

(1) 43
(1) 58

DECLARATIONS
EXE\$SETSTK - SET STACK LIMITS

```

0000 1      .TITLE  SYSSETSTK - SET STACK LIMITS
0000 2      .IDENT  'V04-000'
0000 3
0000 4      :*****
0000 5      :*
0000 6      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      :*  ALL RIGHTS RESERVED.
0000 9      :*
0000 10     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     :*  TRANSFERRED.
0000 16     :*
0000 17     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     :*  CORPORATION.
0000 20     :*
0000 21     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     :*
0000 24     :*
0000 25     :*****
0000 26
0000 27     :++
0000 28     : FACILITY: EXECUTIVE, SYSTEM SERVICES
0000 29
0000 30     : ABSTRACT:
0000 31     :   This module implements the service to set the stack limits for
0000 32     :   any mode.  Actually, USER mode is a nop since it requires no
0000 33     :   system service to change and is virtually infinite. (Groan!!!)
0000 34
0000 35     : ENVIRONMENT: VAX/VMS
0000 36
0000 37     : AUTHOR: Kerbey T. Altmann,      CREATION DATE: 05-Jun-1981
0000 38
0000 39     : MODIFIED BY:
0000 40
0000 41     :--

```

```
0000 43 .SBTTL DECLARATIONS
0000 44 :
0000 45 : INCLUDE FILES:
0000 46 :
0000 47 $PSLDEF
0000 48 $SSDEF
0000 49
0000 50 :
0000 51 : LOCAL SYMBOLS:
0000 52 :
00000004 0000 53 INADR= 4 ; Address of input array
00000008 0000 54 RETADR= 8 ; Address of output array
0000000C 0000 55 ACMODE= 12 ; The stack mode to set
0000 56
```

```

0000 58      .SBTTL  EXE$SETSTK - SET STACK LIMITS
0000 59
0000 60      ;++
0000 61      ; FUNCTIONAL DESCRIPTION:
0000 62
0000 63      ; The SETSTK system service allows one to change the values in
0000 64      ; the stack limit and base arrays held in per-process P1 space.
0000 65      ; These arrays are used by EXCEPTION and BUGCHECK to validate
0000 66      ; stack address during error processing. The first longword
0000 67      ; is the setting for the low memory (limit or top) address of
0000 68      ; the stack - the check for validity would be less than or
0000 69      ; equal to this value. The second longword is the setting for
0000 70      ; the high memory (base) address of the stack - the check for
0000 71      ; validity would be greater than this value.
0000 72
0000 73      INPUTS:
0000 74
0000 75      INADR(AP)      Pointer to a 2 longword array specifying the
0000 76      new contents of the stack arrays.
0000 77      RETADR(AP)     Pointer to a 2 longword array to receive the
0000 78      contents of the former settings.
0000 79      ACMODE(AP)     A longword containing the value of the access
0000 80      mode for whose stack the settings are to be
0000 81      changed (maximized with the current mode).
0000 82
0000 83      OUTPUTS:
0000 84
0000 85      R0              Status code
0000 86
0000 87      ;--
0000 88
0000 89      .PSECT  YEXEPAGED
0000 90      .ENTRY  EXE$SETSTK, ^M<R2,R3,R4>
0002 91
50  0C AC  02  00  EF 0002 92      EXTZV  #0,#2,ACMODE(AP),R0      ; Get specified access mode
      FFF5' 30 0008 93      BSBW   EXE$MAXACMODE      ; Maximize access mode
51  50  D0 000B 94      MOVL   R0, R1              ; Hold for later
50  03  D1 000E 95      CMPL  #P$LS$C_USER, R0    ; Is it USER mode?
      5B  13 0011 96      BEQL  20$                 ; Yes, just make it a no-op
50  0C  3C 0013 97      MOVZWL #SS$ ACCVIO, R0     ; Assume access violation
52  04 AC  D0 0016 98      MOVL  INADR(AP), R2       ; Pick up pointer to input range
      35  13 001A 99      BEQL  30$                 ; No range specified is ACCVIO
      001C 100      IFNORD #8, (R2), 30$      ; Check accessibility
53  08 AC  D0 0022 101     MOVL  RETADR(AP), R3      ; Pick up pointer to output range
      16  13 0026 102     BEQL  10$                 ; None specified
      0028 103     IFNOWRT #8, (R3), 30$    ; Check accessibility
83  00000000'9F41  D0 002E 104     MOVL  @#CTLSAL_STACKLIM[R1],- ; Return the Lo-address (limit)
      0036 105     (R3)+
63  00000000'9F41  D0 0036 106     MOVL  @#CTLSAL_STACK[R1], (R3); Return the Hi-address (base)
00000000'9F41  82  D0 003E 107 10$: MOVL  (R2)+, -
      0046 108     @#CTLSAL_STACKLIM[R1] ; Set the Lo-address (limit)
00000000'9F41  62  D0 0046 109     MOVL  (R2), @#CTLSAL_STACK[R1]; Set the Hi-address (base)
      50  01  3C 004E 110 20$: MOVZWL #SS$_NORMAL, R0 ; Success
      04  0051 111 30$: RET
      0052 112
      0052 113      .END

