

Table of contents

2-	1	FLGPAG	-- Print a flag page
3-	1	CNVDEC	-- Convert a number to an ascii decimal number
4-	1	CNVR50	-- Convert a RAD-50 word to ascii
5-	1	BORDER	-- Print a border line of the flag page
6-	1	PRTBUF	-- Print a 7 X 5 line of the flag page
7-	1	GCHROF	-- Get the offset into the character table (CHARS)
8-	1	EDTTOD	-- Edit the time of day
9-	1	DIVIDE	-- Divide 32-bit qty by 16-bit
10-	10	LEAP	-- Function to determine if this is a leap year
11-	1	JULIAN	-- Get day of the year for a given date
12-	1	DOW	-- Get the day of the week
13-	1	EDTDAT	-- Print the current date
14-	1	SETSTN	-- KMON emt to copy site name into overlay
15-	1	GTLICN	-- Get site information EMT
16-	1	SPLSFF	-- Set or reset flagpage flag for device
17-	1	SPLSWF	-- Set or reset wide flagpage flag for device
18-	1	FNDSB	-- Locate SDCB for spooled channel
19-	1		5 X 7 Character tables

```

1          .TITLE  TSSPL2  T/S SPOOLER FLAG-PAGE ROUTINES
2          ;
3          ;   COPYRIGHT (C) 1988
4          ;
5          ;   S&H COMPUTER SYSTEMS, INC.
6          ;   NASHVILLE, TENNESSEE
7          ;
8          ;   THESE ROUTINES IMPLEMENT AUTOMATIC PRODUCTION OF A FLAG PAGE
9          ;   FOR FILES SENT TO SPOOLED DEVICES.
10         ;
11 000000          .CSECT  TSSPL2
12 000000 074540  TSSPL2: .RAD50  /SP2/
13         ;
14         ; MACROS
15         ;
16         000040  SPACE = 40          ; OCTAL VALUE FOR THE SPACE CHAR.
17         ;
18         ;
19         ;   Macro to begin a new character in the bit pattern character table
20         ;
21         .MACRO  BEGINC
22                 .DSABL  CRF
23                 BIT = 1
24                 C1 = 0
25                 C2 = 0
26                 C3 = 0
27                 C4 = 0
28                 C5 = 0
29                 .ENABL  CRF
30         .ENDM   BEGINC
31         ;
32         ;
33         ;   Macro to perform the or operation for the macro to build the bit
34         ;   pattern for a character
35         ;
36         ;
37         .MACRO  DOOR      CNUM
38                 .DSABL  CRF
39                 C'CNUM = C'CNUM ! BIT
40                 .ENABL  CRF
41         .ENDM   DOOR
42         ;
43         ;
44         ;   Macro to build the bit pattern for a character
45         ;
46         ;
47         .MACRO  LINE      PIC
48                 .DSABL  CRF
49                 COL = 0
50                 .IRPC  X, <PIC>
51                 .DSABL  CRF
52                 COL = COL + 1
53                 .IIF  NE, <<'X>-SPACE>  DOOR  \COL
54                 .ENABL  CRF
55                 .ENDR
56                 BIT = BIT#2
57                 .ENABL  CRF

```

```

58          . ENDM LINE
59
60          ;
61          ; Macro to end a character in the bit pattern character table
62          ;
63
64          . MACRO  ENDCHR
65                  . DSABL  CRF
66                  . BYTE   C1
67                  . BYTE   C2
68                  . BYTE   C3
69                  . BYTE   C4
70                  . BYTE   C5
71                  . ENABL  CRF
72          . ENDM ENDCHR
73
74          ; Macro definition to call global routines residing in mapped system regions.
75          ;
76          . MACRO  OCALL  ENTADD
77          . IF    B, ENTADD
78                  . ERROR  ; OCALL SPECIFIED WITH NO ENTRY ADDRESS
79          . MEXIT
80          . ENDC
81          CALL    OVRHC          ; CALL THE OVERLAY HANDLER
82          . WORD  ENTADD        ; SPECIFY THE ENTRY POINT
83          . ENDM
84
85          ; Global references and definitions
86          ;
87          . GLOBL  CSIBUF, CSIBND, SD$WID, SDFLAG, SITNAM, SFUSER, SD$RSV
88          . GLOBL  LUNAME, SFFILE, LPROG, LPROJ, SFLGPG, SYTIMH
89          . GLOBL  SYTIML, TK1SEC, SYSDAT, FLOPAG, EMTXIT, SETSTN
90          . GLOBL  SFCB, SFCBND, CORUSR, SFUSER, CHNNUM, SFCHAN
91          . GLOBL  SFCBSZ, SFSDCB, SDFLAG, SD$WID, SD$FLG, SPLSFF
92          . GLOBL  SPLSWF, TSXSIT, GTLICN, VALADW, EMTBLK, SETC, URO
93          ;

```

FLGPAG -- Print a flag page

```

1
2
3
4
5
6
7
8
9
10 000002 010346
11 000004 010246
12 000006 010446
13 000010 010546
14 000012 010046
15 000014 010146
16
17 000016 012704 000000G
18 000022 032760 000000G 000000G
19 000030 001403
20 000032 012702 000204
21 000036 000402
22 000040 012702 000120
23
24
25
26
27
28
29
30 000044 016705 004322
31 000050 012703 004314'
32 000054 004767 000666
33
34
35
36 000060 011601
37 000062 010603
38 000064 112743 000040
39 000070 162706 000016
40 000074 116100 000000G
41 000100 006300
42 000102 010046
43 000104 006300
44 000106 062600
45 000110 062700 000000G
46 000114 010005
47 000116 062705 000014
48 000122 114543
49 000124 020500
50 000126 101375
51 000130 010305
52 000132 062705 000014
53 000136 124527 000040
54 000142 001003
55 000144 020503
56 000146 001406
57 000150 000772

```

```

.SBTTL FLGPAG -- Print a flag page
-----
;
; FLGPAG causes a flag page to be printed for each file copied to
; spooled device if that device has this feature enabled
;
; Inputs
; R0 = Beginning of the spool device control block (SDCB)
; R1 = Beginning of the spool file control block (SFCB)
;
FLGPAG: MOV R3, -(SP) ;SAVE REGISTERS
MOV R2, -(SP)
MOV R4, -(SP)
MOV R5, -(SP)
MOV R0, -(SP)
MOV R1, -(SP)
MOV #CSIBUF, R4 ;GET ADDRESS OF STOLEN BUFFER
BIT #SD$WID, SDFLAG(R0) ;132 COLUMNS?
BEQ 11$ ;BR IF NOT
MOV #132, R2 ;SET PAGE WIDTH
BR 12$
11$: MOV #80, R2 ;SET PAGE WIDTH
;
; Print the top border with site name
;
12$: MOV #SITNAM, R5 ;GET END OF SITE NAME
13$: CMPB (R5)+, #200 ;END OF NAME?
; BNE 13$ ;BR IF NOT
; SUB #SITNAM+1, R5 ;GET LENGTH OF SITE NAME
12$: MOV SITLEN, R5 ;Get length of site name
MOV #SITNAM, R3 ;GET SITE NAME
14$: CALL BORDER ;PRINT TOP BORDER
;
; Print user name (if known)
;
MOV (SP), R1 ;GET POINTER INTO SFCB
MOV SP, R3 ;POINT TO USER NAME BUFFER
MOVB #' , -(R3) ;SAVE ROOM FOR TERMINATER
SUB #14, SP ;RESERVE THAT BUFFER
MOVB SFUSER(R1), R0 ;GET JOB INDEX #
ASL R0 ;GET POINTER TO USER NAME FOR THIS JOB
MOV R0, -(SP) ;(MULTIPLY JOB INDEX BY 6)
ASL R0
ADD (SP)+, R0
ADD #LUNAME, R0
MOV R0, R5 ;GET END OF USER NAME
ADD #12, R5
1$: MOVB -(R5), -(R3) ;MOVE NAME INTO BUFFER
CMP R5, R0 ;DONE?
BHI 1$ ;BR IF SO
2$: MOV R3, R5 ;GET END OF USER NAME
ADD #12, R5
22$: CMPB -(R5), #' ;IS THIS A SPACE?
; BNE 21$ ;BR IF NOT
CMP R5, R3 ;IS THE WHOLE NAME BLANK?
BEQ 3$ ;BR IF SO
BR 22$

```

```

58 000152 005205          21$:   INC      R5          ; TERMINATE NAME
59 000154 112715 000200   MOVB    #200,(R5)
60 000160 160305          SUB     R3,R5      ; GET NAME LENGTH
61 000162 000401          BR      6$
62 000164 005005          3$:   CLR     R5          ; NO NAME BUT NEED A BLANK LINE (CENTERING)
63 000166 005001          6$:   CLR     R1          ; SAY SINGLE HIGH & WIDE
64 000170 004767 001036   CALL   PRTBUF     ; PRINT USER NAME
65 000174 062706 000016   ADD    #14.,SP    ; FINISHED W/ BUFFER
66
67
68
69 000200 011601          4$:   MOV    (SP),R1    ; GET POINTER INTO SFCB
70 000202 010603          MOV    SP,R3      ; POINT TO FILE NAME BUFFER
71 000204 162706 000016   SUB    #14.,SP    ; RESERVE THAT BUFFER
72 000210 112743 000200   MOVB   #200,-(R3) ; TERMINATE BUFFER
73 000214 010305          MOV    R3,R5      ; GET END OF FILE NAME
74 000216 016100 000002G  MOV    SFFILE+2(R1),R0 ; GET SECOND HALF OF FILE NAME
75 000222 004767 000364   CALL   CNVR50     ; CONVERT TO RAD-50
76 000226 016100 000000G  MOV    SFFILE(R1),R0 ; GET FIRST HALF OF FILE NAME
77 000232 004767 000354   CALL   CNVR50     ; CONVERT TO RAD-50
78 000236 012701 000001   MOV    #1,R1      ; SAY DOUBLE HIGH & WIDE
79 000242 160305          SUB    R3,R5      ; GET LENGTH OF FILE NAME
80 000244 004767 000762   CALL   PRTBUF     ; PRINT FILE NAME
81 000250 062706 000016   ADD    #14.,SP    ; FINISHED W/ BUFFER
82
83
84
85 000254 011601          MOV    (SP),R1    ; GET POINTER INTO SFCB
86 000256 116100 000000G  MOVB   SFUSER(R1),R0 ; GET JOB INDEX #
87 000262 012705 000003   MOV    #3,R5      ; SET MINIMUM STRING LENGTH
88 000266 016001 000000G  MOV    LPROG(R0),R1 ; GET PROGRAMMER NUMBER
89 000272 010603          MOV    SP,R3      ; POINT TO PPN BUFFER
90 000274 162706 000024   SUB    #20.,SP    ; RESERVE THAT BUFFER
91 000300 112743 000200   MOVB   #200,-(R3) ; TERMINATE BUFFER
92 000304 112743 000135   MOVB   #'J,-(R3)  ; END OF STRING
93 000310 010046          MOV    R0,-(SP)   ; SAVE R0
94 000312 004767 000224   CALL   CNVDEC     ; GET NUMBER
95 000316 060005          ADD    R0,R5      ; ADD LENGTH OF NUMBER
96 000320 112743 000054   MOVB   #'.,-(R3)  ;
97 000324 012600          MOV    (SP)+,R0   ; RESTORE R0
98 000326 016001 000000G  MOV    LPROJ(R0),R1 ; GET PROJECT NUMBER
99 000332 004767 000204   CALL   CNVDEC     ;
100 000336 060005          ADD    R0,R5      ; ADD LENGTH OF NUMBER
101 000340 112743 000133   MOVB   #'L,-(R3)  ;
102 000344 005001          CLR    R1          ; SAY SINGLE HIGH & WIDE
103 000346 004767 000660   CALL   PRTBUF     ; PRINT PPN
104 000352 062706 000024   ADD    #20.,SP    ; FINISHED WITH BUFFER
105
106
107
108 000356 011601          MOV    (SP),R1    ; GET POINTER INTO SFCB
109 000360 012705 000007   MOV    #7,R5      ; GET MINIMUM LENGTH OF LINE
110 000364 010603          MOV    SP,R3      ; POINT TO PPN BUFFER
111 000366 162706 000016   SUB    #14.,SP    ; RESERVE THAT BUFFER
112 000372 112743 000200   MOVB   #200,-(R3) ; TERMINATE BUFFER
113 000376 116101 000000G  MOVB   SFUSER(R1),R1 ; GET JOB INDEX #
114 000402 006201          ASR    R1          ; GET LINE NUMBER

