: CMP     R1,SXSFCB       ;HAVE WE GOT CONTROL OF BUFFER?
001236  001412                        BEQ     9$              ;BRANCH IF YES
001240  005767  000000G                TST     SXSFCB          ;IS BUFFER IN USE BY ANYONE?
001244  001402                        BEQ     8$              ;BRANCH IF NOT
                                ;  BUFFER IS IN USE BY ANOTHER FILE.
                                ;  WRITE OUT INFO IN BUFFER.
001246  004767  177706                CALL    SBUFOT          ;DUMP CURRENT BUFFER FULL
                                ;  NOW CLAIM BUFFER FOR OUR FILE
001252  010167  000000G        8$:     MOV     R1,SXSFCB       ;SAY WE'VE GOT BUFFER
001256  012767  000004G 000000G        MOV     #SPUBUF+4,SXBPNT;INITIALIZE SPOOL BUFFER POINTER
001264  000207                9$:     RETURN
```

```
                                        .SBTTL  SFLUSH -- Write current buffer to file
                                ;-------------------------------------------------------
                                ;  SFLUSH IS CALLED TO WRITE THE CURRENT SPOOL BUFFER TO
                                ;  THE SPOOL FILE.  SFLUSH SETS THE CORRECT WORD COUNT
                                ;  IN THE BUFFER CONTROL WORD AND WRITES THE BUFFER TO
                                ;  THE DISK BLOCK WHOSE ADDRESS IS IN SFFLNK.
                                ;  IT THEN SETS SFFLNK TO THE VALUE WHICH IS IN THE
                                ;  FLINK FIELD OF THE BUFFER.  THE BUFFER POINTERS
                                ;  ARE RESET.  ALL REGESTERS ARE PRESERVED.
                                ;
001266  010046                  SFLUSH: MOV     R0,-(SP)
001270  010146                          MOV     R1,-(SP)
001272  010246                          MOV     R2,-(SP)
001274  010346                          MOV     R3,-(SP)
001276  012702  000000G                 MOV     #SPUBUF,R2      ;POINT TO BUFFER
001302  016701  000000G                 MOV     SXSFCB,R1       ;POINT TO CURRENT SFCB
001306  001456                          BEQ     8$              ;BRANCH IF NO FILE USING BUFFER
                                ;  SET WORD COUNT IN BUFFER
001310  016700  000000G                 MOV     SXBPNT,R0       ;GET CURRENT POINTER INTO BUFFER
001314  032700  000001                  BIT     #1,R0           ;IS ADDRESS ODD?
001320  001401                          BEQ     1$              ;BRANCH IF NOT
001322  105020                          CLRB    (R0)+           ;PUT NULL AS ODD CHAR
001324  160200                  1$:     SUB     R2,R0           ;SUBTRACT BUFFER BASE
001326  162700  000004                  SUB     #4,R0           ;DON'T COUNT CONTROL WORDS
001332  006200                          ASR     R0              ;CONVERT COUNT TO WORDS
001334  010062  000002                  MOV     R0,2(R2)        ;STORE WORD COUNT INTO BUFFER
                                ;  WRITE OUT THE BUFFER.
001340                          2$:     .WRITW  #SPLARG,#USPLCH,R2,#256.,SFFLNK(R1) ;WRITE BLOCK TO SPOOL FILE
001376  103003                          BCC     3$              ;BR IF NO WRITE ERROR OCCURED
001400                                  .PRINT  #SPLERR         ;SIGNIFY ERROR ON SPOOL WRITE
001406  011261  0000000G          3$:     MOV     (R2),SFFLNK(R1) ;SET NEW FILE FLINK
                                ;  TELL SPOOLER THAT ANOTHER BLOCK IS AVAILABLE.
001412  005261  0000000G                  INC     SFNMBL(R1)      ;SAY ANOTHER BLOCK IS IN FILE
001416  016100  0000000G                  MOV     SFSDCB(R1),R0   ;POINT TO SDCB
001422  132761  0000000G 0000000G        BITB    #SF$BSY,SFFLAG(R1);IS FILE ACTIVE NOW?
001430  001403                          BEQ     7$              ;BRANCH IF NOT
001432  004767  000712                  CALL    SPOLRD          ;TELL SPOOLER IT CAN READ BLOCK
001436  000402                          BR      8$
001440  004767  000100          7$:     CALL    SPOLGO          ;TRY TO START SPOOLER
                                ;  RESET BUFFER POINTERS
001444  062702  000004          8$:     ADD     #4,R2           ;POINT PAST CONTROL WORDS
001450  010267  0000000G                  MOV     R2,SXBPNT
001454  012603                          MOV     (SP)+,R3
001456  012602                          MOV     (SP)+,R2
001460  012601                          MOV     (SP)+,R1
001462  012600                          MOV     (SP)+,R0
001464  000207                          RETURN
```

```
                                        .SBTTL  SPLSHF -- Set or reset hold flag for file
                                ;-----------------------------------------------------------------------
                                ;  SPLSHF is called to set or reset the hold flag for a spool file.
                                ;
                                ;  Inputs:
                                ;   CHNNUM = User's channel number
                                ;   R2 = 0==>nohold; 1==>hold file.
                                ;
001466  010146                  SPLSHF: MOV     R1,-(SP)
001470  012767  000000G 000000G        MOV     #SD$RSV,URO     ;Assume the device is not spooled
                                ;
                                ;  Find SFCF associated with this channel
                                ;
001476  004767  003220                  CALL    FNDSD           ;Locate SFCF associated with this channel
001502  103416                          BCS     9$              ;Br if cannot find the SFCF
                                ;
                                ;  Set or reset the flag in the SFCB
                                ;
001504  116167  000000G 000000G        MOVB    SFFLAG(R1),URO  ;Get current setting of hold mode
001512  042767  000000C 000000G        BIC     #^CSF$HLD,URO   ;And return to user
001520  152761  000000G 000000G        BISB    #SF$HLD,SFFLAG(R1);Assume we want to set the flag
001526  005702                          TST     R2              ;Do we want to reset the flag?
001530  001003                          BNE     9$              ;Br if not
001532  142761  000000G 000000G        BICB    #SF$HLD,SFFLAG(R1);Clear the hold flag
                                ;
                                ;  Finished
                                ;
001540  012601                  9$:     MOV     (SP)+,R1
001542  000207                          RETURN
```

```
                                        .SBTTL  SPOLGO -- Try to start spooled device
                                ;-------------------------------------------------------
                                ;  SPOLGO IS CALLED TO TRY TO START UP THE SPOOLED DEVICE
                                ;  WHOSE SDCB IS POINTED TO BY R0.
                                ;  ALL REGISTERS ARE PRESERVED.
                                ;
001544                          SPOLGO: DISABL                  ;** DISABLE **
001552  005760  000000G                TST     SDSFCB(R0)      ;IS DEVICE BUSY NOW?
001556  001402                        BEQ     7$              ;BR IF NOT
001560  000167  000554                JMP     SGX2
001564  010146                  7$:   MOV     R1,-(SP)
                                ;
                                ;  LOOK FOR WAITING FORM ALIGNMENT PRINT FILE.
                                ;  (ALIGNMENT FILES USE THE FORM NAME "**    ")
                                ;
001566  016001  000000G               MOV     SDFHD(R0),R1    ;POINT TO 1ST WAITING SFCB
001572  001002                        BNE     1$              ;BR IF THERE ARE WAITING FILES
001574  000167  000536                JMP     SGX1
001600  005761  000000G          1$:  TST     SFNMBL(R1)      ;ANY DATA IN FILE YET?
001604  001404                        BEQ     5$              ;BR IF NOT
001606  026127  000000G 025052        CMP     SFFORM(R1),#"** ;ALIGNMENT FILE?
001614  001576                        BEQ     SGOTFL          ;BR IF FOUND ALIGNMENT FILE
001616  016101  000000G          5$:  MOV     SFQLNK(R1),R1   ;LINK FORWARD TO NEXT SFCB
001622  001366                        BNE     1$              ;BR IF MORE TO CHECK
                                ;
                                ;  SEE IF WE ARE WAITING FOR A FORM MOUNT.
                                ;
001624  032760  000000G 000000G       BIT     #SD$WFM,SDFLAG(R0);WAITING FOR A FORM MOUNT?
001632  001402                        BEQ     9$              ;BR IF NOT
001634  000167  000476                JMP     SGX1            ;EXIT IF WAITING FOR FORM MOUNT
                                ;
                                ;  SEE IF WE CAN FIND A FILE WHICH NEEDS THE FORM WHICH
                                ;  IS CURRENTLY MOUNTED.
                                ;
001640  016001  000000G          9$:  MOV     SDFHD(R0),R1    ;POINT TO 1ST WAITING SFCB
001644  132761  000000G 000000G  3$:  BITB    #SF$HLD,SFFLAG(R1);IS THIS FILE BEING HELD?
001652  001031                        BNE     2$              ;BR IF YES
001654  005761  000000G               TST     SFNMBL(R1)      ;ANY DATA IN FILE YET?
001660  001426                        BEQ     2$              ;BR IF NOT
001662  132761  000000G 000000G       BITB    #SF$1ST,SFFLAG(R1);WAS 1ST WRITE EVER DONE TO FILE
001670  001002                        BNE     8$              ;BR IF YES
001672  000167  000314                JMP     SGOTFL          ;IF NOT THEN IGNORE FORM NAME (EMPTY FILE)
001676  132761  000000G 000000G  8$:  BITB    #SF$DEL,SFFLAG(R1) ;Is this file to be deleted?
001704  001142                        BNE     SGOTFL          ;Br if yes -- Delete the file now
001706  026160  000000G 000000G       CMP     SFFORM(R1),SDFORM(R0);SEE IF SAME FORM NAME
001714  001010                        BNE     2$
001716  026160  000002G 000002G       CMP     SFFORM+2(R1),SDFORM+2(R0)
001724  001004                        BNE     2$
001726  026160  000004G 000004G       CMP     SFFORM+4(R1),SDFORM+4(R0)
001734  001424                        BEQ     SGFSF           ;BR IF FOUND ELIGIBLE FILE
001736  016101  000000G          2$:  MOV     SFQLNK(R1),R1   ;LINK TO NEXT SFCB
001742  001340                        BNE     3$              ;BR IF MORE TO CHECK
                                ;
                                ;  CAN'T FIND FILE NEEDING CURRENTLY MOUNTED FORM.
                                ;  SEE IF WE ARE ALLOWED TO REQUEST NEW FORM.
                                ;
001744  032760  000000G 000000G       BIT     #SD$FLK,SDFLAG(R0);IS CURRENT FORM LOCKED ON PRINTER?
```

SPOLGO -- Try to start spooled device

