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

```
                                        .TITLE  TSKM2D -- System once-only code
                                        .ENABL  LC
                                        .DSABL  GBL
000000                                  .PSECT  TSKM2D
000000                          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,VMNUAO
                                        .GLOBL  NUMDEV,PNAME,HANDSK,HANENT,DMYDEV,SYNAME
                                        .GLOBL  SITE,SPLND,CSIBUF,TSXSIT,PROFLG
                                ;
                                ;  Local definitions
                                ;
000404                          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
```

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

```
                                ;-------------------------------------------------------------------------------
                                ;  Set max # jobs that can be logged on at one time.
                                ;  This is set in TSNAME during the distribution process to be
                                ;  3 + maxjob*975.  If maxjob = 0, then there are no restrictions.
                                ;
000006                          SETMXJ:
000006  012701  000000G                MOV     #MNUAOT,R1      ;Get lightly encrypted max # jobs
000012  162701  000003                SUB     #3,R1           ;Remove offset
000016  001413                        BEQ     11$             ;If no restrictions, skip divide
000020  005000                        CLR     R0              ;set for divide
000022  071027  001717                DIV     #975.,R0        ;Remove factor
000026  005701                        TST     R1              ;Shouldn't be any remainder
000030  001004                        BNE     1$              ;If there is, then # is invalid
000032  010001                        MOV     R0,R1           ;Get maxjob into R1
000034  020127  000005                CMP     R1,#5           ;Should be <= 5 for micro
000040  101402                        BLOS    11$             ;Br if in valid range (0-5)
000042  012701  000001        1$:     MOV     #1,R1           ;If out of range, this becomes 1 user
000046  012700  000123        11$:    MOV     #123,R0         ;Set for XOR
000052  074001                        XOR     R0,R1           ;Encrypt max # jobs again
000054  110167  000000G               MOVB    R1,VMNUAO       ;Set re-encrypted maxjob into TSGEN
```

```
                                ;-----------------------------------------------------------------------
                                ;  Set table of device handler disk block numbers (HANDSK).
                                ;
                                ;  Determine virtual address of this job's RMON HANDSK table
                                ;
000060                          SETHBT:
000060  016701  0000006                 MOV     CXTRMN,R1           ;Get virtual address of our own RMON
000064  062701  000002                  ADD     #2,R1               ;CXTRMN points to VECBAS, bump to MONVEC
000070  016100  000404                  MOV     PNPTR(R1),R0        ;Get offset to RMON PNAME table
000074  060001                          ADD     R0,R1               ;Convert to virtual address of RMON PNAME
000076  012700  0000006                 MOV     #HANDSK,R0          ;Get kernel HANDSK address
000102  162700  0000006                 SUB     #PNAME,R0           ;Get kernel offset from PNAME to HANDSK
000106  060001                          ADD     R0,R1               ;Convert to virtual address of RMON HANDSK
                                ;
                                ;  Scan through kernel device tables to correct the HANDSK table of
                                ;  handler block 1 offsets from offsets relative to start of the handler
                                ;  file (1) required during TSINIT to offsets relative to the start of
                                ;  the disk which may be required later by utilities.
                                ;
000110  016767  0000006 000270          MOV     SYNAME,DEVSPC       ;Set boot device name into filspc
000116  016703  0000006                 MOV     NUMDEV,R3           ;Get index of last device loaded
000122  026327  0000006 000002 3$:      CMP     HANENT(R3),#2       ;Does this device have a handler file?
000130  101445                          BLOS    7$                  ;Br if not, ignore handler
000132  016300  0000006                 MOV     PNAME(R3),R0        ;Get handler name
000136  001442                          BEQ     7$                  ;Ignore if empty device name
000140  020027  0000006                 CMP     R0,#DMYDEV          ;Dummy device name?
000144  001437                          BEQ     7$                  ;Br if so, ignore dummy device entries
000146  010067  000236                  MOV     R0,FILNAM           ;Put name into device spec
000152                                  .LOOKUP #AREA,#1,#DEVSPC    ;Try to open channel to handler file
000172  103424                          BCS     7$                  ;Ignore device if can't find handler file
000174                                  .CSTAT  #AREA,#1,#INFO      ;Ask for channel information
000214  103410                          BCS     6$                  ;Ignore on error but close file
000216  016700  000176                  MOV     INFO+2,R0           ;Get starting disk block number of handler
000222  005200                          INC     R0                  ;Bump up to block 1
000224  010063  0000006                 MOV     R0,HANDSK(R3)       ;Set into device tables
                                ;
                                ;  Now, the simulated RMON of this job got a copy of the system tables
                                ;  as set up during TSINIT, which still have the relative block numbers,
                                ;  so we need to correct our own tables also.
