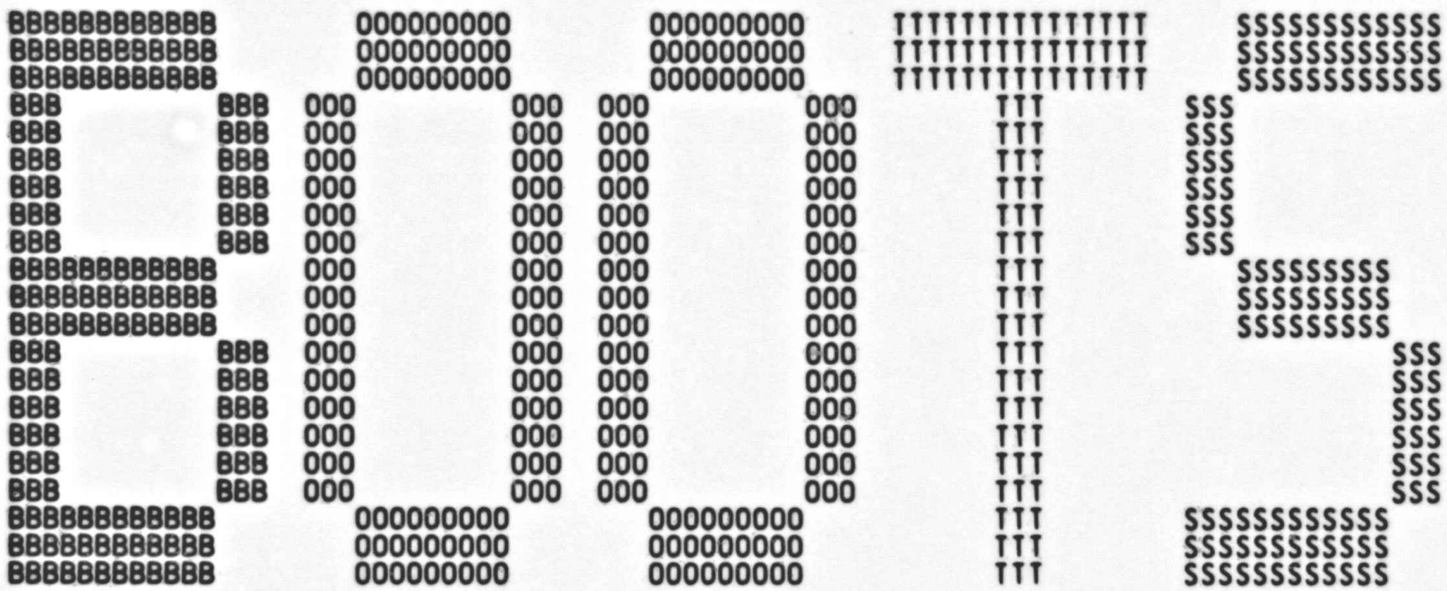
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTTTTTTTTTTTTTT SSSSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOO OOO OOO OOO TTT SSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTT SSSSSSSSSSS

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA    LL          OOOOOO     CCCCCCCC  KK      KK
  SSSSSSSS  TTTTTTTTTT    AAAAAA    LL          OOOOOO     CCCCCCCC  KK      KK
SS              TT       AA    AA  LL        OO      OO  CC        KK      KK
SS              TT       AA    AA  LL        OO      OO  CC        KK      KK
SS              TT       AA    AA  LL        OO      OO  CC        KK    KK
SS              TT       AA    AA  LL        OO      OO  CC        KK    KK
  SSSSSS        TT       AA    AA  LL        OO      OO  CC        KKKKKK
  SSSSSS        TT       AA    AA  LL        OO      OO  CC        KKKKKK
        SS      TT       AAAAAAAAAA  LL      OO      OO  CC        KK    KK
        SS      TT       AAAAAAAAAA  LL      OO      OO  CC        KK    KK
        SS      TT       AA    AA  LL        OO      OO  CC        KK      KK      ....
        SS      TT       AA    AA  LL        OO      OO  CC        KK      KK      ....
SSSSSSSS        TT       AA    AA  LLLLLLLLLL  OOOOOO     CCCCCCCC  KK      KK      ....
SSSSSSSS        TT       AA    AA  LLLLLLLLLL  OOOOOO     CCCCCCCC  KK      KK      ....