```
     58 001752  001171                          BNE     SGX1            ;BR IF FORM LOCKED
     59                                ;
     60                                ;  WE CAN REQUEST FORM CHANGE.
     61                                ;  LOOK FOR 1ST FILE WAITING TO BE PRINTED.
     62                                ;
     63 001754  016001  000000G                 MOV     SDFHD(R0),R1    ;POINT TO 1ST SFCB
     64 001760  005761  000000G          4$:   TST     SFNMBL(R1)      ;ANY DATA IN FILE YET?
     65 001764  001404                          BEQ     6$              ;BR IF NOT
     66 001766  132761  000000G 000000G        BITB    #SF$HLD,SFFLAG(R1);IS FILE BEING HELD?
     67 001774  001414                          BEQ     SGNFMT          ;BR IF GOT ELIGIBLE FILE
     68 001776  016101  000000G          6$:   MOV     SFQLNK(R1),R1   ;LINK FORWARD TO NEXT SFCB
     69 002002  001366                          BNE     4$              ;BR IF MORE TO CHECK
     70 002004  000554                          BR      SGX1            ;COULD FIND NO FILE TO PRINT
     71                                ;
     72                                ;  FOUND FILE FOR CURRENTLY MOUNTED FORM.
     73                                ;  SEE IF IN SINGLE FILE MODE WHICH REQUIRES A MOUNT MESSAGE
     74                                ;  EVEN FOR THIS FILE.
     75                                ;
     76 002006  032760  000000G 000000G SGFSF: BIT     #SD$SNG,SDFLAG(R0);IN SINGLE FILE MODE?
     77 002014  001476                          BEQ     SGOTFL          ;BR IF NOT
     78 002016  032760  000000G 000000G        BIT     #SD$SMS,SDFLAG(R0);ALREADY SENT MESSAGE?
     79 002024  001072                          BNE     SGOTFL          ;BR IF YES
     80                                ;
     81                                ;  PUT OUT MESSAGE REQUESTING FORM MOUNT.
     82                                ;
     83 002026                         SGNFMT: ENABL                   ;** ENABLE **
     84 002034  010246                          MOV     R2,-(SP)
     85 002036  010446                          MOV     R4,-(SP)
     86 002040  010546                          MOV     R5,-(SP)
     87 002042  010046                          MOV     R0,-(SP)        ;SAVE SDCB ADDRESS ON TOP OF STACK
     88 002044  116702  000000G                 MOVB    CTRLTT,R2       ;GET JOB # OF OPERATOR'S CONSOLE
     89 002050  001445                          BEQ     2$              ;BR IF NO OPERATOR
     90 002052  032762  000000G 000000G        BIT     #$DILUP,LSW(R2) ;IS OPERATOR LOGGED ON NOW?
     91 002060  001441                          BEQ     2$              ;BR IF NOT
     92 002062  004767  000000G                 CALL    GTSYMB          ;GET A FREE SYSTEM MESSAGE BUFFER
     93 002066  103436                          BCS     2$              ;BR IF NONE AVAILABLE
     94 002070  010402                          MOV     R4,R2           ;GET ADDRESS OF MESSAGE BUFFER
     95 002072  062702  000000G                 ADD     #SB$TXT,R2      ;POINT TO TEXT STORAGE AREA
     96 002076  012705  000000G                 MOV     #FMMSG,R5       ;MOVE MESSAGE TEXT TO BUFFER
     97 002102  012700  000021                  MOV     #17.,R0         ;MOVE 17 WORDS
     98 002106  012522                   1$:   MOV     (R5)+,(R2)+
     99 002110  077002                          SOB     R0,1$
    100 002112  010402                          MOV     R4,R2           ;GET ADDRESS OF MESSAGE BUFFER BLOCK
    101 002114  062702  000014G                 ADD     #SB$TXT+12.,R2  ;POINT INTO MESSAGE TEXT
    102 002120  062701  000000G                 ADD     #SFFORM,R1      ;MOVE FORM NAME INTO MESSAGE TEXT
    103 002124  012122                          MOV     (R1)+,(R2)+
    104 002126  012122                          MOV     (R1)+,(R2)+
    105 002130  012122                          MOV     (R1)+,(R2)+
    106 002132  011600                          MOV     (SP),R0         ;GET SDCB ADDRESS
    107 002134  062700  000000G                 ADD     #SDANAM,R0      ;Point to ASCII device name
    108 002140  062702  000012                  ADD     #10.,R2         ;Point to position where dev name goes
    109 002144  112022                          MOVB    (R0)+,(R2)+     ;Move device name into message
    110 002146  112022                          MOVB    (R0)+,(R2)+
    111 002150  112022                          MOVB    (R0)+,(R2)+
    112 002152  116701  000000G                 MOVB    CTRLTT,R1       ;GET # OF OPERATOR'S CONSOLE
    113 002156                                  OCALL   QMSG            ;QUEUE MESSAGE FOR OPERATOR
    114 002164  012600                   2$:   MOV     (SP)+,R0
```

SPOLGO -- Try to start spooled device

```
    115 002166  052760  000000G 000000G          BIS     #SD$WFM,SDFLAG(R0);WAITING FOR FORM MOUNT
    116 002174  052760  000000G 000000G          BIS     #SD$SMS,SDFLAG(R0);FORM MOUNT MESSAGE SENT
    117 002202  012605                          MOV     (SP)+,R5
    118 002204  012604                          MOV     (SP)+,R4
    119 002206  012602                          MOV     (SP)+,R2
    120 002210  000452                          BR      SGX1
    121                                 ;
    122                                 ;  FOUND ELIGIBLE FILE -- SAY DEVICE IS BUSY WITH FILE.
    123                                 ;
    124 002212  010160  000000G         SGOTFL: MOV     R1,SDSFCB(R0)   ;ASSOCIATE FILE WITH DEVICE
    125 002216  152761  000000G 000000G        BISB    #SF$BSY,SFFLAG(R1);SAY FILE IS BUSY
    126 002224                                  ENABL                   ;** ENABLE **
    127 002232  010146                          MOV     R1,-(SP)
    128 002234  012701  000000G                 MOV     #DF$ENT,R1      ;Get function code for .ENTER
    129 002240  004767  001304                  CALL    SDFUNC          ;Tell device handler we are doing .ENTER
    130 002244  012601                          MOV     (SP)+,R1
    131 002246  132761  000000G 000000G        BITB    #SF$DEL,SFFLAG(R1) ;Is file marked to be deleted?
    132 002254  001403                          BEQ     1$              ;Br if not
    133 002256  052760  000000G 000000G        BIS     #SD$DEL,SDFLAG(R0) ;Set delete flag for spooler
    134 002264  016160  000000G 000000G 1$:    MOV     SFSTRT(R1),SDFLNK(R0);SET DISK ADDR FOR 1ST READ
    135 002272  005060  000000G                 CLR     SDBLK(R0)       ;Say 1st block written will be # 0
    136 002276  132761  000000G 000000G        BITB    #SF$BN1,SFFLAG(R1);Should 1st block written be # 1?
    137 002304  001402                          BEQ     2$              ;Br if not
    138 002306  005260  000000G                 INC     SDBLK(R0)       ;Make 1st block written be # 1
    139 002312  005367  000000G         2$:     DEC     NESB            ;ONE LESS EXTRA SPOOL BUFFER
    140 002316  004767  000026                  CALL    SPOLRD          ;READ IN 1ST BLOCK FROM FILE
    141                                 ;  RESTART USERS WAITING FOR DISK BLOCKS
    142 002322  010046                          MOV     R0,-(SP)
    143 002324  012700  000000G                 MOV     #S$SPDB,R0      ;POINT TO WAITING Q HEAD
    144 002330  004767  000000G                 CALL    UREGO           ;RESTART 1ST USER IN Q
    145 002334  012600                          MOV     (SP)+,R0
    146 002336  012601                  SGX1:   MOV     (SP)+,R1
    147 002340                          SGX2:   ENABL                   ;** ENABLE **
    148 002346  000207                          RETURN
```