```

SYSSETSTK
Symbol table

- SET STACK LIMITS

M 10

16-SEP-1984 02:34:44 VAX/VMS Macro V04-00
5-SEP-1984 03:57:33 [SYS.SRC]SYSSETSTK.MAR;1

Page 4
(1)

SYS
V04

ACMODE	=	0000000C		
CTLSAL_STACK		*****	X	02
CTLSAL_STACKLIM		*****	X	02
EXESMAXACMODE		*****	X	02
EXESSETSTK		00000000	RG	02
INADR	=	00000004		
PSL\$C_USER	=	00000003		
RETADR	=	00000008		
SS\$_ACCVIO	=	0000000C		
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
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YEXEPAGED	00000052 (82.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.10	00:00:00.27
Command processing	107	00:00:00.56	00:00:01.28
Pass 1	207	00:00:04.05	00:00:05.27
Symbol table sort	0	00:00:00.66	00:00:00.73
Pass 2	38	00:00:00.75	00:00:00.91
Symbol table output	2	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	388	00:00:06.17	00:00:08.53

The working set limit was 1200 pages.
 22191 bytes (44 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 3 local symbols.
 113 source lines were read in Pass 1, producing 16 object records in Pass 2.
 11 pages of virtual memory were used to define 10 macros.

! Macro library statistics !

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

520 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

SYSSETSTK
VAX-11 Macro Run Statistics

- SET STACK LIMITS

N 10

16-SEP-1984 02:34:44 VAX/VMS Macro V04-00
5-SEP-1984 03:57:33 [SYS.SRC]SYSSETSTK.MAR;1

Page 5
(1)

SYS
V04

MACRO/LIS=LIS\$:SYSSETSTK/OBJ=OBJ\$:SYSSETSTK MSRC\$:SYSSETSTK/UPDATE=(ENH\$:SYSSETSTK)+EXECMLS/LIB

This image displays a grid of 120 small terminal window screenshots arranged in 12 rows and 10 columns. Each window shows a different system utility or diagnostic tool. Many of the windows include the following labels:

- SYSSETPRG LIS
- SYSSETSSP LIS
- SYSSNDJBC LIS
- SYSSETIME LIS
- SYSSETPPM LIS
- SYSUNWIND LIS
- SYSSETPRV LIS
- SYSSETSTK LIS
- SYSSETMOD LIS
- SYSSETEXU LIS
- SYSSETPRI LIS
- SYSSCHEUT LIS
- SYSUNWIND LIS
- SYSUNWIND LIS

The screenshots themselves contain various data, including lists of files, system parameters, and status information, though the text is too small to read clearly.