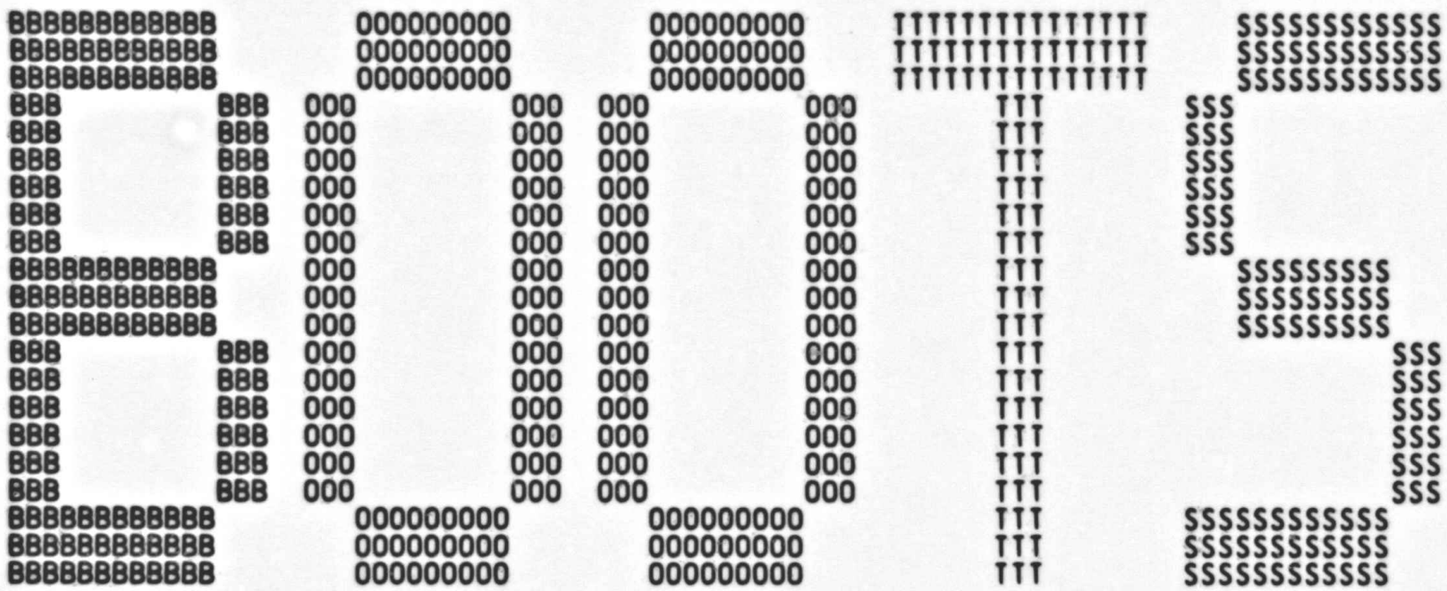
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTTTTTTTTTTTTTT SSSSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOO OOO OOO OOO TTT SSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTT SSSSSSSSSSS