```

```

115 000404 004767 000132      CALL    CNVDEC      ; CONVERT TO DECIMAL
116 000410 060005              ADD     RO,R5      ; ADD LENGTH OF # TO LENGTH OF PRT STRING
117 000412 012700 000541'     MOV     #LINEND,RO ; GET END OF LINE STRING
118 000416 114043              MOVVB  -(RO),-(R3) ; GET THE REST OF THE LINE
119 000420 020027 000532'     CMP     RO,#LINSTR ; ALL IN BUFFER?
120 000424 101374              BHI    5$         ; BR IF NOT
121 000426 005001              CLR    R1         ; SAY SINGLE HIGH & WIDE
122 000430 004767 000576     CALL   PRTBUF     ; PRINT PPN
123 000434 062706 000016     ADD    #14.,SP    ; FINISHED WITH BUFFER
124
125
126                          ; Print the bottom border with time, day, date
127
128 000440 010603              MOV     SP,R3     ; GET POINTER TO TIME, DAY, DATE BUFFER
129 000442 162706 000050     SUB    #40.,SP   ; RESERVE THAT BUFFER
130 000446 012743 000200     MOV    #200,-(R3); TERMINATE THAT BUFFER
131 000452 010305              MOV    R3,R5     ; SAVE BEGINNING OF THAT BUFFER
132 000454 004767 001756     CALL   EDTDAT    ; GET DATE AND DAY
133 000460 004767 001270     CALL   EDTTOD    ; GET TIME OF DAY
134 000464 160305              SUB    R3,R5     ; GET LENGTH OF PRINT STRING
135 000466 004767 000254     CALL   BORDER    ; PRINT BOTTOM BORDER
136 000472 062706 000050     ADD    #40.,SP   ; DELETE TIME, DAY, DATE BUFFER
137
138 000476 112724 000014     MOVVB  #14,(R4)+ ; NEW PAGE
139 000502 112714 000200     MOVVB  #200,(R4); TERMINATE BUFFER
140 000506              OCALL  SFLGPG    ; PRINT BUFFER
141
142 000514 012601              MOV    (SP)+,R1  ; RESTORE REGISTERS
143 000516 012600              MOV    (SP)+,R0
144 000520 012605              MOV    (SP)+,R5
145 000522 012604              MOV    (SP)+,R4
146 000524 012602              MOV    (SP)+,R2
147 000526 012603              MOV    (SP)+,R3
148 000530 000207              RETURN
149
150 000532      114      111      116 LINSTR: .ASCII /LINE = /
      000535      105      040      075
      000540      040
151 000541              LINEND:
152                          .EVEN
153

```

CNVDEC -- Convert a number to an ascii decimal number

```

1          .SBTTL  CNVDEC  -- Convert a number to an ascii decimal number
2          ;-----
3          ; CNVDEC
4          ;
5          ; Inputs:
6          ;   R1 = Number to convert to decimal
7          ;   R3 = End of the buffer to hold the ascii number
8          ; Outputs:
9          ;   R0 = Length of the number
10         ;
11 000542 010246  CNVDEC: MOV     R2,-(SP)      ;SAVE R2
12
13 000544 005002          CLR     R2
14
15 000546 005000 1$:     CLR     R0             ;SET UP FOR DIV
16 000550 071027 000012  DIV     #10.,R0      ;GET LEAST SIGNIFICANT DIGIT
17 000554 062701 000060  ADD     #'0,R1       ;MAKE IT ASCII
18 000560 110143          MOVVB  R1,-(R3)     ;PUSH ONTO STACK
19 000562 005202          INC     R2             ;INC # OF CHARS. PRINT THIS PASS
20 000564 010001          MOV     R0,R1       ;SET UP FOR NEXT DIV
21 000566 001367          BNE     1$          ;UNTIL NOTHING LEFT
22 000570 020227 000001  CMP     R2,#1       ;PRINT AT LEAST 2 DIGITS.?
23 000574 003003          BGT     2$          ;BR IF YES
24 000576 112743 000060  MOVVB  #'0,-(R3)   ;ELSE DO SO
25 000602 005202          INC     R2             ;INC # OF CHARS.
26 000604
27 000604 010200 2$:     MOV     R2,R0       ;GET LENGTH OF NUMBER
28 000606 012602          MOV     (SP)+,R2    ;RESTORE R2
29 000610 000207          RETURN
30

```

```

1                                     .SBTTL  CNVR50  -- Convert a RAD-50 word to ascii
2                                     ;-----
3                                     ;   CNVR50
4                                     ;
5                                     ; Inputs:
6                                     ;   R0 = Word to convert
7                                     ;   R3 = Address of end of buffer to hold ascii value
8                                     ;
9 000612 010146  CNVR50: MOV      R1,-(SP)      ;SAVE REGISTERS
10 000614 010446  MOV      R4,-(SP)
11
12 000616 010001  MOV      R0,R1      ;GET WORD TO CONVERT
13 000620 005000  CLR      R0          ;ZERO HIGH WORD FOR DIV
14 000622 071027 000050  DIV     #50,R0      ;GET LAST CHAR.
15 000626 010104  MOV     R1,R4        ;SAVE LAST CHAR.
16 000630 010001  MOV     R0,R1        ;SET UP FOR NEXT DIV
17 000632 005000  CLR     R0          ;ZERO HIGH WORD FOR DIV
18 000634 071027 000050  DIV     #50,R0      ;GET 2nd AND 3rd CHARS.
19 000640 005704  TST     R4          ;IS CHAR. A SPACE
20 000642 001402  BEQ     1$          ;BR IF SO
21 000644 116443 000676'  MOVB   R50TBL(R4),-(R3); MOVE IN THREE CHARS.
22 000650 005701  1$:    TST     R1          ;IS CHAR. A SPACE
23 000652 001402  BEQ     2$          ;BR IF SO
24 000654 116143 000676'  MOVB   R50TBL(R1),-(R3); MOVE IF TWO CHARS.
25 000660 005700  2$:    TST     R0          ;IS CHAR. A SPACE
26 000662 001402  BEQ     3$          ;BR IF SO
27 000664 116043 000676'  MOVB   R50TBL(R0),-(R3)
28 000670
29 000670 012604  MOV     (SP)+,R4     ;RESTORE REGISTERS
30 000672 012601  MOV     (SP)+,R1
31 000674 000207  RETURN
32
33 000676      040      101      102  R50TBL: .ASCII / ABCDEFGHIJKLMNOPQRSTUVWXYZ$. 0123456789/
      000701      103      104      105
      000704      106      107      110
      000707      111      112      113
      000712      114      115      116
      000715      117      120      121
      000720      122      123      124
      000723      125      126      127
      000726      130      131      132
      000731      044      056      040
      000734      060      061      062
      000737      063      064      065
      000742      066      067      070
      000745      071
34                                     . EVEN
35

```


BORDER -- Print a border line of the flag page

```

1          .SBTTL  BORDER  -- Print a border line of the flag page
2          ;-----
3          ; BORDER - PRINT HEADER LINE AT BEGINNING OF PAGE
4          ;
5          ; Inputs:
6          ;   R2 = Page width
7          ;   R3 = Address of site name
8          ;   R4 = Address of stolen buffer
9          ;   R5 = Length of site name
10         ;
11 000746 010046          BORDER: MOV     R0,-(SP)      ;SAVE REGISTERS
12 000750 010146          MOV     R1,-(SP)
13 000752 010246          MOV     R2,-(SP)
14 000754 062705 000004  ADD     #4,R5      ;ADD 2 SPACES ON SIDES OF SITE NAME
15 000760 010546          MOV     R5,-(SP)      ;SAVE SITE NAME LENGTH
16 000762 160205          SUB     R2,R5      ;GET LENGTH OF LINE - SITE NAME
17 000764 005405          NEG     R5
18 000766 006205          ASR     R5
19
20 000770 012701 001222'  MOV     #TSXSTR,R1   ;GET ADDR OF STRING TO PRINT
21 000774 012702 000003  MOV     #3,R2      ;SOB COUNTER
22 001000 010500          11$: MOV     R5,R0      ;GET # CHARS. TO PRINT
23 001002 004767 000166  1$: CALL  TSTEND     ;TEST FOR END OF BUFFER
24 001006 020127 001232'  2$: CMP     R1,#STREND ;END IF STRING?
25 001012 103402          BLD     3$         ;BR IF NOT
26 001014 012701 001222'  MOV     #TSXSTR,R1   ;ELSE RE-INIT STRING POINTER
27 001020 112124          3$: MOVB   (R1)+,(R4)+ ;MOVE HEADER INTO BUFFER
28 001022 077011          SOB     R0,1$
29 001024 011600          MOV     (SP),R0     ;GET LENGTH OF SITE NAME
30 001026 020227 000002  CMP     R2,#2      ;MIDDLE LINE?
31 001032 001021          BNE     4$         ;BR IF NOT
32 001034 112724 000040  MOVB   #' ,(R4)+   ;SPACE OVER
33 001040 112724 000040  MOVB   #' ,(R4)+
34 001044 162700 000004  SUB     #4,R0      ;GET ACTUAL LENGTH OF SITE NAME
35 001050 004767 000120  31$: CALL  TSTEND     ;TEST FOR END OF BUFFER
36 001054 112324          32$: MOVB   (R3)+,(R4)+ ;MOVE IN SITE NAME
37 001056 121327 000200  CMPB   (R3),#200   ;END OF SITE NAME?
38 001062 001372          BNE     31$
39 001064 112724 000040  MOVB   #' ,(R4)+   ;SPACE OVER
40 001070 112724 000040  MOVB   #' ,(R4)+
41 001074 000405          BR     5$
42 001076 004767 000072  4$: CALL  TSTEND     ;TEST FOR END OF BUFFER
43 001102 112724 000040  41$: MOVB   #' ,(R4)+ ;MOVE IN SPACES
44 001106 077005          SOB     R0,4$
45 001110 010500          5$: MOV     R5,R0      ;GET # CHARS. TO PRINT
46 001112 004767 000056  6$: CALL  TSTEND     ;TEST FOR END OF BUFFER
47 001116 020127 001232'  7$: CMP     R1,#STREND ;END IF STRING?
48 001122 103402          BLD     8$         ;BR IF NOT
49 001124 012701 001222'  MOV     #TSXSTR,R1   ;ELSE RE-INIT STRING POINTER
50 001130 112124          8$: MOVB   (R1)+,(R4)+ ;MOVE HEADER INTO BUFFER
51 001132 077011          SOB     R0,6$
52 001134 112724 000012  MOVB   #12,(R4)+
53 001140 112724 000015  MOVB   #15,(R4)+
54 001144 077263          SOB     R2,11$
55 001146 112724 000012  MOVB   #12,(R4)+
56 001152 112724 000012  MOVB   #12,(R4)+
57 001156 112724 000012  MOVB   #12,(R4)+

