L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
_$

Ps
--
YZ

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$
```

```
**FILE**ID**SYSSETIME                                   C  3                                                               SYS
                                                                                                                          V04






            SSSSSSSS  YY      YY    SSSSSSSS    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    IIIIII    MM      MM  EEEEEEEEEE
            SSSSSSSS  YY      YY    SSSSSSSS    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT    IIIIII    MM      MM  EEEEEEEEEE
          SS          YY      YY  SS          SS          EE              TT          II      MMMM  MMMM  EE
          SS          YY      YY  SS          SS          EE              TT          II      MMMM  MMMM  EE
          SS            YY  YY    SS          SS          EE              TT          II      MM  MM  MM  EE
          SS            YY  YY    SS          SS          EE              TT          II      MM  MM  MM  EE
            SSSSSS        YY        SSSSSS      SSSSSS    EEEEEEEE        TT          II      MM      MM  EEEEEEEE
            SSSSSS        YY        SSSSSS      SSSSSS    EEEEEEEE        TT          II      MM      MM  EEEEEEEE
                  SS      YY              SS          SS  EE              TT          II      MM      MM  EE
                  SS      YY              SS          SS  EE              TT          II      MM      MM  EE
                  SS      YY              SS          SS  EE              TT          II      MM      MM  EE              ....
                  SS      YY              SS          SS  EE              TT          II      MM      MM  EE              ....
          SSSSSSSS        YY      SSSSSSSS    SSSSSSSS    EEEEEEEEEE      TT        IIIIII    MM      MM  EEEEEEEEEE      ....
          SSSSSSSS        YY      SSSSSSSS    SSSSSSSS    EEEEEEEEEE      TT        IIIIII    MM      MM  EEEEEEEEEE      ....

          LL            IIIIII      SSSSSSSS
          LL            IIIIII      SSSSSSSS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II        SSSSSS
          LL              II        SSSSSS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LLLLLLLLLL    IIIIII    SSSSSSSS
          LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000         .TITLE  SYSSETIME - SYSTEM SERVICE TO SET CURRENT SYSTEM TIME
0000         .IDENT  'V04-000'
0000
0000   ;
0000   ;****************************************************************************
0000   ;*                                                                          *
0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000   ;*  ALL RIGHTS RESERVED.                                                    *
0000   ;*                                                                          *
0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000   ;*  TRANSFERRED.                                                            *
0000   ;*                                                                          *
0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000   ;*  CORPORATION.                                                            *
0000   ;*                                                                          *
0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000   ;*                                                                          *
0000   ;*                                                                          *
0000   ;****************************************************************************
0000   ;
0000   ; LEN KAWELL 14-FEB-79
0000   ;
0000   ; MODIFICATION HISTORY:
0000   ;
0000   ;       V03-012 SRB0125         Steve Beckhardt         6-Jul-1984
0000   ;               Zero distributed deadlock detection timestamp
0000   ;               expirations when a SET_TIME is performed.  The approach
0000   ;               taken in change SRB0117 may cause false deadlocks.
0000   ;
0000   ;       V03-011 SRB0117         Steve Beckhardt         9-Apr-1984
0000   ;               Recompute distributed deadlock detection timestamp
0000   ;               expirations when a SET TIME is performed.
0000   ;
0000   ;       V03-010 CWH3010         CW Hobbs                6-Dec-1983
0000   ;               Change references to EXE$READ_TODR and EXE$WRITE_TODR
0000   ;               to use the physical register routines EXE$READP_TODR and
0000   ;               EXE$WRITEP_TODR.  This is so that the non-p routines for
0000   ;               Nautilus can fabricate the TODR from the quadword system
0000
0000   ;       V03-009 KDM0086         Kathleen D. Morse       13-Oct-1983
0000   ;               Rearrange order in which PR$_TODR and EXE$GQ_SYSTIME
0000   ;               are set so that the same code works for MicroVAX I and
0000   ;               for all other VAX systems.
0000   ;
0000   ;       V03-008 CWH3008         CW Hobbs                10-Sep-1983
0000   ;               If parameter map in BOOTCB is null, then return success
0000   ;               without writing time out.  If running standalone and
0000   ;               system disk has been removed, then map will be null.
0000   ;
0000   ;       V03-007 DWT0125         David W. Thiel          23-Aug-1983
```

```
0000 ;              Add internal entry point EXE$SETIME_INT that avoids writing
0000 ;              to the system disk.  Restructure code to support this new
0000 ;              entry point.
0000 ;
0000 ;     V03-006 KDM0056          Kathleen D. Morse       12-Jul-1983
0000 ;              Change references to IPR TODR to use cpu-dependent
0000 ;              routines:  EXE$READ_TODR and EXE$WRITE_TODR.
0000 ;
0000 ;     V03-005 ROW0121          Ralph O. Weber          24-AUG-1982
0000 ;              Change JSB to EXE$BOOTCB_CHK so that it is performed at
0000 ;              IPL$_SYNCH.  A revision to EXE$BOOTCB_CHK requires that the
0000 ;              routine be called at or above this IPL to prevent modification
0000 ;              of volitile protions of the SYS.EXE window control block which
0000 ;              is a part of the area checksummed.
0000 ;
0000 ;     V03-004 KDM0002          Kathleen D. Morse       28-Jun-1982
0000 ;              Added $IODEF and $SSDEF.
0000 ;
0000 ;     V03-003 PHL0101          Peter H. Lipman         20-Jun-1982
0000 ;              $QIOW now synchronizes the EFN and IOSB parameters
0000 ;              correctly.  Eliminate the synchronization code here.
0000 ;
0000 ;
0000 ; SYSTEM SERVICE TO SET THE CURRENT SYSTEM TIME
0000 ;
0000 ; DOES ANYBODY REALLY KNOW WHAT TIME IT IS?
0000 ; DOES ANYBODY REALLY CARE?
0000 ; IF SO I CAN'T IMAGINE WHY,
0000 ; WE'VE ALL GOT TIME ENOUGH TO DIE.
0000 ;                               CHICAGO 1971
0000 ;
```