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
```

Table of contents

```
00000001  0000     1          STASW = 1                   ; SET SWITCH FOR STANDALONE CODE VERSIONS
          0000     1          .IF     NDF,STASW
          0000     2          .TITLE  LOCKDATA - Routines to lock/unlock SYSGEN database
          0000     3          .IFF
          0000     4          .TITLE  STALOCK  - Dummy routines for STASYSGEN
          0000     5          .ENDC
          0000     6          .IDENT  'V04-000'
          0000     7  ;
          0000     8  ;*****************************************************************************
          0000     9  ;*                                                                           *
          0000    10  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
          0000    11  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
          0000    12  ;*  ALL RIGHTS RESERVED.                                                     *
          0000    13  ;*                                                                           *
          0000    14  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
          0000    15  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
          0000    16  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
          0000    17  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
          0000    18  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
          0000    19  ;*  TRANSFERRED.                                                             *
          0000    20  ;*                                                                           *
          0000    21  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
          0000    22  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
          0000    23  ;*  CORPORATION.                                                             *
          0000    24  ;*                                                                           *
          0000    25  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
          0000    26  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
          0000    27  ;*                                                                           *
          0000    28  ;*                                                                           *
          0000    29  ;*****************************************************************************
          0000    30
          0000    31  ;++
          0000    32  ;
          0000    33  ; Facility:  System generation and initialization
          0000    34  ;
          0000    35  ; Abstract:  LOCKDATA provides subroutines which allow synchronization of
          0000    36  ;            access to the SYSGEN device database.
          0000    37  ;
          0000    38  ; Environment:
          0000    39  ;
          0000    40  ; Author:  Maryann Hinden, Creation date: 08-June-1983
          0000    41  ;
          0000    42  ; Modification History:
          0000    43  ;
          0000    44  ;       V03-002 WHM0001         Bill Matthews           29-Feb-1984
          0000    45  ;               Enable queueing of the SYSGEN database lock request.
          0000    46  ;
          0000    47  ;       V03-001 MSH0001         Maryann Hinden          13-Jul-1983
          0000    48  ;               Use general addressing mode for EXE$GL_SYSID_LOCK.
          0000    49  ;--
          0000    50
          0000    51  ;
          0000    52  ; Include files:
          0000    53  ;
          0000    54          $LCKDEF                         ; Define lock manager symbols
          0000    55          $SSDEF                          ; Define system status values
          0000    56          $SYSGMSGDEF                     ; Sysgen messages
```

```
                                0000     57
                                0000     58 ;
                                0000     59 ; Equated Symbols:
                                0000     60 ;
                     00000002   0000     61          LOCK_EFN = 2                        ; Event flag number used for lock request
                     00000018   0000     62          LOCK_FLAGS = LCK$M_SYNCSTS!-        ; Flags specified for lock request
                                0000     63                       LCK$M_SYSTEM
                                0000     64
                            00000000     65          .PSECT PAGED_DATA       NOEXE
                                0000     66 ;
                                0000     67 ;  Data for SYSGEN database lock
                                0000     68 ;
                                0000     69 BOO$GB_RESDSC::                              ; Descriptor for resource name
4E 45 47 53 59 53 00000008'010E0000'    0000     70          .ASCID  /SYSGEN$_DATABASE/
      45 53 41 42 41 54 41 44 5F 24     000E
                                0018     71
                                0018     72 BOO$LOCK_STATUS::                            ; LOCK_STATUS and LOCK_ID form
                     00000000   0018     73          .LONG   0                           ; the lock status block for the
                                001C     74 BOO$LOCK_ID::                                ; lock request
                     00000000   001C     75          .LONG   0
                                0020     76
                            00000000     77          .PSECT PAGED_CODE       NOWRT
```