```
                                        .SBTTL  SPOLRD -- Read next block from spool file
                                ;-------------------------------------------------------
                                ;  SPOLRD IS CALLED TO ATTEMPT TO READ THE NEXT BLOCK
                                ;  FROM THE SPOOL FILE.  WHEN CALLED, R0 MUST POINT TO THE
                                ;  SDCB FOR THE DEVICE ATTACHED TO THE FILE.
                                ;  ALL REGISTERS ARE PRESERVED.
                                ;
002350  010146                  SPOLRD: MOV     R1,-(SP)
002352  010246                          MOV     R2,-(SP)
002354  016001  000000G                 MOV     SDSFCB(R0),R1   ;POINT TO SFCB
002360  001534                          BEQ     SRX             ;BRANCH IF DEVICE INACTIVE
                                ;  SEE IF WE SHOULD START ANOTHER READ.
002362                                  DISABL                  ;** DISABLE **
002370  005760  000000G                 TST     SDBUF2(R0)      ;INPUT BUFFER ALREADY BUSY?
002374  001123                          BNE     SREX            ;BR IF YES
002376  032760  000000G 000000G         BIT     #SD$BWT,SDFLAG(R0) ;ALREADY WAITING FOR A BUFFER?
002404  001117                          BNE     SREX            ;BR IF YES
002406  032760  000000G 000000G         BIT     #SD$BAK,SDFLAG(R0);IS BACKUP WANTED?
002414  001006                          BNE     3$              ;BR IF YES
002416  005761  000000G                 TST     SFNMBL(R1)      ;ANY BLOCKS IN THIS FILE?
002422  001510                          BEQ     SREX            ;BR IF FILE IS EMPTY
002424  005760  000000G                 TST     SDFLNK(R0)      ;DO WE HAVE DISK FLINK?
002430  001505                          BEQ     SREX            ;BR IF NOT
002432  005760  000000G          3$:    TST     SDBUF1(R0)      ;OUTPUT BUFFER ALLOCATED?
002436  001403                          BEQ     2$              ;IF NOT DEFINITELY START READ
                                ;  SEE IF THERE ARE ENOUGH BUFFERS AVAILABLE FOR US TO
                                ;  DOUBLE BUFFER.  ONLY DOUBLE BUFFER IF THERE ARE ENOUGH FREE
                                ;  BUFFERS FOR EACH ACTIVE SPOOLER TO GET AT LEAST ONE.
002440  005727  000000C                 TST     #<SPLNB+1-<2*SPLND>>;ENOUGH BUFFERS TO DOUBLE BUFFER?
002444  002477                          BLT     SREX            ;BR IF TOO FEW BUFFERS TO DOUBLE BUFFER
                                ;  WE WANT TO START THE NEXT READ.
                                ;  SEE IF WE CAN GET A SPOOL BUFFER.
002446  016702  000000G          2$:    MOV     SPLBHD,R2       ;HEAD OF FREE BUFFER LIST
002452  001010                          BNE     1$              ;BRANCH IF BUFFER AVAILABLE
                                ;  THERE ARE NO FREE BUFFERS.
                                ;  QUEUE SPOLRD TO BE RECALLED WHEN A BUFFER IS RELEASED.
002454  052760  000000G 000000G         BIS     #SD$BWT,SDFLAG(R0) ;SAY WE ARE WAITING ON A BUFFER
002462  012701  000006'                 MOV     #SPBWHD,R1      ;POINT TO WAIT LIST HEAD
002466  004767  000166                  CALL    SPOOLQ          ;QUEUE FOR RECALL ** ENABLE **
002472  000467                          BR      SRX
                                ;  WE FOUND A FREE BUFFER -- START READ INTO IT
002474  011267  000000G          1$:    MOV     (R2),SPLBHD     ;REMOVE BUFFER FROM FREE LIST
002500  010260  000000G                 MOV     R2,SDBUF2(R0)   ;SET BUFFER ADDRESS
002504  052760  000000G 000000G         BIS     #SD$INR,SDFLAG(R0) ;SAY READ IS IN PROGRESS
002512                                  ENABL                   ;** ENABLE **
                                ;
                                ;  SEE IF OPERATOR WANTS TO BACK UP IN SPOOL FILE
                                ;
002520  032760  000000G 000000G         BIT     #SD$BAK,SDFLAG(R0);HAS BACKUP BEEN REQUESTED?
002526  001417                          BEQ     4$              ;BR IF NOT
                                ;  BACK UP HAS BEEN REQUESTED.  BACK UP TO OLDEST REMEMBERED BLOCK.
002530  010346                          MOV     R3,-(SP)
002532  010003                          MOV     R0,R3           ;POINT TO CELLS WITH SAVED
002534  062703  000000G                 ADD     #SDBU,R3        ;BACK-UP DISK ADDRESSES
002540  011360  000000G                 MOV     (R3),SDFLNK(R0) ;SET OLDEST AS NEW FLINK
002544  005261  000000G          5$:    INC     SFNMBL(R1)      ;WE JUST ADDED 1 BLOCK TO FILE
002550  005023                          CLR     (R3)+           ;CLEAR SAVE AREA
```

SPOLRD -- Read next block from spool file

```
     58 002552  005713                          TST     (R3)                ;MORE TO ADD?
     59 002554  001373                          BNE     5$                  ;ADD ALL WE REMEMBER
     60 002556  042760  000000G 000000G         BIC     #SD$BAK,SDFLAG(R0);SAY BACKUP HAS BEEN DONE
     61 002564  012603                          MOV     (SP)+,R3
     62                                 ;
     63                                 ;  START THE DISK READ
     64                                 ;
     65 002566  010046                  4$:     MOV     R0,-(SP)            ;SAVE ADDRESS OF SDCB
     66                                 ;
     67                                 ;  Get a system I/O queue element
     68                                 ;
     69 002570  012701  000000G                 MOV     #SPLCHN,R1          ;POINT TO SPOOL FILE CHANNEL BLOCK
     70 002574  004767  000000G                 CALL    GETSYQ              ;GET SYSTEM I/O QUEUE ELEMENT
     71                                 ;
     72                                 ;  Set up I/O queue element for read
     73                                 ;
     74 002600  011600                          MOV     (SP),R0             ;GET ADDRESS OF SDCB
     75 002602  010061  000000G                 MOV     R0,Q.CHAN(R1)       ;SDCB ADDRESS WILL BE IN R1 FOR COMPLETION ROUTINE
     76 002606  066061  000000G 000000G         ADD     SDFLNK(R0),Q.BLKN(R1);SET DISK BLOCK #
     77 002614  010200                          MOV     R2,R0               ;GET BUFFER ADDRESS
     78 002616  004767  000000G                 CALL    SYBFAD              ;SET BUFFER ADDRESS IN I/O QUEUE ELEMENT
     79 002622  012761  000000G 000000G         MOV     #SBUFWD,Q.WCNT(R1);SET WORD COUNT FOR TRANSFER
     80 002630  012761  002734' 000000G         MOV     #SFRCMP,Q.COMP(R1);SET ADDRESS OF COMPLETION ROUTINE
     81 002636  012600                          MOV     (SP)+,R0
     82                                 ;
     83                                 ;  Start the I/O
     84                                 ;
     85 002640  004767  000000G                 CALL    SYQIO               ;QUEUE THE I/O REQUEST
     86                                 ;
     87                                 ;  Finished
     88                                 ;
     89 002644                          SREX:   ENABL                       ;*** ENABLE ***
     90 002652  012602                  SRX:    MOV     (SP)+,R2
     91 002654  012601                          MOV     (SP)+,R1
     92 002656  000207                          RETURN
```

```
                                        .SBTTL  SPOOLQ -- Put spooler in wait queue
                                ;-------------------------------------------------------
                                ;  SPOOLQ IS CALLED TO ADD A SPOOLED DEVICE CONTROL BLOCK
                                ;  TO THE END OF SOME QUEUE WHOSE LIST HEAD IS POINTED
                                ;  TO BY R1.  WHEN CALLED R0 MUST CONTAIN THE ADDRESS
                                ;  OF THE SDCB.
                                ;  R1 IS DESTROYED -- ALL OTHER REGISTERS ARE PRESERVED.
                                ;
002660  010246                  SPOOLQ: MOV     R2,-(SP)
002662                                  DISABL                  ;** DISABLE **
002670  011102                          MOV     (R1),R2         ;GET CONTENTS OF LIST HEAD
002672  001410                          BEQ     1$              ;BRANCH IF LIST IS EMPTY
                                ;  LIST NOT EMPTY -- PUT US AT TAIL
002674  016201  000000G         3$:     MOV     SDWLST(R2),R1   ;LOOK FOR END OF LIST
002700  001402                          BEQ     2$
002702  010102                          MOV     R1,R2           ;FOLLOW CHAIN
002704  000773                          BR      3$
002706  010062  000000G         2$:     MOV     R0,SDWLST(R2)   ;ADD US TO END OF LIST
002712  000401                          BR      4$
                                ;  EMPTY LIST -- PUT US AT HEAD.
002714  010011                  1$:     MOV     R0,(R1)         ;STORE INTO LIST HEAD
002716  005060  000000G         4$:     CLR     SDWLST(R0)      ;SAY WE ARE TAIL OF LIST
002722                                  ENABL                   ;** ENABLE **
002730  012602                          MOV     (SP)+,R2
002732  000207                          RETURN
```

```
                                        .SBTTL  SFRCMP -- Spool read completion routine
                                ;-------------------------------------------------
                                ;  SPOOL READ COMPLETION ROUTINE -- ENTERED WHEN A READ
                                ;  FROM THE SPOOL FILE FINISHES.
                                ;
                                ;  Inputs:
                                ;   R1 = Address of SDCB.
                                ;
                                ;
                                ;  ADD THE DISK ADDRESS OF THE BLOCK JUST READ TO THE END
                                ;  OF THE LIST OF BLOCKS WHICH ARE BEING REMEMBERED IN CASE
                                ;  THE OPERATOR WANTS TO BACK UP IN THE SPOOL FILE.
                                ;  SEE IF THERE IS ROOM LEFT FOR ANOTHER BLOCK ADDRESS.
                                ;
002734  010100                  SFRCMP: MOV     R1,R0           ;MOVE SDCB ADDRESS TO R0
002736  005760  000000G                 TST     SDBULS(R0)      ;IS TABLE FULL?
002742  001412                          BEQ     1$              ;BR IF TABLE NOT FULL
                                ;  TABLE IS FULL.  FORGET OLDEST ENTRY.
002744  016001  000000G                 MOV     SDBU(R0),R1     ;GET OLDEST DISK ADDRESS IN TABLE
002750  004767  001544                  CALL    RLSBLK          ;RELEASE THE BLOCK
                                ;  SHOVE UP ALL TABLE ENTRIES
002754  010001                          MOV     R0,R1           ;POINT INTO TABLE
002756  062701  000002G                 ADD     #SDBU+2,R1
002762  012161  177774          2$:     MOV     (R1)+,-4(R1)    ;SHIFT UP OVER 1ST ENTRY
002766  001375                          BNE     2$              ;LOOP UNTIL ALL MOVED
                                ;  ADD NEW ENTRY TO 1ST FREE SLOT IN TABLE
002770  010001                  1$:     MOV     R0,R1
002772  062701  000000G                 ADD     #SDBU,R1
002776  005721                  3$:     TST     (R1)+           ;IS THIS SLOT EMPTY?
003000  001376                          BNE     3$              ;SEARCH FOR 1ST FREE SLOT
003002  016061  000000G 177776          MOV     SDFLNK(R0),-2(R1);REMEMBER THIS DISK ADDRESS
003010  016001  000000G                 MOV     SDSFCB(R0),R1   ;POINT TO SFCB
003014  005361  000000G                 DEC     SFNMBL(R1)      ;SAY 1 LESS BLOCK IN FILE
003020  017060  000000G 000000G         MOV     @SDBUF2(R0),SDFLNK(R0)  ;SET NEW DISK FLINK
003026  042760  000000G 000000G         BIC     #SD$INR,SDFLAG(R0) ;SAY READ IS FINISHED
                                ;  SEE IF A WRITE TO SPOOLED DEVICE IS IN PROGRESS.
                                ;  IF NOT, START ONE.
003034  005760  000000G                 TST     SDBUF1(R0)      ;OUTGOING BUFFER SET UP?
003040  001002                          BNE     9$              ;BRANCH IF YES
003042  004767  000002                  CALL    SPOLWR          ;START WRITE
003046  000207                  9$:     RETURN
```

