```
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
```

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA      SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  NN      NN
  SSSSSSSS  TTTTTTTTTT    AAAAAA      SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  NN      NN
SS              TT      AA      AA  SS          YY      YY  SS          GG            EE          NN      NN
SS              TT      AA      AA  SS          YY      YY  SS          GG            EE          NN      NN
SS              TT      AA      AA  SS            YY  YY    SS          GG            EE          NNNN    NN
SS              TT      AA      AA  SS            YY  YY    SS          GG            EE          NNNN    NN
  SSSSSS        TT      AA      AA    SSSSSS        YY        SSSSSS    GG            EEEEEEEE    NN  NN  NN
  SSSSSS        TT      AA      AA    SSSSSS        YY        SSSSSS    GG            EEEEEEEE    NN  NN  NN
        SS      TT      AAAAAAAAAA          SS      YY              SS  GG  GGGGGG    EE          NN    NNNN
        SS      TT      AAAAAAAAAA          SS      YY              SS  GG  GGGGGG    EE          NN    NNNN
        SS      TT      AA      AA          SS      YY              SS  GG      GG    EE          NN      NN
        SS      TT      AA      AA          SS      YY              SS  GG      GG    EE          NN      NN
SSSSSSSS        TT      AA      AA  SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE  NN      NN
SSSSSSSS        TT      AA      AA  SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE  NN      NN

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

```
0000     1          .TITLE  STASYSGEN - MAIN PROGRAM FOR STANDALONE SYSGEN
0000     2          .IDENT  'V04-000'
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY:     STANDALONE SYSGEN
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;       THIS ROUTINE IS THE MAIN PROGRAM AND SUBROUTINES FOR
0000    34 ;       STANDALONE SYSGEN.  STANDALONE SYSGEN IS USED IN STANDALONE
0000    35 ;       BACKUP TO AUTOCONFIGURE ALL DISKS AND TAPES.
0000    36 ;
0000    37 ; ENVIRONMENT:  USER, EXEC, AND KERNEL MODE
0000    38 ;
0000    39 ; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  18-SEP-1979
0000    40 ;
0000    41 ; MODIFIED BY:
0000    42 ;
0000    43 ;       V03-016 CWH3016         CW Hobbs                    26-Aug-1984
0000    44 ;               Move the fix in KDM0091 so that the device will be
0000    45 ;               available when the attempt is made to read its volume
0000    46 ;               label for the 'Resuming operation on volume "" message'
0000    47 ;
0000    48 ;       V03-015 WHM0001         Bill Matthews               14-Apr-1984
0000    49 ;               Added definitions of dummy entry points for BOO$SETASCII,
0000    50 ;               BOO$GT_SYSPARFILE, and BOO$WRTSYSPARFILE.
0000    51 ;
0000    52 ;       V03-014 CWH3014         CW Hobbs                    15-Mar-1984
0000    53 ;               Require a Yes response to the prompt requesting the next
0000    54 ;               piece of media.  Print a message telling which piece to
0000    55 ;               remove, which has the side effect of rewinding the
0000    56 ;               previous volume.  This is necessary to prevent timeouts
0000    57 ;               while a VAX-11/750 is rewinding the TU58 device.
```

```
0000     58 ;
0000     59 ;       V03-013 KDM0091         Kathleen D. Morse       11-Nov-1983
0000     60 ;               For MSCP devices, the drive must be set back online after
0000     61 ;               replacing one piece of media with another.
0000     62 ;
0000     63 ;       V03-012 KDM0090         Kathleen D. Morse       10-Nov-1983
0000     64 ;               Align psect BOO$SYSGEN on word boundry so that STASYSGEN
0000     65 ;               works on MicroVAX I.
0000     66 ;
0000     67 ;       V03-011 KDM0089         Kathleen D. Morse       09-Nov-1983
0000     68 ;               Add MicroVAX I and II support to the test for a small
0000     69 ;               console as a boot device.
0000     70 ;
0000     71 ;       V03-010 CWH3010         CW Hobbs                25-Sep-1983
0000     72 ;               Change messages about the configure process.  Start
0000     73 ;               STANDCONF with the process name "STANDLOAD" and wait
0000     74 ;               until it changes its name to "STANDCONF" before proceeding.
0000     75 ;               This is to prevent asking the user to insert the next
0000     76 ;               console volume while still loading STANDCONF.
0000     77 ;
0000     78 ;       V03-009 CWH3009         CW Hobbs                29-Aug-1983
0000     79 ;               Add call to BOO$USEACT to perform an implicit USE /ACTIVE
0000     80 ;               Other changes to support V3b packaging changes (i.e. 3 pieces
0000     81 ;               of console media).
0000     82 ;
0000     83 ;       V03-008 WMC0002         Wayne Cardoza           01-Aug-1983
0000     84 ;               Fix a typo.
0000     85 ;
0000     86 ;       V03-007 KDM0057         Kathleen D. Morse       28-Jul-1983
0000     87 ;               Replace SIP_SETTIME logic with a call to a loadable,
0000     88 ;               cpu-dependent routine, EXE$INIT_TODR.
0000     89 ;
0000     90 ;       V03-006 WMC0001         Wayne Cardoza           27-Jul-1983
0000     91 ;               Definitions for BOO$FILCLOSE, BOO$UFOOPEN, EXE$LOAD_CODE
0000     92 ;
0000     93 ;       V03-005 MSH0003         Maryann Hinden          07-Jul-1983
0000     94 ;               Move PUTERROR, IOGEN$READDRIV to separate modules.
0000     95 ;
0000     96 ;       V03-004 MSH0002         Maryann Hinden          23-Jun-1983
0000     97 ;               Use $BOOCMDDEF in other modules - global definition here
0000     98 ;               not needed.
0000     99 ;
0000    100 ;       V03-003 MSH0001         Maryann Hinden          15-Feb-1983
0000    101 ;               Configure TU81.
0000    102 ;
0000    103 ;       V03-002 ACG53600        Andrew C. Goldstein,    10-Feb-1983  16:57
0000    104 ;               Add code to prompt for system date
0000    105 ;
0000    106 ;       V03-001 MLJ0086         Martin L. Jack, 5-Apr-1982  10:56
0000    107 ;               Add PU to device select list.
0000    108 ;
0000    109 ;       V02-011 MLJ0065         Martin L. Jack, 23-Dec-1981  3:07
0000    110 ;               Rearrange order of operations so that volume switching occurs
0000    111 ;               before autoconfigure, to support repackaging of most drivers on
0000    112 ;               second volume.