FILEID**SCSLOADER

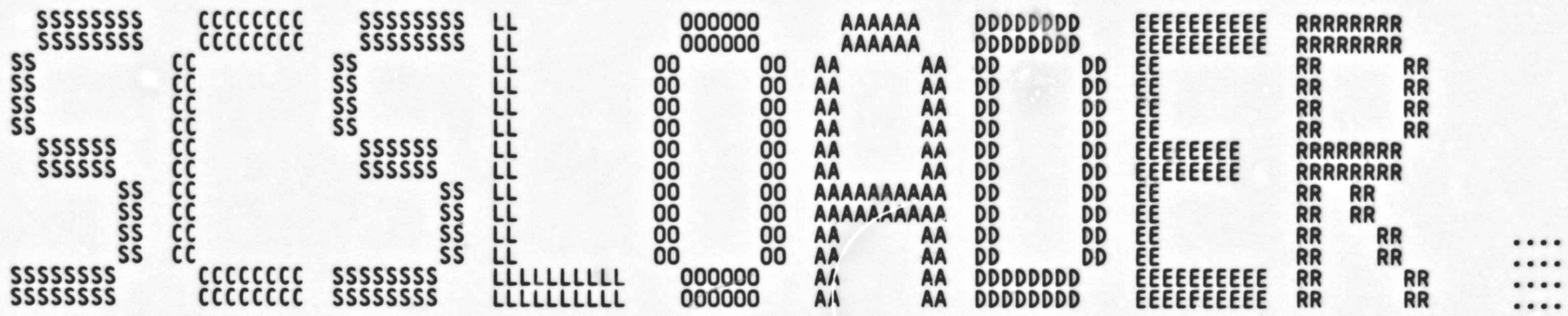

Table of contents

```
0000     1          .TITLE  SCSLOADER - Loader for SCS code
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 ;
0000    31 ; FACILITY:
0000    32 ;
0000    33 ; ABSTRACT:
0000    34 ;
0000    35 ;       This module loads SCS code for driver that require it.
0000    36 ;
0000    37 ; ENVIRONMENT:
0000    38 ;
0000    39 ;       VAX/VMS  user and kernel mode, raises IPL.
0000    40 ;
0000    41 ;--
0000    42 ;
0000    43 ; AUTHOR:       Kerbey T. Altmann, CREATION DATE: August, 1981
0000    44 ;
0000    45 ; MODIFIED BY:
0000    46 ;
0000    47 ;       V03-005 KPL0001         Peter Lieberwirth       24-Feb-1984
0000    48 ;               Fix truncation error.
0000    49 ;
0000    50 ;       V03-004 JWH0222         Jeffrey W. Horn         27-Apr-1983
0000    51 ;               Replace LINK_VEC with EXE$LINK_VEC.
0000    52 ;
0000    53 ;       V03-003 MSH0001         Maryann Hinden          31-Jan-1983
0000    54 ;               Change IOGEN$GL_DPT to ACF$GL_DPT.
0000    55 ;
0000    56 ;       V03-002 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    57 ;               Added $IPLDEF, $SSDEF, and $PRDEF.
```

```
                     ;
                     ;**
                     
                             .SBTTL  DECLARATIONS
                     ;
                     ; INCLUDE FILES:
                     ;
                     
                             $DPTDEF
                             $IPLDEF
                             $PRDEF
                             $SSDEF
                             $SLVDEF
                             $SYSGMSGDEF
                     
                     ;++
                     ;
                     ; FUNCTIONAL DESCRIPTION:
                     ;
                     ;       SCSLOADER is called to check for presence of SCS code, if
                     ;       not present, it loads it into non-paged pool.
                     ;
                     ; CALLING SEQUENCE:
                     ;
                     ;       CALLS   #0,BOO$SCSLOADER
                     ;
                     ; INPUT PARAMETERS:
                     ;       NONE
                     ;
                     ; IMPLICIT INPUTS:
                     ;
                     ;       SCS$AL_LOAVEC - Address of entry point into SCS code.
                     ;
                     ; OUTPUT PARAMETERS:
                     ;       NONE
                     ;
                     ; IMPLICIT OUTPUTS:
                     ;
                     ;       Non-paged pool is allocated.
                     ;
                     ; COMPLETION CODES:
                     ;
                     ;       SS$_NORMAL - All errors are signaled from with this module.
                     ;
                     ; SIDE EFFECTS:
                     ;       NONE
                     ;
                     ;
                     ;--
                     
                     ;
                     ; OWN STORAGE
                     ;
                     
                 00000000            .PSECT  NONPAGED_DATA,rd,wrt,noexe,quad
                     
41 4F 4C 53 43 53 00'                SCS_NAME:       .ASCIC  /SCSLOA/        ; Input file spec
                  06
```

```
00000000  0007   114 SCS_CHAN:      .LONG   0               ; Channel number
00000013  000B   115 SCS_ADDR:      .BLKB   8               ; Return addresses
          0013   116
```

```
                                0013   118 ;
                                0013   119 ; BEGIN
                                0013   120 ;
                                0013   121
                          00000000   122          .PSECT  NONPAGED_CODE,rd,nowrt,exe,long
                                0000   123
                           0804 0000   124          .ENTRY  BOO$SCSLOADER,^M<R2,R11>
                                0002   125
                                0002   126          $CMKRNL_S W^SCS_CHECK          ; See if SCS needs to be loaded
                      47 50  E9 000F   127          BLBC    R0,100$                ; Branch if not necessary
                                0012   128 ;
                                0012   129 ; Set up inputs for reading file in
                                0012   130 ; (Can't use RMS directly because of STASYSGEN)
                                0012   131 ;
              50  00000000'EF  9E 0012   132          MOVAB   SCS_NAME,R0            ; Address of counted string
              53  00000007'EF  9E 0019   133          MOVAB   SCS_CHAN,R3            ; Address of channel
              54  0000000B'EF  9E 0020   134          MOVAB   SCS_ADDR,R4            ; Address of quadword region descriptor
                                0027   135