```

```

58
59 001162 005726          TST      (SP)+
60
61 001164 012602          MOV      (SP)+,R2
62 001166 012601          MOV      (SP)+,R1
63 001170 012600          MOV      (SP)+,R0
64 001172 000207          RETURN
65
66 001174 020427 177766G  TSTEND: CMP      R4,#<<CSIBND-10.>;END OF BUFFER?
67 001200 103407          BLO     1$          ;BR IF NOT
68 001202 112714 000200  MOVB    #200,(R4)   ;TERMINATE BUFFER
69 001206          OCALL   SFLGPG     ;MOVE BUFFER INTO SPOOL BUFFER
70 001214 012704 000000G  MOV     #CSIBUF,R4 ;RE-INIT POINTER TO BUFFER
71 001220 000207          1$:      RETURN
72
73 001222          124      123      130  TSXSTR: .ASCII /TSX-Plus/
   001225          055      120      154
   001230          165      163
74 001232          STREND:
75

```

```

1                                     .SBTTL PRTBUF -- Print a 7 X 5 line of the flag page
2                                     ;-----
3                                     ; PRTBUF - PRINT A LINE OF THE FLAGPAGE
4                                     ;
5                                     ; Inputs:
6                                     ;   R1 = Desired height and width.
7                                     ;   R2 = Page width
8                                     ;   R3 = Address of buffer being printed
9                                     ;   R4 = Address of stolen buffer
10                                    ;   R5 = Lenth of the line
11                                    ;
12 001232 010246 PRTBUF: MOV      R2,-(SP)      ;SAVE R2
13
14 001234 010346          MOV      R3,-(SP)      ;SAVE ADDRESS OF PRINT BUFFER
15 001236 005705          TST      R5          ;ANY TEXT ON THIS LINE?
16 001240 001513          BEQ      99$         ;BR IF NO
17 001242 006305          ASL      R5          ;GET WIDTH OF LINE
18 001244 010546          MOV      R5,-(SP)
19 001246 006305          ASL      R5
20 001250 062605          ADD      (SP)+,R5
21 001252 005701          TST      R1          ;DHDW?
22 001254 001401          BEQ      1$         ;BR IF NO
23 001256 006305          ASL      R5          ;GET # WIDTH OF DHDW
24 001260 160205          1$:      SUB      R2,R5      ;GET # OF BLANKS IN LINE
25 001262 005405          2$:      NEG      R5
26 001264 006205          ASR      R5          ;GET # SPACES TO SKIP
27 001266 012700 000001  22$:     MOV      #1,R0      ;SET INITIAL BIT TO TEST
28 001272 010146          21$:     MOV      R1,-(SP)     ;SAVE HEIGHT AND WIDTH
29
30                                    ; Print the line of output
31                                    ;
32 001274 010502          3$:      MOV      R5,R2      ;COUNTER FOR SOB LOOP
33 001276 005702          31$:     TST      R2          ;IS COUNTER ZERO?
34 001300 001405          BEQ      43$         ;BR IF SO
35
36                                    ; Loop to center the line (print leading spaces)
37                                    ;
38 001302 004767 177666  4$:      CALL     TSTEND      ;TEST FOR END OF BUFFER
39 001306 112724 000040  41$:     MOVB     #'',(R4)+    ;MOVE IN SPACE
40 001312 077205          SOB      R2,4$
41
42 001314 016603 000002  43$:     MOV      2(SP),R3    ;GET ADDRESS OF PRINT LINE
43 001320 010346          MOV      R3,-(SP)     ;SAVE IT
44
45                                    ; Get the next character to print
46                                    ;
47 001322 004767 000230  5$:      CALL     GCHROF      ;GET OFFSET INTO CHAR. TABLE
48 001326 116246 001646'  MOVB     CHARS(R2),-(SP) ;SAVE CHAR.
49 001332 010246          MOV      R2,-(SP)     ;GET OFFSET INTO BIT PATTERN TABLE
50 001334 006302          ASL      R2
51 001336 006302          ASL      R2
52 001340 062602          ADD      (SP)+,R2
53 001342 012703 000005  MOV      #5,R3          ;INIT COUNTER FOR SOB LOOP
54
55                                    ; Print one row of an individual character
56                                    ;
57 001346 004767 177622  6$:      CALL     TSTEND      ;TEST FOR END OF BUFFER

```

```

58 001352 111646          7$:   MOVB   (SP),-(SP)      ;GET CHAR. TO PRINT
59 001354 130062 003556'   BITB   R0,CHRTBL(R2)    ;SPACE OR LETTER?
60 001360 001002          BNE    71$              ;BR IF LETTER
61 001362 112716 000040    MOVB   #' ,(R4)         ;ELSE WANT TO PRINT SPACE
62 001366 111624          71$:  MOVB   (SP),(R4)+       ;PRINT DESIRED CHAR.
63 001370 005701          TST    R1               ;DOUBLE WIDE?
64 001372 001401          BEQ    8$               ;BR IF NOT
65 001374 111624          MOVB   (SP),(R4)+       ;ELSE PRINT LETTER
66 001376 005726          8$:   TST    (SP)+        ;CLEAR CHAR. TO PRINT
67 001400 005202          INC    R2               ;POINT TO NEXT COLUMN
68 001402 077317          SOB    R3,6$           ;PRINT NEXT COLUMN
69
70                          ; Done with 1 row of 1 character, next char
71
72 001404 005726          TST    (SP)+           ;POP OFF LETTER
73 001406 012603          MOV    (SP)+,R3        ;GET CURRENT LETTER IN PRINT LINE
74 001410 005203          INC    R3              ;GET NEXT LETTER IN PRINT LINE
75 001412 121327 000200    CMPB   (R3),#200       ;END OF LINE?
76 001416 001410          BEQ    9$              ;BR IF YES
77 001420 112724 000040    MOVB   #' ,(R4)+       ;ELSE SPACE BETWEEN LETTERS
78 001424 005701          TST    R1              ;DHDW?
79 001426 001402          BEQ    81$            ;BR IF NOT
80 001430 112724 000040    MOVB   #' ,(R4)+       ;MAKE DOUBLE WIDE
81 001434 010346          81$:  MOV    R3,-(SP)      ;SAVE POINTER INTO PRINT LINE
82 001436 000731          BR     5$              ;GO PRINT NEXT LETTER
83 001440 112724 000012    9$:   MOVB   #12,(R4)+   ;NEW LINE
84 001444 112724 000015    MOVB   #15,(R4)+
85 001450 005716          TST    (SP)            ;DOUBLE HIGH?
86 001452 001402          BEQ    91$            ;BR IF NOT
87 001454 005016          CLR    (SP)            ;CLR HEIGHT COUNTER
88 001456 000706          BR     3$              ;MAKE DOUBLE HIGH
89
90                          ; Next Row
91
92 001460 005726          91$:  TST    (SP)+       ;POP OFF DHDW COUNTER
93 001462 106300          ASLB   R0               ;GET NEXT ROW
94 001464 100302          BPL    21$             ;IF ANOTHER ROW, GO PRINT IT
95 001466 000422          BR     98$
96
97                          ; No text was on this line; print a 7 line feeds.
98
99 001470 005201          99$:  INC    R1           ;INC R1 TO USE FOR DHDW SOB
100
101 001472 112724 000012    97$:  MOVB   #12,(R4)+
102 001476 112724 000012    MOVB   #12,(R4)+
103 001502 112724 000012    MOVB   #12,(R4)+
104 001506 112724 000012    MOVB   #12,(R4)+
105 001512 004767 177456    CALL   TSTEND           ;TEST FOR END OF BUFFER
106 001516 112724 000012    MOVB   #12,(R4)+
107 001522 112724 000012    MOVB   #12,(R4)+
108 001526 112724 000012    MOVB   #12,(R4)+
109 001532 077121          SOB    R1,97$          ;DO IT AGAIN IF DHDW
110
111                          ; Done with line
112
113 001534 112724 000012    98$:  MOVB   #12,(R4)+   ;NEW LINE
114 001540 112724 000012    MOVB   #12,(R4)+   ;NEW LINE

```

```
115 001544 112724 000012      MOVB    #12,(R4)+      ;NEW LINE
116 001550 005726              TST     (SP)+         ;POP OFF PRINT LINE ADDRESS
117                               ;
118 001552 012602      MOV     (SP)+,R2      ;RESTORE R2
119 001554 000207      RETURN
120
```