0000    113 ;
0000    114 ;       V02-010 MLJ0034         Martin L. Jack, 27-Aug-1981  16:06
```

```
0000   115 ;                   Improve wording of volume switch prompt.
0000   116 ;
0000   117 ;       V02-009 MLJ0032         Martin L. Jack, 10-Aug-1981  14:35
0000   118 ;               Change image name to STABACKUP.  Autoconfigure line printers.
0000   119 ;               Issue prompt to switch volumes only if booting from console.
0000   120 ;
0000   121 ;       V02-008 JLV0037         Jake VanNoy             7-Jul-1981
0000   122 ;               Added dummy entry point for LBR$OUTPUT_HELP, the new
0000   123 ;               SYSGEN help routine.
0000   124 ;
0000   125 ;       V02-007 JLV0030         Jake VanNoy             27-June-1981
0000   126 ;               Added RIO_INPNAM symbol to resolve SYSGEN reference.
0000   127 ;
0000   128 ;       V006    TCM0001         Trudy C. Matthews       18-Jun-1981
0000   129 ;               Added code to disable and deallocate boot file cacheing,
0000   130 ;               so that the DSC.EXE image on the second floppy can be
0000   131 ;               found.
0000   132 ;
0000   133 ;       V005    JLV0011         Jake VanNoy             7-May-1981
0000   134 ;               ADDED BOO$V&M_AUTOLOG OFFSETS TO RESOLVE SYSGEN SYMBOLS.
0000   135 ;
0000   136 ;       V004    JLV0003         JAKE VANNOY             27-FEB-1981
0000   137 ;               ADDED BOO$_ OFFSETS TO RESOLVE SYSGEN SYMBOLS.
0000   138 ;
0000   139 ;       V003    JLV0002         Jake VanNoy             11-Feb-1981
0000   140 ;               Added a BOO$GL_DOT label so stand alone sysgen can
0000   141 ;               link with SYSGEN modules. Connected to a bug fix in
0000   142 ;               the USE <filespec> in SYSGEN.
0000   143 ;
0000   144 ;       V002    LMK0001         LEN KAWELL              09-FEB-1980
0000   145 ;               CHANGE IOGEN$READDRIV TO READ THE IMAGE HEADER.
0000   146 ;--
```

```
                        0000   148          .SBTTL  DECLARATIONS
                        0000   149 ;
                        0000   150 ; INCLUDE FILES:
                        0000   151 ;
                        0000   152
                        0000   153          $BTDDEF
                        0000   154          $DPTDEF
                        0000   155          $JPIDEF
                        0000   156          $PQLDEF
                        0000   157          $PRDEF
                        0000   158          $RPBDEF                         ; Define RPB offsets
                        0000   159          $UCBDEF                         ; Define UCB offsets
                        0000   160
                        0000   161 ;
                        0000   162 ; MACROS:
                        0000   163 ;
                        0000   164          .MACRO  ERROR   MSG
                        0000   165          BSBW    ERROR_HALT
                        0000   166          .ASCIZ  \'MSG'\
                        0000   167          .ENDM   ERROR
                        0000   168
                        0000   169 ;
                        0000   170 ; EQUATED SYMBOLS:
                        0000   171 ;
                        0000   172
0000000D                0000   173 CR=^XD                                  ; ASCII carriage return
0000000A                0000   174 LF=^XA                                  ; ASCII line feed
                        0000   175
                        0000   176 ;
                        0000   177 ; OWN STORAGE:
                        0000   178 ;
                        0000   179
                00000000  0000   180          .PSECT  BOO$SYSGEN,WRT,WORD
                        0000   181
                00000000  0000   182 BOO$GL_CMDOPT:: .LONG 0                ; Options longword
                00000000  0004   183 BOO$GL_DOT::    .LONG 0                ; Dummy label, "."
                00000000  0008   184 RIO_INPNAM::    .LONG 0                ; Dummy label
                          000C   185                                        ; not used in stand alone version
                          000C   186
                          000C   187 BOOTFLAGS:                             ; Boot flags
                00000010  000C   188          .BLKL   1
                          0010   189 BOOT_SMALL_CONSOLE:
                00000014  0010   190          .BLKL   1                      ; Will have to do volume switch
                          0014   191
                00000018  0014   192 INPBUF: .BLKL   1                      ; Dummy input buffer
                          0018   193
                          0018   194 SYSDISK_CHAN:
                0000001C  0018   195          .BLKL   1
                          001C   196
                          001C   197 SYSDISK_NAME:
49 44 24 53 59 53 00000024'010E0000'  001C   198          .ASCID  "SYS$DISK:"
                        3A 4B 53  002A
                          002D   199
                          002D   200 ; Prompt string for switching console media
                          002D   201
68 74 20 74 72 65 73 6E 49 07 0A 0D  002D   202 SWITCHPROMPT:   .ASCII  <CR><LF><7>/Insert the standalone application volume /
65 6E 6F 6C 61 64 6E 61 74 73 20 65  0039
```

```
6E 6F 69 74 61 63 69 6C 70 70 61 20  0045
            20 65 6D 75 6C 6F 76 20  0051
59 22 20 72 65 74 6E 65 20 64 6E 61  0059   203                 .ASCIZ  /and enter "YES" when ready: /
61 65 72 20 6E 65 68 77 20 22 53 45  0065
                  00 20 3A 79 64     0071
                                     0076   204
65 72 20 65 73 61 65 6C 50 07 0A 0D  0076   205 REMOVEPROMPT1:  .ASCIZ  <CR><LF><7>/Please remove the volume "/<7>
6C 6F 76 20 65 68 74 20 65 76 6F 6D  0082
               00 07 22 20 65 6D 75  008E
63 20 65 68 74 20 6D 6F 72 66 20 22  0095   206 REMOVEPROMPT2:  .ASCIZ  /" from the console device./<7><CR><LF>
63 69 76 65 64 20 65 6C 6F 73 6E 6F  00A1
               00 0A 0D 07 2E 65     00AD
                                     00B3   207
20 67 6E 69 6D 75 73 65 52 0A 0A 0D  00B3   208 RESUMEPROMPT1:  .ASCIZ  <CR><LF><LF>/Resuming load operation on volume "/
69 74 61 72 65 70 6F 20 64 61 6F 6C  00BF
65 6D 75 6C 6F 76 20 6E 6F 20 6E 6F  00CB
                  00 22 20           00D7
74 73 20 65 73 61 65 6C 70 20 2C 22  00DA   209 RESUMEPROMPT2:  .ASCIZ  /", please stand by . . ./<CR><LF><LF>
2E 20 2E 20 2E 20 79 62 20 64 6E 61  00E6
               00 0A 0A 0D           00F2
                                     00F6   210
                                     00F6   211 IMAGENAME:                              ; Image name to activate after
41 44 4E 41 54 53 000000FE'010E0000' 00F6   212         .ASCID  /STANDALON.EXE/         ; SYSGEN is finished
            45 58 45 2E 4E 4F 4C     0104
                                     010B   213
                                     010B   214 SELECT_LIST:                           ; Select list for AUTOCONFIGURE ALL
                              44 00' 010B   215         .ASCIC  /D/                     ; All disks
                                 01  010B
                              4D 00' 010D   216         .ASCIC  /M/                     ; All tapes
                                 01  010D
                           50 4C 00' 010F   217         .ASCIC  /LP/                    ; Line printers
                                 02  010F
                           43 4C 00' 0112   218         .ASCIC  /LC/                    ;
                                 02  0112
                              50 00' 0115   219         .ASCIC  /P/                     ; All port drivers plus paper tape
                                 01  0115
                                 00  0117   220         .BYTE   0                       ; End of list
                                     0118   221
                                     0118   222 ;
                                     0118   223 ; Data used by the autoconfigure.
