

TSKM2D -- System once-only code MACRO V05.05 Thursday 19-Jan-89 14:57
Table of contents

2- 1 KMINIT - System once only code
7- 1 Select Pro or ordinary site name

1
2
3
4 000000
5 000000
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

000404

```

        .TITLE  TSKM2D -- System once-only code
        .ENABL  LC
        .DSABL  GBL
        .PSECT  TSKM2D
TSKM2D:
;
;   TSKM2D is the portion of TSKMON that performs system initialization
;   code that is executed only once each time the system is started.
;   The first job to enter TSKMON is the one that executes this code.
;
;   Copyright 1988.
;   S&H Computer Systems, Inc.
;   Nashville, Tennessee
;
;   Macro calls
;
;       .MCALL  .LOOKUP, .CSTAT, .CLOSE
;
;   Global definitions
;
;       .GLOBL  KMINIT
;
;   Global references
;
;       .GLOBL  KMONCE, EXCJOB, DOSCHD, CXTRMN, MNUAOT, VMNUAD
;       .GLOBL  NUMDEV, PNAME, HANDSK, HANENT, DMYDEV, SYNAME
;       .GLOBL  SITE, SPLND, CSIBUF, TSXSIT, PROFLG
;
;   Local definitions
;
;   PNPTR  = 404           ;Fixed offset to cell holding offset to PNAME
;
;   Define macro to encrypt a text string
;
;       .MACRO  ENC      TEXT
;       .IRPC  X, <TEXT>
;       .BYTE  -<'X>
;       .ENDR
;       .ENDM  ENC
    
```

KMINIT - System once only code

```

1           .SBTTL  KMINIT - System once only code
2           ;-----
3           ; This routine is called once each time the system is restarted by the
4           ; first job to enter kmon. That job has disabled the scheduler by putting
5           ; its own job index into EXCJOB so that this code has a chance to execute
6           ; before any other job can get started.
7           ;
8           ; On entry, R1 and EXCJOB contain the job index
9           ;
10          ; On exit, EXCJOB is cleared, re-enabling the job scheduler
11          ;
12 000000 010046 KMINIT: MOV     R0,-(SP)      ;Save registers
13 000002 010146          MOV     R1,-(SP)
14 000004 010346          MOV     R3,-(SP)

```

KMINIT - System once only code

```

1
2 ; -----
3 ; Set max # jobs that can be logged on at one time.
4 ; This is set in TSNAME during the distribution process to be
5 ; 3 + maxjob*975. If maxjob = 0, then there are no restrictions.
6 ;
7 000006 SETMXJ:
8 000006 012701 000000G MOV #MNUAOT,R1 ;Get lightly encrypted max # jobs
9 000012 162701 000003 SUB #3,R1 ;Remove offset
10 000020 005000 BEQ 11$ ;If no restrictions, skip divide
11 000022 071027 001717 CLR R0 ;set for divide
12 000026 005701 DIV #975.,R0 ;Remove factor
13 000030 001004 TST R1 ;Shouldn't be any remainder
14 000032 010001 BNE 1$ ;If there is, then # is invalid
15 000034 020127 000005 MOV R0,R1 ;Get maxjob into R1
16 000040 101402 CMP R1,#5 ;Should be <= 5 for micro
17 000042 012701 000001 BLOS 11$ ;Br if in valid range (0-5)
18 000046 012700 000123 1$: MOV #1,R1 ;If out of range, this becomes 1 user
19 000052 074001 11$: MOV #123,R0 ;Set for XOR
20 000054 110167 000000G XOR R0,R1 ;Encrypt max # jobs again
MOV R1,VMNUAD ;Set re-encrypted maxjob into TSGEN