GCHROF -- Get the offset into the character table (CHARS)

```

1          .SBTTL  GCHROF  -- Get the offset into the character table (CHARS)
2          ;-----
3          ; GCHROF - Get offset into character table for a byte value
4          ;
5          ; Inputs:
6          ;       R3 = Address of the byte
7          ;
8          ; Outputs:
9          ;       R2 = Offset into character table
10         ;
11         GCHROF: MOVB   (R3),R2          ;GET BYTE VALUE
12         CMPB   R2,#'          ;IS IT < THAN A SPACE
13         BGE   2$             ;BR IF NOT
14         MOVB   #69.,R2        ;UNPRINTABLE; SET OFFSET INTO TABLE
15         BR    9$             ;DONE
16         2$:  CMPB   R2,#'`          ;IS IT <= AN "`"
17         BGT   21$            ;BR IF NOT
18         SUB    #' ,R2         ;ELSE, SET OFFSET INTO TABLE
19         BR    9$             ;DONE
20         21$: CMPB   R2,#'{'        ;IS IT < A "{"
21         BGE   3$             ;BR IF NOT
22         SUB    #64.,R2        ;ELSE, SET OFFSET INTO TABLE
23         BR    9$             ;DONE
24         3$:  CMPB   R2,#'~        ;IS IT > A "~"
25         BLE   4$             ;BR IF NOT
26         MOVB   #69.,R2        ;ELSE, SET OFFSET INTO TABLE
27         BR    9$             ;DONE
28         4$:  SUB    #58.,R2        ;SET OFFSET INTO TABLE
29         9$:  RETURN
30
31
32         001646      040      041      042  CHARS:  . ASCII  | !"#%&'()*+,-./0123456789:;<=>?@!
33         001651      043      044      045
34         001654      046      047      050
35         001657      051      052      053
36         001662      054      055      056
37         001665      057      060      061
38         001670      062      063      064
39         001673      065      066      067
40         001676      070      071      072
41         001701      073      074      075
42         001704      076      077      100
43         001707      101      102      103  . ASCII  /ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_`/
44         001712      104      105      106
45         001715      107      110      111
46         001720      112      113      114
47         001723      115      116      117
48         001726      120      121      122
49         001731      123      124      125
50         001734      126      127      130
51         001737      131      132      133
52         001742      134      135      136
53         001745      137      140
54         001747      173      174      175  . ASCII  /{ } ~ ? /
55         001752      176      077
56
57         . EVEN

```

```

1                                     .SBTTL  EDTTOD  -- Edit the time of day
2                                     ;-----
3                                     ; EDTTOD IS CALLED TO EDIT THE TIME OF DAY.
4                                     ; ALL REGISTERS ARE PRESERVED.
5                                     ;
6 001754 010046 EDTTOD: MOV      R0,-(SP)
7 001756 010246      MOV      R2,-(SP)
8 001760 010446      MOV      R4,-(SP)
9 001762 010546      MOV      R5,-(SP)
10 001764 016704 000000G  MOV      SYTIMH,R4      ;GET HIGH-ORDER VALUE
11 001770 016705 000000G  MOV      SYTIML,R5      ;GET LOW-ORDER VALUE
12 001774 016702 000000G  MOV      TK1SEC,R2      ;Get # clock ticks per second
13 002000 004767 000054   CALL      DIVIDE        ;Get # seconds past midnight
14 002004 012702 000074   MOV      #60,R2        ;DIVIDE BY 60 TO SPLIT INTO SEC & MIN
15 002010 004767 000044   CALL      DIVIDE
16 002014 004767 176522   CALL      CNVDEC        ;PRINT SECONDS
17 002020 112743 000072   MOVB     #'',(R3)      ;PUT IN COLON SEPARATOR
18 002024 004767 000030   CALL      DIVIDE        ;GET # MINUTES & # HOURS
19 002030 004767 176506   CALL      CNVDEC        ;PRINT MINUTES VALUE
20 002034 112743 000072   MOVB     #'',(R3)      ;PUT IN COLON SEPARATOR
21 002040 010501         MOV      R5,R1         ;GET # HOUR VALUE
22 002042 004767 176474   CALL      CNVDEC        ;PRINT # HOURS
23 002046 012605         MOV      (SP)+,R5
24 002050 012604         MOV      (SP)+,R4
25 002052 012602         MOV      (SP)+,R2
26 002054 012600         MOV      (SP)+,R0
27 002056 000207         RETURN

```

DIVIDE -- Divide 32-bit qty by 16-bit

```

1
2
3
4
5
6
7
8 002060 010346
9 002062 005001
10 002064 012703 000037
11 002070 006305
12 002072 006104
13 002074 006101
14 002076 020102
15 002100 103402
16 002102 160201
17 002104 005205
18 002106 005303
19 002110 100367
20 002112 012603
21 002114 000207
22

```

```

.SBTTL DIVIDE -- Divide 32-bit qty by 16-bit
-----
; SUBROUTINE DIVIDE IS CALLED TO DIVIDE THE 32-BIT QUANTITY
; IN R4 (HIGH ORDER) AND R5 (LOW ORDER) BY THE 16-BIT
; QUANTITY IN R2. ON RETURN THE QUOTIENT IS IN R4-R5
; AND THE REMAINDER IS IN R1. ALL OTHER REGISTERS ARE PRESERVED.
;
DIVIDE: MOV R3, -(SP)
        CLR R1 ; INITIALIZE REMAINDER
        MOV #31, R3 ; GET SHIFT COUNT
1$: ASL R5 ; SHIFT BIT OUT OF LOW ORDER
    ROL R4 ; SHIFT THROUGH HIGH ORDER
    ROL R1 ; INTO R0
    CMP R1, R2 ; GOT ENOUGH TO SUBTRACT YET?
    BLO 2$ ; BR IF NOT
    SUB R2, R1 ; SUBTRACT DIVISOR
    INC R5 ; INCREASE QUOTIENT
2$: DEC R3 ; COUNT # BITS SHIFTED
    BPL 1$ ; BR IF MORE TO DO
    MOV (SP)+, R3
    RETURN

```



```

1          ; -----
2          ; Miscellaneous date related routines
3          ;
4          000037 YEAR$ = 000037 ; YEAR PART OF DATE WORD (YEAR - 1972)
5          001740 DAY$  = 001740 ; DAY PART OF DATE WORD
6          036000 MONTH$ = 036000 ; MONTH PART OF DATE WORD
7          ;
8          .DSABL GBL
9
10         .SBTTL LEAP -- Function to determine if this is a leap year
11        ; LEAP -----
12        ; Function to determine if a year is a leap year
13        ;
14        ; Inputs:      RO contains date in RT-11 format
15        ; Outputs:    Carry set if leap year, carry clear if not
16        ;
17        ; Leap years have 366 days, add Feb 29; non-leap years have 365 days.
18        ; Leap year rule:
19        ;   If year is not evenly divisible by 4, it is not a leap year.
20        ;   If year is evenly divisible by 4, it is a leap year -
21        ;     except if evenly divisible by 100, it is not -
22        ;     except if evenly divisible by 400, it is after all.
23        ; Thus, in 400 years there will be
24        ;   303 non-leap years      110595 days
25        ;   97 leap years           35502 days
26        ;
27        ;
28        ;   146097 / 400 = 365.2425 avg days/year
29        ; That means a year is 365 days, 5 hr, 49 min and 12 sec.
30        ;
31        ; First do a fast check to eliminate 75% of years
32        ;
33        LEAP: BIT #3,RO ; POSSIBLE LEAP YEAR?
34              BEQ 2$ ; BR TO FURTHER CHECKS IF MAYBE
35        1$: CLC ; IF NOT MULTIPLE OF 4, THIS IS NOT A LEAP YEAR
36              RETURN
37        ;
38        ; If year is multiple of 4, then need to check further
39        ;
40        2$: MOV RO,-(SP) ; SAVE REGISTERS
41              MOV R1,-(SP)
42              BIC #^C<YEAR$>,RO ; GET RID OF ALL BUT YEAR
43              ADD #1972,RO ; ADD BACK BASE YEAR
44              MOV RO,R1 ; SET FOR DIVIDE
45              CLR RO
46              DIV #100,RO ; BREAK INTO CENTURIES
47        ; QUOTIENT GOES IN RO AND REMAINDER IN R1
48              TST R1 ; IS THIS A NEW CENTURY?
49              BNE 5$ ; IF NOT, GO SAY THIS IS A LEAP YEAR
50              BIT #3,RO ; IS THIS YEAR A MULTIPLE OF 400?
51              BEQ 5$ ; IF SO, GO SAY THIS IS A LEAP YEAR
52              CLC ; 100,200,300 MULTIPLES ARE NOT LEAP YEARS
53              BR 9$
54        5$: SEC ; 4 AND 400 MULTIPLES ARE LEAP YEARS
55        9$: MOV (SP)+,R1 ; RESTORE REGISTERS
56              MOV (SP)+,RO
57              RETURN

```

JULIAN -- Get day of the year for a given date

```

1          .SBTTL JULIAN  -- Get day of the year for a given date
2          ; JULIAN -----
3          ; Returns the day of the year for a given date
4          ;
5          ; Inputs:      R0 contains date in RT-11 format
6          ; Outputs:    R0 contains day of the year (1-366; Jan 1 is day 1)
7          ;
8 002202 010246          JULIAN: MOV      R2, -(SP)
9 002204 010346          MOV      R3, -(SP)
10 002206 010046         MOV      R0, -(SP)
11 002210 005002         CLR      R2          ; CLEAR RESULT COUNTER
12 002212 010003         MOV      R0, R3       ; GET COPY OF DATE
13 002214 072327 177766  ASH      #-10, R3    ; SHIFT MONTH INTO LOW BITS
14 002220 020327 000003  CMP      R3, #3      ; PAST FEBRUARY?
15 002224 002404         BLT      1$          ; NO, DON'T WORRY ABOUT LEAP YEAR
16 002226 004767 177664  CALL     LEAP        ; >FEB, SEE IF LEAP YEAR (REQ'S DATE IN R0)
17 002232 103001         BCC      1$          ; IF NOT LEAP YEAR, FEB HAS 28 DAYS
18 002234 005202         INC      R2          ; IF LEAP YEAR, AND PAST FEB, ADD FEB 29
19 002236 012700 002276' 1$:      MOV      #10$, R0    ; POINT TO MONTH LENGTH TABLE
20 002242 005303         2$:      DEC      R3          ; DON'T ADD IN CURRENT MONTH
21 002244 003402         BLE      3$          ; DONE WITH PAST MONTHS?
22 002246 062002         ADD      (R0)+, R2    ; ADD MONTH TO TOTAL
23 002250 000774         BR       2$          ; LOOP UP TO CURRENT MONTH
24 002252 012603         3$:      MOV      (SP)+, R3    ; GET DATE BACK
25 002254 042703 176037  BIC      #^C<DAY$>, R3 ; CLEAR OUT ALL BUT DAY BITS
26 002260 072327 177773  ASH      #-5, R3     ; SHIFT DAY TO LOW BITS
27 002264 060302         ADD      R3, R2      ; ADD DAYS THIS MONTH TO GET TOTAL
28 002266 010200         MOV      R2, R0      ; RETURN RESULT IN R0
29 002270 012603         MOV      (SP)+, R3
30 002272 012602         MOV      (SP)+, R2
31 002274 000207         RETURN
32          ;
33          ; Table of # days per month
34 002276 000037 000034 000037 10$:      .word   31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31.
          002304 000036 000037 000036
          002312 000037 000037 000036
          002320 000037 000036 000037