```
                                  0000            .SBTTL  DECLARATIONS
                                  0000    ;
                                  0000    ; MACRO LIBRARY CALLS
                                  0000    ;
                                  0000
                                  0000            $BOODEF                         ;DEFINE BOOT CONTROL BLOCK OFFSETS
                                  0000            $IODEF                          ;DEFINE I/O FUNCTION CODES
                                  0000            $IPLDEF                         ;DEFINE INTERRUPT PRIORITY LEVELS
                                  0000            $PCBDEF                         ;DEFINE PCB OFFSETS
                                  0000            $PRDEF                          ;DEFINE PROCESSOR REGISTERS
                                  0000            $PRVDEF                         ;DEFINE PRIVILEGES
                                  0000            $SSDEF                          ;DEFINE SYSTEM STATUS CODES
                                  0000            $TQEDEF                         ;DEFINE TIMER QUEUE ENTRIES
                                  0000            $VADEF                          ;DEFINE VIRTUAL ADDRESSES
                                  0000
                                  0000    ;
                                  0000    ; LOCAL SYMBOLS
                                  0000    ;
                                  0000    ; ARGUMENT LIST OFFSET DEFINITIONS
                                  0000    ;
                                  0000
                        00000004  0000    TIMADR=4                                ;NEW TIME
                                  0000
                                  0000    ;
                                  0000    ; OWN STORAGE
                                  0000    ;
                              00000000    .PSECT  YEXEPAGED                       ;PAGED DATA AND CODE
                                  0000
                                  0000    ;
                                  0000    ; BASE TIME FOR TIME-OF-DAY PROCESSOR REGISTER (TODR). THE TODR CAN'T
                                  0000    ; CONTAIN THE ENTIRE SYSTEM TIME, SO IT IS DEFINED TO CONTAIN A VALUE
                                  0000    ; RELATIVE TO THE BASE TIME.  THE BASE TIME IS DEFINED AS:
                                  0000    ;
                                  0000    ;               01-JAN-CURRENT_YEAR 00:00:00.00
                                  0000    ;
                                  0000    ; TO ALLOW CONVERSION OF THE TODR VALUE TO SYSTEM TIME FORMAT,
                                  0000    ; THE TODR AND ITS CORRESPONDING SYSTEM TIME ARE STORED IN THE SYSTEM
                                  0000    ; IMAGE FILE (SYS$SYSTEM:SYS.EXE).  BECAUSE ALL TODR'S HAVE THE SAME
                                  0000    ; BASE, A DISK THAT IS BOOTED ON ONE SYSTEM CAN BE TRANSPORTED TO
                                  0000    ; ANOTHER SYSTEM AND THE TIME WILL BE SET CORRECTLY.
                                  0000    ;
                                  0000    BASETIME:                               ;BASE TIME FOR TIME-OF-DAY REGISER
              2D 4E 41 4A 2D 31 30  0000            .ASCII  /01-JAN-/               ;
                        00000007  0007    BASEYEAR = .-BASETIME                   ;(OFFSET TO YEAR)
30 3A 30 30 3A 30 30 20 58 58 58 58  0007            .ASCII  /XXXX 00:00:00.00/      ;
                     30 30 2E 30  0013
                        00000017  0017    BASETIMESZ = .-BASETIME                 ;BASE TIME STRING SIZE
                                  0017    ;
                                  0017    ; TO DETERMINE WHETHER OR NOT THE PROCESSOR TIME-OF-YEAR CLOCK (PR$_TODR) HAS
                                  0017    ; ROLLED OVER THE YEAR, WE MUST KNOW HOW LONG THE BASE YEAR WAS.  TO CALULATE
                                  0017    ; THAT TIME, WE NEED A DEFINITION OF THE END OF THE YEAR.  USING THE ABOVE
                                  0017    ; DEFINITION OF THE FIRST OF THE YEAR AND THE FOLLOWING END OF YEAR, WE CAN
                                  0017    ; USE THE APPROPRIATE SYSTEM SERVICES TO CALCULATE THAT SPAN OF TIME - ALWAYS
                                  0017    ; PROPERLY ALLOWING FOR LEAP YEARS.
                                  0017    ;
                                  0017    ENDTIME:
20 58 58 58 58 2D 43 45 44 2D 31 33  0017            .ASCII  /31-DEC-XXXX 23:59:59.99/       ; END OF YEAR (- 10ms)
```

```
SYS
Sym

$$T
BAS
BAS
BAS
BEG
BOO
BUG
COM
CVT
END
ERR
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXE
EXI
IO$
IPL
IPL
IPL
LCK
LOC
LOC
LOC
PCB
PR$
PRV
PRV
REC
SAV
SCA
SCA
SET
SET
SS$
SS$
SS$
SS$
SYS
SYS
SYS
SYS
SYS
SYS
TIM
TQE
TQE
TQE
TQE
```

```
   39 39 2E 39 35 3A 39 35 3A 33 32  0023
                                     002E    146
                                     002E    147 ;
                                     002E    148 ; SYSTEM DISK LOGICAL NAME
                                     002E    149 ;
                                     002E    150 SYSDISK:                                           ;SYSTEM DISK LOGICAL NAME
59 53 24 53 59 53 00000036'010E0000' 002E    151         .ASCID  /SYS$SYSDEVICE/
               45 43 49 56 45 44 53  003C
```