```
                                        .SBTTL  SPOLWR -- Write to spooled device
                                ;-----------------------------------------------
                                ;  SPOLWR IS CALLED TO START A WRITE OPERATION TO A
                                ;  SPOOLED DEVICE.  WHEN CALLED R0 MUST POINT TO
                                ;  THE SDCB FOR THE DEVICE.  ALL REGISTERS ARE PRESERVED.
                                ;
003050  010146                  SPOLWR: MOV     R1,-(SP)
003052  010246                          MOV     R2,-(SP)
003054                                  DISABL                  ;** DISABLE **
003062  005760  000000G                 TST     SDBUF1(R0)      ;IS WRITE IN PROGRESS NOW?
003066  001102                          BNE     8$              ;BRANCH IF YES
003070  016002  000000G                 MOV     SDBUF2(R0),R2   ;IS A BUFFER READY TO GO?
003074  001466                          BEQ     7$              ;BRANCH IF NOT
003076  032760  000000G 000000G         BIT     #SD$INR,SDFLAG(R0) ;IS INPUT READ STILL GOING?
003104  001073                          BNE     8$              ;BRANCH IF YES
                                ;  THERE IS A BUFFER READY TO GO.  START WRITE.
003106  010260  000000G                 MOV     R2,SDBUF1(R0)   ;SET ADDR OF OUTPUT BUFFER
003112  005060  000000G                 CLR     SDBUF2(R0)      ;SAY NO INPUT BUFFER
003116                                  ENABL                   ;** ENABLE **
003124  032760  000000G 000000G         BIT     #SD$DEL,SDFLAG(R0) ;ARE WE DELETING THIS FILE?
003132  001055                          BNE     1$              ;BRANCH IF YES
003134  005760  000000G                 TST     SDSKIP(R0)      ;ARE WE SKIPPING FORWARD?
003140  001403                          BEQ     3$              ;BR IF NOT
003142  005360  000000G                 DEC     SDSKIP(R0)      ;DEC REMAINING SKIP COUNT
003146  000447                          BR      1$              ;SKIP THIS BLOCK
003150  005762  000002          3$:     TST     2(R2)           ;IS WORD COUNT IN BUFFER 0?
003154  001444                          BEQ     1$              ;BRANCH IF ZERO
                                ;
                                ;  Write this block to the spooled device.
                                ;
003156  010046                  2$:     MOV     R0,-(SP)        ;SAVE SDCB ADDRESS
                                ;
                                ;  Get a system I/O queue entry.
                                ;
003160  010001                          MOV     R0,R1           ;POINT TO SDCB
003162  062701  000000G                 ADD     #SDCHAN,R1      ;POINT TO SPOOLED DEVICE CHANNEL BLOCK IN SDCB
003166  004767  000000G                 CALL    GETSYQ          ;GET SYSTEM I/O QUEUE ELEMENT
                                ;
                                ;  Set up I/O queue element for the write
                                ;
003172  011600                          MOV     (SP),R0         ;GET ADDRESS OF SDCB
003174  066061  000000G 000000G         ADD     SDBLK(R0),Q.BLKN(R1);SET BLOCK NUMBER
003202  005260  000000G                 INC     SDBLK(R0)       ;ADVANCE BLOCK NUMBER
003206  010200                          MOV     R2,R0           ;GET BUFFER ADDRESS
003210  062700  000004                  ADD     #4,R0           ;SKIP OVER BUFFER HEADER
003214  004767  000000G                 CALL    SYBFAD          ;SET BUFFER ADDRESS IN I/O QUEUE ELEMENT
003220  016200  000002                  MOV     2(R2),R0        ;GET # WORDS TO WRITE
003224  005400                          NEG     R0              ;NEGATIVE ==> WRITE OPERATION
003226  010061  000000G                 MOV     R0,Q.WCNT(R1)
003232  012761  003310' 000000G         MOV     #SWCMP,Q.COMP(R1);SET ADDRESS OF COMPLETION ROUTINE
003240  011661  000000G                 MOV     (SP),Q.CHAN(R1) ;PUT SDCB ADDRESS IN R1 FOR COMPL ROUTINE
                                ;
                                ;  Start the write
                                ;
003244  004767  000000G                 CALL    SYQIO           ;QUEUE THE I/O REQUEST
003250  012600                          MOV     (SP)+,R0        ;GET BACK SDCB ADDRESS
                                ;  TRY TO START READ OF NEXT BLOCK FROM FILE.
```

```
      58 003252                              7$:     ENABL                   ;** ENABLE **
      59 003260  004767  177064                      CALL    SPOLRD          ;TRY TO DO READ
      60 003264  000406                              BR      9$
      61                                    ;  WE ARE DELETING THIS FILE SO IGNORE THIS BLOCK AND READ NEXT
      62                                    ;  FAKE CALL TO WRITE COMPLETION ROUTINE.
      63 003266  010001                      1$:     MOV     R0,R1           ;GET SDCB ADDRESS
      64 003270  004767  000014                      CALL    SWCMP           ;CALL WRITE COMPL ROUTINE
      65 003274                              8$:     ENABL                   ;** ENABLE **
      66 003302  012602                      9$:     MOV     (SP)+,R2
      67 003304  012601                              MOV     (SP)+,R1
      68 003306  000207                              RETURN
```

```
                                        .SBTTL  SWCMP  -- Spool write completion routine
                                ;-------------------------------------------------
                                ;  SWCMP IS THE COMPLETION ROUTINE WHICH IS ENTERED
                                ;  ON COMPLETION OF A WRITE TO A SPOOLED DEVICE.
                                ;  R1 IS DESTROYED (ALLOWED FOR COMPL ROUTINES).
                                ;  ALL OTHER REGISTERS ARE PRESERVED.
                                ;  Inputs:
                                ;   R1 = Address of SDCB
                                ;
003310  010100                  SWCMP:  MOV     R1,R0           ;MOVE SDCB ADDRESS TO R0
003312  016001  000000G                 MOV     SDBUF1(R0),R1   ;POINT TO SPOOL BUFFER
003316  011146                          MOV     (R1),-(SP)      ;SAVE DISK FLINK FROM BUFFER
003320  004767  001310                  CALL    RLSBUF          ;RELEASE SPOOL BUFFER
003324  005060  000000G                 CLR     SDBUF1(R0)      ;SAY WRITE IS FINISHED
003330  005726                          TST     (SP)+           ;IS FILE FLINK ZERO?
003332  001403                          BEQ     5$              ;BRANCH IF ZERO -- END OF FILE
                                ;  THIS IS NOT THE END OF THE FILE.
                                ;  TRY TO START NEXT WRITE.
003334  004767  177510                  CALL    SPOLWR          ;TRY FOR WRITE
003340  000207                          RETURN
                                ;  THIS IS THE END OF THE FILE.
003342  032760  000000G 000000G 5$:     BIT     #SD$DEL,SDFLAG(R0);ARE WE DELETING THE FILE?
003350  001402                          BEQ     8$              ;BR IF NOT
                                ;  WE ARE FINISHED DELETING A FILE.
                                ;  OUTPUT A CR-LF TO MAKE SURE THE PRINTER BUFFER GETS EMPTIED.
003352  004767  000104                  CALL    SWCRLF          ;WRITE CR-LF TO SPOOLED DEVICE
                                ;  Call routine to see if we need to signal EOF to device handler
003356  012701  000000G         8$:     MOV     #DF$CLS,R1      ;Get .CLOSE special function code
003362  004767  000162                  CALL    SDFUNC          ;See if we need to signal EOF to device
                                ;  RELEASE ANY DISK BLOCKS THAT WE ARE HOLDING FOR BACKUP
003366  010246                          MOV     R2,-(SP)
003370  010002                          MOV     R0,R2           ;POINT TO CELLS HOLDING ADDRESSES
003372  062702  000000G                 ADD     #SDBU,R2
003376  012201                  6$:     MOV     (R2)+,R1        ;GET SAVED DISK ADDRESS
003400  001405                          BEQ     7$              ;BR IF REACHED END OF LIST
003402  005062  177776                  CLR     -2(R2)          ;CLEAR THE TABLE
003406  004767  001106                  CALL    RLSBLK          ;RELEASE THE DISK BLOCK
003412  000771                          BR      6$
003414  012602                  7$:     MOV     (SP)+,R2
                                ;  RELEASE DEVICE AND FILE.
003416  016001  000000G         1$:     MOV     SDSFCB(R0),R1   ;POINT TO ACTIVE SFCB
003422  042760  000000G 000000G         BIC     #SD$CLR,SDFLAG(R0);CLEAR SOME DEVICE CONTROL FLAGS
003430  005060  000000G                 CLR     SDSKIP(R0)      ;FINISHED SKIPPING
003434  005060  000000G                 CLR     SDBLK(R0)
003440  005060  000000G                 CLR     SDSFCB(R0)
003444  005267  000000G                 INC     NESB            ;ONE MORE EXTRA BUFFER AVAILABLE
                                ;  Free the SFCB
003450  004767  000504                  CALL    FRESFB          ;Free the SFCB
                                ;  TRY TO START UP NEXT FILE ON THIS DEVICE
003454  004767  176064          4$:     CALL    SPOLGO          ;TRY TO START SPOOLER
003460  000207                          RETURN
```

```
                                        .SBTTL  SWCRLF -- Write cr-lf to spooled device
                                ;-------------------------------------------------------------------
                                ;  SWCRLF is called to write carriage-return, line-feed to a spooled device.
                                ;
                                ;  Inputs:
                                ;   R0 = Address of SDCB for the device.
                                ;
                                ;  Outputs:
                                ;   R0 is preserved.
                                ;
003462  010046                  SWCRLF: MOV     R0,-(SP)
003464  010146                          MOV     R1,-(SP)
003466  010246                          MOV     R2,-(SP)
003470  010002                          MOV     R0,R2           ;GET ADDRESS OF SDCB
                                ;
                                ;  Get system I/O queue element
                                ;
003472  010001                          MOV     R0,R1           ;GET ADDRESS OF SDCB
003474  062701  000000G                ADD     #SDCHAN,R1      ;POINT TO CHANNEL BLOCK FOR SPOOLED DEV IN SDCB
003500  004767  000000G                CALL    GETSYQ          ;GET SYSTEM I/O QUEUE
                                ;
                                ;  Set up queue element for the write
                                ;
003504  062761  000001  000000G        ADD     #1,Q.BLKN(R1)   ;SAY BLOCK # = 1
003512  012700  000010'                MOV     #CRTXT,R0       ;GET ADDRESS OF BUFFER TO BE WRITTEN
003516  004767  000000G                CALL    SYBFAD          ;STORE BUFFER ADDRESS IN I/O QUEUE ELEMENT
003522  012761  177777  000000G        MOV     #-1,Q.WCNT(R1)  ;SET WORD COUNT ( - ==> WRITE)
003530  005061  000000G                CLR     Q.COMP(R1)      ;NO COMPLETION ROUTINE
                                ;
                                ;  Start the I/O
                                ;
003534  004767  000000G                CALL    SYQIO           ;QUEUE THE I/O REQUEST
                                ;
                                ;  Finished
                                ;
003540  012602                          MOV     (SP)+,R2
003542  012601                          MOV     (SP)+,R1
003544  012600                          MOV     (SP)+,R0
003546  000207                          RETURN
```

