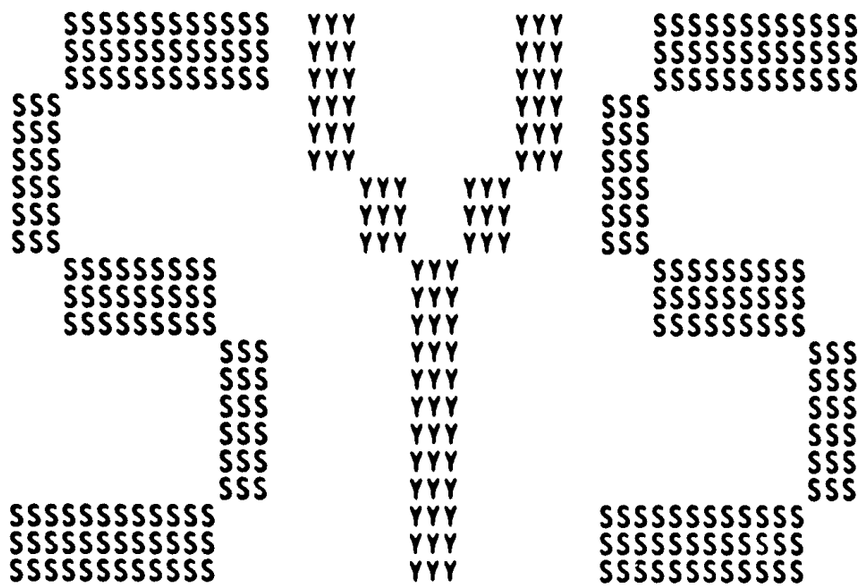

```
  SSSSSSSS  YY      YY    SSSSSSSS  DDDDDDDD      CCCCCCCC  LL          EEEEEEEEEE  XX      XX  HH      HH
  SSSSSSSS  YY      YY    SSSSSSSS  DDDDDDDD      CCCCCCCC  LL          EEEEEEEEEE  XX      XX  HH      HH
SS          YY      YY  SS          DD      DD  CC          LL          EE          XX      XX  HH      HH
SS          YY      YY  SS          DD      DD  CC          LL          EE          XX      XX  HH      HH
SS            YY  YY    SS          DD      DD  CC          LL          EE            XX  XX    HH      HH
SS            YY  YY    SS          DD      DD  CC          LL          EE            XX  XX    HH      HH
  SSSSSS        YY        SSSSSS    DD      DD  CC          LL          EEEEEEEE        XX      HHHHHHHHHH
  SSSSSS        YY        SSSSSS    DD      DD  CC          LL          EEEEEEEE        XX      HHHHHHHHHH
        SS      YY              SS  DD      DD  CC          LL          EE            XX  XX    HH      HH
        SS      YY              SS  DD      DD  CC          LL          EE            XX  XX    HH      HH
        SS      YY              SS  DD      DD  CC          LL          EE          XX      XX  HH      HH      ....
        SS      YY              SS  DD      DD  CC          LL          EE          XX      XX  HH      HH      ....
SSSSSSSS        YY      SSSSSSSS    DDDDDDDD      CCCCCCCC  LLLLLLLLLL  EEEEEEEEEE  XX      XX  HH      HH      ....
SSSSSSSS        YY      SSSSSSSS    DDDDDDDD      CCCCCCCC  LLLLLLLLLL  EEEEEEEEEE  XX      XX  HH      HH      ....

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  SYSDCLEXH - SYSTEM SERVICES CANCEL AND DECLARE EXIT HANDLER
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  ; FACILITY: EXECUTIVE, SYSTEM SERVICES
0000    30  ;
0000    31  ;       SYSTEM SERVICES CANCEL AND DECLARE EXIT HANDLER
0000    32  ;
0000    33  ; ENVIRONMENT:
0000    34  ;       MODE=KERNEL, PAGED CODE
0000    35  ;
0000    36  ; AUTHOR:       D. N. CUTLER     CREATION DATE: 10-AUG-76
0000    37  ;
0000    38  ; MODIFIED BY:
0000    39  ;
0000    40  ;       V03-001 LJK0287         Lawrence J. Kenah       26-Jun-1984
0000    41  ;               Count the number of exit handlers declared for each mode for
0000    42  ;               use by $EXIT and $CANEXH.
0000    43  ;--
```

```
                     0000     45          .SUBTITLE       DECLARATIONS
                     0000     46 ;
                     0000     47 ; MACRO LIBRARY CALLS
                     0000     48 ;
                     0000     49
                     0000     50          $PSLDEF                         ;DEFINE PROCESSOR STATUS FIELDS
                     0000     51          $SSDEF                          ;DEFINE SYSTEM STATUS VALUES
                     0000     52
                     0000     53 ;
                     0000     54 ; LOCAL SYMBOLS
                     0000     55 ;
                     0000     56 ; ARGUMENT LIST OFFSET DEFINITIONS
                     0000     57 ;
                     0000     58
00000004             0000     59 DESBLK=4                                ;EXIT CONTROL BLOCK ADDRESS
```

```
                     0000     61          .SBTTL  CANCEL EXIT HANDLER
                     0000     62 ;+
                     0000     63 ; EXE$CANEXH - CANCEL EXIT HANDLER
                     0000     64 ;
                     0000     65 ; THIS SERVICE PROVIDES THE CAPABILITY TO DELETE AN EXIT CONTROL BLOCK OR
                     0000     66 ; ALL EXIT CONTROL BLOCKS FROM THE EXIT HANDLER LIST FOR THE CALLING ACCESS
                     0000     67 ; MODE.
                     0000     68 ;
                     0000     69 ; INPUTS:
                     0000     70 ;
                     0000     71 ;       DESBLK(AP) = ADDRESS OF EXIT CONTROL BLOCK (ZERO IMPLIES ALL).
                     0000     72 ;
                     0000     73 ;       R4 = CURRENT PROCESS PCB ADDRESS.
                     0000     74 ;