```

```

1          ; -----
2          ;   Set table of device handler disk block numbers (HANDSK).
3          ;
4          ;   Determine virtual address of this job's RMON HANDSK table
5          ;
6 000060   SETHBT:
7 000060   016701 000000G   MOV     CXTRMN,R1      ;Get virtual address of our own RMON
8 000064   062701 000002   ADD     #2,R1         ;CXTRMN points to VECBAS, bump to MONVEC
9 000070   016100 000404   MOV     PNPTR(R1),R0  ;Get offset to RMON PNAME table
10 000074   060001          ADD     R0,R1         ;Convert to virtual address of RMON PNAME
11 000076   012700 000000G   MOV     #HANDSK,R0   ;Get kernel HANDSK address
12 000102   162700 000000G   SUB     #PNAME,R0    ;Get kernel offset from PNAME to HANDSK
13 000106   060001          ADD     R0,R1         ;Convert to virtual address of RMON HANDSK
14          ;
15          ;   Scan through kernel device tables to correct the HANDSK table of
16          ;   handler block 1 offsets from offsets relative to start of the handler
17          ;   file (1) required during TSINIT to offsets relative to the start of
18          ;   the disk which may be required later by utilities.
19          ;
20 000110   016767 000000G 000270   MOV     SYNAME,DEVSPC ;Set boot device name into filspc
21 000116   016703 000000G          MOV     NUMDEV,R3    ;Get index of last device loaded
22 000122   026327 000000G 000002 3#:   CMP     HANENT(R3),#2 ;Does this device have a handler file?
23 000130   101445          BLOS    7#           ;Br if not, ignore handler
24 000132   016300 000000G          MOV     PNAME(R3),R0 ;Get handler name
25 000136   001442          BEQ     7#           ;Ignore if empty device name
26 000140   020027 000000G          CMP     R0,#DMYDEV   ;Dummy device name?
27 000144   001437          BEQ     7#           ;Br if so, ignore dummy device entries
28 000146   010067 000236          MOV     R0,FILNAM    ;Put name into device spec
29 000152          .LOOKUP #AREA,#1,#DEVSPC ;Try to open channel to handler file
30 000172   103424          BCS     7#           ;Ignore device if can't find handler file
31 000174          .CSTAT #AREA,#1,#INFO  ;Ask for channel information
32 000214   103410          BCS     6#           ;Ignore on error but close file
33 000216   016700 000176          MOV     INFO+2,R0    ;Get starting disk block number of handler
34 000222   005200          INC     R0           ;Bump up to block 1
35 000224   010063 000000G          MOV     R0,HANDSK(R3) ;Set into device tables
36          ;
37          ;   Now, the simulated RMON of this job got a copy of the system tables
38          ;   as set up during TSINIT, which still have the relative block numbers,
39          ;   so we need to correct our own tables also.
40          ;
41 000230   060103          ADD     R1,R3        ;Index to current HANDSK entry address
42 000232   010013          MOV     R0,(R3)      ;Correct our own copy
43 000234   160103          SUB     R1,R3        ;Convert back to current device index
44          ;
45 000236          6#: .CLOSE #1         ;Close channel to handler file
46 000244   162703 000002 7#:   SUB     #2,R3        ;Drop down to next device index
47 000250   002324          BGE     3#           ;Repeat through entire device table (numdev-0)

```

KMINIT - System once only code

```

1
2 ; -----
3 ; Initialize site name string in spool overlay
4
5 SETSIT:
6     TST     #SPLND           ;Was the spooling system included?
7     BEQ     DONE            ;Skip if not (overlay probably not loaded)
8     MOV     #CSIBUF,R3      ;Point to intermediate buffer in context block
9     CALL    PIKSIT          ;Get ptr to encrypted site name in R1
10    1$:     MOVB    (R1)+,R0   ;Get next char in site name
11           NEG     R0         ;Decrypt it
12           BLE    2$         ;0 or 200 is done
13           CMP     R0,#15     ;Carriage return is also end
14           BEQ    2$
15           CMP     R3,#CSIBUF+45. ;Also truncate at max string length
16           BHIS   2$
17           MOVB   R0,(R3)+    ;Keep this char
18           BR     1$         ;Loop thru site name string
19
20 ; Body of site name is moved, make sure it is at least 1 char long
21 ; and is terminated with a 200.
22
23 2$:     CMP     R3,#CSIBUF    ;Did we get anything?
24           BHI    3$         ;Br if so
25           MOVB   #40,(R3)+   ;Set in 1 space char if site name is empty
26           MOVB   #200,(R3)+ ;And always terminate string
27
28 ; Now call emt in TSSPL2 to copy site name into flag page overlay
29
30     MOV     #SETSTN,R0      ;Point to emt arg block to
31     EMT    375              ;Set site name
                               ;Ignore errors