```
                                        .SBTTL  SDFUNC -- Send special function code to spooled device
                                ;-------------------------------------------------------------------------
                                ;  SDFUNC is called to simulate a .SPFUN special function operation to
                                ;  the spooled device.  This is used to simulate a .ENTER or .CLOSE
                                ;  operation to devices that accept these functions.
                                ;
                                ;  Inputs:
                                ;   R1 = .SPFUN function code to be sent to handler.
                                ;   R0 = Address of SDCB
                                ;
003550  010046                  SDFUNC: MOV     R0,-(SP)
003552  010246                          MOV     R2,-(SP)
003554  010102                          MOV     R1,R2           ;Carry function code in R2
                                ;
                                ;  See if spooled device is directory structured and can accept .SPFUN
                                ;  requests.
                                ;
003556  116001  000000G                 MOVB    SDDVU(R0),R1    ;Get index number for spooled device
003562  032761  000000G 000000G         BIT     #DS$NRD,DVSTAT(R1) ;Special directory structured device?
003570  001004                          BNE     1$              ;Br if yes
003572  032761  000000G 000000G         BIT     #DX$NRD,DVFLAG(R1) ;Hidden special dir dev flag set?
003600  001411                          BEQ     9$              ;Br if not
                                ;
                                ;  Send special function to handler
                                ;
003602  010001                  1$:     MOV     R0,R1           ;Get address of SDCB
003604  062701  000000G                 ADD     #SDCHAN,R1      ;Point to channel block in SDCB
003610  004767  000000G                 CALL    GETSYQ          ;Get an I/O queue element
003614  110261  000000G                 MOVB    R2,Q.FUNC(R1)   ;Set .SPFUN function code
003620  004767  000000G                 CALL    SYQIO           ;Queue the operation for the handler
                                ;
                                ;  Finished
                                ;
003624  012602                  9$:     MOV     (SP)+,R2
003626  012600                          MOV     (SP)+,R0
003630  000207                          RETURN
```

```
                                        .SBTTL  SDCLOS -- Close spooled channel
                                ;-----------------------------------------------
                                ;  SDCLOS IS CALLED TO SEE IF A CHANNEL WHICH IS BEING
                                ;  CLOSED IS ASSOCIATED WITH A SPOOLED DEVICE.
                                ;
                                ;  Inputs:
                                ;   CHNNUM = User channel number.
                                ;   CHNADR = Address of current channel block.
                                ;   CORUSR = Job index number
                                ;
003632  010146                  SDCLOS: MOV     R1,-(SP)
003634  010246                          MOV     R2,-(SP)
003636  016701  000000G                 MOV     CHNADR,R1       ;GET ADDRESS OF CURRENT CHANNEL BLOCK
003642  032761  000000G 000000G         BIT     #CS$SPL,C.CSW(R1);IS CHANNEL OPENED TO SPOOLED DEVICE?
003650  001427                          BEQ     9$              ;BR IF NOT
003652  004767  001044                  CALL    FNDSD           ;IS THIS A SPOOLED CHANNEL?
003656  103424                          BCS     9$              ;BRANCH IF NOT
                                ;  THIS IS A SPOOLED CHANNEL -- CLOSE OUT THE FILE.
003660  010046                          MOV     R0,-(SP)
003662  016100  000000G                 MOV     SFSDCB(R1),R0   ;GET ADDRESS OF SDCB
                                ;  WRITE BLOCK WITH FLINK = 0 TO SIGNAL END OF FILE
003666  004767  175340                  CALL    SGETBF          ;MAKE SURE WE'VE GOT BUFFER
003672  005067  000000G                 CLR     SPUBUF          ;SET FILE FLINK = 0
003676  004767  175364                  CALL    SFLUSH          ;WRITE OUT BUFFER
003702  005067  000000G                 CLR     SXSFCB          ;SAY BUFFER IS FREE
003706  112761  177777  000000G         MOVB    #-1,SFCHAN(R1)  ;DISASSOCIATE FILE AND CHANNEL
                                ;  TRY TO START SPOOLER
003714  142761  000000G 000000G         BICB    #SF$HLD,SFFLAG(R1) ;SAY FILE IS FREE TO START
003722  004767  175616                  CALL    SPOLQ0          ;TRY TO START SPOOLER
003726  012600                          MOV     (SP)+,R0
003730  012602                  9$:     MOV     (SP)+,R2
003732  012601                          MOV     (SP)+,R1
003734  000207                          RETURN
```

```
                                        .SBTTL  SPLDEL -- Delete a spool file
                                ;-------------------------------------------------------------------------
                                ;  SPLDEL is called to delete a spool file.
                                ;  There are three cases to contend with:
                                ;   1. Spool file is currently being processed by spooler.
                                ;      -- Set SD$DEL flag to cause spooler to delete rest of file.
                                ;   2. Spool file is still open.
                                ;      -- Set SF$DEL flag to cause spooler to delete file when it is closed.
                                ;   3. Spool file is closed but spooler hasn't started working on it yet.
                                ;      -- Delete the spool file immediately.
                                ;
                                ;  Inputs:
                                ;   R3 = ID number of spool file to delete.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Spool file found and taken care of.
                                ;   C-flag set     ==> Could not find file with specified ID.
                                ;
003736  010146                  SPLDEL: MOV     R1,-(SP)
003740  010246                          MOV     R2,-(SP)
003742  010346                          MOV     R3,-(SP)
                                ;
                                ;  Try to find the spool file with the specified ID
                                ;
003744  012700  000000G                 MOV     #SDCB,R0        ;Point to first SDCB
003750  020027  000000G         5$:     CMP     R0,#SDCBND      ;Checked all?
003754  103014                          BHIS    6$              ;Br if yes -- could not find file
003756  016002  000000G                 MOV     SDFHD(R0),R2    ;Get address of 1st SFCB for this device
003762  001406                          BEQ     7$              ;Br if none
003764  020362  000000G         8$:     CMP     R3,SFID(R2)     ;Is this the SFCB for the correct file?
003770  001410                          BEQ     10$             ;Br if yes
003772  016202  000000G                 MOV     SFQLNK(R2),R2   ;Get address of next SFCB for this device
003776  001372                          BNE     8$              ;Loop if more to check
004000  062700  000000G         7$:     ADD     #SDCBSZ,R0      ;Point to next SDCB
004004  000761                          BR      5$              ;Go check it
                                ;
                                ;  Error: We cannot find a spool file with the specified ID
                                ;
004006  000261                  6$:     SEC                     ;Signal error on return
004010  000457                          BR      9$
                                ;
                                ;  Found spool file with the specified ID
                                ;    R0 = SDCB address
                                ;    R2 = SFCB address
                                ;  Set flag in SFCB saying file is to be deleted
                                ;
004012  152762  000000G 000000G 10$:    BISB    #SF$DEL,SFFLAG(R2) ;Set delete-file flag
                                ;
                                ;  If file is currently being processed by the spooler, set the
                                ;  delete-file flag for the spooler.
                                ;
004020  020260  000000G                 CMP     R2,SDSFCB(R0)   ;Is this file the active one for this device?
004024  001004                          BNE     1$              ;Br if not
004026  052760  000000G 000000G        BIS     #SD$DEL,SDFLAG(R0) ;Tell spooler to delete the file
004034  000444                          BR      11$
                                ;
                                ;  File is not currently being processed by the spooler.
```

```
                                        ;  See if file has been closed yet.
                                        ;
004036  126227  0000006 177777  1$:      CMPB    SFCHAN(R2),#-1  ;Is file still open?
004044  001040                          BNE     11$             ;Br if yes -- spooler will delete it later
                                        ;
                                        ;  File is closed and is not yet being processed by the spooler.
                                        ;  We will delete the file and free its blocks.
                                        ;
004046  152762  0000006 0000006         BISB    #SF$HLD,SFFLAG(R2) ;Set flag so spooler won't start on it
                                        ;
                                        ;  Get number of 1st block in this spool file
                                        ;
004054  016201  0000006                 MOV     SFSTRT(R2),R1   ;Get # of 1st block in spool file
                                        ;
                                        ;  Read in 1st word of the block to get the flink to the next block
                                        ;
004060  010046                  2$:     MOV     R0,-(SP)        ;Save SDCB address
004062                                  .READW  #SPLARG,#USPLCH,#SPDLBF,#1.,R1 ;Read 1st word from block
004120  012600                          MOV     (SP)+,R0        ;Recover SDCB address
                                        ;
                                        ;  Free the block
                                        ;
004122  004767  000372                  CALL    RLSBLK          ;Release this block
                                        ;
                                        ;  Get number of next block in file
                                        ;
004126  016701  0000006                 MOV     SPDLBF,R1       ;Get number of next block
004132  010162  0000006                 MOV     R1,SFSTRT(R2)   ;Set as starting block in case we are aborted
004136  001350                          BNE     2$              ;Loop if more blocks to delete
                                        ;
                                        ;  We have finished freeing all of the blocks in the file.
                                        ;  Release the SFCB.
                                        ;
004140  010201                          MOV     R2,R1           ;Get address of SFCB to R1 for FRESFB
004142  004767  000012                  CALL    FRESFB          ;Free the SFCB
                                        ;
                                        ;  We successfully deleted the file
                                        ;
004146  000241                  11$:    CLC                     ;Signal success on return
                                        ;
                                        ;  Finished
                                        ;
004150  012603                  9$:     MOV     (SP)+,R3
004152  012602                          MOV     (SP)+,R2
004154  012601                          MOV     (SP)+,R1
004156  000207                          RETURN
```

```
                                        .SBTTL  FRESFB -- Release a SFCB
                                ;-----------------------------------------------------------------------
                                ;  FRESFB is called to release a SFCB after we have finished processing
                                ;  the spool file.
                                ;
                                ;  Inputs:
                                ;   R1 = Address of the SFCB
                                ;
004160  010046                  FRESFB: MOV     R0,-(SP)
004162  010246                          MOV     R2,-(SP)
                                ;
                                ;  Get address of associated SDCB
                                ;
004164  016100  000000G                 MOV     SFSDCB(R1),R0
                                ;
                                ;  Clean out some cells in the SFCB
                                ;
004170  105061  000000G                 CLRB    SFUSER(R1)      ;No associated user
004174  005061  000000G                 CLR     SFSDCB(R1)      ;No associated SDCB