                     0000     75 ; OUTPUTS:
                     0000     76 ;
                     0000     77 ;       R0 LOW BIT CLEAR INDICATES FAILURE TO CANCEL EXIT HANDLER.
                     0000     78 ;
                     0000     79 ;               R0 = SS$_ACCVIO - FIRST LONGWORD OF EXIT CONTROL BLOCK
                     0000     80 ;                    CANNOT BE READ BY THE CALLING ACCESS MODE, FIRST
                     0000     81 ;                    LONGWORD OF A EXIT CONTROL BLOCK IN THE LIST CANNOT
                     0000     82 ;                    BE READ BY THE CALLING ACCESS MODE, OR THE FIRST
                     0000     83 ;                    LONGWORD OF THE PREVIOUS EXIT CONTROL BLOCK CANNOT
                     0000     84 ;                    BE WRITTEN.
                     0000     85 ;
                     0000     86 ;               R0 = SS$_NOHANDLER - SPECIFIED HANDLER CANNOT BE FOUND IN
                     0000     87 ;                    EXIT HANLDER CONTROL BLOCK LIST.
                     0000     88 ;
                     0000     89 ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                     0000     90 ;
                     0000     91 ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                     0000     92 ;-
                     0000     93
                     0000     94          .ENABL  LSB
                00000000      95          .PSECT  Y$EXEPAGED
               003C  0000     96          .ENTRY  EXE$CANEXH,^M<R2,R3,R4,R5>
          62   10   0002     97          BSBB    GETPARM                 ;GET EXIT CONTROL BLOCK PARAMETERS
          3A   13   0004     98          BEQL    25$                     ;IF EQL CANCEL ALL EXIT HANDLERS
     52   50   D0   0006     99          MOVL    R0,R2                   ;COPY ADDRESS OF LISTHEAD
     55   64   D0   0009    100          MOVL    (R4),R5                 ;GET COUNT FOR LOOP CONTROL
     53   52   D0   000C    101 10$:     MOVL    R2,R3                   ;SAVE ADDRESS OF PREVIOUS ENTRY
                    000F    102          IFNORD  #4,(R3),50$             ;CAN NEXT EXIT CONTROL BLOCK ADDRESS BE READ
     52   63   D0   0015    103          MOVL    (R3),R2                 ;GET ADDRESS OF NEXT EXIT CONTROL BLOCK
          46   13   0018    104          BEQL    60$                     ;IF EQL END OF LIST
     52   51   D1   001A    105          CMPL    R1,R2                   ;EXIT CONTROL BLOCK ADDRESS MATCH?
          09   13   001D    106          BEQL    15$                     ;IF EQUAL, ADDRESSES MATCH
        EA 55   F4   001F    107          SOBGEQ  R5,10$                  ;GO BACK FOR NEXT BLOCK IN LIST
50   22B4 8F   3C   0022    108          MOVZWL  #SS$_INVEXHLIST,R0      ;REPORT INCONSISTENT LIST
               04   0027    109          RET                             ;RETURN FAILURE STATUS TO CALLER
                    0028    110
                    0028    111 15$:     IFNORD  #4,(R2),50$             ;CAN NEXT EXIT CONTROL BLOCK ADDRESS BE READ
     53   50   D1   002E    112          CMPL    R0,R3                   ;PREVIOUS EXIT CONTROL BLOCK LISTHEAD?
          06   13   0031    113          BEQL    20$                     ;IF EQL YES
                    0033    114          IFNOWRT #4,(R3),50$             ;CAN PREVIOUS EXIT CONTROL BLOCK BE WRITTEN?
     63   62   D0   0039    115 20$:     MOVL    (R2),(R3)               ;REMOVE EXIT CONTROL BLOCK FROM LIST
          64   D7   003C    116          DECL    (R4)                    ;INDICATE THAT HANDLER WAS REMOVED
          18   11   003E    117          BRB     40$                     ;EXIT WITH SUCCESS
```

```
                        0040   118
64   D4   0040   119 25$:    CLRL    (R4)                    ;SET COUNT TO REFLECT EMPTY LIST
11   11   0042   120         BRB     30$                     ;EXIT BY CLEARING LISTHEAD
```

```
                     0044   122          .SBTTL  DECLARE EXIT HANDLER