```

```

1          . SBTTL DOW          -- Get the day of the week
2          ; DOW -----
3          ; returns the day of the week Sun=0, Mon=1, ... Sat=6
4          ;
5          ; Inputs:           RO contains system date in RT-11 format
6          ; Outputs:          RO contains weekday Sun-Sat == 0-6
7          ;
8 002326   010146
9 002330   010246
10 002332  010046
11 002334  004767 177642
12 002340  005300
13 002342  012602
14 002344  042702 177740
15 002350  062702 002376'
16 002354  111201
17 002356  060001
18 002360  005000
19 002362  071027 000007
20 002366  010100
21 002370  012602
22 002372  012601
23 002374  000207
24
25          ;
26          ; Table of days on which years start 1972-2003 (valid range of RT-11 dates)
27 002376   006   001   002   10$: . BYTE 6, 1, 2, 3, 4, 6, 0, 1, 2, 4, 5, 6, 0, 2, 3, 4, 5, 0 ; 1972-1989
   002401   003   004   006
   002404   000   001   002
   002407   004   005   006
   002412   000   002   003
   002415   004   005   000
28 002420   001   002   003   . BYTE 1, 2, 3, 5, 6, 0, 1, 3, 4, 5, 6, 1, 2, 3 ; 1990-2003
   002423   005   006   000
   002426   001   003   004
   002431   005   006   001
   002434   002   003
29          . EVEN
30

```

EDTDTAT -- Print the current date

```

1          .SBTTL EDTDTAT -- Print the current date
2          ;-----
3          ; EDTDTAT IS CALLED TO PRINT THE CURRENT DATE.
4          ; ALL REGISTERS ARE PRESERVED.
5          ;
6 002436 010046 EDTDTAT: MOV R0,-(SP)
7 002440 010146          MOV R1,-(SP)
8 002442 010246          MOV R2,-(SP)
9 002444 010446          MOV R4,-(SP)
10 002446 010546         MOV R5,-(SP)
11 002450 016700 000000G MOV SYSDAT,R0          ;GET CURRENT DATE
12 002454 010046         MOV R0,-(SP)          ;SAVE DATE
13 002456 006300         ASL R0          ;LEFT JUSTIFY VALUE
14 002460 012702 000005 MOV #5,R2          ;SHIFT OFF 5 BITS
15 002464 005001         CLR R1          ;INTO R1
16 002466 006100 1$:    ROL R0          ;SHIFT MONTH VALUE INTO R1
17 002470 006101         ROL R1
18 002472 077203         SOB R2,1$          ;LOOP IF MORE BITS TO SHIFT
19 002474 010046         MOV R0,-(SP)
20 002476 005301         DEC R1          ;GET MONTH VALUE IN RANGE 0-11
21 002500 010146         MOV R1,-(SP)          ;SAVE MONTH VALUE
22          ; PRINT YEAR
23 002502 016601 000004 MOV 4(SP),R1          ;GET THE ORIGINAL DATE VALUE
24 002506 042701 177740 BIC #^C37>,R1          ;CLEAR ALL BUT YEAR FIELD
25 002512 062701 000110 ADD #72.,R1          ;YEAR # IS RELATIVE TO 1972
26 002516 004767 176020 CALL CNVDEC          ;PRINT YEAR VALUE
27 002522 016601 000004 MOV 4(SP),R1          ;GET ORIGINAL DATE VALUE
28 002526 042701 177740 BIC #^C<YEAR#>,R1          ;GET RID OF ALL BUT YEAR
29 002532 062701 003664 ADD #1972.,R1          ;ADD BACK BASE YEAR
30 002536 005000         CLR R0
31 002540 071027 000144 DIV #100.,R0          ;BREAK INTO CENTURIES
32 002544 010001         MOV R0,R1          ;GET RIGHT CENTURY
33 002546 004767 175770 CALL CNVDEC          ;PRINT CENTURY
34 002552 112743 000040 MOVB #' ,-(R3)          ;PUT IN SEPARATOR
35 002556 112743 000054 MOVB #' , ,-(R3)          ;PUT IN SEPARATOR
36          ; PRINT DAY NUMBER
37 002562 016600 000002 MOV 2(SP),R0          ;GET ORIGINAL DATE VALUE
38 002566 012702 000005 MOV #5,R2          ;GET 5 BITS OF DAY VALUE
39 002572 005001         CLR R1
40 002574 006100 2$:    ROL R0          ;SHIFT BITS INTO R1
41 002576 006101         ROL R1
42 002600 077203         SOB R2,2$
43 002602 004767 175734 CALL CNVDEC          ;PRINT DAY-OF-MONTH
44 002606 112743 000040 MOVB #' ,-(R3)          ;PUT IN SEPARATOR
45          ; PRINT MONTH NAME
46 002612 012601         MOV (SP)+,R1          ;GET MONTH INDEX
47 002614 006301         ASL R1          ;*2
48 002616 016104 002714' MOV MONTAB(R1),R4          ;POINT TO MONTH NAME
49 002622 016102 002716' MOV MONTAB+2(R1),R2;POINT TO END OF THE MONTH NAME
50 002626 114243 3$:    MOVB -(R2),-(R3)          ;PRINT NAME OF MONTH
51 002630 020204         CMP R2,R4          ;GOT THE WHOLE NAME?
52 002632 101375         BHI 3$          ;BR IF NOT
53 002634 112743 000040 MOVB #' ,-(R3)          ;PUT IN SEPARATOR SPACE
54 002640 112743 000054 MOVB #' , ,-(R3)
55          ; PRINT DAY OF WEEK
56 002644 005726         TST (SP)+
57 002646 012600         MOV (SP)+,R0          ;ORIGINAL DATE FORMAT

```

```

58 002650 004767 177452          CALL   DOW           ;GET DAY OF WEEK (0-6)
59 002654 006300                ASL    RO           ;DAY OF WEEK INDEX (*2)
60 002656 016004 003060'        MOV    DAYTAB(RO),R4 ;POINT TO DAY OF WEEK
61 002662 016002 003062'        MOV    DAYTAB+2(RO),R2 ;POINT TO END OF THE MONTH NAME
62 002666 114243                4$:   MOVB  -(R2),-(R3) ;PRINT NAME OF MONTH
63 002670 020204                CMP    R2,R4        ;GOT THE WHOLE NAME?
64 002672 101375                BHI   4$           ;BR IF NOT
65 002674 112743 000040        MOVB  #' ,-(R3)    ;PUT IN SEPARATOR SPACE
66
67 002700 012605                MOV    (SP)+,R5    ;RESTORE REGISTERS
68 002702 012604                MOV    (SP)+,R4
69 002704 012602                MOV    (SP)+,R2
70 002706 012601                MOV    (SP)+,R1
71 002710 012600                MOV    (SP)+,R0
72 002712 000207                RETURN
73

```

```

74 002714 002746'        MONTAB: .WORD  JANURY
75 002716 002755'        .WORD  FEBURY
76 002720 002765'        .WORD  MARCH
77 002722 002772'        .WORD  APRIL
78 002724 002777'        .WORD  MAY
79 002726 003002'        .WORD  JUNE
80 002730 003006'        .WORD  JULY
81 002732 003012'        .WORD  AUGUST
82 002734 003020'        .WORD  SPTMBR
83 002736 003031'        .WORD  OCTOBR
84 002740 003040'        .WORD  NVEMBR
85 002742 003050'        .WORD  DCEMBR
86 002744 003060'        .WORD  ENDMTB
87

```

```

88 002746      112      141      156  JANURY: .ASCII  /January/
      002751      165      141      162
      002754      171
89 002755      106      145      142  FEBURY: .ASCII  /February/
      002760      162      165      141
      002763      162      171
90 002765      115      141      162  MARCH:  .ASCII  /March/
      002770      143      150
91 002772      101      160      162  APRIL:  .ASCII  /April/
      002775      151      154
92 002777      115      141      171  MAY:    .ASCII  /May/
93 003002      112      165      156  JUNE:   .ASCII  /June/
      003005      145
94 003006      112      165      154  JULY:   .ASCII  /July/
      003011      171
95 003012      101      165      147  AUGUST: .ASCII  /August/
      003015      165      163      164
96 003020      123      145      160  SPTMBR: .ASCII  /September/
      003023      164      145      155
      003026      142      145      162
97 003031      117      143      164  OCTOBR: .ASCII  /October/
      003034      157      142      145
      003037      162
98 003040      116      157      166  NVEMBR: .ASCII  /November/
      003043      145      155      142
      003046      145      162
99 003050      104      145      143  DCEMBR: .ASCII  /December/

```

	003053	145	155	142	
	003056	145	162		
100	003060				ENDMTB:
101					
102	003060	003100'			DAYTAB: . WORD SUNDAY
103	003062	003106'			. WORD MONDAY
104	003064	003114'			. WORD TUESDY
105	003066	003123'			. WORD WDNSDY
106	003070	003134'			. WORD THRSYD
107	003072	003144'			. WORD FRIDAY
108	003074	003152'			. WORD SATRDY
109	003076	003162'			. WORD ENDDTB
110					
111	003100	123	165	156	SUNDAY: . ASCII /Sunday/
	003103	144	141	171	
112	003106	115	157	156	MONDAY: . ASCII /Monday/
	003111	144	141	171	
113	003114	124	165	145	TUESDY: . ASCII /Tuesday/
	003117	163	144	141	
	003122	171			
114	003123	127	145	144	WDNSDY: . ASCII /Wednesday/
	003126	156	145	163	
	003131	144	141	171	
115	003134	124	150	165	THRSYD: . ASCII /Thursday/
	003137	162	163	144	
	003142	141	171		
116	003144	106	162	151	FRIDAY: . ASCII /Friday/
	003147	144	141	171	
117	003152	123	141	164	SATRDY: . ASCII /Saturday/
	003155	165	162	144	
	003160	141	171		
118	003162				ENDDTB:

SETSTN -- KMON emt to copy site name into overlay

```

1          .SBTTL  SETSTN -- KMON emt to copy site name into overlay
2          ;-----
3          ; This is done once only, during the first time TSKMON is initialized.
4          ;
5 003162  012701  000000G  SETSTN: MOV      #CSIBUF,R1      ;KMONCE has put decrypted site name here
6 003166  012702  004314'      MOV      #SITNAM,R2      ;Copy it into overlay here
7 003172  112122                1$:  MOVVB   (R1)+,(R2)+    ;Get next char
8 003174  003403                BLE      2$          ;Stop if NUL or terminating 200
9 003176  020227  004372'      CMP      R2,#<SITNAM+46.> ;Have we gone past end of buffer?
10 003202  103773                BLO     1$          ;Move entire site name if not
11 003204  005302                2$:  DEC      R2          ;Point back at terminating byte
12 003206  112712  000200      MOVVB   #200,(R2)     ;Always terminate with 200
13 003212  162702  004314'      SUB      #SITNAM,R2    ;Calculate site name length
14 003216  010267  001150      MOV      R2,SITLEN    ;And save it
15 003222  000167  000000G      JMP      EMTXIT      ;Done