```
SYS
Pse

PSE
---
. 
$AB
YEX

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
649
The
560
31

Mac
---
_$2
_$2
TOT

136

The

MAC
```

```
                         0043          .SBTTL  SET TIME
                         0043   ;+
                         0043   ; EXE$SETIME - SET CURRENT SYSTEM TIME
                         0043   ;
                         0043   ; THIS SERVICE PROVIDES THE CAPABILITY TO RESET THE CURRENT SYSTEM TIME.
                         0043   ;
                         0043   ; EXE$SETIME_INT - INTERNAL ENTRY POINT THAT RECOMPUTES INTERNAL TIME
                         0043   ;                  AND WRITES NOTING TO THE DISK.
                         0043   ;
                         0043   ; INPUTS:
                         0043   ;
                         0043   ;       TIMADR(AP) = ADDRESS OF NEW TIME VALUE.  ZERO FOR ADDRESS OR
                         0043   ;               TIME VALUE CAUSES THE PROCESSOR'S TIME-OF-DAY CLOCK
                         0043   ;               TO BE USED TO RECALIBRATE THE CURRENT SYSTEM TIME.
                         0043   ;
                         0043   ;       R4 = ADDRESS OF CURRENT PROCESS PCB.
                         0043   ;
                         0043   ; OUTPUTS:
                         0043   ;
                         0043   ;       R0 LOW BIT CLEAR INDICATES FAILURE TO SET SYSTEM TIME.
                         0043   ;
                         0043   ;               R0 = SS$_ACCVIO - TIME VALUE CANNOT BE READ BY CALLING
                         0043   ;                       ACCESS MODE.
                         0043   ;
                         0043   ;               R0 = SS$_IVTIME - A NEGATIVE TIME VALUE WAS SPECIFIED,
                         0043   ;                       OR AN INVALID PROCESSOR CLOCK VALUE WAS FOUND.
                         0043   ;
                         0043   ;               R0 = SS$_NOIOCHAN - NO I/O CHANNEL IS AVAILABLE FOR
                         0043   ;                       USE BY THE SERVICE.
                         0043   ;
                         0043   ;               R0 = SS$_NOPRIV - THE CALLING PROCESS DOES NOT HAVE
                         0043   ;                       OPER OR LOG_IO PRIVILEGE.
                         0043   ;
                         0043   ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                         0043   ;
                         0043   ;               R0 = SS$_NORMAL - NORMAL COMPLETION, SYSTEM TIME SET.
                         0043   ;
                         0043   ;
                         0043   ;       CURRENT SYSTEM TIME (EXE$GQ_SYSTIME) IS SET TO SPECIFIED
                         0043   ;       TIME.  PROCESSOR'S TIME-OF-DAY CLOCK IS UPDATED TO NEW TIME.
                         0043   ;
                         0043   ;       EXE$SETIME ONLY -- NEW TIME IN SYSTEM TIME FORMAT AND TIME-OF-DAY
                         0043   ;               CLOCK FORMAT IS WRITTEN TO SYSTEM IMAGE FILE
                         0043   ;               (SYS$SYSTEM:SYS.EXE).
                         0043   ;-
                         0043          .ENABLE LSB
                         0043   EXE$SETIME_INT::
                   0FFC  0043          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>     ; ENTRY MASK
      51   04 AC     D0  0045          MOVL    TIMADR(AP),R1           ;GET ADDR OF NEW TIME ARGUMENT
              01     DD  0049          PUSHL   #1                      ;STORE FLAG
              22     11  004B          BRB     20$
                         004D
                   0FFC  004D          .ENTRY  EXE$SETIME,-            ;SET CURRENT SYSTEM TIME
                         004F                  ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;(ENTRY MASK)
                         004F
        50    24     3C  004F          MOVZWL  S^#SS$_NOPRIV,R0        ;ASSUME NO PRIVILEGE
                         0052          IFNPRIV OPER,90$                ;BR IF NO OPER PRIVILEGE
```

```
                           0058   210          IFNPRIV LOG_IO,90$                ;BR IF NO LOG_IO PRIVILEGE
         50   0C     3C    005E   211          MOVZWL  S^#SS$_ACCVIO,R0          ;ASSUME ACCESS VIOLATION
   51      04 AC     D0    0061   212          MOVL    TIMADR(AP),R1             ;GET ADDR OF NEW TIME ARG
              06     13    0065   213          BEQL    10$                       ;BR IF NONE
                           0067   214          IFNORD  #8,(R1),90$               ;CAN TIME BE READ?
              7E     D4    006D   215 10$:     CLRL    -(SP)                     ;STORE FLAG ON STACK
         56   7E     7E    006F   216 20$:     MOVAQ   -(SP),R6                  ;ALLOCATE BINARY TIME BUFFER
         58   7E     7E    0072   217          MOVAQ   -(SP),R8                  ;ALLOCATE ASCII TIME BUFFER DESCRIPTOR
         5E   18     C2    0075   218          SUBL2   #<BASETIMESZ+3>&^C3,SP    ;ALLOCATE ASCII TIME BUFFER
         57   5E     D0    0078   219          MOVL    SP,R7                     ;HOLD POINTER TO IT
   04    A8   57     D0    007B   220          MOVL    R7,4(R8)                  ;CREATE DESCR
         68   17     D0    007F   221          MOVL    #BASETIMESZ,(R8)          ;LENGTH FIELD
              51     D5    0082   222          TSTL    R1                        ;IS A TIME SPECIFIED?
              5C     13    0084   223          BEQL    RECAL                     ;BR IF NO - RECALIBRATE
 50    0184   8F     3C    0086   224          MOVZWL  #SS$_IVTIME,R0            ;ASSUME INVALID TIME
         5A   61     7D    008B   225          MOVQ    (R1),R10                  ;GET TIME - R10,R11
              03     14    008E   226          BGTR    COMPUTE_TODR              ;BR IF VALID TIME
              50     13    0090   227          BEQL    RECAL                     ;BR IF ZERO - RECALIBRATE
                           0092   228 90$:
                     04    0092   229          RET                               ;RFTURN WITH FAILURE
                           0093   230
                           0093   231          .DISABLE        LSB
```

```
                                   0093             .SBTTL  COMPUTE TIME-OF-DAY CLOCK VALUE
                                   0093      ;++
                                   0093      ;
                                   0093      ; COMPUT_TODR - COMPUTE NEW TIME-OF-DAY CLOCK VALUE.
                                   0093      ;
                                   0093      ;          COMPUTE NEW TIME-OF-DAY CLOCK VALUE BY COMPUTING THE NEW
                                   0093      ;          TIME AFTER THE BASE TIME AND SCALING TO 10 MSEC UNITS.
                                   0093      ;
                                   0093      ; INPUTS:
                                   0093      ;
                                   0093      ;          R10,R11 = NEW SYSTEM TIME.
                                   0093      ;
                                   0093      ; OUTPUTS:
                                   0093      ;
                                   0093      ;          R0 = NEW TIME-OF-DAY CLOCK VALUE.
                                   0093      ;
                                   0093      ;--
                                   0093     COMPUTE_TODR:                                     ;COMPUTE TIME-OF-DAY CLOCK VALUE
               52   FF69 CF    9E  0093             MOVAB   BASETIME,R2                       ;GRAB CORRECT TIME STRING
                           1E  10  0098             BSBB    CVT_TIME                          ;CONVERT THE TIME
                      50   5A  7D  009A             MOVQ    R10,R0                            ;GET NEW TIME
                      50   86  C2  009D             SUBL    (R6)+,R0                          ;COMPUTE YEAR'S DELTA
                      51   66  D9  00A0             SBWC    (R6),R1                           ;IN 100 NANOSECOND UNITS
     51     50     50   0185'CF  7B  00A3           EDIV    W^SCALE,R0,R0,R1                  ;SCALE TO 10 MS UNITS
               50     50   01  78  00AA             ASHL    #1,R0,R0                          ; (SINCE NO UNSIGNED EDIV)
          50   10000000 8F     C0  00AE             ADDL    #<1@28>,R0                        ;COMPUTE BIASED VALUE FOR TODR
                                   00B5                                                       ;VALUES LESS THAN BIAS ARE USED
                                   00B5                                                       ;TO DETECT CLOCK POWERFAIL
                         00D5  31  00B5             BRW     SET_TODR                          ;GO SET TIME-OF-DAY CLOCK
```