```
                0000    79
```

```
0000     81          .SBTTL  BOO$LOCK_GEN - Lock SYSGEN database
0000     82 ;++
0000     83 ;  Functional description
0000     84 ;       Lock the SYSGEN database for create/modify, in order to
0000     85 ;       synchronize I/O database building.  Used by LOAD, RELOAD,
0000     86 ;       AUTOCONFIGURE, CONNECT.
0000     87 ;
0000     88 ;       This routine attempts to acquire an exclusive mode system lock on
0000     89 ;       the SYSGEN$_DATABASE resource in executive mode.  So that this resource
0000     90 ;       is only specific to the local system, rather than a cluster, the $ENQ
0000     91 ;       request specifies a parent (resource) which identifies the system on
0000     92 ;       which the request is being made.
0000     93 ;
0000     94 ;       The $ENQW entry point is used, so that if the resource is not
0000     95 ;       available immediately, the process will wait, with the assumption being
0000     96 ;       that it will get the resource soon.  An event flag number is specified
0000     97 ;       to avoid possible interference with CONFIGURE.
0000     98 ;
0000     99 ;       An alternate version of this routine (STALOCK) is provided in
0000    100 ;       which the calls to the entry points simply return a success status.
0000    101 ;       This is provided for use by STASYSGEN which runs standalone, and
0000    102 ;       therefore no locking is necessary.
0000    103 ;
0000    104 ;  Calling sequence
0000    105 ;       JSB/BSBx   BOO$LOCK_GEN
0000    106 ;
0000    107 ;  Inputs
0000    108 ;       NONE
0000    109 ;
0000    110 ;  Outputs
0000    111 ;       R0 - If LBS, locked database
0000    112 ;                  LBC, then no current access to database (SYSG$_NOLOCK)
0000    113 ;--
0000    114
0000    115 BOO$LOCK_GEN::
0000    116
0000    117          .IF     NDF, STASW
0000    118          $CMEXEC_S  LOCK                                ; Change mode to take out lock
0000    119          RSB                                            ; Return
0000    120
0000    121 LOCK:    .WORD   ^M<R2>                                 ; Null entry mask
0000    122          MOVL    G^EXE$GL_SYSID_LOCK,R0                 ; Get ID of parent lock
0000    123          $ENQW_S efn     = #LOCK_EFN,-
0000    124                  lkmode  = #LCK$K_EXMODE,-
0000    125                  lksb    = BOO$LOCK_STATUS,-
0000    126                  flags   = #LOCK_FLAGS,-
0000    127                  resnam  = BOO$GB_RESDSC,-
0000    128                  parid   = R0
0000    129          BLBC    R0,10$                                 ; If LBC, error
0000    130          MOVZWL  BOO$LOCK_STATUS,R0                     ; Get final status
0000    131          BLBC    R0,10$                                 ; Success?
0000    132          RET                                            ; Yes
0000    133 10$:     MOVL    #SYSG$_NOLOCK,R0                       ; Indicate error
0000    134          RET
0000    135
0000    136          .IFF
0000    137
```

```
                50   01   3C  0000   138          MOVZWL  #SS$_NORMAL, R0                 ; Force success
                          05  0003   139          RSB
                              0004   140
                              0004   141          .ENDC
                              0004   142
```