```

GTLICN -- Get site information EMT

```

1                                     .SBTTL  GTLICN -- Get site information EMT
2                                     ;-----
3                                     ; The form of the argument block is:
4                                     ;
5                                     ;     .BYTE   n,124           ;n specifies subfunction
6                                     ;     .WORD   RTNADR        ;For site name only, user buffer addr
7                                     ;
8 003226 116700 000000G GTLICN: MOV    EMTBLK,R0      ;Get function code
9 003232 020027 000002  CMP    R0,#GSFMAX      ;Valid subfunction?
10 003236 103402  BLD    1$              ;Br if OK
11 003240 000167 000000G  JMP    SETC            ;Return with error
12 003244 006300 1$: ASL    R0              ;Convert function code to table index
13 003246 000170 003252'  JMP    @GSFTBL(R0)    ;Jump into appropriate processing routine.
14                                     ;
15                                     ; Table of Get Site inFormation routines
16                                     ;
17 003252 003256' GSFTBL: .WORD   GETLIC      ;0,124 Return site incremental license number
18 003254 003270'  .WORD   GETSIT      ;1,124 Return site license name
19          000002  GSFMAX = <.-GSFTBL>/2
20                                     ;
21                                     ;-----
22                                     ; Return site incremental license number to user in R0
23                                     ;
24 003256 016767 000000G 000000G GETLIC: MOV    TSXSIT,URO    ;Put license # in user's R0
25 003264 000167 000000G  JMP    EMTXIT
26                                     ;
27                                     ;-----
28                                     ; Return site name in user buffer
29                                     ;
30 003270 GETSIT:
31 003270 016702 000002G  MOV    EMTBLK+2,R2    ;Get user buffer ptr !!! Word aligned !!!
32 003274 010200  MOV    R2,R0          ;Validate user buffer address
33 003276 004767 000000G  CALL   VALADW        ;User ptr must be word aligned (for MTPD's)
34 003302 012701 004314'  MOV    #SITNAM,R1     ;Get ptr to site name
35 003306 016703 001060  MOV    SITLEN,R3      ;How long is actual site name?
36 003312 005203  INC    R3             ;Round up
37 003314 006203  ASR    R3             ;Convert to words
38 003316 012146 1$: MOV    (R1)+,-(SP)  ;Get next char from site name
39 003320 106622  MTPD   (R2)+          ;Return value to user
40 003322 077303  SOB    R3,1$          ;Full site length
41 003324 016767 001042 000000G  MOV    SITLEN,URO    ;Return string length in user R0
42 003332 000167 000000G  JMP    EMTXIT

```



```

1                                     .SBTTL  SPLSFF -- Set or reset flagpage flag for device
2                                     ;-----
3                                     ; SPLSFF is called to set/reset the flagpage flag for a spooled device.
4                                     ;
5                                     ; Inputs:
6                                     ;   CHNNUM = User's channel number
7                                     ;   R2 = 0==>noflagpage 1==>flagpage
8                                     ;
9 003336 010146                       SPLSFF: MOV      R1, -(SP)
10 003340 012767 000000G 000000G      MOV      #SD$RSV, URO      ; Assume device is not spooled
11                                     ;
12                                     ; Find SDCB associated with this channel
13                                     ;
14 003346 004767 000124                CALL     FNDSB      ; Locate SDCB associated with this channel
15 003352 103416                       BCS     9$          ; Br if cannot find the SDCB
16                                     ;
17                                     ; Set or reset the flag in the SDCB
18                                     ;
19 003354 016167 000000G 000000G      MOV      SDFLAG(R1), URO ; Get current flag values
20 003362 042767 000000C 000000G      BIC     #^CSD$FLG, URO ; And return flag page status to user
21 003370 052761 000000G 000000G      BIS     #SD$FLG, SDFLAG(R1); Assume we want to set the flag
22 003376 005702                       TST     R2          ; Do we want to reset the flag?
23 003400 001003                       BNE     9$          ; Br if not
24 003402 042761 000000G 000000G      BIC     #SD$FLG, SDFLAG(R1); Clear the hold flag
25                                     ;
26                                     ; Finished
27                                     ;
28 003410 012601                       9$:   MOV      (SP)+, R1
29 003412 000167 000000G                JMP     EMTXIT

```

SPLSWF -- Set or reset wide flagpage flag for device

```

1          .SBTTL  SPLSWF -- Set or reset wide flagpage flag for device
2          ;-----
3          ; SPLSWF is called to set/reset the wide flagpage flag for a spooled
4          ; device.
5          ;
6          ; Inputs:
7          ;   CHNUM = User's channel number
8          ;   R2 = 0==>narrow flagpage 1==>wide flagpage
9          ;
10         SPLSWF:  MOV     R1, -(SP)
11         003420  012767  000000G 000000G      MOV     #SD$RSV, URO      ; Assume the device is not spooled
12         ;
13         ; Find SDCB associated with this channel
14         ;
15         003426  004767  000044                CALL    FNDSO             ; Locate SDCB associated with this channel
16         003432  103416                        BCS    9$                ; Br if cannot find the SDCB
17         ;
18         ; Set or reset the flag in the SDCB
19         ;
20         003434  016167  000000G 000000G      MOV     SDFLAG(R1),URO   ; Get current flag settings
21         003442  042767  000000C 000000G      BIC    #^CSD$WID,URO    ; And return current width to user
22         003450  052761  000000G 000000G      BIS    #SD$WID, SDFLAG(R1); Assume we want to set the flag
23         003456  005702                        TST    R2                ; Do we want to reset the flag?
24         003460  001003                        BNE    9$                ; Br if not
25         003462  042761  000000G 000000G      BIC    #SD$WID, SDFLAG(R1); Clear the hold flag
26         ;
27         ; Finished
28         ;
29         003470  012601      9$:  MOV     (SP)+, R1
30         003472  000167  000000G      JMP    EMTXIT

```

FNDSO -- Locate SDCB for spooled channel

```

1          .SBTTL  FNDSO -- Locate SDCB for spooled channel
2          ;-----
3          ; FNDSO is called to locate the SDCB associated with the current I/O channel.
4          ;
5          ; Inputs:
6          ;   CHNUM = User channel number.
7          ;   CORUSR = Job index number
8          ;
9          ; Outputs:
10         ;   C-flag cleared if SDCB found, set otherwise.
11         ;   R1 = Address of SDCB.
12         ;
13 003476 010246  FNDSO:  MOV      R2,-(SP)      ;SAVE R2
14
15 003500 016702 000000G      MOV      SFCB,R2      ;POINT TO 1ST SFCB
16 003504 020267 000000G  4$:  CMP      R2,SFCBND    ;CHECKED ALL SFCB'S?
17 003510 103013          BHS      1$          ;BR IF YES
18 003512 126762 000000G 000000G  CMPB    CORUSR,SFUSER(R2); IS THIS SFCB IN USE BY THIS USER?
19 003520 001004          BNE      2$          ;BR IF NOT
20 003522 126762 000000G 000000G  CMPB    CHNUM,SFCHAN(R2); IS SFCB ASSIGNED TO THIS CHANNEL?
21 003530 001405          BEQ      3$          ;BR IF YES
22 003532 062702 000000G  2$:  ADD      #SFCBSZ,R2    ;POINT TO NEXT SFCB
23 003536 000762          BR       4$          ;GO CHECK IT
24         ; Can't find SFCB.
25 003540 000261  1$:  SEC          ;SIGNAL ERROR ON RETURN
26 003542 000403          BR       5$
27         ; Found the SFCB. So get SDCB
28 003544 000241  3$:  CLC          ;SIGNAL SUCCESS ON RETURN
29 003546 016201 000000G      MOV      SFSDCB(R2),R1 ;GET SDCB
30
31 003552 012602  5$:  MOV      (SP)+,R2    ;RESTORE R2
32 003554 000207          RETURN

```

```
1 .SBTTL 5 X 7 Character tables  
2  
3 CHRTBL:  
4 BEGINC  
5 LINE < >  
6 LINE < >  
7 LINE < >  
8 LINE < >  
9 LINE < >  
10 LINE < >  
11 LINE < >  
12 ENDCHR  
13 BEGINC  
14 LINE < !! >  
15 LINE < !! >  
16 LINE < !! >  
17 LINE < !! >  
18 LINE < !! >  
19 LINE < !! >  
20 LINE < !! >  
21 ENDCHR  
22 BEGINC  
23 LINE < " " >  
24 LINE < " " >  
25 LINE < " " >  
26 LINE < >  
27 LINE < >  
28 LINE < >  
29 LINE < >  
30 ENDCHR  
31 BEGINC  
32 LINE < # # >  
33 LINE < # # >  
34 LINE <#####>  
35 LINE < # # >  
36 LINE <#####>  
37 LINE < # # >  
38 LINE < # # >  
39 ENDCHR  
40 BEGINC  
41 LINE < $$$ >  
42 LINE <$ $ $ >  
43 LINE <$ $ >  
44 LINE < $$$ >  
45 LINE < $ $ >  
46 LINE <$ $ $ >  
47 LINE < $$$ >  
48 ENDCHR  
49 BEGINC  
50 LINE <% % >  
51 LINE <% % >  
52 LINE < % >  
53 LINE < % >  
54 LINE < % >  
55 LINE <% %>  
56 LINE <% %>  
57 ENDCHR
```

58 003614	BEGINC
59 003614	LINE < && >
60 003614	LINE < & & >
61 003614	LINE < & & >
62 003614	LINE < & && >
63 003614	LINE < & & & >
64 003614	LINE < & & >
65 003614	LINE < && & >
66 003614	ENDCHR
67 003621	BEGINC
68 003621	LINE < ' ' >
69 003621	LINE < ' ' >
70 003621	LINE < ' ' >
71 003621	LINE < ' ' >
72 003621	LINE < ' ' >
73 003621	LINE < ' ' >
74 003621	LINE < ' ' >
75 003621	ENDCHR
76 003626	BEGINC
77 003626	LINE < (>
78 003626	LINE < (>
79 003626	LINE < (>
80 003626	LINE < (>
81 003626	LINE < (>
82 003626	LINE < (>
83 003626	LINE < (>
84 003626	ENDCHR
85 003633	BEGINC
86 003633	LINE <) >
87 003633	LINE <) >
88 003633	LINE <) >
89 003633	LINE <) >
90 003633	LINE <) >
91 003633	LINE <) >
92 003633	LINE <) >
93 003633	ENDCHR
94 003640	BEGINC
95 003640	LINE < * >
96 003640	LINE < * * * >
97 003640	LINE < *** >
98 003640	LINE < ***** >
99 003640	LINE < *** >
100 003640	LINE < * * * >
101 003640	LINE < * >
102 003640	ENDCHR
103 003645	BEGINC
104 003645	LINE < + >
105 003645	LINE < + >
106 003645	LINE < + >
107 003645	LINE < + + + + + >
108 003645	LINE < + >
109 003645	LINE < + >
110 003645	LINE < + >
111 003645	ENDCHR
112 003652	BEGINC
113 003652	LINE < >
114 003652	LINE < >