                                ;
                                ;  Remove this SFCB from the list associated with the SDCB
                                ;  and put the SFCB on the free list.
                                ;
004200  010002                          MOV     R0,R2           ;Point to SDCB
004202  062702  000000G                 ADD     #SDFHD,R2       ;Point to list head for this SDCB
004206                                  DISABL                  ;** Disable interrupts **
004214  020112                  1$:     CMP     R1,(R2)         ;Is next entry the one we want?
004216  001404                          BEQ     2$              ;Br if yes
004220  011202                          MOV     (R2),R2         ;Chain to next one
004222  062702  000000G                 ADD     #SFQLNK,R2      ;Point to flink field
004226  000772                          BR      1$              ;Go check next one
004230  016112  000000G         2$:     MOV     SFQLNK(R1),(R2) ;Remove SFCB from chain
004234  016761  000000G 000000G         MOV     SFCBFH,SFQLNK(R1);Add SFCB to free list
004242  010167  000000G                 MOV     R1,SFCBFH
004246  020167  000000G                 CMP     R1,SDSFCB       ;Is our SFCB the current SFCB?
004252  001002                          BNE     3$              ;Br if not
004254  005067  000000G                 CLR     SDSFCB          ;Say no current SFCB
004260                          3$:     ENABL                   ;** Enable interrupts **
                                ;
                                ;  Restart any users waiting for a free SFCB
                                ;
004266  012700  000000G                 MOV     #S$SPCB,R0      ;Get wait state
004272  004767  000000G                 CALL    UREGO           ;Restart waiting users
                                ;
                                ;  Finished
                                ;
004276  012602                          MOV     (SP)+,R2
004300  012600                          MOV     (SP)+,R0
004302  000207                          RETURN
```

```
                                        .SBTTL  GTSBLK -- Get free spool file disk block
                                ;-------------------------------------------------------------------------
                                ;  GTSBLK IS CALLED TO GET A FREE SPOOL FILE DISK BLOCK.
                                ;  IF NONE ARE AVAILABLE THE USER IS SUPENDED UNTIL
                                ;  ONE IS RELEASED.  WHEN CALLED, R1 MUST POINT TO
                                ;  THE SFCB.  ON RETURN THE FREE DISK BLOCK NUMBER
                                ;  IS IN R0.  ALL OTHER REGISTERS ARE PRESERVED.
                                ;
004304  010246                  GTSBLK: MOV     R2,-(SP)
004306  010346                          MOV     R3,-(SP)
004310                          5$:     DISABL                  ;** DISABLE **
                                ;  SEE IF THERE ARE ANY FREE PUBLIC SPOOL BLOCKS.
004316  005767  000000G                 TST     NFRESB          ;ANY FREE PUBLIC BLOCKS?
004322  001017                          BNE     6$              ;IF YES THEN GO USE ONE
                                ;  THERE ARE NO FREE PUBLIC SPOOL BLOCKS.
                                ;  EACH SPOOL DEVICE HAS A SMALL NUMBER OF PRIVATE BLOCKS.
                                ;  SEE IF WE CAN FIND A FREE PRIVATE BLOCK TO USE.
004324  016100  000000G                 MOV     SFSDCB(R1),R0   ;POINT TO SDCB
004330  016002  000000G                 MOV     SDFRBL(R0),R2   ;GET # FREE PRIVATE BLOCKS
004334  001450                          BEQ     1$              ;GIVE UP IF NONE AVAIL
                                ;  SAVE THE LAST FEW PRIVATE BLOCKS FOR ACTIVE FILES.
004336  020227  000002G                 CMP     R2,#<SNBUX+2>   ;ENOUGH FOR INACTIVE FILES?
004342  101004                          BHI     9$              ;BR IF YES
004344  132761  000000G 000000G         BITB    #SF$BSY,SFFLAG(R1);IS THIS FILE ACTIVE?
004352  001441                          BEQ     1$              ;CAN'T GET BLOCK IF INACTIVE
004354  005360  000000G          9$:     DEC     SDFRBL(R0)      ;USE A PRIVATE BLOCK
004360  000402                          BR      8$
                                ;  THERE IS A FREE BLOCK -- FIND IT.
004362  005367  000000G          6$:     DEC     NFRESB          ;SAY ONE LESS BLOCK THERE
004366  016700  173410           8$:     MOV     GRNTBL,R0       ;POINT TO BIT MAP TABLE
004372  005720                   7$:     TST     (R0)+           ;ANY FREE BLOCKS IN THIS WORD?
004374  001776                          BEQ     7$              ;BRANCH IF NOT
004376  014002                          MOV     -(R0),R2        ;GET BITS FROM WORD
004400  010046                          MOV     R0,-(SP)        ;SAVE WORD LOCATION
004402  166700  173374                 SUB     GRNTBL,R0       ;GET BYTE INDEX
004406  072027  000003                 ASH     #3,R0           ;CONVERT BYTE TO BIT INDEX
004412  005003                          CLR     R3              ;COUNT BITS
004414  006102                   3$:     ROL     R2              ;SEARCH FOR FREE GRAN BIT
004416  103402                          BCS     2$              ;BRANCH IF FREE BIT FOUND
004420  005203                          INC     R3              ;KEEP TRACK OF ADDRESS
004422  000774                          BR      3$
004424  060300                   2$:     ADD     R3,R0           ;GET FINAL BLOCK ADDRESS
004426  000241                          CLC                     ;MARK BLOCK AS USED NOW
004430  006002                   4$:     ROR     R2
004432  005303                          DEC     R3
004434  002375                          BGE     4$
004436  010236                          MOV     R2,@(SP)+       ;PUT BACK BIT MAP WORD
004440                                  ENABL                   ;** ENABLE **
004446  005200                          INC     R0              ;1ST BLOCK IN FILE IS # 1
004450  012603                          MOV     (SP)+,R3
004452  012602                          MOV     (SP)+,R2
004454  000207                          RETURN
                                ;  THERE ARE NO FREE BLOCKS -- SUSPEND USER UNTIL ONE IS RELEASED.
                                ;  DON'T HOLD FILE FOR CHANNEL CLOSE IF FILE IS FULL.
004456  132761  000000G 000000G  1$:     BITB    #SF$HLD,SFFLAG(R1)      ;IS FILE BEING HELD?
004464  001406                          BEQ     10$             ;BR IF NOT
004466  142761  000000G 000000G         BICB    #SF$HLD,SFFLAG(R1);RELEASE THE FILE
```

```
      58 004474  004767  175044                    CALL    SPOLGO          ;TRY TO START PRINTING FILE
      59 004500  000703                            BR      5$              ;NOW GO TRY TO GET SPACE AGAIN
      60                                    ;  SUSPEND USER UNTIL A BLOCK IS FREED
      61 004502  012700  000000G            10$:    MOV     #S$SPDB,R0      ;QUEUE FOR SPOOL BLOCKS
      62 004506  004767  000000G                    CALL    QNSPND          ;SUSPEND USER  ** ENABLE **
      63 004512  004767  000000G                    CALL    CHKABT          ;SEE IF WE'VE BEEN ABORTED
      64 004516  000674                            BR      5$
```

```
                                        .SBTTL  RLSBLK -- Free spool file disk block
                                ;-------------------------------------------------------
                                ;  RLSBLK IS CALLED TO RELEASE A SPOOL FILE DISK BLOCK.
                                ;  WHEN CALLED, R0 MUST POINT TO THE SDCB.
                                ;  WHEN CALLED R1 MUST CONTAIN THE SPOOL FILE BLOCK NUMBER.
                                ;  THE FIRST USER WAITING FOR A BLOCK IS RESTARTED BY RLSBLK.
                                ;  ALL REGISTERS ARE PRESERVED EXCEPT R1.
                                ;
004520  010246                  RLSBLK: MOV     R2,-(SP)
004522  010046                          MOV     R0,-(SP)
004524  005301                          DEC     R1                      ;1ST BLOCK IS # 1
004526  010100                          MOV     R1,R0                   ;SAVE DISK BLOCK #
004530  042700  177760                  BIC     #177760,R0              ;GET # OF BIT IN WORD
004534  005002                          CLR     R2
004536  000261                          SEC                             ;SET BUSY BIT
004540  006002                  1$:     ROR     R2                      ;FORM MASK FOR SINGLE BIT
004542  005300                          DEC     R0
004544  002375                          BGE     1$
004546  072127  177774                  ASH     #-4,R1                  ;GET WORD INDEX
004552  006301                          ASL     R1
004554  066701  173222                  ADD     GRNTBL,R1               ;POINT INTO BIT TABLE
004560  020167  173220                  CMP     R1,GRNEND               ;MAKE SURE STORE IS OK
004564  103401                          BLO     2$                      ;BRANCH IF OK
004566  000000                          HALT                            ;SAFETY STOP
004570  050211                  2$:     BIS     R2,(R1)                 ;MARK BLOCK AS FREE
                                ;  INCREMENT NUMBER OF FREE PUBLIC OR PRIVATE BLOCKS
004572  011600                          MOV     (SP),R0                 ;POINT TO SDCB
004574  026027  000000G 000000G        CMP     SDFRBL(R0),#PVSPBL;ADD FREE BLOCK TO PUBLIC OR PRIVATE?
004602  103003                          BHIS    3$                      ;BR IF PUBLIC
004604  005260  000000G                 INC     SDFRBL(R0)              ;ADD TO PRIVATE
004610  000402                          BR      4$
004612  005267  000000G          3$:    INC     NFRESB                  ;SAY ANOTHER FREE BLOCK
                                ;  RESTART ANY USER WAITING FOR DISK BLOCK
004616  012700  000000G          4$:    MOV     #S$SPDB,R0              ;POINT TO WAIT QUEUE HEAD
004622  004767  000000G                 CALL    UREGO                   ;REACTIVATE 1ST USER
004626  012600                          MOV     (SP)+,R0
004630  012602                          MOV     (SP)+,R2
004632  000207                          RETURN
                                ;
```