                                     0118   224 ;
                                     0118   225
                     FFFFFFFF FF676980 0118   226 TIMER:          .LONG   -10*1000*1000,-1        ; One second
                              00000000 0120   227 PA_POLL_INT:    .LONG   0                       ; Number of seconds in port poll int
                              00000000 0124   228 PRC_POLL_INT:   .LONG   0                       ; Number of seconds in process poll
                              00 0A 0D 0128   229 CONFCRLF:       .ASCIZ  <CR><LF>
6F 72 70 20 67 6E 69 74 61 65 72 43  012B   230 CONFLOAD:       .ASCIZ  /Creating process to configure remote devices . . ./<CR><LF>
66 6E 6F 63 20 6F 74 20 73 73 65 63  0137
65 74 6F 6D 65 72 20 65 72 75 67 69  0143
2E 20 2E 20 73 65 63 69 76 65 64 20  014F
               00 0A 0D 2E 20        015B
72 75 67 69 66 6E 6F 63 20 77 6F 4E  0160   231 CONFWAIT:       .ASCIZ  /Now configuring HSC and MSCP-served devices . . ./<CR><LF>
20 64 6E 61 20 43 53 48 20 67 6E 69  016C
20 64 65 76 72 65 73 2D 50 43 53 4D  0178
20 2E 20 2E 20 73 65 63 69 76 65 64  0184
               00 0A 0D 2E           0190
                                     0194   232
```

```
                                        0194    ;
                                        0194    ; The following data is used in creating the stand-alone Configure Process
                                        0194    ;
43 44 4E 41 54 53 0000019C'010E0000'    0194    STAC_IMAGE:     .ASCID  /STANDCONF.EXE/ ; Image name
          45 58 45 2E 46 4E 4F          01A2
3A 30 41 50 4F 5F 000001B1'010E0000'    01A9    STAC_OPER:      .ASCID  /_OPA0:/        ; Input/output device
                FFFFFFFF FFFFFFFF       01B7    STAC_PRV_MSK:   .LONG   -1,-1           ; Let it do anything it wants to do
4C 44 4E 41 54 53 000001C7'010E0000'    01BF    STAC_PRC:       .ASCID  /STANDLOAD/     ; Initial process name
                         44 41 4F       01CD
                         00000000       01D0    STAC_PID:       .LONG   0               ; Gets the process pid
                                        01D4    STAC_QLIST:                             ; And a generous list of quotas
                               01       01D4            .BYTE   PQL$_ASTLM
                         000000C8       01D5            .LONG   200
                               02       01D9            .BYTE   PQL$_BIOLM
                         000000C8       01DA            .LONG   200
                               03       01DE            .BYTE   PQL$_BYTLM
                         000186A0       01DF            .LONG   100000
                               04       01E3            .BYTE   PQL$_CPULM
                         00000000       01E4            .LONG   0
                               05       01E8            .BYTE   PQL$_DIOLM
                         000000C8       01E9            .LONG   200
                               0C       01ED            .BYTE   PQL$_ENQLM
                         000000C8       01EE            .LONG   200
                               06       01F2            .BYTE   PQL$_FILLM
                         000000C8       01F3            .LONG   200
                               07       01F7            .BYTE   PQL$_PGFLQUOTA
                         00005000       01F8            .LONG   20480
                               08       01FC            .BYTE   PQL$_PRCLM
                         000000C8       01FD            .LONG   200
                               09       0201            .BYTE   PQL$_TQELM
                         000000C8       0202            .LONG   200
                               0B       0206            .BYTE   PQL$_WSDEFAULT
                         000003E8       0207            .LONG   1000
                               0A       020B            .BYTE   PQL$_WSQUOTA
                         000003E8       020C            .LONG   1000
                               00       0210            .BYTE   PQL$_LISTEND
                                        0211    ;
                                        0211    ; Some data needed so that we will see when STANDCONF changes it's name
                                        0211    ;
       46 4E 4F 43 44 4E 41 54 53       0211    STANDCONF:      .ASCII  "STANDCONF"
                         00000000       021A    JPI_LENGTH:     .LONG   0
                         0000022E       021E    JPI_NAME:       .BLKB   16
                             0010       022E    JPI_ITEM:       .WORD   16              ; Buffer is 16 bytes long
                             031C       0230                    .WORD   JPI$_PRCNAM     ; Item desired is process name
                        0000021E'       0232                    .ADDRESS JPI_NAME       ; Output buffer
                        0000021A'       0236                    .ADDRESS JPI_LENGTH     ; Output length
                         00000000       023A                    .LONG   0               ; End of list
```

```
                     023E    279          .SBTTL  BOO$SYSGEN - main program
                     023E    280 ;++
                     023E    281 ; FUNCTIONAL DESCRIPTION:
                     023E    282 ;       This is the main program for standalone SYSGEN.  It does the
                     023E    283 ;       following:
                     023E    284 ;
                     023E    285 ;               1)  Locks the entire image into the working set.
                     023E    286 ;               2)  Sets up the system time and saves the current
                     023E    287 ;                   SYSGEN parameters in SYS.EXE.
                     023E    288 ;               3)  Autoconfigures disks, tapes, and line printers.
                     023E    289 ;               4)  Optionally does a SHOW/DEVICES.