                                ;
000230  060103                          ADD     R1,R3               ;Index to current HANDSK entry address
000232  010013                          MOV     R0,(R3)             ;Correct our own copy
000234  160103                          SUB     R1,R3               ;Convert back to current device index
                                ;
000236                          6$:     .CLOSE  #1                  ;Close channel to handler file
000244  162703  000002          7$:     SUB     #2,R3               ;Drop down to next device index
000250  002324                          BGE     3$                  ;Repeat through entire device table (numdev-0)
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Initialize site name string in spool overlay
                                ;
000252                          SETSIT:
000252  005727  000000G                 TST     #SPLND          ;Was the spooling system included?
000256  001431                          BEQ     DONE            ;Skip if not (overlay probably not loaded)
000260  012703  000000G                 MOV     #CSIBUF,R3      ;Point to intermediate buffer in context block
000264  004767  000146                  CALL    PIKSIT          ;Get ptr to encrypted site name in R1
000270  112100                  1$:     MOVB    (R1)+,R0        ;Get next char in site name
000272  005400                          NEG     R0              ;Decrypt it
000274  003410                          BLE     2$              ;0 or 200 is done
000276  020027  000015                  CMP     R0,#15          ;Carriage return is also end
000302  001405                          BEQ     2$
000304  020327  000055G                 CMP     R3,#CSIBUF+45.  ;Also truncate at max string length
000310  103002                          BHIS    2$
000312  110023                          MOVB    R0,(R3)+        ;Keep this char
000314  000765                          BR      1$              ;Loop thru site name string
                                ;
                                ;  Body of site name is moved, make sure it is at least 1 char long
                                ;  and is terminated with a 200.
                                ;
000316  020327  000000G         2$:     CMP     R3,#CSIBUF      ;Did we get anything?
000322  101002                          BHI     3$              ;Br if so
000324  112723  000040                  MOVB    #40,(R3)+       ;Set in 1 space char if site name is empty
000330  112723  000200          3$:     MOVB    #200,(R3)+      ;And always terminate string
                                ;
                                ;  Now call emt in TSSPL2 to copy site name into flag page overlay
                                ;
000334  012700  000432'                 MOV     #SETSTN,R0      ;Point to emt arg block to
000340  104375                          EMT     375             ;Set site name
                                                                ;Ignore errors
```

```
                                ;-----------------------------------------------------------------------
                                ;  Finish
                                ;
000342                          DONE:
000342  012603                          MOV     (SP)+,R3        ;Restore registers
000344  012601                          MOV     (SP)+,R1
000346  012600                          MOV     (SP)+,R0
                                ;
000350  105067  000000G                 CLRB    KMONCE          ;Remember once-only code has been done
000354  105067  000000G                 CLRB    EXCJOB          ;Re-enable job-scheduling
000360  105267  000000G                 INCB    DOSCHD          ;Make sure scheduler gets called
000364  000207                          RETURN
                                ;
000366                          AREA:   .BLKW   10              ;General EMT arg block
000406  075250                  DEVSPC: .RAD50  /SY /            ;Replaced with real boot device
000410  031066                  FILNAM: .RAD50  /HAN/            ;Replaced from PNAME table
000412  000000  100020                  .RAD50  /   TSX/        ;Should always be empty
000416                          INFO:   .BLKW   6               ;CSTAT return information block
000432     000      126         SETSTN: .BYTE   0,126           ;KMON EMT arg block
000434  000021                          .WORD   21              ; sub-function: set site name into TSSPL2
                                ;
```

```
                                        .SBTTL  Select Pro or ordinary site name
                                ;-------------------------------------------------------------------
                                ;  Pro distributions don't have a "real" embedded site name, so if we are
                                ;  running on a Pro, we have to fake a site name; if not, just point to the
                                ;  real site name.
                                ;
000436  010446                  PIKSIT: MOV     R4,-(SP)
000440  010546                          MOV     R5,-(SP)
                                ;
000442  012701  000000G                 MOV     #SITE,R1        ;Point to encrypted site name
000446  105767  000000G                 TSTB    PROFLG          ;Are we running on a Pro?
000452  001425                          BEQ     9$              ;If not, that's all we have to do!