```
                                        .SBTTL  RLSBUF -- Release spool buffer
                                ;-------------------------------------------------
                                ;  RLSBUF IS CALLED TO RELEASE A SPOOL CORE BUFFER.
                                ;  WHEN CALLED, R1 MUST POINT TO THE BUFFER.
                                ;  SPOLRD IS CALLED TO RESTART ANY SPOOLERS WHO
                                ;  ARE WAITING FOR A BUFFER.
                                ;  ALL REGISTERS ARE PRESERVED EXCEPT R1.
                                ;
004634  010046                  RLSBUF: MOV     R0,-(SP)
004636                                  DISABL                  ;** DISABLE **
                                ;  RELEASE THE BUFFER
004644  016711  000000G                 MOV     SPLBHD,(R1)     ;LINK INTO FREE CHAIN
004650  010167  000000G                 MOV     R1,SPLBHD
                                ;  RESTART ANY WAITING SPOOL BUFFER REQUESTS.
004654  016700  173126                  MOV     SPBWHD,R0       ;HEAD OF WAIT LIST
004660  001413                          BEQ     1$              ;BR IF NOONE WAITING
004662  016067  000000G 173116          MOV     SDWLST(R0),SPBWHD ;REMOVE 1ST FROM LIST
004670  042760  000000G 000000G         BIC     #SD$BWT,SDFLAG(R0) ;SAY NO LONGER WAITING FOR BUFFER
004676                                  ENABL                   ;** ENABLE **
004704  004767  175440                  CALL    SPOLRD          ;SAY IT CAN DO A READ
004710                          1$:     ENABL                   ;** ENABLE **
004716  012600                          MOV     (SP)+,R0
004720  000207                          RETURN
```

```
                                        .SBTTL  FNDSD  -- Locate SFCB for spooled channel
                                ;-----------------------------------------------------------------------
                                ;  FNDSD is called to locate the SFCB associated with the current I/O channel.
                                ;
                                ;  Inputs:
                                ;   CHNNUM = User channel number.
                                ;   CORUSR = Job index number
                                ;
                                ;  Outputs:
                                ;   C-flag cleared if SFCB found, set otherwise.
                                ;   R1 = Address of SFCB.
                                ;
004722  016701  000000G       FNDSD:  MOV     SFCB,R1         ;POINT TO 1ST SFCB
004726  020167  000000G       4$:     CMP     R1,SFCBND       ;CHECKED ALL SFCB'S?
004732  103013                        BHIS    1$              ;BR IF YES
004734  126761  000000G 000000G       CMPB    CORUSR,SFUSER(R1);IS THIS SFCB IN USE BY THIS USER?
004742  001004                        BNE     2$              ;BR IF NOT
004744  126761  000000G 000000G       CMPB    CHNNUM,SFCHAN(R1);IS SFCB ASSIGNED TO THIS CHANNEL?
004752  001405                        BEQ     3$              ;BR IF YES
004754  062701  000000G       2$:     ADD     #SFCBSZ,R1      ;POINT TO NEXT SFCB
004760  000762                        BR      4$              ;GO CHECK IT
                                ;  Can't find SFCB.
004762  000261                1$:     SEC                     ;SIGNAL ERROR ON RETURN
004764  000401                        BR      5$
                                ;  Found the SFCB.
004766  000241                3$:     CLC                     ;SIGNAL SUCCESS ON RETURN
004770  000207                5$:     RETURN
```

```
                                        .SBTTL  CVTDVU -- Convert device name to dev index and unit #
                                ;-----------------------------------------------------------------------
                                ;  CVTDVU is called to convert a RAD50 device name into the corresponding
                                ;  device index number and unit number.
                                ;
                                ;  Inputs:
                                ;   R1 = RAD50 device name.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Conversion successful.
                                ;   C-flag set     ==> Unable to find device name in tables.
                                ;   R1 = Device index number (low byte), device unit number (high byte).
                                ;
004772  010246                  CVTDVU: MOV     R2,-(SP)
004774  010346                          MOV     R3,-(SP)
                                ;
                                ;  Split the unit number off of the full device name
                                ;
004776  010103                          MOV     R1,R3           ;Get full device name
005000  005002                          CLR     R2              ;Set up for divide
005002  071227  000050                  DIV     #50,R2          ;Split name and unit (R2=name, R3=unit)
005006  005703                          TST     R3              ;Was a unit number specified?
005010  001402                          BEQ     1$              ;Br if not
005012  162703  000036                  SUB     #36,R3          ;Convert unit number to binary value
005016  010301                  1$:     MOV     R3,R1           ;Get unit number
005020  000301                          SWAB    R1              ;Position to high-order byte
                                ;
                                ;  Look up the device name to get the device index
                                ;
005022  070227  000050                  MUL     #50,R2          ;Now get the device name without unit number
005026  016702  0000000G                MOV     NUMDEV,R2       ;Get index number of last device
005032  020362  0000000G        2$:     CMP     R3,PNAME(R2)    ;Search for device in name table
005036  001407                          BEQ     3$              ;Br if found it
005040  162702  000002                  SUB     #2,R2           ;Try next device
005044  002372                          BGE     2$              ;Loop if more to check
                                ;
                                ;  Error, cannot find device name in tables
                                ;
005046  012701  177777                  MOV     #-1,R1          ;Set device # = unit # = -1
005052  000261                          SEC                     ;Signal error on return
005054  000402                          BR      9$
                                ;
                                ;  Found the device in the tables
                                ;
005056  050201                  3$:     BIS     R2,R1           ;Combine device # and unit #
005060  000241                          CLC                     ;Signal success on return
                                ;
                                ;  Finished
                                ;
005062  012603                  9$:     MOV     (SP)+,R3
005064  012602                          MOV     (SP)+,R2
005066  000207                          RETURN
```

```
                                        .SBTTL  EMSPHL -- EMT to set spool HOLD mode
                                ;-------------------------------------------------------------------
                                ;  Kmon EMT to set spool hold mode.
                                ;
                                ;  Inputs:
                                ;   EMT arg block+4 = Address of SDCB for spooled device.
                                ;
005070                          EMSPHL:
                                ;
                                ;  Set hold flag in SDCB
                                ;
005070  016702  000004G                 MOV     EMTBLK+4,R2     ;Point to SDCB
005074  052762  000000G 000000G         BIS     #SD$HLD,SDFLAG(R2);Set hold flag
                                ;
                                ;  Set SF$HLD flag in each spool file
                                ;
005102  016203  000000G                 MOV     SDFHD(R2),R3    ;Point to 1st SFCB for the device
005106  001406                          BEQ     9$              ;Br if there are none
005110  152763  000000G 000000G 1$:     BISB    #SF$HLD,SFFLAG(R3);Set hold flag for file
005116  016303  000000G                 MOV     SFQLNK(R3),R3   ;Get address of next SFCB
005122  001372                          BNE     1$              ;Br if there is another file
                                ;
                                ;  Finished
                                ;
005124  000167  000000G         9$:     JMP     EMTXIT
```

```
                                        .SBTTL  EMSPNH -- EMT to set spool NOHOLD mode
                                ;-------------------------------------------------------------------
                                ;  Kmon EMT to set spool nohold mode.
                                ;
                                ;  Inputs:
                                ;   EMT arg block+4 = Address of SDCB for spooled device.
                                ;
005130                          EMSPNH:
                                ;
                                ;  Reset hold flag in SDCB
                                ;
005130  016702  000004G                 MOV     EMTBLK+4,R2     ;Point to SDCB
005134  042762  000000G 000000G         BIC     #SD$HLD,SDFLAG(R2);Reset hold flag
                                ;
                                ;  Clear SF$HLD flag in each spool file
                                ;
005142  016203  000000G                 MOV     SDFHD(R2),R3    ;Point to 1st SFCB for the device
005146  001411                          BEQ     9$              ;Br if there are none
005150  142763  000000G 000000G 1$:     BICB    #SF$HLD,SFFLAG(R3);Clear hold flag for file
005156  016303  000000G                 MOV     SFQLNK(R3),R3   ;Get address of next SFCB
005162  001372                          BNE     1$              ;Br if there is another file
                                ;
                                ;  Try to start spooler
                                ;
005164  010200                          MOV     R2,R0           ;Get address of SDCB
005166  004767  174352                  CALL    SPOLGO
                                ;
                                ;  Finished
                                ;
005172  000167  000000G         9$:     JMP     EMTXIT
```

```
      1                                                .SBTTL  SFLGPG  -- COPY FLAG PAGE TO SPOOL BUFFER
      2                                          ;-----------------------------------------------------------------
      3                                          ;  SFLGPG copies the contents of the stolen CSI buffer into the
      4                                          ;  spool buffer.
      5                                          ;
      6 005176  010246                           SFLGPG: MOV     R2,-(SP)        ;SAVE R2
      7
      8 005200  012704  000000G                          MOV     #CSIBUF,R4      ;GET BEGINNING OF STOLEN BUFFER
      9 005204  016702  000000G                          MOV     SXBPNT,R2       ;GET POINTER INTO SPOOL BUFFER
     10 005210  020227  000000G                  1$:     CMP     R2,#SPUBND      ;BUFFER FULL?
     11 005214  103406                                   BLO     2$              ;BR IF NOT
     12 005216  010267  000000G                          MOV     R2,SXBPNT       ;SAVE POINTER INTO SPOOL BUFFER
     13 005222  004767  173732                           CALL    SBUFOT          ;WRITE TO SPOOL FILE
     14 005226  016702  000000G                          MOV     SXBPNT,R2       ;RESTORE POINTER INTO SPOOL BUFFER
     15 005232  112422                           2$:     MOVB    (R4)+,(R2)+     ;MOVE INTO SPOOL BUFFER
     16 005234  121427  000200                           CMPB    (R4),#200       ;END?
     17 005240  001363                                   BNE     1$              ;BR IF NOT
     18
     19 005242  010267  000000G                          MOV     R2,SXBPNT       ;SAVE POINTER INTO SPOOL BUFFER
     20 005246  012602                                   MOV     (SP)+,R2        ;RESTORE R2
     21 005250  000207                                   RETURN
     22
     23                                          ;  Top of TSSPOL module
     24                                          ;
     25 005252                                  SPLTOP:
     26         000001                                   .END
```

```
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9631 Words  ( 38 Pages)
Size of core pool: 18176 Words  ( 71 Pages)
Operating  system: RT-11

Elapsed time: 00:00:42.53
,LP:TSSPOL=DK:TSSPOL/C/N:SYM
```

Cross reference table (CREF V05.05)