```
                     00B8          .SBTTL  CVT_TIME
                     00B8     ;++
                     00B8     ;
                     00B8     ; CVT_TIME - CONVERT AN ASCII STRING TO A BINARY TIME
                     00B8     ;
                     00B8     ; INPUTS:
                     00B8     ;
                     00B8     ;        R2 - POINTER TO TIME STRING
                     00B8     ;        R10, R11 - NEW TIME IN BINARY
                     00B8     ;
                     00B8     ; OUTPUTS:
                     00B8     ;
                     00B8     ;        R6 - POINTER TO QUAD WORD TIME
                     00B8     ;        R7 - POINTER TO ASCII TIME
                     00B8     ;        R8 - POINTER TO DESCRIPTOR FOR ASCII TIME
                     00B8     ;        R9 - ASCII VALUE OF BASE YEAR
                     00B8     ;
                     00B8     ;--
                     00B8
                     00B8     CVT_TIME:
      66    5A   7D  00B8             MOVQ    R10,(R6)                ;SET NEW TIME IN BUFFER
                     00BB             $ASCTIM_S TIMADR=(R6),-         ;CONVERT NEW TIME TO ASCII
                     00BB                     TIMBUF=(R8)             ;
    59   07 A7   D0  00CA             MOVL    BASEYEAR(R7),R9         ;GET NEW TIME'S YEAR
 67   62    17   28  00CE             MOVC    #BASETIMESZ,(R2),(R7)   ;MOVE TODR BASE TO ASCII BUFFER
   07 A7    59   D0  00D2             MOVL    R9,BASEYEAR(R7)         ;MOVE NEW TIME'S YEAR TO BUFFER
                     00D6             $BINTIM_S TIMADR=(R6),-         ;COMPUTE TODR BASE IN BINARY
                     00D6                     TIMBUF=(R8)             ;
                 05  00E1             RSB                             ;LEAVE
```

```
                                    00E2           .SBTTL  RECALIBRATE
                                    00E2    ;++
                                    00E2    ;
                                    00E2    ; RECAL - RECALIBRATE USING TIME-OF-DAY CLOCK
                                    00E2    ;
                                    00E2    ;       THE TIME-OF-DAY CLOCK IS SPECIFIED TO BE MORE ACCURATE
                                    00E2    ;       THAN THE INTERVAL TIMER. IF THE SYSTEM HAS BEEN UP AWHILE, THIS
                                    00E2    ;       FUNCTION CAN BE USED TO RECALIBRATE THE SYSTEM TIME.  THIS
                                    00E2    ;       FUNCTION IS ALSO USED BY SYSINIT TO SET THE SYSTEM TIME AT
                                    00E2    ;       EACH RE-BOOT.
                                    00E2    ;
                                    00E2    ;       IT IS IMPORTANT TO NOTE THAT THE HARDWARE CLOCK HAS A LIMIT
                                    00E2    ;       OF ABOUT 15 MONTHS.  IT SHOULD NEVER BE ALLOWED TO REACH THIS
                                    00E2    ;       LIMIT; THUS ANY VMS SYSTEM SHOULD BE BOOTED, OR A SET TIME
                                    00E2    ;       REQUEST SHOULD BE DONE AT LEAST ONCE BETWEEN 1-JAN AND 1-APR
                                    00E2    ;       OF EACH YEAR.
                                    00E2    ;
                                    00E2    ;       DELTA-TOD = PR$_TODR - EXE$GL_TODR
                                    00E2    ;       DELTA-TIME = DELTA-TOD * SCALE
                                    00E2    ;       EXE$GQ_SYSTIME, EXE$GQ_TODCBASE = EXE$GQ_TODCBASE + DELTA-TIME
                                    00E2    ;       PR$_TODR = PR$_TODR - BASE-YEAR IF PR$_TODR > BASE_YEAR
                                    00E2    ;
                                    00E2    ; INPUTS:
                                    00E2    ;
                                    00E2    ;       TIME-OF-DAY PROCESSOR CLOCK.
                                    00E2    ;
                                    00E2    ; OUTPUTS:
                                    00E2    ;
                                    00E2    ;       R0 = NEW TIME-OF-DAY CLOCK VALUE
                                    00E2    ;       R10,R11 = NEW SYSTEM TIME COMPUTED FROM TIME-OF-DAY CLOCK
                                    00E2    ;
                                    00E2    ;--
                                    00E2
                                    00E2   RECAL:                                       ;RECALIBRATE USING TOD CLOCK
                                    00E2    ;
                                    00E2    ; CALCULATE HOW LONG THE BASE YEAR (DEFINED BY THE VALUE IN EXE$GQ_TODCBASE)
                                    00E2    ; IS.  THIS VALUE MAY BE NEEDED LATER TO CORRECT THE YEAR ROLLOVER IN THE
                                    00E2    ; HARDWARE TIME OF YEAR CLOCK.  THE CALCULATION IS DONE EARLY BECAUSE
                                    00E2    ; 1) IT DOES NOT DEPEND ON ANY CHANGING REGISTERS, AND 2) IT CAN BE DONE
                                    00E2    ; AT ZERO IPL ($BINTIM IS PAGED).
                                    00E2    ;
        5C   00000000'GF      7E    00E2           MOVAQ   G^EXE$GQ_TODCBASE,AP     ;GET A POINTER
                   5A   6C    7D    00E9           MOVQ    (AP),R10                 ;PICK UP CURRENT BASE TIME
            52   FF27 CF      9E    00EC           MOVAB   ENDTIME,R2               ;GRAB PROPER TIME STRING
                        C5    10    00F1           BSBB    CVT_TIME                 ;CONVERT THE TIME
                   5A   66    7D    00F3           MOVQ    (R6),R10                 ;HOLD YEAR-END TIME
      67   FF05 CF      17    28    00F6           MOVC    #BASETIMESZ,BASETIME,(R7) ;MOVE BASE STRING TO BUFFER
                07 A7   59    D0    00FC           MOVL    R9,BASEYEAR(R7)          ;MOVE NEW TIME'S YEAR TO BUFFER
                                    0100           $BINTIM_S TIMADR=(R6),-          ;COMPUTE TODR BASE IN BINARY
                                    0100                   TIMBUF=(R8)              ;
                   5A   86    C2    010B           SUBL    (R6)+,R10                ;COMPUTE BASE YEAR'S LENGTH(-10 ms)
                   5B   66    D9    010E           SBWC    (R6),R11                 ;IN 100 NANOSECOND UNITS
               5A   89'AF     C0    0111           ADDL    B^SCALE2,R10             ;ADD IN THE MISSING 10 MS
                   5B   00    D8    0115           ADWC    #0,R11                   ;
                   50   5A    7D    0118           MOVQ    R10,R0                   ;COPY
   51    50    50   85'AF     7B    011B           EDIV    B^SCALE,R0,R0,R1         ;SCALE TO 10 MS UNITS
            52     50   01    78    0121           ASHL    #1,R0,R2                 ; (SINCE NO UNSIGNED EDIV)
```