                                ;  Convert internal license number to ascii
000454  005000                          CLR     R0              ;Count number of digits
000456  016705  000000G                 MOV     TSXSIT,R5       ;Get site license number
000462  005004                  1$:     CLR     R4              ;Set for divide
000464  071427  000012                  DIV     #10.,R4         ;Split out low digit R5, quotient in R4
000470  062705  000060                  ADD     #'0,R5          ;Convert least significant digit to ascii
000474  110546                          MOVB    R5,-(SP)        ;Save binary digit on stack
000476  005200                          INC     R0              ;Count a digit saved
000500  010405                          MOV     R4,R5           ;Copy quotient for next divide
000502  001367                          BNE     1$              ;Loop until insignificant
                                ;  Now hang the license number at the end of the fake site name
000504  012701  000574'                 MOV     #PSEND,R1       ;Point to end of name string
000510  112605                  2$:     MOVB    (SP)+,R5        ;Get back ascii digit
000512  005405                          NEG     R5              ;Encrypt it
000514  110521                          MOVB    R5,(R1)+        ;And hang at string end
000516  077004                          SOB     R0,2$           ;Loop thru all
000520  105011                          CLRB    @R1             ;Terminate string
                                ;  And end by pointing to the fake pro site name string
000522  012701  000534'                 MOV     #PROSIT,R1      ;Point to encrypted site name
                                ;
000526  012605                  9$:     MOV     (SP)+,R5
000530  012604                          MOV     (SP)+,R4
000532  000207                          RETURN
                                ;
                                ;  Define a fake site name to be filled in for Pro systems
                                ;
000534                          PROSIT:
                                        .NLIST  BEX
000534                                  ENC     <PRO/TSX-Plus  License # 999-TPS->
000574                          PSEND:  .BLKB   6               ;Leave room for 5 digit number and terminator
                                        .EVEN
                                        .LIST   BEX

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
```

Select Pro or ordinary site name

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

Cross reference table (CREF V05.05)

```
...V1    4-29     4-29     4-29     4-31     4-45     4-45
...V2    4-29     4-29     4-29#    4-29#    4-31     4-31     4-31#    4-31#    4-45     4-45#
AREA     4-29     4-31     6-14#
CSIBUF   1-28     5-7      5-14     5-22
CXTRMN   1-26     4-7
DEVSPC   4-20*    4-29     6-15#
DMYDEV   1-27     4-26
DONE     5-6      6-4#
DOSCHD   1-26     6-11*
EXCJOB   1-26     6-10*
FILNAM   4-28*    6-16#
HANDSK   1-27     4-11     4-35*
HANENT   1-27     4-22
INFO     4-31     4-33     6-18#
KMINIT   1-22     2-12#
KMONCE   1-26     6-9*
MNUAOT   1-26     3-7
NUMDEV   1-27     4-21
PIKSIT   5-8      7-7#
PNAME    1-27     4-12     4-24
PNPTR    1-32#    4-9
PROFLG   1-28     7-11
PROSIT   7-31     7-39#
PSEND    7-24     7-42#
SETHBT   4-6#
SETMXJ   3-6#
SETSIT   5-4#
SETSTN   5-29     6-19#
SITE     1-28     7-10
SPLND    1-28     5-5
SYNAME   1-27     4-20
TSKM2D   1-5#
TSXSIT   1-28     7-15
VMNUAO   1-26     3-20*
```

```
...CM1    4-29       4-31
...CM2    4-29       4-29       4-31
...CM3    4-45
...CM5    4-29       4-31
.CLOSE    1-18#      4-45
.CSTAT    1-18#      4-31
.LOOKU    1-18#      4-29
ENC       1-36#      7-41
```

```
TSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX                              -PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-Pl
usTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusT  S&H Computer Systems, Inc.  SX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-
PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-Plu                              sTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTS
```

```
TTTTT  SSS  K   K  SSS  TTTTT   1
  T   S   S K  K  S   S   T    11
  T   S     K K   S       T   1 1
  T    SSS  KK     SSS    T     1
  T       S K K       S   T     1
  T   S   S K  K  S   S   T     1
  T    SSS  K   K  SSS    T   11111


TTTTTTTTTT    SSSSSS    KK        KK    SSSSSS    TTTTTTTTTT        11
TTTTTTTTTT    SSSSSS    KK        KK    SSSSSS    TTTTTTTTTT        11
    TT      SS      SS  KK      KK    SS      SS      TT          1111
    TT      SS      SS  KK      KK    SS      SS      TT          1111
    TT      SS          KK    KK      SS              TT        11  11
    TT      SS          KK    KK      SS              TT        11  11
    TT        SSSSSS    KKKK            SSSSSS        TT            11
    TT        SSSSSS    KKKK            SSSSSS        TT            11
    TT              SS  KK  KK                SS      TT            11
    TT              SS  KK  KK                SS      TT            11
    TT      SS      SS  KK      KK    SS      SS      TT            11
    TT      SS      SS  KK      KK    SS      SS      TT            11
    TT        SSSSSS    KK        KK    SSSSSS        TT        1111111111
    TT        SSSSSS    KK        KK    SSSSSS        TT        1111111111


[[[   000  55555       000   333    ]]]
[    0   0 5          0   0 3   3     ]
[    0  00 5          0  00     3     ]
[    0 0 0 5555       0 0 0   33      ]
[    00  0     5  ,,  00  0     3     ]
[    0   0 5   5  ,,  0   0 3   3     ]
[[[   000   555  ,,    000   333    ]]]


L     IIIII N   N EEEEE              1   77777
L       I   NN  N E                 11       7
L       I   N N N E        =====   1 1      7
L       I   N  NN EEEE               1     7
L       I   N   N E        =====     1    7
L       I   N   N E                  1   7
LLLLL IIIII N   N EEEEE            11111 7
```