                     023E    290 ;               5)  Activates the next image, STABACKUP.
                     023E    291 ;
                     023E    292 ; CALLING SEQUENCE:
                     023E    293 ;
                     023E    294 ;       Called by the Image activator
                     023E    295 ;
                     023E    296 ; INPUT PARAMETERS:
                     023E    297 ;
                     023E    298 ;       NONE
                     023E    299 ;
                     023E    300 ; OUTPUT PARAMETERS:
                     023E    301 ;
                     023E    302 ;       R0      Completion code
                     023E    303 ;
                     023E    304 ; COMPLETION CODES:
                     023E    305 ;
                     023E    306 ;       Various errors
                     023E    307 ;
                     023E    308 ;--
                     023E    309
                0000 023E    310          .ENTRY  BOO$SYSGEN,^M<>
                     0240    311          $LKWSET_S -                     ; Lock entire image in working set
                     0240    312                  INADR=BOO$GQ_LIMITS, -  ;
                     0240    313                  RETADR=BOO$GQ_RETADR    ;
                     0255    314 ;
                     0255    315 ; Set up the system time
                     0255    316 ;
                     0255    317          $CMKRNL_S W^SIP_SETTIME         ; Set up the system time
                     0262    318 ;
                     0262    319 ; Get boot flags and boot device type
                     0262    320 ;
                     0262    321          $CMKRNL_S  W^GET_RPB_INFO       ; Get boot flags and device type
                     026F    322 ;
                     026F    323 ; Initialize various other things which need kernel mode
                     026F    324 ;
                     026F    325          $CMKRNL_S  W^KERNEL_INIT
                     027C    326 ;
                     027C    327 ; Autoconfigure disks, tapes, and line printers.
                     027C    328 ;
    00000000'EF   00 FB  027C    329          CALLS   #0,BOO$USEACT           ; Get current values for local SYSPARAM
00000000'EF   FE84 CF 9E 0283    330          MOVAB   SELECT_LIST,BOO$GL_SELECT ; Use builtin select list
    00000000'EF   00 FB  028C    331          CALLS   #0,BOO$CONFIGALL        ; Autoconfigure all adapters
                     0293    332 ;
                     0293    333 ; Check if there is a CI on system, if so wait for devices to show up.
                     0293    334 ;
                     0293    335          $CMKRNL_S  W^FINDCI             ; Find any CI adapter
```

```
                03 50   E8  02A0   336          BLBS    R0,10$                  ; None
                 0103   31  02A3   337          BRW     30$
                        02A6   338 10$:     $CREPRC_S -                         ; Start up the STANDCONF process to
                        02A6   339                IMAGE = STAC_IMAGE,-          ;  asynchronously find and configure
                        02A6   340                INPUT = STAC_OPER,-           ;  all devices served by HSC and MSCP-server
                        02A6   341                OUTPUT = STAC_OPER,-          ;  hosts.
                        02A6   342                ERROR = STAC_OPER,-
                        02A6   343                PRVADR = STAC_PRV_MSK,-
                        02A6   344                QUOTA = STAC_QLIST,-
                        02A6   345                PRCNAM = STAC_PRC,-
                        02A6   346                PIDADR = STAC_PID,-
                        02A6   347                BASPRI = #8,-
                        02A6   348                UIC = #^x10004
                6B 50   E9  02DC   349          BLBC    R0,23$
                        02DF   350 ;
                        02DF   351 ; STANDCONF has been created with the process name "STANDLOAD".  We will now
                        02DF   352 ; wait for it to change its process name to "STANDCONF".  This will let us know
                        02DF   353 ; that it is loaded and executing, so that we won't ask the human to load the
                        02DF   354 ; next piece of console media while the STANDCONF image is still being read.
                        02DF   355 ;
                   7E   7C  02DF   356 20$:     CLRQ    -(SP)                   ; No input buffer
                FE43 CF   9F  02E1   357          PUSHAB  CONFCRLF                ; Output a blank line
00000000'EF        03   FB  02E5   358          CALLS   #3,BOO$READPROMPT       ; Print it
                FD20 CF   95  02EC   359          TSTB    BOOT_SMALL_CONSOLE      ; Booting from a small console?
                   0D   13  02F0   360          BEQL    21$                     ; Disk boot, no need for encouragement
                   7E   7C  02F2   361          CLRQ    -(SP)                   ; No input buffer
                FE33 CF   9F  02F4   362          PUSHAB  CONFLOAD                ; Output a line to tickle the user
00000000'EF        03   FB  02F8   363          CALLS   #3,BOO$READPROMPT       ; Print it
                   55   D4  02FF   364 21$:     CLRL    R5                      ; Clear a counter
                        0301   365 22$:     $SETIMR_S EFN=#3,DAYTIM=TIMER       ; Set a timer for one second
                        0310   366          $WAITFR_S EFN=#3                    ;  ... and wait for it.
                        0319   367          $GETJPIW_S EFN=#3,-                 ; Get the process name using the PID
                        0319   368                    PIDADR=STAC_PID,-         ;  returned by creprc.
                        0319   369                    ITMLST=JPI_ITEM
                17 50   E9  0330   370          BLBC    R0,23$                  ; Blow up if we can't do the getjpi
00   FED9 CF    09   2D  0333   371          CMPC5   #9,STANDCONF,#0,-           ; Compare the name against "STANDCONF"
FEDF CF    FEDE CF      0339   372                  JPI_LENGTH,JPI_NAME        ;  to see if the process is active
                   33   13  033F   373          BEQL    25$                     ; The name we want, go do something
                   55   D6  0341   374          INCL    R5                      ; Increment our count
                55 012C 8F   B1  0343   375          CMPW    #300,R5             ; Have we been doing this for 5 minutes?
                   B7   14  0348   376          BGTR    22$                     ; Less than 5 minutes, do it again
                        034A   377 23$:     ERROR   <Configure process failed to initialize>
                        0374   378 ;
                        0374   379 ; STANDCONF is up and running, now wait for several polling intervals so that most
                        0374   380 ; devices should be configured before we give the BACKUP prompt.  Print a message
                        0374   381 ; that we are configuring so that the human won't give up hope.