```
                                  0125  ;
                                  0125  ; AT THIS POINT:
                                  0125  ;       R10:R11 = THE NUMBER OF 10 NANOSECONDS IN THE BASE YEAR
                                  0125  ;       R2 = THE NUMBER OF 10 MILLISECONDS IN THE BASE YEAR
                                  0125  ;
                                  0125  ; WE MUST CHECK THE PROCESSOR'S TIME OF YEAR CLOCK (PR$_TODR) TO SEE IF
                                  0125  ; IT HAS HAS MORE THAN A YEAR'S WORTH OF TICKS IN IT.  IF SO, WE SUBTRACT
                                  0125  ; OUT THE YEAR FROM THE CLOCK. IF NOT, WE DO NOT CHANGE THE CLOCK.
                                  0125  ; IN EITHER CASE, WE THEN UPDATE THE SYSTEM TIME FROM THE CORRECT
                                  0125  ; VALUE OF THE TODC.
                                  0125  ;
                                  0125  LOCK_START:                             ;START OF LOCKED-DOWN CODE
                                  0125          DSBINT  W^LOCK_IPL              ;DISABLE ALL INTERRUPTS AND
                                  012D                                          ; LOCK CODE IN MEMORY
           00000000'EF     16     012D          JSB     EXE$READP_TODR          ;GET TIME OF PHYSICAL DAY CLOCK VALUE
  51   50  00000000'GF     C3     0133          SUBL3   G^EXE$GL_TODR,R0,R1     ;GET TOD DELTA TIME (10 MS UNITS)
                    11     1A     013B          BGTRU   5$                      ;ZERO OR POSITIVE IS OKAY
        FF7C2A00 8F  51    D1     013D          CMPL    R1,#-<24*60*60*100>     ;IF NEGATIVE, THEN IF MORE THAN A DAY
                    36     1F     0144          BLSSU   ERROR                   ;INVALID TIME IN PROCESSOR CLOCK
  5A   00   51   89'AF     7A     0146          EMUL    B^SCALE2,R1,#0,R10      ;ELSE COVERT TO 100 NANOSECONDS
                    25     11     014C          BRB     20$                     ;AND CONTINUE ON
                                  014E
  53   52  10000000 8F     C1     014E  5$:     ADDL3   #<1@28>,R2,R3           ;ADD IN THE BIAS TO THE YEAR
               53   50     D1     0156          CMPL    R0,R3                   ;CORRECTION NEEDED FOR CLOCK?
                    03     1F     0159          BLSSU   10$                     ;NO CORRECTION NEEDED
                                  015B  ;
                                  015B  ; THE CLOCK HAS ONE YEAR TOO MANY TICKS.  WE MUST SUBTRACT OUT
                                  015B  ; THE BASE YEAR FROM THE PROCESSOR TODC. AFTER THE YEAR IS SUBTRACTED OUT,
                                  015B  ; THE DELTA WILL BE CORRECT.
                                  015B  ;
               50   52     C2     015B          SUBL    R2,R0                   ;SUBTRACT OUT A YEAR
                                  015E  10$:
                                  015E  ;
                                  015E  ; AT THIS POINT:
                                  015E  ;       R0 = CORRECT TODR + BIAS TO GO INTO PROCESSOR REGISTER
                                  015E  ;       R1 = CORRECT TODR DELTA TO FIGURE TIME
                                  015E  ;
       5A   51   1F   01   EF     015E          EXTZV   #1,#31,R1,R10           ;DIVIDE BY 2
                                  0163                                          ;(SINCE NO UNSIGNED EMUL)
  5A   00   5A   85'AF     7A     0163          EMUL    B^SCALE,R10,#0,R10      ;COMPUTE 100 NANOSECOND DELTA
                 07 51     E9     0169          BLBC    R1,20$                  ;BR IF NO CORRECTION NEEDED
            5A   89'AF     C0     016C          ADDL    B^SCALE2,R10            ;ADD LAST 10MS UNIT
               5B   00     D8     0170          ADWC    #0,R11                  ;TO QUADWORD
               5A   6C     C0     0173  20$:    ADDL    (AP),R10                ;FORM ABSOLUTE TIME
            5B   04 AC     D8     0176          ADWC    4(AP),R11               ;
                    19     11     017A          BRB     SET_TODR1               ;SET THE TODR REGISTER
                                  017C  ;
                                  017C  ; THE VALUE IN THE PROCESSOR CLOCK WAS LESS THAN THE BIAS (2^28) OR
                                  017C  ; IT WAS GREATER THAN THE LAST VALUE STORED IN EXE$GL_TODR - INDICATING
                                  017C  ; THAT THE SYSTEM DISK HASN'T BEEN BOOTED IN A VERY LONG TIME AND THUS
                                  017C  ; THE YEAR WITHIN DATE WILL NOT BE CORRECT.
                                  017C  ;
                                  017C  ERROR:  ENBINT
           50   0184 8F    3C     017F          MOVZWL  #SS$_IVTIME,R0          ;INVALID PROCESSOR CLOCK VALUE
                           04     0184          RET                             ;LEAVE AND REPORT IT
                                  0185  ;
                                  0185  ; SCALING VALUE FOR CALCULATIONS - MUST BE HERE SINCE IT IS REFERENCED
```