                  00000000'EF  16 0027   136          JSB     IOGEN$READDRIV         ; Read in SCS code
                        18 50  E9 002D   137          BLBC    R0,99$                 ; Branch if error!
                                0030   138
           5B  64  00000200 8F  C1 0030   139          ADDL3   #512,(R4),R11          ; Set address to load, skip image header
                                0038   140
                                0038   141 ;
                                0038   142 ; Now execute the remaining code in KERNEL mode.
                                0038   143 ;
                                0038   144
                                0038   145 90$:     $CMKRNL_S W^SCSPOOL            ; Execute kernel routine
                      11 50  E8 0045   146          BLBS    R0,100$                ; Branch if OK
                                0048   147
                         50  DD 0048   148 99$:     PUSHL   R0                     ; Set error
                         7E  D4 004A   149          CLRL    -(SP)                  ; zero longword
                   007C9038  8F DD 004C   150          PUSHL   #SYSG$_NOSCS           ; No SCS code
          00000000'GF      03 FB 0052   151          CALLS   #3,G^LIB$SIGNAL        ; Error
                                0059   152
                      50  01  D0 0059   153 100$:    MOVL    #SS$_NORMAL,R0         ; Always return success
                                04 005C   154          RET                            ; Return
                                005D   155
                                005D   156
```

```
                        005D   158
                 0004   005D   159 SCS_CHECK:       .WORD ^M<R2>
                        005F   160
                        005F   161 ;
                        005F   162 ; Check driver to see if SCS is required
                        005F   163 ;
                        005F   164
                   50   D4  005F   165           CLRL    R0                      ; Assume "Don't load"
52 00000000'EF     D0  0061   166           MOVL    ACF$GL_DPT,R2           ; Set address of DPT
                   15   13  0068   167           BEQL    10$                     ; Branch if none
                   03   E1  006A   168           BBC     #DPT$V_SCS,-
                10 0D A2       006C   169                   DPT$B_FLAGS(R2),10$     ; Branch if SCS not specified
                        006F   170 ;
                        006F   171 ;  Driver says to load SCS, now check to see if it's loaded
                        006F   172 ;
00000000'8F        D1  006F   173           CMPL    #EXE$LOAD_ERROR,-
00000002'GF            0075   174                   G^SCS$AL_LOAVEC+2       ; See if vector has changed from
                        007A   175                                           ;  initial value
                   03   12  007A   176           BNEQ    10$                     ; Branch if loaded
                        007C   177
                50 01   D0  007C   178           MOVL    #SS$_NORMAL,R0          ; Set "Load SCS"
                   04   007F   179 10$:    RET                             ; Return
                        0080   180
```

```
                        0080   182 SCSPOOL:
                067C    0080   183         .WORD   ^M<R2,R3,R4,R5,R6,R9,R10>
                        0082   184
                        0082   185 ;
                        0082   186 ; Grab enough non-paged pool for the code
                        0082   187 ;
      51    6B    D0    0082   188         MOVL    (R11), R1                       ; Size needed for code
00000000'GF     16    0085   189         JSB     G^EXE$ALONONPAGED               ; Get the space
      06    50    E8    008B   190         BLBS    R0, 10$                         ; Okay
50    0124  8F    3C    008E   191         MOVZWL  #SS$_INSFMEM, R0
                  04    0093   192         RET                                     ; Error
                        0094   193 10$:
      59    51    7D    0094   194         MOVQ    R1, R9                          ; Save size and address
6A    6B    59    28    0097   195         MOVC3   R9, (R11), (R10)                ; Move the code to non-paged pool
                        009B   196 ;
                        009B   197 ; Fix up the vectors
                        009B   198 ;
                        009B   199         SETIPL  #IPL$_SYNCH                     ; Synchronize
                        009E   200 ;
                        009E   201 ;*****
                        009E   202 ; THE FOLLOWING IS STOLEN FROM INIT.MAR
                        009E   203 ;
                        009E   204 ; The loadable SCS code image (SCSLOA.EXE) is now allocated non-paged
                        009E   205 ; pool space and read into it by SYSBOOT.  The address of this code