                     0044   123 ;+
                     0044   124 ; EXE$DCLEXH - DECLARE EXIT HANDLER
                     0044   125 ;
                     0044   126 ; THIS SERVICE PROVIDES THE CAPABILITY TO INSERT AN EXIT CONTROL BLOCK AT
                     0044   127 ; THE FRONT OF THE EXIT HANDLER LIST FOR THE CALLING ACCESS MODE.
                     0044   128 ;
                     0044   129 ; INPUTS:
                     0044   130 ;
                     0044   131 ;       DESBLK(AP) = ADDRESS OF EXIT CONTROL BLOCK.
                     0044   132 ;
                     0044   133 ;       R4 = CURRENT PROCESS PCB ADDRESS.
                     0044   134 ;
                     0044   135 ; OUTPUTS:
                     0044   136 ;
                     0044   137 ;       R0 LOW BIT CLEAR INDICATES FAILURE TO DECLARE EXIT HANDLER.
                     0044   138 ;
                     0044   139 ;               R0 = SS$_ACCVIO - FIRST LONGWORD OF EXIT CONTROL BLOCK
                     0044   140 ;                    CANNOT BE WRITTEN BY CALLING ACCESS MODE.
                     0044   141 ;
                     0044   142 ;               R0 = SS$_NOHANDLER - NO EXIT CONTROL BLOCK ADDRESS SPECIFIED.
                     0044   143 ;
                     0044   144 ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                     0044   145 ;
                     0044   146 ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                     0044   147 ;-
                     0044   148
                001C 0044   149          .ENTRY  EXE$DCLEXH,^M<R2,R3,R4>
             1E   10 0046   150          BSBB    GETPARM                 ;GET EXIT CONTROL BLOCK PARAMETERS
             16   13 0048   151          BEQL    60$                     ;IF EQL NO EXIT CONTROL BLOCK SPECIFIED
                     004A   152          IFNOWRT #4,(R1),50$             ;CAN FIRST LONGWORD BE WRITTEN?
        61   60   D0 0050   153          MOVL    (R0),(R1)               ;LINK OLD FIRST TO NEW FIRST
             64   D6 0053   154          INCL    (R4)                    ;COUNT ANOTHER DECLARED HANDLER
        60   51   D0 0055   155 30$:     MOVL    R1,(R0)                 ;SET ADDRESS OF NEW FIRST ENTRY
        50   01   3C 0058   156 40$:     MOVZWL  #SS$_NORMAL,R0          ;SET NORMAL COMPLETION STATUS
                  04 005B   157          RET                             ;
        50   0C   3C 005C   158 50$:     MOVZWL  #SS$_ACCVIO,R0          ;SET ACCESS VIOLATION
                  04 005F   159          RET                             ;
  50    08F8 8F   3C 0060   160 60$:     MOVZWL  #SS$_NOHANDLER,R0       ;SET NO HANLDER STATUS
                  04 0065   161          RET                             ;
                     0066   162          .DSABL  LSB
```

```
                     0066   164          .SBTTL  GET EXIT CONTROL BLOCK PARAMETERS
                     0066   165 ;+
                     0066   166 ; This routine obtains the parameters that characterize this call to
                     0066   167 ; SYS$DCLEXH or SYS$CANEXH.
                     0066   168 ;
                     0066   169 ; Input Parameter:
                     0066   170 ;
                     0066   171 ;       DESBLK(AP) - Address of termination handler control block
                     0066   172 ;
                     0066   173 ; Implicit Input:
                     0066   174 ;
                     0066   175 ;       PSL<PRVMOD> - Access mode of caller of system service
                     0066   176 ;
                     0066   177 ;       CTL$GL_THEXEC - Address of three-longword array of listheads of
                     0066   178 ;               termination handler control blocks, one each for exec,
                     0066   179 ;               supervisor, and user mode.
                     0066   180 ;
                     0066   181 ;       CTL$GL_THCOUNT - Address of three-longword array of counts of