```
                      0185   407 ; FROM WITHIN A LOCKED-DOWN SECTION OF CODE.
                      0185   408 ;
            00030D40  0185   409 SCALE:  .LONG   <100*1000*2>            ;NUMBER OF 100 NS IN 10 MS * 2
            000186A0  0189   410 SCALE2: .LONG   <100*1000>              ;NUMBER OR 100 NS IN 10 MS
```

```
ENBF
EXE$
EXE$
EXE$
PCB$
PCB$
PCB$
PCB$
PCB$
PRV$
PSL$
PSL$
SETM
SS$_
SS$_
SS$_
SWPF
WATF



PSEC
----
.  A
$ABS
Y$EX



Phas
----
Init
Comm
Pass
Symb
Pass
Symb
Psec
Cros
Asse

The
3249
Ther
162
12 p
```

```
                                    018D            .SBTTL  SET TODR
                                    018D      ;++
                                    018D      ;
                                    018D      ; SET_TODR - SET NEW TIME-OF-DAY CLOCK VALUE
                                    018D      ;
                                    018D      ;         SET NEW TIME IN TIME-OF-DAY CLOCK, SYSTEM TIME, AND SAVE BOTH
                                    018D      ;         IN SYSTEM PARAMETERS.  ALSO RECOMPUTE REMAINING TIME INTERVAL
                                    018D      ;         IN TIMER QUEUE ENTRIES.
                                    018D      ;
                                    018D      ; INPUTS:
                                    018D      ;
                                    018D      ;         R0 = NEW TIME-OF-DAY CLOCK VALUE
                                    018D      ;         R10,R11 = NEW SYSTEM TIME
                                    018D      ;--
                                    018D      SET_TODR:
                                    018D            DSBINT  W^LOCK_IPL                  ;DISABLE ALL INTERRUPTS AND
                                    0195                                                ;LOCK CODE IN MEMORY
                                    0195      SET_TODR1:
          54    00000000'GF     7D  0195            MOVQ    G^EXE$GQ_SYSTIME,R4         ;GET CURRENT SYSTEM TIME
          00000000'GF     5A    7D  019C            MOVQ    R10,G^EXE$GQ_SYSTIME        ;SET NEW SYSTEM TIME
          00000000'GF     5A    7D  01A3            MOVQ    R10,G^EXE$GQ_TODCBASE       ;SAVE SYSTEM TIME OF TOD REG
                                    01AA      ;
                                    01AA      ; At this point we should write the new value.  We use the physical routine
                                    01AA      ; to force this to written into any hardware TODR registers that may exist.
                                    01AA      ; On Nautilus, this forces the slow, non-interruptible write into the clock
                                    01AA      ; of the console processor.  On other processors, the physical function is
                                    01AA      ; identical to the non-physical function.
                                    01AA      ;
                00000000'EF     16  01AA            JSB     EXE$WRITEP_TODR             ;SET AS CURRENT TIME INTO PHYSICAL REG
                                    01B0      ;
                                    01B0      ; Resetting the processor TODR value must be done after a new time is
                                    01B0      ; written into EXE$GQ_SYSTIME in order for this routine to work on
                                    01B0      ; MicroVAX I.  This is because the routine EXE$WRITE_TODR re-computes
                                    01B0      ; the value to write based on a base year determined from EXE$GQ_SYSTIME.
                                    01B0      ; We use the physical function to make sure that we pick up any translations
                                    01B0      ; which might have been performed.  One example would be the conversion to
                                    01B0      ; one-second time resolution for the Nautilus physical clock register.
                                    01B0      ;
                00000000'EF     16  01B0            JSB     EXE$READP_TODR              ;GET TIME (MAY BE DIFFERENT THAN
                                    01B6                                                ; WHAT WAS WRITTEN)
          00000000'GF     50    D0  01B6            MOVL    R0,G^EXE$GL_TODR            ;SAVE TOD REG OF BASE
                                    01BD      ;
                                    01BD      ; Recompute expiration times that were not specified as absolute so that the
                                    01BD      ; remaining interval is the same for the new system time.  Absolute expirations
                                    01BD      ; will occur with respect to the absolute value of the new system time.
                                    01BD      ;
                      5A  54    C2  01BD            SUBL    R4,R10                      ;COMPUTE DELTA OF
                      5B  55    D9  01C0            SBWC    R5,R11                      ; OLD AND NEW TIMES
          53    00000000'GF     9E  01C3            MOVAB   G^EXE$GL_TQFL,R3            ;GET ADDRESS OF TIMER QUEUE HEAD
                      50  63    7D  01CA            MOVQ    (R3),R0                     ; Get existing timer queue
                      7E  50    7D  01CD            MOVQ    R0,-(SP)                    ; Save existing timer queue
                      61  5E    D0  01D0            MOVL    SP,TQE$L_TQFL(R1)           ; Readjust links of endmost entries
                   04 A0  5E    D0  01D3            MOVL    SP,TQE$L_TGBL(R0)           ;
                      63  53    D0  01D7            MOVL    R3,(R3)                     ; Reinitialize timer queue to empty
                   04 A3  53    D0  01DA            MOVL    R3,4(R3)                    ;
               55     00 BE     0F  01DE      10$:  REMQUE  @0(SP),R5                   ; Remove an entry from saved queue
                          1E    1D  01E2            BVS     30$                         ; Br if no more to remove
```