115 003652	LINE	< >
116 003652	LINE	< >
117 003652	LINE	< ,, >
118 003652	LINE	< ,, >
119 003652	LINE	< ,, >
120 003652	ENDCHR	
121 003657	BEGINC	
122 003657	LINE	< >
123 003657	LINE	< >
124 003657	LINE	< >
125 003657	LINE	<----->
126 003657	LINE	< >
127 003657	LINE	< >
128 003657	LINE	< >
129 003657	ENDCHR	
130 003664	BEGINC	
131 003664	LINE	< >
132 003664	LINE	< >
133 003664	LINE	< >
134 003664	LINE	< >
135 003664	LINE	< >
136 003664	LINE	< .. >
137 003664	LINE	< .. >
138 003664	ENDCHR	
139 003671	BEGINC	
140 003671	LINE	< >
141 003671	LINE	< / >
142 003671	LINE	< / >
143 003671	LINE	< / >
144 003671	LINE	< / >
145 003671	LINE	< / >
146 003671	LINE	< / >
147 003671	ENDCHR	
148 003676	BEGINC	
149 003676	LINE	< 000 >
150 003676	LINE	<0 0>
151 003676	LINE	<0 00>
152 003676	LINE	<0 0 0>
153 003676	LINE	<00 0>
154 003676	LINE	<0 0>
155 003676	LINE	< 000 >
156 003676	ENDCHR	
157 003703	BEGINC	
158 003703	LINE	< 1 >
159 003703	LINE	< 11 >
160 003703	LINE	<1 1 >
161 003703	LINE	< 1 >
162 003703	LINE	< 1 >
163 003703	LINE	< 1 >
164 003703	LINE	<11111>
165 003703	ENDCHR	
166 003710	BEGINC	
167 003710	LINE	< 222 >
168 003710	LINE	<2 2>
169 003710	LINE	< 2 >
170 003710	LINE	< 2 >
171 003710	LINE	< 2 >

172 003710	LINE	<2 >
173 003710	LINE	<22222>
174 003710	ENDCHR	
175 003715	BEGINC	
176 003715	LINE	< 333 >
177 003715	LINE	<3 3>
178 003715	LINE	< 3>
179 003715	LINE	< 33 >
180 003715	LINE	< 3>
181 003715	LINE	<3 3>
182 003715	LINE	< 333 >
183 003715	ENDCHR	
184 003722	BEGINC	
185 003722	LINE	< 4 >
186 003722	LINE	< 44 >
187 003722	LINE	< 4 4 >
188 003722	LINE	<44444>
189 003722	LINE	< 4 >
190 003722	LINE	< 4 >
191 003722	LINE	< 4 >
192 003722	ENDCHR	
193 003727	BEGINC	
194 003727	LINE	<55555>
195 003727	LINE	<5 >
196 003727	LINE	<5 >
197 003727	LINE	<5555 >
198 003727	LINE	< 5>
199 003727	LINE	<5 5>
200 003727	LINE	< 555 >
201 003727	ENDCHR	
202 003734	BEGINC	
203 003734	LINE	< 666 >
204 003734	LINE	<6 >
205 003734	LINE	<6 >
206 003734	LINE	<6666 >
207 003734	LINE	<6 6>
208 003734	LINE	<6 6>
209 003734	LINE	< 666>
210 003734	ENDCHR	
211 003741	BEGINC	
212 003741	LINE	<77777>
213 003741	LINE	< 7>
214 003741	LINE	< 7 >
215 003741	LINE	< 7 >
216 003741	LINE	< 7 >
217 003741	LINE	<7 >
218 003741	LINE	<7 >
219 003741	ENDCHR	
220 003746	BEGINC	
221 003746	LINE	< 888 >
222 003746	LINE	<8 8>
223 003746	LINE	<8 8>
224 003746	LINE	< 888 >
225 003746	LINE	<8 8>
226 003746	LINE	<8 8>
227 003746	LINE	< 888 >
228 003746	ENDCHR	

229	003753	BEGINC
230	003753	LINE < 999 >
231	003753	LINE < 9 9 >
232	003753	LINE < 9 9 >
233	003753	LINE < 9999 >
234	003753	LINE < 9 >
235	003753	LINE < 9 >
236	003753	LINE < 999 >
237	003753	ENDCHR
238	003760	BEGINC
239	003760	LINE < >
240	003760	LINE < :: >
241	003760	LINE < :: >
242	003760	LINE < >
243	003760	LINE < :: >
244	003760	LINE < :: >
245	003760	LINE < >
246	003760	ENDCHR
247	003765	BEGINC
248	003765	LINE < >
249	003765	LINE < ; >
250	003765	LINE < ; >
251	003765	LINE < >
252	003765	LINE < ; >
253	003765	LINE < ; >
254	003765	LINE < ; >
255	003765	ENDCHR
256	003772	BEGINC
257	003772	LINE < () >
258	003772	LINE < () >
259	003772	LINE < () >
260	003772	LINE < () >
261	003772	LINE < () >
262	003772	LINE < () >
263	003772	LINE < () >
264	003772	ENDCHR
265	003777	BEGINC
266	003777	LINE < >
267	003777	LINE < >
268	003777	LINE < ===== >
269	003777	LINE < >
270	003777	LINE < ===== >
271	003777	LINE < >
272	003777	LINE < >
273	003777	ENDCHR
274	004004	BEGINC
275	004004	LINE <) >
276	004004	LINE <) >
277	004004	LINE <) >
278	004004	LINE <) >
279	004004	LINE <) >
280	004004	LINE <) >
281	004004	LINE <) >
282	004004	ENDCHR
283	004011	BEGINC
284	004011	LINE < ??? >
285	004011	LINE < ? ? >

286 004011	LINE	< ? >
287 004011	LINE	< ? >
288 004011	LINE	< ? >
289 004011	LINE	< ? >
290 004011	LINE	< ? >
291 004011	ENDCHR	
292 004016	BEGINC	
293 004016	LINE	< @@@ >
294 004016	LINE	<@ @>
295 004016	LINE	<@ @>
296 004016	LINE	<@ @@@>
297 004016	LINE	<@ @@@>
298 004016	LINE	<@ @>
299 004016	LINE	< @@@ >
300 004016	ENDCHR	
301 004023	BEGINC	
302 004023	LINE	< A >
303 004023	LINE	< A A >
304 004023	LINE	<A A>
305 004023	LINE	<AAAAA>
306 004023	LINE	<A A>
307 004023	LINE	<A A>
308 004023	LINE	<A A>
309 004023	ENDCHR	
310 004030	BEGINC	
311 004030	LINE	<BBBB >
312 004030	LINE	<B B>
313 004030	LINE	<B B>
314 004030	LINE	<BBBB>
315 004030	LINE	<B B>
316 004030	LINE	<B B>
317 004030	LINE	<BBBB>
318 004030	ENDCHR	
319 004035	BEGINC	
320 004035	LINE	< CCCC >
321 004035	LINE	<C >
322 004035	LINE	<C >
323 004035	LINE	<C >
324 004035	LINE	<C >
325 004035	LINE	<C >
326 004035	LINE	< CCCC >
327 004035	ENDCHR	
328 004042	BEGINC	
329 004042	LINE	<DDD >
330 004042	LINE	<D D >
331 004042	LINE	<D D>
332 004042	LINE	<D D>
333 004042	LINE	<D D>
334 004042	LINE	<D D >
335 004042	LINE	<DDD >
336 004042	ENDCHR	
337 004047	BEGINC	
338 004047	LINE	<EEEEEE>
339 004047	LINE	<E >
340 004047	LINE	<E >
341 004047	LINE	<EEEE >
342 004047	LINE	<E >

343	004047	LINE	<E >
344	004047	LINE	<EEEEEE>
345	004047	ENDCHR	
346	004054	BEGINC	
347	004054	LINE	<FFFFFF>
348	004054	LINE	<F >
349	004054	LINE	<F >
350	004054	LINE	<FFFF >
351	004054	LINE	<F >
352	004054	LINE	<F >
353	004054	LINE	<F >
354	004054	ENDCHR	
355	004061	BEGINC	
356	004061	LINE	< GGGG>
357	004061	LINE	<G >
358	004061	LINE	<G >
359	004061	LINE	<G GG>
360	004061	LINE	<G G>
361	004061	LINE	<G G>
362	004061	LINE	< GGG >
363	004061	ENDCHR	
364	004066	BEGINC	
365	004066	LINE	<H H>
366	004066	LINE	<H H>
367	004066	LINE	<H H>
368	004066	LINE	<HHHHH>
369	004066	LINE	<H H>
370	004066	LINE	<H H>
371	004066	LINE	<H H>
372	004066	ENDCHR	
373	004073	BEGINC	
374	004073	LINE	<IIIII>
375	004073	LINE	< I >
376	004073	LINE	< I >
377	004073	LINE	< I >
378	004073	LINE	< I >
379	004073	LINE	< I >
380	004073	LINE	<IIIII>
381	004073	ENDCHR	
382	004100	BEGINC	
383	004100	LINE	< J>
384	004100	LINE	< J>
385	004100	LINE	< J>
386	004100	LINE	< J>
387	004100	LINE	<J J>
388	004100	LINE	<J J>
389	004100	LINE	< JJJ >
390	004100	ENDCHR	
391	004105	BEGINC	
392	004105	LINE	<K K>
393	004105	LINE	<K K >
394	004105	LINE	<K K >
395	004105	LINE	<KK >
396	004105	LINE	<K K >
397	004105	LINE	<K K >
398	004105	LINE	<K K>
399	004105	ENDCHR	