                     0066   182 ;               termination handlers for each of the three access modes.
                     0066   183 ;
                     0066   184 ; Output Parameters:
                     0066   185 ;
                     0066   186 ;       R0 - Address of listhead for the correct access mode
                     0066   187 ;       R1 - Address of termination handler control block
                     0066   188 ;       R4 - Address of count cell for the correct access mode.
                     0066   189 ;
                     0066   190 ; Implicit Ouptut:
                     0066   191 ;
                     0066   192 ;       PSL<Z> - Indicates whether R1 contains a nonzero termination
                     0066   193 ;               handler control block address.
                     0066   194 ;-
                     0066   195
                     0066   196 GETPARM:
                50   DC  0066   197          MOVPSL  R0                              ;READ CURRENT PSL
 50   50   02   16   EF  0068   198          EXTZV   #PSL$V_PRVMOD,#PSL$S_PRVMOD,R0,R0 ;EXTRACT PREVIOUS MODE FIELD
                50   D7  006D   199          DECL    R0                              ;SUBTRACT OUT EXECUTIVE MODE BIAS
                15   19  006F   200          BLSS    10$                             ;IF LSS CALL FROM KERNEL MODE
54   00000000'9F40   DE  0071   201          MOVAL   @#CTL$GL_THCOUNT[R0],R4         ;GET ADDRESS OF COUNT CELL
50   00000000'9F40   DE  0079   202          MOVAL   @#CTL$GL_THEXEC[R0],R0          ;GET ADDRESS OF EXIT HANDLER LISTHEAD
       51   04 AC   D0  0081   203          MOVL    DESBLK(AP),R1                   ;GET ADDRESS OF EXIT CONTROL BLOCK
                05   05  0085   204          RSB                                     ;
                     0086   205
                     0086   206 10$:     BUG_CHECK IVSSRVRQST                        ;INVALID SYSTEM SERVICE REQUEST
        50   0174 8F   3C  008A   207          MOVZWL  #SS$_IVSSRQ,R0                  ;SET INVALID SYSTEM SERVICE REQUEST
                04   04  008F   208          RET                                     ;
                     0090   209
                     0090   210          .END
```

```
BUG$_IVSSRVRQST                 ********   X   02
CTL$GL_THCOUNT                  ********   X   02
CTL$GL_THEXEC                   ********   X   02
DESBLK                        = 00000004
EXE$CANEXH                      00000000 RG    02
EXE$DCLEXH                      00000044 RG    02
GETPARM                         00000066 R     02
PSL$S_PRVMOD                  = 00000002
PSL$V_PRVMOD                  = 00000016
SS$_ACCVIO                    = 0000000C
SS$_INVEXHLIST                = 000022B4
SS$_IVSSRQ                    = 00000174
SS$_NOHANDLER                 = 000008F8
SS$_NORMAL                    = 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
Y$EXEPAGED	00000090 (144.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.08	00:00:01.27
Command processing	129	00:00:00.56	00:00:05.54
Pass 1	207	00:00:04.28	00:00:15.04
Symbol table sort	0	00:00:00.69	00:00:01.34
Pass 2	55	00:00:00.90	00:00:03.87
Symbol table output	3	00:00:00.03	00:00:00.03
Psect synopsis output	1	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	432	00:00:06.58	00:00:27.19

```
The working set limit was 1200 pages.
23713 bytes (47 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 453 non-local and 9 local symbols.
210 source lines were read in Pass 1, producing 19 object records in Pass 2.
12 pages of virtual memory were used to define 11 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	3
_255DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	8

525 GETS were required to define 8 macros.

There were no errors, warnings or information messages.

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

SYSCRMPSC
LIS

SYSDCLEXH
LIS

SYSDEVALC
LIS

SYSCVTIIM
LIS

SYSDGBLSC
LIS

SYSENQDEQ
LIS

SYSDCLCMH
LIS

SYSDERLMB
LIS

SYSDASSGN
LIS

SYSDELPRC
LIS