```
                 1C A5    D5  01E4                TSTL    TQE$Q_TIME+4(R5)        ;IS THIS A VALID TIME?
                              01E7                                                ;(PERMANENT ENTRY IS INVALID)
                    0D    19  01E7                BLSS    20$                     ;BR IF NOT
           08 0B A5 03    E0  01E9                BBS     #TQE$V_ABSOLUTE,TQE$B_RQTYPE(R5),20$
                              01EE                                                ; Br if specified as absolute time
                 18 A5    5A  C0  01EE            ADDL    R10,TQE$Q_TIME(R5)      ;ADD SYSTEM TIME DELTA
                 1C A5    5B  D8  01F2            ADWC    R11,TQE$Q_TIME+4(R5)    ; TO REMAINING INTERVAL
              50   18 A5  7D  01F6      20$:      MOVQ    TQE$Q_TIME(R5),R0       ; Get time value
             00000000'GF  16  01FA                JSB     G^EXE$INSTIMQ           ; Reinsert in queue
                    DC    11  0200                BRB     10$                     ;
                 5E 08    C0  0202      30$:      ADDL2   #8,SP                   ; Clean up temporary queue
     50      00000000'GF  7E  0205                MOVAQ   G^LCK$GQ_BITMAP_EXP,R0  ; Get address of deadlock timestamps
                    80    7C  020C                CLRQ    (R0)+                   ; Reset deadlock expiration timestamps
                    60    7C  020E                CLRQ    (R0)
                              0210                ENBINT                          ;RE-ENABLE INTERRUPTS
                    04    11  0213                BRB     SAVE_TIME               ;SAVE TIME IN SYSTEM IMAGE
                              0215
                              0215      ;
                              0215      ; THE PRECEDING CODE MUST DISABLE ALL INTERRUPTS, BUT SINCE IT IS
                              0215      ; PAGED IT MUST BE LOCKED IN MEMORY.  THE USAGE OF THE DSBINT MACRO
                              0215      ; ABOVE, BOTH DISABLES INTERRUPTS AND FAULTS INTO MEMORY THE CODE
                              0215      ; BETWEEN THE DSBINT AND LOCK_IPL.
                              0215      ;
                              0215      LOCK_IPL:                                 ;IPL NEEDED TO BLOCK EVERYTHING
                      0000001F  0215              .LONG   IPL$_POWER              ;
                              0219      LOCK_END:                                 ;END OF LOCKED-DOWN CODE
                              0219                ASSUME  <LOCK_END-LOCK_START> LE 512 ;ONE PAGE TO A CUSTOMER
                              0219          .SBTTL  SAVE TIME
                              0219  ;++
                              0219  ;
                              0219  ; SAVE_TIME - SAVE TIME-OF-DAY CLOCK VALUE AND SYSTEM TIME
                              0219  ;
                              0219  ;       SAVE TIME OF DAY REGISTER AND CORRESPONDING SYSTEM TIME IN THE
                              0219  ;       SYSTEM IMAGE FILE.  THESE ARE LOCATED IN THE SYSTEM PARAMETERS
                              0219  ;       SO WE WILL JUST WRITE THE SYSTEM PARAMETER PAGE(S).
                              0219  ;
                              0219  ; INPUTS:
                              0219  ;
                              0219  ;       EXE$GL_TODR = TIME-OF-DAY CLOCK VALUE.
                              0219  ;       EXE$GQ_TODCBASE = SYSTEM CORRESPONDING TO TIME-OF-DAY CLOCK.
                              0219  ;--
                              0219          .ENABLE LSB
                              0219  SAVE_TIME:                                  ;SAVE TIME IN SYSTEM IMAGE FILE
         20 FC AD       E8    0219          BLBS    -4(FP),5$                   ;BRANCH TO SKIP WRITING DISK
  52   00000000'GF      9E    021D          MOVAB   G^EXE$A_SYSPARAM,R2         ;GET SYSTEM PARAMETER ADDRESS
                              0224  BEGIN_NOPAGE:                               ;PROTECT CHECKSUM OPER. FROM OTHERS
                              0224          DSBINT  IPL  SYNCH                  ;AND MAKE THIS CODE NOT PAGE FAULT.
       00000000'EF      16    022E          JSB     EXE$BOOTCB_CHK              ;IS THE BOOT CONTROL BLOCK OK?
                0C      13    0234          BEQL    10$                         ;BRANCH IF YES
                              0236          ENBINT                              ;RESTORE PREVIOUS IPL
                              0239          BUG_CHECK BADBOOTCB                 ;ISSUE NON-FATAL BUGCHECK
         50     01      D0    023D  5$:     MOVL    S^#SS$_NORMAL,R0            ;FAKE A SUCCESSFUL COMPLETION
                67      11    0240          BRB     EXIT                        ;AND EXIT
                              0242  10$:    ENBINT                              ;RESTORE PREVIOUS IPL
      53     04 A1      D0    0245          MOVL    BOO$L_PARAM_MAP(R1),R3      ;ADR OF VIRTUAL TO LOGICAL MAP
                F2      13    0249          BEQL    5$                          ;NO MAP, RETURN SUCCESS.  NO MAP
                              024B                                              ; IF RUNNING WITHOUT SYS DISK
  56    83    FD 8F     78    024B          ASHL    #-3,(R3)+,R6                ;RETRIEVAL POINTER COUNT
         54     7E      DE    0250          MOVAL   -(SP),R4                    ;ALLOCATE BUFF FOR CHANNEL
         55     7E      7E    0253          MOVAQ   -(SP),R5                    ;ALLOCATE BUFF FOR IOSB
                              0256          $ASSIGN_S DEVNAM=SYSDISK,-          ;ASSIGN A CHANNEL TO SYSTEM DISK
                              0256                  CHAN=(R4)                   ;
             41 50      E9    0265          BLBC    R0,EXIT                     ;BR IF ERROR
    57   83     09      78    0268  20$:    ASHL    #9,(R3)+,R7                 ;BYTE COUNT FOR THIS RTRV PTR
         58     83      D0    026C          MOVL    (R3)+,R8                    ;STARTING LBN FOR THIS PIECE
                              026F          $QIOW_S CHAN=(R4),-                 ;SAVE NEW TOD WITH SYSTEM PARAMS
                              026F                  FUNC=S^#IO$_WRITELBLK,-     ;
                              026F                  EFN=S^#EXE$C_SYSEFN,-       ;EVENT FLAG
                              026F                  P1=(R2),-                   ;VIRTUAL ADDRESS OF BUFFER
                              026F                  P2=R7,-                     ;BYTE COUNT TO TRANSFER
                              026F                  P3=R8,-                     ;LBN
                              026F                  IOSB=(R5)                   ;
             0C 50      E9    028C          BLBC    R0,60$                      ;BRANCH IF FAILED TO QUEUE REQUEST
         50     65      3C    028F          MOVZWL  (R5),R0                     ;GET STATUS FROM IOSB
             06 50      E9    0292          BLBC    R0,60$                      ;IF ERROR, THEN QUIT
         52     57      C0    0295          ADDL    R7,R2                       ;SET UP TO WRITE FROM NEXT VA
             CD 56      F5    0298          SOBGTR  R6,20$                      ;LOOP THROUGH THE RETRIEVAL POINTERS
                50      DD    029B  60$:    PUSHL   R0                          ;SAVE STATUS
                              029D          $DASSGN_S CHAN=(R4)                 ;DEASSIGN CHANNEL
                01      BA    02A7          POPR    #^M<R0>                     ;RESTORE STATUS
                              02A9  EXIT:
                        04    02A9          RET                                 ;AND RETURN
                              02AA
                              02AA  IPL__SYNCH:
```