```

```

1      ; -----
2      ;   Finish
3      ;
4 000342      ;
5 000342 012603      ;
6 000344 012601      ;
7 000346 012600      ;
8      ;
9 000350 105067 0000000      ;
10 000354 105067 0000000      ;
11 000360 105267 0000000      ;
12 000364 000207      ;
13      ;
14 000366      ;
15 000406 075250      ;
16 000410 031066      ;
17 000412 000000 100020      ;
18 000416      ;
19 000432      000      126      ;
20 000434 000021      ;
21      ;

```

```

;
;   Finish
;
DONE:
      MOV      (SP)+,R3      ;Restore registers
      MOV      (SP)+,R1
      MOV      (SP)+,R0
;
      CLRB     KMONCE      ;Remember once-only code has been done
      CLRB     EXCJOB      ;Re-enable job-scheduling
      INCB     DOSCHD      ;Make sure scheduler gets called
      RETURN
;
AREA:      .BLKW      10      ;General EMT arg block
DEVSPC:    .RAD50     /SY /    ;Replaced with real boot device
FILNAM:    .RAD50     /HAN/    ;Replaced from PNAME table
          .RAD50     /  TSX/    ;Should always be empty
INFO:      .BLKW      6      ;CSTAT return information block
SETSTN:    .BYTE     0,126     ;KMON EMT arg block
          .WORD     21      ; sub-function: set site name into TSSPL2
;

```

Select Pro or ordinary site name

```

1          .SBTTL  Select Pro or ordinary site name
2          ;-----
3          ; Pro distributions don't have a "real" embedded site name, so if we are
4          ; running on a Pro, we have to fake a site name; if not, just point to the
5          ; real site name.
6          ;
7 000436 010446  PIKSIT: MOV      R4, -(SP)
8 000440 010546          MOV      R5, -(SP)
9          ;
10 000442 012701 000000G  MOV      #SITE, R1      ;Point to encrypted site name
11 000446 105767 000000G  TSTB    PROFLG         ;Are we running on a Pro?
12 000452 001425          BEQ     9$              ;If not, that's all we have to do!
13          ; Convert internal license number to ascii
14 000454 005000          CLR     R0              ;Count number of digits
15 000456 016705 000000G  MOV     TSXSIT, R5     ;Get site license number
16 000462 005004 1$: CLR     R4              ;Set for divide
17 000464 071427 000012  DIV     #10, R4        ;Split out low digit R5, quotient in R4
18 000470 062705 000060  ADD     #'0, R5        ;Convert least significant digit to ascii
19 000474 110546          MOVVB   R5, -(SP)      ;Save binary digit on stack
20 000476 005200          INC     R0              ;Count a digit saved
21 000500 010405          MOV     R4, R5         ;Copy quotient for next divide
22 000502 001367          BNE    1$              ;Loop until insignificant
23          ; Now hang the license number at the end of the fake site name
24 000504 012701 000574'  MOV     #PSEND, R1     ;Point to end of name string
25 000510 11260b 2$: MOVVB  (SP)+, R5     ;Get back ascii digit
26 000512 005405          NEG     R5              ;Encrypt it
27 000514 110521          MOVVB  R5, (R1)+      ;And hang at string end
28 000516 077004          SOB   R0, 2$         ;Loop thru all
29 000520 105011          CLRB   @R1            ;Terminate string
30          ; And end by pointing to the fake pro site name string
31 000522 012701 000534'  MOV     #PROSIT, R1    ;Point to encrypted site name
32          ;
33 000526 012605 9$: MOV     (SP)+, R5
34 000530 012604          MOV     (SP)+, R4
35 000532 000207          RETURN
36          ;
37          ; Define a fake site name to be filled in for Pro systems
38          ;
39 000534          PROSIT:
40          .NLIST  BEX
41 000534          ENC    <PRO/TSX-Plus License # 999-TPS->
42 000574          PSEND: .BLKB  6              ;Leave room for 5 digit number and terminator
43          .EVEN
44          .LIST  BEX
45
46          000001          .END

```

Errors detected: 0

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 9533 Words (38 Pages)
Size of core pool: 18176 Words (71 Pages)
Operating system: RT-11

Elapsed time: 00:00:07.11

.LP: TSKM2D=DK: TSKM2D/C/N: SYM