                        0374   382 ;
                   7E   7C  0374   383 25$:     CLRQ    -(SP)                   ; No input buffer
                FDE6 CF   9F  0376   384          PUSHAB  CONFWAIT                ; Output string
 00000000'EF        03   FB  037A   385          CALLS   #3,BOO$READPROMPT       ; Print it
50   FD9A CF    03   C5  0381   386          MULL3   #3,PA_POLL_INT,R0           ; Thrice the port poll interval
     50   FD99 CF    C0  0387   387          ADDL2   PRC_POLL_INT,R0             ; Plus the process poll interval
     FD87 CF    50   C4  038C   388          MULL2   R0,TIMER                    ; Make into delta time
                        0391   389          $SETIMR_S #3,TIMER                  ; Set a timer
                        03A0   390          $WAITFR_S #3                        ; ... and wait for it.
                        03A9   391 ;
                        03A9   392 ; If XDELTA is mapped into the system, do a SHOW/DEVICES.
```

```
                                03A9    393 ;
 OF FC5E CF    01    E1         03A9    394 30$:    BBC     #RPB$V_DEBUG,BOOTFLAGS,40$ ; Branch if no XDELTA
                7E    7C         03AF    395         CLRQ    -(SP)                      ; Set up fake TPARSE parameter block
                7E    7C         03B1    396         CLRQ    -(SP)                      ;
                7E    7C         03B3    397         CLRQ    -(SP)                      ;
                7E    7C         03B5    398         CLRQ    -(SP)                      ;
00000000'EF     08    FB         03B7    399         CALLS   #8,BOO$SHODEV_ALL          ; Show devices
                                03BE    400 ;
                                03BE    401 ; Request operator to switch to second floppy or cartridge if necessary.
                                03BE    402 ;
          FC4E  CF    95         03BE    403 40$:    TSTB    BOOT_SMALL_CONSOLE         ; Do we have to switch volumes?
                4C    13         03C2    404         BEQL    50$                        ; Branch if no
          FCCD  CF    9F         03C4    405         PUSHAB  REMOVEPROMPT2              ; Pass address of second half of message
          FCAA  CF    9F         03C8    406         PUSHAB  REMOVEPROMPT1              ; Pass address of first half of message
0000041C'EF     02    FB         03CC    407         CALLS   #2,LABELMSG                ; Tell s/he to remove
          FC3D  CF    9F         03D3    408 45$:    PUSHAB  INPBUF                     ; Push address of input buffer
                04    DD         03D7    409         PUSHL   #4                         ; Push size of input buffer
          FC50  CF    9F         03D9    410         PUSHAB  SWITCHPROMPT               ; Push address of message
00000000'EF     03    FB         03DD    411         CALLS   #3,BOO$READPROMPT          ; Prompt and wait for return
     FC2C CF    20    8A         03E4    412         BICB    #32,INPBUF+1               ; Force to upper case (ASCIC string returned
 59 8F    FC28 CF    91         03E9    413         CMPB    INPBUF+1,#^A"Y"            ; Look for the Y
                E2    12         03EF    414         BNEQ    45$                        ; Loop until a Y is entered
                                03F1    415 ;
                                03F1    416 ; If device is MSCP, then it goes offline when we replace it with
                                03F1    417 ; another piece of media.  A PACKACK must be sent to the device to
                                03F1    418 ; allow us to continue.  Note:  mount verification cannot take care
                                03F1    419 ; of this for us, because there is no VCB and the system disk does not
                                03F1    420 ; have the "mounted" characteristic set.
                                03F1    421 ;
             011B   30          03F1    422         BSBW  W^MSCP_PACKACK               ; Set device back on line
          FCE2  CF    9F         03F4    423         PUSHAB  RESUMEPROMPT2              ; Pass address of second half of message
          FCB7  CF    9F         03F8    424         PUSHAB  RESUMEPROMPT1              ; Pass address of first half of message
0000041C'EF     02    FB         03FC    425         CALLS   #2,LABELMSG                ; Tell that we are resuming
                                0403    426 ;
                                0403    427 ; Disable file caching so FIL$OPENFILE will be forced to look
                                0403    428 ; at the new floppy or cartridge.
                                0403    429 ;
                                0403    430         $CMKRNL_S  W^DISABLE_FILE_CACHE ; Disable and de-allocate file cache
                                0410    431 ;
                                0410    432 ; Activate STABACKUP image.
                                0410    433 ;
          FCE2  CF    7F         0410    434 50$:    PUSHAQ  IMAGENAME                  ; Push address of image name descriptor
00000000'EF     01    FB         0414    435         CALLS   #1,BOO$ACTIMAGE            ; Activate next image
                04              041B    436         RET                                ; Should never return here
```

```
                                041C   438 ;
                                041C   439 ; Print a message, giving the volume label in the middle
                                041C   440 ; Inputs:       4(AP)  Pointer to ASCIZ string for front of message
                                041C   441 ;               8(AP)  Pointer to second half of message
                                041C   442 ;
                                041C   443 LABELMSG:
                           0FFC 041C   444         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                     5E   FE00 CE  9E  041E   445         MOVAB   -512(SP),SP             ; Make a buffer on the stack
                          56   5E  D0  0423   446         MOVL    SP,R6                   ; Save the buffer address
                                       0426   447         $ASSIGN_S -                     ; Assign a channel to sys$disk
                                       0426   448                 DEVNAM=SYSDISK_NAME,-
                                       0426   449                 CHAN=SYSDISK_CHAN
                                79 50  E9  0437   450         BLBC    R0,100$
                                       043A   451         $QIOW_S -                       ; Read the home block
                                       043A   452                 EFN=#16,-               ;  Use event flag 16, easy to see
                                       043A   453                 CHAN=SYSDISK_CHAN,-     ;  Use the channel we just assigned
                                       043A   454                 FUNC=#IO$_READLBLK,-    ;  Read logical block
                                       043A   455                 P1=(R6),-               ;  Buffer space reserved on the stack
                                       043A   456                 P2=#512,-               ;  Read one block
                                       043A   457                 P3=#1                   ;  Logical block number 1
                                51 50  E9  045F   458         BLBC    R0,100$
                                       0462   459         $DASSGN_S CHAN=SYSDISK_CHAN     ; Return the channel
                                42 50  E9  046E   460         BLBC    R0,100$
                                   7E  7C  0471   461         CLRQ    -(SP)                   ; No buffer means print only
                                04 AC  DD  0473   462         PUSHL   4(AP)                   ; Push address of first part of message
                    00000000'EF    03  FB  0476   463         CALLS   #3,BOO$READPROMPT       ; Prompt and wait for return
                        55 01D8 C6  9E  047D   464         MOVAB   472(R6),R5              ; Point R5 at the volume label
                           50   0C  D0  0482   465         MOVL    #12,R0                  ; Length of volume label
                              51 6540  9E  0485   466         MOVAB   (R5)[R0],R1             ; Point R1 one past the end
                                52   71  90  0489   467 20$:    MOVB    -(R1),R2                ; R2 contains the byte
                                     05  13  048C   468         BEQL    30$                     ; Byte is null, skip it
                                52   20  91  048E   469         CMPB    #^A" ",R2               ; Is it a space
                                     05  12  0491   470         BNEQ    40$                     ; End of name found, print it
                                   F3 50  F5  0493   471 30$:    SOBGTR  R0,20$                  ; Do the whole string
                                      0E  11  0496   472         BRB     50$                     ; No name, don't bother not printing it
                                   01 A1  94  0498   473 40$:    CLRB    1(R1)                   ; Put a null after the string
                                      7E  7C  049B   474         CLRQ    -(SP)                   ; Null buffer is print only
                                      55  DD  049D   475         PUSHL   R5                      ; R5 -> front of volume label
                    00000000'EF       03  FB  049F   476         CALLS   #3,BOO$READPROMPT       ; Type the first string
                                      7E  7C  04A6   477 50$:    CLRQ    -(SP)                   ; Null buffer is print only
                                   08 AC  DD  04A8   478         PUSHL   8(AP)                   ; Push address of message
                    00000000'EF       03  FB  04AB   479         CALLS   #3,BOO$READPROMPT       ; Prompt and wait for return
                                      04  04B2   480         RET
                                          04B3   481 100$:    ERROR   <Unable to read application volume>
```

```
                                            04D8     ;
                                            04D8     ; Local kernel mode routine to get RPB information.