```
$DILUP    1-58      9-90
. . . V1  7-28      7-28      7-30      18-75     18-75
. . . V2  7-28      7-28      7-28#     7-28#     18-75     18-75     18-75#    18-75#
BELL      1-94#
C.CSW     1-63      3-106*    17-14
CHKABT    1-66      3-50      20-63
CHKSD     1-44      3-21#
CHNADR    1-63      3-105     17-13
CHNNUM    1-76      3-84      23-18
CORUSR    1-52      3-60      3-72      23-16
CR        1-88      1-92#     4-62
CRTXT     1-88#     15-25
CS$SPL    1-62      3-106     17-14
CSIBND    1-52
CSIBUF    1-52      27-8
CTRLTT    1-71      9-88      9-112
CVTDVU    3-25      24-14#
DF$CLS    1-50      14-28
DF$ENT    1-50      9-128
DMYDEV    1-55      3-29
DS$NRD    1-50      16-19
DVFLAG    1-51      16-21
DVSTAT    1-50      16-19
DX$NRD    1-51      16-21
EMSPHL    1-44      25-8#
EMSPNH    1-44      26-8#
EMTBLK    1-51      4-27      4-28      4-42      25-12     26-12
EMTXIT    1-50      25-25     26-30
FILSPC    1-62      3-24      3-39      3-40
FLGPAG    1-59      4-83
FMMSG     1-59      9-96
FNDSD     4-18      8-14      17-16     23-13#
FRESFB    14-48     18-92     19-9#
FREUSR    1-71      3-42
GETSYQ    1-77      10-70     13-37     15-20     16-28
GETUCH    1-76      4-49      4-61      4-71      4-93
GRNEND    1-86#     2-21*     21-22
GRNTBL    1-85#     2-14*     20-30     20-35     21-21
GTSBLK    3-99      5-26      20-9#
GTSYMB    1-54      9-92
INTPRI    1-63      3-58      3-97      9-83      9-126     9-147     10-45     10-89     11-23     13-19     13-58     13-65
          19-39     20-48     22-19     22-21
LF        1-88      1-93#     4-72
LPRG1     1-75      3-73
LPRG2     1-75      3-74
LPROG     1-60
LPROJ     1-60
LSW       1-58      9-90
LUNAME    1-54
NESB      1-59      9-139*    14-46*
NFRESB    1-54      2-57*     20-13     20-29*    21-32*
NSPLBL    1-56      2-15      2-55
NSPLFL    1-56      2-42
NUMDEV    1-51      24-31
OVRHC     1-80      3-42      4-83      9-113
PNAME     1-51      24-32
```

Cross reference table (CREF V05.05)

```
PSW       1-63     3-54*    3-58*    3-88*    3-97*    9-7*     9-83*    9-126*   9-147*   10-13*   10-45*   10-89*
         11-10*   11-23*   13-9*    13-19*   13-58*   13-65*   19-27*   19-39*   20-11*   20-48*   22-10*   22-19*
         22-21*
PVSPBL    1-53     21-28
Q.BLKN    1-77     10-76*   13-42*   15-24*
Q.CHAN    1-78     10-75*   13-51*
Q.COMP    1-78     10-80*   13-50*   15-28*
Q.FUNC    1-50     16-29*
Q.WCNT    1-78     10-79*   13-49*   15-27*
QMSG      1-54      9-113
QNSPND    1-66      3-49    20-62
RLSBLK   12-20     14-37    18-80    21-9#
RLSBUF   14-13     22-9#
RSQBR     1-95#     4-51
S$SPCB    1-64      3-48    19-43
S$SPDB    1-64      9-143   20-61    21-34
SB$TXT    1-54      9-95     9-101
SBUFOT    4-91      5-16#    6-17    27-13
SBUFWD    1-73     10-79
SD$BAK    1-74     10-18    10-49    10-60
SD$BWT    1-59     10-16    10-37    22-18
SD$CLR    1-74     14-42
SD$DEL    1-73      9-133   13-20    14-22    18-54
SD$FLG    1-57      4-81
SD$FLK    1-57      9-57
SD$HLD    1-60      3-81    25-13    26-13
SD$INR    1-72     10-44    12-35    13-14
SD$RSV    1-74      8-10
SD$SMS    1-56      9-78     9-116
SD$SNG    1-56      9-76
SD$WFM    1-57      9-28     9-115
SDANAM    1-57      9-107
SDBLK     1-72      9-135*   9-138*   13-42    13-43*   14-44*
SDBU      1-75     10-54    12-19    12-23    12-28    14-33
SDBUF1    1-70     10-24    12-38    13-10    13-17*   14-11    14-14*
SDBUF2    1-70     10-14    10-43*   12-34    13-12    13-18*
SDBULS    1-75     12-16
SDCB      1-51      3-26    18-25
SDCBND    1-51      3-27    18-26
SDCBSZ    1-73      3-33    18-34
SDCHAN    1-69     13-36    15-19    16-27
SDCLOS    1-44     17-11#
SDDVU     1-61      3-31    16-18
SDFHD     1-72      3-89     9-16     9-35     9-63    18-28    19-26    25-17    26-17
SDFLAG    1-69      3-81     4-81     9-28     9-57     9-76     9-78     9-115*   9-116*   9-133*   10-16    10-18
         10-37*   10-44*   10-49    10-60*   12-35*   13-14    13-20    14-22    14-42*   18-54*   22-18*   25-13*
         26-13*
SDFLNK    1-70      9-134*  10-22    10-55*   10-76    12-31    12-34*
SDFORM    1-57      9-45     9-47     9-49
SDFRBL    1-53     20-19    20-26*   21-28    21-30*
SDFUNC    9-129    14-29    16-11#
SDMOVE    1-46      4-10#
SDMV1     4-37      4-75     4-79     4-82     4-87#
SDMVX     4-19      4-30     4-104#
SDNAME    1-55      3-29
SDSFCB    1-70      9-8      9-124*   10-10    12-32    14-41    14-45*   18-52    19-36    19-38*
```

Cross reference table (CREF V05.05)

```
SDSKIP   1-75     13-22     13-24*    14-43*
SDWLST   1-71     11-14     11-18*    11-22*    22-17
SF$1ST   1-55      4-36      4-48      9-40
SF$BN1   1-52      4-44      4-78      9-136
SF$BSY   1-69      7-35      9-125     20-24
SF$DEL   1-55      9-43      9-131     18-47
SF$HLD   1-60      3-83      8-20      8-21      8-24      9-36      9-66      17-28     18-66     20-55     20-57     25-19
        26-19
SFCB     1-65      2-40      23-13
SFCBFH   1-74      2-41*     3-55      3-57*     19-34     19-35*
SFCBND   1-65     23-14
SFCBSZ   1-55      2-44      23-20
SFCHAN   1-67      3-84*     17-26*    18-60     23-18
SFFILE   1-67      3-69*     3-70*     3-73*     3-74*
SFFLAG   1-67      3-61*     3-83*     4-36      4-44*     4-48*     4-78      7-35      8-19      8-21*     8-24*     9-36
         9-40      9-43      9-66      9-125*    9-131     9-136     17-28*    18-47*    18-66*    20-24     20-55     20-57*
        25-19*    26-19*
SFFLNK   1-68      3-101*    7-28      7-31*
SFFORM   1-56      3-77*     3-78*     3-79*     4-58      9-21      9-45      9-47      9-49      9-102
SFID     1-58      3-67*     18-30
SFLGPG   1-45     27-6#
SFLUSH   5-29      7-11#     17-24
SFNMBL   1-68      3-86*     7-33*     9-19      9-38      9-64      10-20     10-56*    12-33*
SFQLNK   1-69      2-49*     3-57      3-93      3-96*     9-23      9-51      9-68      18-32     19-31     19-33     19-34*
        25-20     26-20
SFRCMP  10-80     12-15#
SFSDCB   1-68      3-85*     4-80      7-34      17-20     19-14     19-19*    20-18
SFSTRT   1-68      3-100*    9-134     18-70     18-85*
SFUSER   1-67      3-60*     19-18*    23-16
SGETBF   4-23      6-11#     17-22
SGFSF    9-50      9-76#
SGNFMT   9-67      9-83#
SGOTFL   9-22      9-42      9-44      9-77      9-79      9-124#
SGX1     9-18      9-30      9-58      9-70      9-120     9-146#
SGX2     9-10      9-147#
SNBUX    1-60      2-56      20-22
SPBWHD   1-87#    10-38      22-15     22-17*
SPDLBF   1-53     18-75      18-84
SPLARG   1-62      7-28      18-75
SPLBHD   1-65      2-28*     10-33     10-42*    22-12     22-13*
SPLCHN   1-62     10-69
SPLDEL   1-46     18-19#
SPLERR   1-59      7-30
SPLINI   1-45      2-7#
SPLNB    1-52      2-26      10-29
SPLND    1-52      2-56      10-29
SPLSHF   1-45      8-9#
SPLTOP   1-46      2-13      27-25#
SPOLGO   1-45      7-39       9-7#     14-50     17-29     20-58     26-26
SPOLID   1-58      3-63       3-65*     3-66*     3-67
SPOLRD   7-37      9-140     10-8#     13-59     22-20
SPOLWR  12-40     13-7#      14-19
SPOOLQ  10-39     11-9#
SPUBND   1-76      4-88      27-10
SPUBUF   1-62      5-22       5-27*     6-20      7-15      17-23*
SREX    10-15     10-17      10-21     10-23     10-30     10-89#
```

Cross reference table (CREF V05.05)

```
SRX      10-11     10-40     10-90#
SWCMP    13-50     13-64     14-10#
SWCRLF   14-26     15-11#
SXBPNT    1-61      4-87      4-90*     4-92      4-100*    5-21      6-20*     7-19      7-42*     27-9      27-12*    27-14
         27-19*
SXSFCB    1-61      5-18      6-11      6-13      6-19*     7-16      17-25*
SYBFAD    1-65     10-78     13-46     15-26
SYQIO     1-78     10-85     13-55     15-32     16-30
TSSPOL    1-13#     1-44
UFORM     1-53      3-77      3-78      3-79
URO       1-76      4-29*     8-10*     8-19*     8-20*
UREGO     1-66      9-144     19-44     21-35
USPLCH    1-64      7-28      7-28      18-75     18-75
```

Cross reference table (CREF V05.05)

...CM1	7-28	18-75										
...CM2	7-28	7-28	7-28	7-28	18-75	18-75	18-75	18-75				
...CM5	7-28	7-30	18-75									
...CM7	7-28	18-75										
.PRINT	1-15#	7-30										
.READW	1-15#	18-75										
.WRITW	1-15#	7-28										
DISABL	1-22#	3-54	3-88	9-7	10-13	11-10	13-9	19-27	20-11	22-10		
ENABL	1-26#	3-58	3-97	9-83	9-126	9-147	10-45	10-89	11-23	13-19	13-58	13-65
	19-39	20-48	22-19	22-21								
OCALL	1-32#	3-42	4-83	9-113								