```
00000008  02AA   554         .LONG   IPL$_SYNCH
          02AE   555
          02AE   556         ASSUME  <IPL__SYNCH-BEGIN_NOPAGE> LE 512
          02AE   557
          02AE   558         .DISABLE        LSB
          02AE   559
          02AE   560         .END
```

```
$$T1                           = 00000001                       TQE$V_ABSOLUTE                = 00000003
BASETIME                         00000000 R       02
BASETIMESZ                     = 00000017
BASEYEAR                       = 00000007
BEGIN_NOPAGE                     00000224 R       02
BOO$L_PARAM_MAP                = 00000004
BUG$_BADBOOTCB                   ********      X  02
COMPUTE_TODR                     00000093 R       02
CVT_TIME                         000000B8 R       02
ENDTIME                          00000017 R       02
ERROR                            0000017C R       02
EXE$A_SYSPARAM                   ********      X  02
EXE$BOOTCB_CHK                   ********      X  02
EXE$C_SYSEFN                     ********      X  02
EXE$GL_TODR                      ********      X  02
EXE$GL_TQFL                      ********      X  02
EXE$GQ_SYSTIME                   ********      X  02
EXE$GQ_TODCBASE                  ********      X  02
EXE$INSTIMQ                      ********      X  02
EXE$READP_TODR                   ********      X  02
EXE$SETIME                       0000004D RG      02
EXE$SETIME_INT                   00000043 RG      02
EXE$WRITEP_TODR                  ********      X  02
EXIT                             000002A9 R       02
IO$_WRITELBLK                  = 00000020
IPL$_POWER                     = 0000001F
IPL$_SYNCH                     = 00000008
IPL__SYNCH                       000002AA R       02
LCK$GQ_BITMAP_EXP                ********      X  02
LOCK_END                         00000219 R       02
LOCK_IPL                         00000215 R       02
LOCK_START                       00000125 R       02
PCB$Q_PRIV                     = 00000084
PR$_IPL                        = 00000012
PRV$V_LOG_IO                   = 00000007
PRV$V_OPER                     = 00000012
RECAL                            000000E2 R       02
SAVE_TIME                        00000219 R       02
SCALE                            00000185 R       02
SCALE2                           00000189 R       02
SET_TODR                         0000018D R       02
SET_TODR1                        00000195 R       02
SS$_ACCVIO                     = 0000000C
SS$_IVTIME                     = 00000184
SS$_NOPRIV                     = 00000024
SS$_NORMAL                     = 00000001
SYS$ASCTIM                       ********     GX  02
SYS$ASSIGN                       ********     GX  02
SYS$BINTIM                       ********     GX  02
SYS$DASSGN                       ********     GX  02
SYS$QIOW                         ********     GX  02
SYSDISK                          0000002E R       02
TIMADR                         = 00000004
TQE$B_RQTYPE                   = 0000000B
TQE$L_TQBL                     = 00000004
TQE$L_TQFL                     = 00000000
TQE$Q_TIME                     = 00000018
```

```
                                                +-----------------+
                                                ! Psect synopsis !
                                                +-----------------+

PSECT name                        Allocation          PSECT No.  Attributes
----------                        ----------          ---------  ----------
.  ABS  .                         00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                             00000000  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
YEXEPAGED                         000002AE  (   686.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                             +------------------------+
                                             ! Performance indicators !
                                             +------------------------+

Phase                      Page ‘aults    CPU Time        Elapsed Time
-----                      -----------    --------        ------------
Initialization                     29     00:00:00.08     00:00:00.27
Command processing                114     00:00:00.55     00:00:01.64
Pass 1                            334     00:00:11.44     00:00:26.49
Symbol table sort                   0     00:00:01.86     00:00:03.70
Pass 2                            117     00:00:02.42     00:00:05.17
Symbol table output                 8     00:00:00.09     00:00:00.22
Psect synopsis output               2     00:00:00.02     00:00:00.03
Cross-reference output              0     00:00:00.00     00:00:00.00
Assembler run totals              606     00:00:16.46     00:00:37.52

The working set limit was 1500 pages.
64976 bytes (127 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1174 non-local and 13 local symbols.
560 source lines were read in Pass 1, producing 18 object records in Pass 2.
31 pages of virtual memory were used to define 30 macros.

                                            +--------------------------+
                                            ! Macro library statistics !
                                            +--------------------------+

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                        10
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     17
TOTALS (all libraries)                                27

1361 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSSETIME/OBJ=OBJ$:SYSSETIME MSRC$:SYSSETIME/UPDATE=(ENH$:SYSSETIME)+EXECML$/LIB
```

0388 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSSETPRA
LIS

SYSSETSSF
LIS

SYSSNDJBC
LIS

SYSSETIME
LIS

SYSSETPFM
LIS

SYSUPDSEC
LIS

SYSSNDMSG
LIS

SYSSETPRT
LIS

SYSSETPRV
LIS

SYSUNWIND
LIS

SYSSETSTK
LIS

SYSSETMOD
LIS

SYSSETEXV
LIS

SYSSETPRI
LIS

SYSSCHEVT
LIS