                                            04D8     ;
                                            04D8     GET_RPB_INFO:
                                     0000   04D8             .WORD   ^M<>
         50    00000000'EF           D0     04DA             MOVL    MMG$A_SYSPARAM+<EXE$GL_RPB-EXE$A_SYSPARAM>,R0
                                            04E1                                             ; Point to RPB (use SYS.EXE copy)
      FB25 CF    30 A0               D0     04E1             MOVL    RPB$L_BOOTR5(R0),BOOTFLAGS  ; Get boot flags
  000003E8 8F   000000B0'GF          D1     04E7             CMPL    G^SYS$GL_BOOTUCB+UCB$L_MAXBLOCK,#1000 ; Is this a small boot
                         05          14     04F2             BGTR    10$                     ; boot device (console)? Br if not.
         FB17 CF         01          90     04F4             MOVB    #1,BOOT_SMALL_CONSOLE   ; Remember that boot console is small.
              50         01          D0     04F9     10$:    MOVL    #1,R0                   ; Return success
                                     04     04FC             RET                             ;
                                            04FD     ;
                                            04FD     ; Local kernel mode routine to do various inits
                                            04FD     ;
                                            04FD     KERNEL_INIT:
                                     0000   04FD             .WORD   ^M<>
                                            04FF     ;
                                            04FF     ; Enable unrestricted locking
                                            04FF     ;
              00000000'GF            94     04FF             CLRB    G^LCK$GB_STALLREQS
                                            0505
              00000000'GF            16     0505             JSB     G^INI$BRK
              50         01          D0     050B             MOVL    #1,R0                   ; Return success
                                     04     050E             RET                             ;
                                            050F     ;
                                            050F     ; For MSCP devices, a PACKACK must be sent to the device before attempting
                                            050F     ; to read the new piece of media.  Putting in the new media has made the
                                            050F     ; device go offline.
                                            050F     ;
                                            050F     MSCP_PACKACK:
              7E         50          7D     050F             MOVQ    R0,-(SP)                ; Save registers.
              7E         52          7D     0512             MOVQ    R2,-(SP)                ; Save registers.
                         7E          D4     0515             CLRL    -(SP)                   ; Place to hold channel number.
              52         5E          D0     0517             MOVL    SP,R2                   ; Remember address of channel.
                                            051A             $ASSIGN_S       -               ; Assign a channel to system disk.
                                            051A                     DEVNAM=DSC,-            ;  Adr of str dsc for dev name str.
                                            051A                     CHAN=(R2)               ;  Adr of a word to receive chan number.
              35 50                  E9     052B             BLBC    R0,10$                  ; Br if error.
                         7E          7C     052E             CLRQ    -(SP)                   ; Set up IOSB.
              53         5E          D0     0530             MOVL    SP,R3                   ; Remember address of IOSB.
                                            0533             $QIOW_S         -               ; Send PACKACK to system device.
                                            0533                     CHAN=(R2),-             ;  Channel number.
                                            0533                     FUNC=#IO$_PACKACK,-     ;  Function code.
                                            0533                     IOSB=(R3)               ;  I/O Status block.
              10 50                  E9     0550             BLBC    R0,10$                  ; Br if error.
              0D 63                  E9     0553             BLBC    (R3),10$                ; Br if error.
                                            0556             $DASSGN_S       -               ; Deassign the channel.
                                            0556                     CHAN=(R2)               ;    Channel number.
              00 50                  E9     0560             BLBC    R0,10$                  ; Br if error.
                                            0563     10$:                                    ; If error, nothing to do...
              5E         0C          C0     0563             ADDL    #12,SP                  ; Clean off IOSB.
              52         8E          7D     0566             MOVQ    (SP)+,R2                ; Save registers.
              50         8E          7D     0569             MOVQ    (SP)+,R0                ; Save registers.
                                     05     056C             RSB                             ; Return
59 53 24 53 59 53 00000575'010E0000'        056D     DSC:    .ASCID  /SYS$SYSDEVICE:/        ; Logical name descriptor for
```

```
   3A 45 43 49 56 45 44 53  057B
                            0583                                                  ; system device for $ASSIGN.
                            0583  ;
                            0583  ; Local kernel mode routine to disable and de-allocate FIL$OPENFILE cache.
                            0583  ;
                            0583  DISABLE_FILE_CACHE:
                      000C  0583          .WORD   ^M<R2,R3>
     51    00000000'GF  7D  0585          MOVQ    G^FIL$GQ_CACHE,R1             ; R1=size, R2=address of cache
                   16   13  058C          BEQL    10$                           ; Branch if cache not present
           00000000'GF  7C  058E          CLRQ    G^FIL$GQ_CACHE                ; Disable the cache
              50   52   D0  0594          MOVL    R2,R0                         ; Copy address of cache
     53    00000004'GF  D0  0597          MOVL    G^EXE$GL_NONPAGED+4,R3        ; Address of free non-paged pool
           00000000'GF  16  059E          JSB     G^EXE$DEALLOCATE              ; Deallocate FIL$OPENFILE cache
              50   01   D0  05A4  10$:    MOVL    #1,R0                         ; Return success
                        04  05A7          RET                                   ;
                            05A8  ;
                            05A8  ; Local kernel mode routine to determine if any CI's on system.