400	004112	BEGINC
401	004112	LINE <L >
402	004112	LINE <L >
403	004112	LINE <L >
404	004112	LINE <L >
405	004112	LINE <L >
406	004112	LINE <L >
407	004112	LINE <LLLLL>
408	004112	ENDCHR
409	004117	BEGINC
410	004117	LINE <M M>
411	004117	LINE <MM MM>
412	004117	LINE <M M M>
413	004117	LINE <M M>
414	004117	LINE <M M>
415	004117	LINE <M M>
416	004117	LINE <M M>
417	004117	ENDCHR
418	004124	BEGINC
419	004124	LINE <N N>
420	004124	LINE <NN N>
421	004124	LINE <N N N>
422	004124	LINE <N NN>
423	004124	LINE <N N>
424	004124	LINE <N N>
425	004124	LINE <N N>
426	004124	ENDCHR
427	004131	BEGINC
428	004131	LINE < 000 >
429	004131	LINE <0 0>
430	004131	LINE <0 0>
431	004131	LINE <0 0>
432	004131	LINE <0 0>
433	004131	LINE <0 0>
434	004131	LINE < 000 >
435	004131	ENDCHR
436	004136	BEGINC
437	004136	LINE <PPPP >
438	004136	LINE <P P>
439	004136	LINE <P P>
440	004136	LINE <PPPP >
441	004136	LINE <P >
442	004136	LINE <P >
443	004136	LINE <P >
444	004136	ENDCHR
445	004143	BEGINC
446	004143	LINE < QQQ >
447	004143	LINE <Q Q>
448	004143	LINE <Q Q>
449	004143	LINE <Q Q>
450	004143	LINE <Q Q Q>
451	004143	LINE <Q QQ>
452	004143	LINE < QQQ>
453	004143	ENDCHR
454	004150	BEGINC
455	004150	LINE <RRRR >
456	004150	LINE <R R>

457 004150	LINE	<R R>
458 004150	LINE	<RRRR >
459 004150	LINE	<R R >
460 004150	LINE	<R R >
461 004150	LINE	<R R>
462 004150	ENDCHR	
463 004155	BEGINC	
464 004155	LINE	< SSS >
465 004155	LINE	<S S>
466 004155	LINE	<S >
467 004155	LINE	< SSS >
468 004155	LINE	< S>
469 004155	LINE	<S S>
470 004155	LINE	< SSS >
471 004155	ENDCHR	
472 004162	BEGINC	
473 004162	LINE	<TTTTT>
474 004162	LINE	< T >
475 004162	LINE	< T >
476 004162	LINE	< T >
477 004162	LINE	< T >
478 004162	LINE	< T >
479 004162	LINE	< T >
480 004162	ENDCHR	
481 004167	BEGINC	
482 004167	LINE	<U U>
483 004167	LINE	<U U>
484 004167	LINE	<U U>
485 004167	LINE	<U U>
486 004167	LINE	<U U>
487 004167	LINE	<U U>
488 004167	LINE	< UUU >
489 004167	ENDCHR	
490 004174	BEGINC	
491 004174	LINE	<V V>
492 004174	LINE	<V V>
493 004174	LINE	<V V>
494 004174	LINE	<V V>
495 004174	LINE	<V V>
496 004174	LINE	< V V >
497 004174	LINE	< V >
498 004174	ENDCHR	
499 004201	BEGINC	
500 004201	LINE	<W W>
501 004201	LINE	<W W>
502 004201	LINE	<W W>
503 004201	LINE	<W W>
504 004201	LINE	<W W W>
505 004201	LINE	<WW WW>
506 004201	LINE	<W W>
507 004201	ENDCHR	
508 004206	BEGINC	
509 004206	LINE	<X X>
510 004206	LINE	<X X>
511 004206	LINE	< X X >
512 004206	LINE	< X >
513 004206	LINE	< X X >

514 004206	LINE	<X	X>
515 004206	LINE	<X	X>
516 004206	ENDCHR		
517 004213	BEGINC		
518 004213	LINE	<Y	Y>
519 004213	LINE	<Y	Y>
520 004213	LINE	<Y	Y>
521 004213	LINE	< Y Y >	
522 004213	LINE	< Y >	
523 004213	LINE	< Y >	
524 004213	LINE	< Y >	
525 004213	ENDCHR		
526 004220	BEGINC		
527 004220	LINE	<ZZZZZ>	
528 004220	LINE	< Z >	
529 004220	LINE	< Z >	
530 004220	LINE	< Z >	
531 004220	LINE	< Z >	
532 004220	LINE	<Z >	
533 004220	LINE	<ZZZZZ>	
534 004220	ENDCHR		
535 004225	BEGINC		
536 004225	LINE	<[[[>	
537 004225	LINE	<[>	
538 004225	LINE	<[>	
539 004225	LINE	<[>	
540 004225	LINE	<[>	
541 004225	LINE	<[>	
542 004225	LINE	<[[[>	
543 004225	ENDCHR		
544 004232	BEGINC		
545 004232	LINE	< >	
546 004232	LINE	< \ >	
547 004232	LINE	< \ >	
548 004232	LINE	< \ >	
549 004232	LINE	< \ >	
550 004232	LINE	< \ >	
551 004232	LINE	< \ >	
552 004232	ENDCHR		
553 004237	BEGINC		
554 004237	LINE	<]]] >	
555 004237	LINE	<] >	
556 004237	LINE	<] >	
557 004237	LINE	<] >	
558 004237	LINE	<] >	
559 004237	LINE	<] >	
560 004237	LINE	<]]] >	
561 004237	ENDCHR		
562 004244	BEGINC		
563 004244	LINE	< ^ ^ >	
564 004244	LINE	< ^ ^ >	
565 004244	LINE	< ^ ^ >	
566 004244	LINE	< >	
567 004244	LINE	< >	
568 004244	LINE	< >	
569 004244	LINE	< >	
570 004244	ENDCHR		

571 004251	BEGINC
572 004251	LINE < >
573 004251	LINE < >
574 004251	LINE < >
575 004251	LINE < >
576 004251	LINE < >
577 004251	LINE < >
578 004251	LINE < >
579 004251	ENDCHR
580 004256	BEGINC
581 004256	LINE < ' ' >
582 004256	LINE < ' ' >
583 004256	LINE < ' ' >
584 004256	LINE < >
585 004256	LINE < >
586 004256	LINE < >
587 004256	LINE < >
588 004256	ENDCHR
589 004263	BEGINC
590 004263	LINE < { } >
591 004263	LINE < { } >
592 004263	LINE < { } >
593 004263	LINE < { { } >
594 004263	LINE < { } >
595 004263	LINE < { } >
596 004263	LINE < { } >
597 004263	ENDCHR
598 004270	BEGINC
599 004270	LINE < >
600 004270	LINE < >
601 004270	LINE < >
602 004270	LINE < >
603 004270	LINE < >
604 004270	LINE < >
605 004270	LINE < >
606 004270	ENDCHR
607 004275	BEGINC
608 004275	LINE < { } } >
609 004275	LINE < } } >
610 004275	LINE < } } >
611 004275	LINE < } } } >
612 004275	LINE < } } >
613 004275	LINE < } } >
614 004275	LINE < { } } >
615 004275	ENDCHR
616 004302	BEGINC
617 004302	LINE < ~ ~ >
618 004302	LINE < ~ ~ ~ >
619 004302	LINE < ~ ~ >
620 004302	LINE < >
621 004302	LINE < >
622 004302	LINE < >
623 004302	LINE < >
624 004302	ENDCHR
625 004307	BEGINC
626 004307	LINE < ? ? ? >
627 004307	LINE < ? ? >

5 X 7 Character tables

```
628 004307      LINE    <? ? ?>
629 004307      LINE    < ? ? >
630 004307      LINE    <? ? ?>
631 004307      LINE    < ? ? >
632 004307      LINE    <? ? ?>
633 004307      ENDCHR
```

634

635

636 004314

000056

637

638

639

640 004372

000056

641

642

643

644

000001

SITNAM: .EVEN

;*** SITNAM must start on an even address ***

;Fill with end-of-string markers

.REPT 46.

.BYTE 200

.ENDR

SITLEN: .WORD .-SITNAM

;Actually filled in when site name is set

.EVEN

.END

Errors detected: 0

*** Assembler statistics

Work file reads: 0

Work file writes: 0

Size of work file: 8593 Words (34 Pages)

Size of core pool: 18176 Words (71 Pages)

Operating system: RT-11

Elapsed time: 00:02:31.31

,LP:TSSPL2=DK:TSSPL2/C/N:SYM

GSFTBL	15-13	15-17#	15-19						
GTLICN	1-92	15-8#							
JANURY	13-74	13-88#							
JULIAN	11-8#	12-11							
JULY	13-80	13-94#							
JUNE	13-79	13-93#							
LEAP	10-32#	11-16							
LINEND	2-117	2-151#							
LINSTR	2-119	2-150#							
LPROG	1-88	2-88							
LPROJ	1-88	2-98							
LUNAME	1-88	2-45							
MARCH	13-76	13-90#							
MAY	13-78	13-92#							
MONDAY	13-103	13-112#							
MONTAB	13-48	13-49	13-74#						
MONTH#	10-6#								
NVEMBR	13-84	13-98#							
OCTOBR	13-83	13-97#							
OVRHC	2-140	5-69							
PRTBUF	2-64	2-80	2-103	2-122	6-12#				
R50TBL	4-21	4-24	4-27	4-33#					
SATRDY	13-108	13-117#							
SD*FLG	1-91	16-20	16-21	16-24					
SD*RSV	1-87	16-10	17-11						
SD*WID	1-87	1-91	2-18	17-21	17-22	17-25			
SDFLAG	1-87	1-91	2-18	16-19	16-21*	16-24*	17-20	17-22*	17-25*
SETC	1-92	15-11							
SETSTN	1-89	14-5#							
SFCB	1-90	18-15							
SFCBND	1-90	18-16							
SFCBSZ	1-91	18-22							
SFCHAN	1-90	18-20							
SFFILE	1-88	2-74	2-76						
SFLGPG	1-88	2-140	5-69						
SFSDCB	1-91	18-29							
SFUSER	1-87	1-90	2-40	2-86	2-113	18-18			
SITLEN	2-30	14-14*	15-35	15-41	19-640#				
SITNAM	1-87	2-31	14-6	14-9	14-13	15-34	19-636#	19-640	
SPACE	1-16#								
SPLSFF	1-91	16-9#							
SPLSWF	1-92	17-10#							
SPTMBR	13-82	13-96#							
STREND	5-24	5-47	5-74#						
SUNDAY	13-102	13-111#							
SYSDAT	1-89	13-11							
SYTIMH	1-88	8-10							
SYTIML	1-89	8-11							
THRSDY	13-106	13-115#							
TK1SEC	1-89	8-12							
TSSPL2	1-12#								
TSTEND	5-23	5-35	5-42	5-46	5-66#	6-38	6-57	6-105	
TSXSIT	1-92	15-24							
TSXSTR	5-20	5-26	5-49	5-73#					
TUESDY	13-104	13-113#							
URO	1-92	15-24*	15-41*	16-10*	16-19*	16-20*	17-11*	17-20*	17-21*

VALADW	1-92	15-33		
WDNSDY	13-105	13-114#		
YEAR\$	10-4#	10-41	12-14	13-28