```
                        0004   144          .SBTTL  BOO$UNLOCK_GEN - Unlock SYSGEN database
                        0004   145 ;++
                        0004   146 ;  Functional description
                        0004   147 ;       Dequeue the lock requested by BOO$LOCK_GEN.
                        0004   148 ;
                        0004   149 ;  Calling sequence
                        0004   150 ;       JSB/BSBx   BOO$UNLOCK_GEN
                        0004   151 ;
                        0004   152 ;  Input
                        0004   153 ;       Lock id in lock status block (implicit).
                        0004   154 ;
                        0004   155 ;  Output
                        0004   156 ;       R0 - If LBS, successful completion
                        0004   157 ;            LBC, error on dequeue (probably serious) - status SYSG$_DEQERR.
                        0004   158 ;--
                        0004   159
                        0004   160 BOO$UNLOCK_GEN::
                        0004   161
                        0004   162          .IF     NDF,STASW
                        0004   163
                        0004   164          $CMEXEC_S  UNLOCK                  ; Change mode to access lock
                        0004   165          RSB                                ; Return
                        0004   166
                        0004   167 UNLOCK: .WORD   0                          ; Null entry mask
                        0004   168          $DEQ_S  lkid = BOO$LOCK_ID         ; Dequeue lock
                        0004   169          BLBS    R0,10$                     ; If LBS, all okay
                        0004   170          MOVL    #SYSG$_DEQERR,R0           ; Indicate error
                        0004   171 10$:     RET
                        0004   172
                        0004   173          .IFF
                        0004   174
 50    01    3C         0004   175          MOVZWL  #SS$_NORMAL,R0             ; Force success
       05               0007   176          RSB
                        0008   177
                        0008   178          .ENDC
                        0008   179          .END
```

```
BOO$GB_RESDSC                   00000000 RG    02
BOO$LOCK_GEN                    00000000 RG    03
BOO$LOCK_ID                     0000001C RG    02
BOO$LOCK_STATUS                 00000018 RG    02
BOO$UNLOCK_GEN                  00000004 RG    03
LCK$M_SYNCSTS                 = 00000008
LCK$M_SYSTEM                  = 00000010
LOCK_EFN                      = 00000002
LOCK_FLAGS                    = 00000018
SS$_NORMAL                    = 00000001
STASW                         = 00000001
```

## 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 |
| PAGED_DATA | 00000020 ( 32.) | 02 ( 2.) | NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE |
| PAGED_CODE | 00000008 ( 8.) | 03 ( 3.) | NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE |

## Performance indicators

| Phase | Page faults | CPU Time | Elapsed Time |
|---|---|---|---|
| Initialization | 31 | 00:00:00.07 | 00:00:00.49 |
| Command processing | 108 | 00:00:00.88 | 00:00:04.00 |
| Pass 1 | 238 | 00:00:05.06 | 00:00:11.27 |
| Symbol table sort | 0 | 00:00:00.77 | 00:00:02.41 |
| Pass 2 | 49 | 00:00:00.92 | 00:00:01.59 |
| Symbol table output | 3 | 00:00:00.02 | 00:00:00.03 |
| Psect synopsis output | 2 | 00:00:00.01 | 00:00:00.02 |
| Cross-reference output | 0 | 00:00:00.00 | 00:00:00.00 |
| Assembler run totals | 433 | 00:00:07.75 | 00:00:19.82 |

```
The working set limit was 1050 pages.
24893 bytes (49 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 508 non-local and 0 local symbols.
180 source lines were read in Pass 1, producing 15 object records in Pass 2.
10 pages of virtual memory were used to define 9 macros.
```

## Macro library statistics

| Macro library name | Macros defined |
|---|---|
| _$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1 | 0 |
| _$255$DUA28:[SYS.OBJ]LIB.MLB;1 | 1 |
| _$255$DUA28:[SYSLIB]STARLET.MLB;2 | 5 |
| TOTALS (all libraries) | 6 |

```
575 GETS were required to define 6 macros.
```

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:STALOCK/OBJ=OBJ$:STALOCK MSRC$:STASW/UPDATE=(ENH$:STASW)+MSRC$:LOCKDATA/UPDATE=(ENH$:LOCKDATA)+EXECML$/LIB+LIB$:BOOTS
```

SYSBOOCMD
LIS

SHOWADAP
LIS

STANDCONF
LIS

SHODEV
LIS

STAPUTERR
LIS

STALOCK
LIS

STASGNMSG
LIS

STACONFIG
LIS

STASYSGEN
LIS

SYSBOOT
LIS

STARDBRIV
LIS