                            05A8  ;
                      0000  05A8  FINDCI: .WORD   ^M<>
                   50   D4  05AA          CLRL    R0
           00000000'GF  D5  05AC          TSTL    G^SCS$GL_BDT                  ; Is the buffer descriptor table there?
                   14   13  05B2          BEQL    10$                           ; No, so no CI
 FB63 CF   00000000'GF  3C  05B4          MOVZWL  G^SCS$GW_PAPOLINT,PA_POLL_INT ; Yes, pick up port poller interval
 FB5E CF   00000000'GF  3C  05BD          MOVZWL  G^SCS$GW_PRCPOLINT,PRC_POLL_INT · Yes, pick up process poller interv
                   50   D6  05C6          INCL    R0
                        04  05C8  10$:    RET
                            05C9  ;+
                            05C9  ; Print an error message and halt
                            05C9  ;-
2D 45 4E 4F 4C 41 44 4E 41 54 53 25  05C9  ERRHDR: .ASCIZ  "%STANDALONE-F-"
                00 2D 46  05D5
                            05D8  ERROR_HALT:
                   7E   7C  05D8          CLRQ    -(SP)                         ; Null buffer, print only
                EC AF   9F  05DA          PUSHAB  ERRHDR                        ; Print the header
       00000000'EF  03  FB  05DD          CALLS   #3,BOO$READPROMPT             ; Now do it
              57   5E   D0  05E4          MOVL    SP,R7                         ; Copy message address
                   7E   7C  05E7          CLRQ    -(SP)                         ; Null buffer, print only
                   57   DD  05E9          PUSHL   R7                            ; Pass the error message address
       00000000'EF  03  FB  05EB          CALLS   #3,BOO$READPROMPT             ; Now do it
                            05F2          $CMKRNL_S B^10$                       ; Change mode to kernel and halt
                      0000  05FE  10$:    .WORD   0
                        00  0600          HALT
```

```
                     0601    581          .SBTTL  SIP_SETTIME - Set system time
                     0601    582 ;++
                     0601    583 ; FUNCTIONAL DESCRIPTION:
                     0601    584 ;
                     0601    585 ;       THIS ROUTINE CALLS THE LOADABLE, CPU-DEPENDENT ROUTINE, EXE$INIT_TODR,
                     0601    586 ;       TO INITIALIZE THE TIME-OF-DAY REGISTER AND SYSTEM TIME.
                     0601    587 ;
                     0601    588 ; INPUT PARAMETERS:
                     0601    589 ;
                     0601    590 ;       NONE
                     0601    591 ;
                     0601    592 ; IMPLICIT INPUTS:
                     0601    593 ;
                     0601    594 ;       TIME-OF-DAY PROCESSOR CLOCK.
                     0601    595 ;
                     0601    596 ; OUTPUT PARAMETERS:
                     0601    597 ;
                     0601    598 ;       R0,R1 - DESTROYED
                     0601    599 ;
                     0601    600 ; IMPLICIT OUTPUTS:
                     0601    601 ;
                     0601    602 ;       EXE$GQ_SYSTIME - SET TO CURRENT TIME IN 100 NANOSECOND UNITS SINCE
                     0601    603 ;                        17-NOV-1858 00:00:00.
                     0601    604 ;
                     0601    605 ;       SYSTEM PARAMETER PAGE(S) ARE WRITTEN BACK TO SYS$SYSTEM:SYS.EXE.
                     0601    606 ;
                     0601    607 ;--
                     0601    608
                     0601    609 SIP_SETTIME:                                ; SET CORRECT TIME
                0000 0601    610          .WORD   0                          ; ENTRY MASK
 00000000'EF      16 0603    611          JSB     EXE$INIT_TODR              ; CALL CPU-DEPENDENT ROUTINE
50    0000'8F    3C 0609    612          MOVZWL  #SS$_NORMAL,R0             ; INDICATE SUCCESS
                04 060E    613          RET                                ;
```

```
           060F          .SBTTL  Dummy entry points
           060F
           060F  ;
           060F  ; These entry points are need to resolve references to routines
           060F  ; not linked with the standalone version of SYSGEN.
           060F  ;
           060F
           060F  BOO$GT_SYSNAME::
           060F  BOO$GT_SYSPARNAME::
           060F  BOO$GQ_FILDESC::
           060F
           060F  LBR$OUTPUT_HELP::
           060F  LBR$INI_CONTROL::
           060F  LBR$OPEN::
           060F  LBR$GET_HELP::
           060F  BOO$SEARCH::
           060F  BOO$SETASCII::
           060F  BOO$SETVALUE::
           060F  BOO$WRTSYSPARFILE::
           060F  EXE$LOAD_CODE::
           060F  LIB$GET_INPUT::
           060F  LIB$PUT_OUTPUT::
      0000 060F          .WORD   0                       ; ENTRY MASK
   50   D4 0611          CLRL    R0
        04 0613          RET
           0614
           0614  BOO$MSGOUT::
           0614  BOO$FILOPEN::
           0614  BOO$FILOPENW::
           0614  BOO$READFILE::
           0614  BOO$WRITEFILE::
           0614  BOO$FILCLOSE::
           0614  BOO$UFOOPEN::
           0614  BOO$EXEOPEN::
   50   D4 0614          CLRL    R0
        05 0616          RSB
           0617
           0617          .END    BOO$SYSGEN
```

```
$$T1                            = 00000001
BOO$ACTIMAGE                      ********    X   02
BOO$CONFIGALL                     ********    X   02
BOO$EXEOPEN                       00000614 RG     02
BOO$FILCLOSE                      00000614 RG     02
BOO$FILOPEN                       00000614 RG     02
BOO$FILOPENW                      00000614 RG     02
BOO$GL_CMDOPT                     00000000 RG     02
BOO$GL_DOT                        00000004 RG     02
BOO$GL_SELECT                     ********    X   02
BOO$GQ_FILDESC                    0000060F RG     02
BOO$GQ_LIMITS                     ********    X   02
BOO$GQ_RETADR                     ********    X   02
BOO$GT_SYSNAME                    000006GF RG     02
BOO$GT_SYSPARNAME                 0000060F RG     02
BOO$MSGOUT                        00000614 RG     02
BOO$READFILE                      00000614 RG     02
BOO$READPROMPT                    ********    X   02
BOO$SEARCH                        0000060F RG     02
BOO$SETASCII                      0000060F RG     02
BOO$SETVALUE                      0000060F RG     02
BOO$SHODEV_ALL                    ********    X   02
BOO$SYSGEN                        0000023E RG     02
BOO$UFOOPEN                       00000614 RG     02
BOO$USEACT                        ********    X   02
BOO$WRITEFILE                     00000614 RG     02
BOO$WRTSYSPARFILE                 0000060F RG     02