                        009E   206 ; is passed in BOO$GL_SCSLOA.  Link the resident system vectors at
                        009E   207 ; SCS$AL_LOAVEC.  The loadable file has self-describing vector and
                        009E   208 ; offset information within it.
                        009E   209 ;
                        009E   210 SCS_LOADCODE:
      52    5A    D0    009E   211         MOVL    R10,R2                          ; Address of SCSLOA image in pool
53    00000000'GF     DE    00A1   212         MOVAL   G^SCS$AL_LOAVEC,R3              ; Address of resident vectors.
      54    52    D0    00A8   213         MOVL    R2,R4                           ; Save
         FF52'    30    00AB   214         BSBW    EXE$LINK_VEC                    ; link vectors
      11    50    E9    00AE   215         BLBC    R0,30$                          ; get out on error
50    04    A4    D0    00B1   216         MOVL    SLV$L_INITRTN(R4),R0            ; Possible initialization routine
                  08    13    00B5   217         BEQL    25$                             ; None, leave
      50    54    C0    00B7   218         ADDL    R4,R0                           ; Absolute address of init
                  0A    10    00BA   219         BSBB    40$                             ; Call it
                        00BC   220 ;
                        00BC   221 ; After init and releasing itself to pool, return here
                        00BC   222 ;
      03    50    E9    00BC   223         BLBC    R0, 30$
      50    01    D0    00BF   224 25$:     MOVL    #1, R0                          ; Set success
                        00C2   225 30$:     SETIPL  #0
                  04    00C5   226         RET
                        00C6   227
                        00C6   228 ;
                        00C6   229 ; Cute little routine to do an REI to code.
                        00C6   230 ;
                  7E    DC    00C6   231 40$:     MOVPSL  -(SP)                           ; Set up PSL
                  50    DD    00C8   232         PUSHL   R0                              ; Address to start at
                  02    00CA   233         REI                                     ; Flush TB and jump to code
                        00CB   234
                        00CB   235         .END
```

```
ACF$GL_DPT                      ********    X   03
BOO$SCSLOADER                   00000000 RG     03
DPT$B_FLAGS                   = 0000000D
DPT$V_SCS                     = 00000003
EXE$ALONONPAGED                 ********    X   03
EXE$LINK_VEC                    ********    X   03
EXE$LOAD_ERROR                  ********    X   03
IOGEN$READDRIV                  ********    X   03
IPL$_SYNCH                    = 00000008
LIB$SIGNAL                      ********    X   03
PR$_IPL                       = 00000012
SCS$AL_LOAVEC                   ********    X   03
SCSPOOL                         00000080 R      03
SCS_ADDR                        0000000B R      02
SCS_CHAN                        00000007 R      02
SCS_CHECK                       0000005D R      03
SCS_LOADCODE                    0000009E R      03
SCS_NAME                        00000000 R      02
SLV$L_INITRTN                 = 00000004
SS$_INSFMEM                   = 00000124
SS$_NORMAL                    = 00000001
SYS$CMKRNL                      ********   GX   03
SYSG$_NOSCS                   = 007C9038
```

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
NONPAGED_DATA	00000013 (19.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
NONPAGED_CODE	000000CB (203.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.81
Command processing	116	00:00:00.69	00:00:03.21
Pass 1	250	00:00:06.34	00:00:11.27
Symbol table sort	0	00:00:00.92	00:00:01.74
Pass 2	55	00:00:01.19	00:00:02.03
Symbol table output	4	00:00:00.04	00:00:00.04
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	459	00:00:09.29	00:00:19.12

```
The working set limit was 1200 pages.
33071 bytes (65 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 630 non-local and 8 local symbols.
235 source lines were read in Pass 1, producing 18 object records in Pass 2.
16 pages of virtual memory were used to define 15 macros.
```

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

Macro library name                 Macros defined
------------------                 --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1             0
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                 5
_$255$DUA28:[SYSLIB]STARLET.MLB;2              7
TOTALS (all libraries)                        12

722 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

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

READLBN
LIS

RTFILREAD
LIS

SHARE
LIS

QUSS
LIS

RWBTDRIVR
LIS

MBBTDRIVR
LIS

SCSLOADER
LIS

RMSCONIO
LIS

READDRIV
LIS

PABTDRIVR
LIS

PUBTDRIVR
LIS

PUTERROR
LIS

READPRMPT
LIS