BOOTFLAGS                         0000000C R      02
BOOT_SMALL_CONSOLE                00000010 R      02
CONFCRLF                          00000128 R      02
CONFLOAD                          0000012B R      02
CONFWAIT                          00000160 R      02
CR                              = 0000000D
DISABLE_FILE_CACHE                00000583 R      02
DSC                               0000056D R      02
ERRHDR                            000005C9 R      02
ERROR_HALT                        000005D8 R      02
EXE$A_SYSPARAM                    ********    X   02
EXE$DEALLOCATE                    ********    X   02
EXE$GL_NONPAGED                   ********    X   02
EXE$GL_RPB                        ********    X   02
EXE$INIT_TODR                     ********    X   02
EXE$LOAD_CODE                     0000060F RG     02
FIL$GQ_CACHE                      ********    X   02
FINDCI                            000005A8 R      02
GET_RPB_INFO                      000004D8 R      02
IMAGENAME                         000000F6 R      02
INI$BRK                           ********    X   02
INPBUF                            00000014 R      02
IO$_PACKACK                       ********    X   02
IO$_READLBLK                      ********    X   02
JPI$_PRCNAM                     = 0000031C
JPI_ITEM                          0000022E R      02
JPI_LENGTH                        0000021A R      02
JPI_NAME                          0000021E R      02
KERNEL_INIT                       000004FD R      02
LABELMSG                          0000041C R      02
LBR$GET_HELP                      0000060F RG     02
LBR$INI_CONTROL                   0000060F RG     02
LBR$OPEN                          0000060F RG     02
LBR$OUTPUT_HELP                   0000060F RG     02
LCK$GB_STALLREQS                  ********    X   02
LF                              = 0000000A
LIB$GET_INPUT                     0000060F RG     02
LIB$PUT_OUTPUT                    0000060F RG     02
MMG$A_SYSPARAM                    ********    X   02
MSCP_PACKACK                      0000050F R      02
PA_POLL_INT                       00000120 R      02
PQL$_ASTLM                      = 00000001
PQL$_BIOLM                      = 00000002
PQL$_BYTLM                      = 00000003
PQL$_CPULM                      = 00000004
PQL$_DIOLM                      = 00000005
PQL$_ENQLM                      = 0000000C
PQL$_FILLM                      = 00000006
PQL$_LISTEND                    = 00000000
PQL$_PGFLQUOTA                  = 00000007
PQL$_PRCLM                      = 00000008
PQL$_TQELM                      = 00000009
PQL$_WSDEFAULT                  = 0000000B
PQL$_WSQUOTA                    = 0000000A
PRC_POLL_INT                      00000124 R      02
REMOVEPROMPT1                     00000076 R      02
REMOVEPROMPT2                     00000095 R      02
RESUMEPROMPT1                     000000B3 R      02
RESUMEPROMPT2                     000000DA R      02
RIO_INPNAM                        00000008 RG     02
RPB$L_BOOTR5                    = 00000030
RPB$V_DEBUG                     = 00000001
SCS$GL_BDT                        ********    X   02
SCS$GW_PAPOLINT                   ********    X   02
SCS$GW_PRCPOLINT                  ********    X   02
SELECT_LIST                       0000010B R      02
SIP_SETTIME                       00000601 R      02
SS$_NORMAL                        ********    X   02
STAC_IMAGE                        00000194 R      02
STAC_OPER                         000001A9 R      02
STAC_PID                          000001D0 R      02
STAC_PRC                          000001BF R      02
STAC_PRV_MSK                      000001B7 R      02
STAC_QLIST                        000001D4 R      02
STANDCONF                         00000211 R      02
SWITCHPROMPT                      0000002D R      02
SYS$ASSIGN                        ********   GX   02
SYS$CMKRNL                        ********   GX   02
SYS$CREPRC                        ********   GX   02
SYS$DASSGN                        ********   GX   02
SYS$GETJPIW                       ********   GX   02
SYS$GL_BOOTUCB                    ********    X   02
SYS$LKWSET                        ********   GX   02
SYS$QIOW                          ********   GX   02
SYS$SETIMR                        ********   GX   02
SYS$WAITFR                        ********   GX   02
SYSDISK_CHAN                      00000018 R      02
```

```
SYSDISK_NAME                    0000001C R      02
TIMER                           00000118 R      02
UCB$L_MAXBLOCK                = 000000B0
```

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
BOO$SYSGEN                      00000617  ( 1559.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC WORD
```

Performance indicators

```
Phase                     Page faults     CPU Time        Elapsed Time
-----                     -----------     --------        ------------
Initialization                 30         00:00:00.09     00:00:00.77
Command processing            110         00:00:00.64     00:00:02.22
Pass 1                        292         00:00:08.87     00:00:16.91
Symbol table sort               0         00:00:01.15     00:00:01.90
Pass 2                        143         00:00:02.27     00:00:04.28
Symbol table output            15         00:00:00.12     00:00:00.60
Psect synopsis output           1         00:00:00.02     00:00:00.02
Cross-reference output          0         00:00:00.00     00:00:00.00
Assembler run totals          593         00:00:13.20     00:00:26.74

The working set limit was 1350 pages.
49632 bytes (97 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 784 non-local and 20 local symbols.
652 source lines were read in Pass 1, producing 22 object records in Pass 2.
29 pages of virtual memory were used to define 27 macros.
```

Macro library statistics

```
Macro library name                              Macros defined
------------------                              --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                       0
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                           4
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       19
TOTALS (all libraries)                                  23

895 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:STASYSGEN/OBJ=OBJ$:STASYSGEN MSRC$:STASYSGEN/UPDATE=(ENH$:STASYSGEN)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB
```

SYSBOOCMD
LIS

SHOWADAP
LIS

STANDCONF
LIS

SHODEV
LIS

STAPUTERR
LIS

STALOCK
LIS

STASGNMSG
LIS

STACONFIG
LIS

STASYSGEN
LIS

SYSBOOT
LIS

STARDBRIV
LIS