

(1)	42
(1)	57
(1)	84

HISTORY	: DETAILED
DECLARATIONS	
EXESSETPRI - SET PRIORITY SYSTEM SERVICE ROUTINE	

```

0000 1 .TITLE SYSSETPRI SET PRIORITY SYSTEM SERVICE
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 ABSTRACT:
0000 32 SYSSETPRI IMPLEMENTS THE SET PRIORITY SYSTEM SERVICE WHICH
0000 33 ENABLES A PROCESS TO ALTER THE PRIORITY OF ITSELF OR ANOTHER
0000 34 PROCESS.
0000 35
0000 36 ENVIRONMENT:
0000 37 MODE=KERNEL, RESIDENT
0000 38
0000 39 --
0000 40
0000 41 .PAGE
0000 42 .SBTTL HISTORY ; DETAILED
0000 43
0000 44 AUTHOR: R. MUSTVEDT CREATION DATE: 6-OCT-76
0000 45
0000 46 MODIFIED BY:
0000 47
0000 48 V03-002 LJK0288 Lawrence J. Kenah 9-Aug-1984
0000 49 Use PRIB and AUTHPRI of target process rather than those of
0000 50 caller as limit on new priority specified by nonprivileged
0000 51 proces. (one lacking ALIPRI privilege).
0000 52
0000 53 V03-001 LJK0196 Lawrence J. Kenah 25-Mar-1983
0000 54 Make rescheduling check if lowering priority of current process.
0000 55 -

```

```

0000 57          .SBTTL  DECLARATIONS
0000 58
0000 59 :
0000 60 : INCLUDE FILES:
0000 61 :
0000 62
0000 63          $IPLDEF          : DEFINE INTERRUPT PRIORIYT LEVELS
0000 64          $PCBDEF          : DEFINE PCB OFFSETS
0000 65          $PHDDEF          : DEFINE PROCESS HEADER OFFSETS
0000 66          $PRDEF          : DEFINE PROCESSOR REGISTERS
0000 67          $PRIDEF         : DEFINE PRIORITY INCREMENTS
0000 68          $PRVDEF         : DEFINE PRIVILEGES
0000 69          $SSDEF          : DEFINE STATUS CODES
0000 70          $STATEDEF       : DEFINE SCHEDULER STATE NUMBERS
0000 71 :
0000 72 : EQUATED SYMBOLS:
0000 73 :
0000 74
00000004 0000 75 PIDADR=4          : PID ADDRESS ARGUMENT OFFSET
00000008 0000 76 LOGNAM=8         : LOGICAL NAME OFFSET
0000000C 0000 77 PRI=12          : NEW PRIORITY
00000010 0000 78 PRVPRI=16       : PREVIOUS PRIORITY SAVE ADDRESS
0000 79 :
0000 80 : OWN STORAGE:
0000 81 :
0000 82

```

```

0000 84      .SBTTL EXE$SETPRI - SET PRIORITY SYSTEM SERVICE ROUTINE
0000 85
0000 86 :++
0000 87 : FUNCTIONAL DESCRIPTION:
0000 88 :
0000 89 : CALLING SEQUENCE:
0000 90 :   CALLG  ARGLIST,EXE$SETPRI
0000 91 :
0000 92 : INPUT PARAMETERS:
0000 93 :   PIDADR(AP) - ADDRESS OF PID FOR TARGET PROCESS (OPTIONAL)
0000 94 :   LOGNAM(AP) - ADDRESS OF PROCESS LOGICAL NAME (OPTIONAL)
0000 95 :   PRI(AP) - BITS 0:4 CONTAIN DESIRED PRIORITY
0000 96 :   PRVPRI(AP) - ADDRESS OF LONGWORD TO RECEIVE PREVIOUS PRIORITY
0000 97 :   R4 - PCB ADDRESS OF CURRENT PROCESS
0000 98 :
0000 99 : IMPLICIT INPUTS:
0000 100 :   CURRENT PCB
0000 101 :
0000 102 : OUTPUT PARAMETERS:
0000 103 :   R0 - COMPLETION STATUS
0000 104 :   @PIDADR(AP) - PID OF PROCESS FOR WHICH PRIORITY WAS CHANGED
0000 105 :
0000 106 : IMPLICIT OUTPUTS:
0000 107 :   NONE
0000 108 :
0000 109 : COMPLETION CODES:
0000 110 :   $$$_NORMAL - SUCCESSFUL COMPLETION
0000 111 :   $$$_NOPRIV - INSUFFICIENT PRIVILEGE FOR REQUESTED OPERATION
0000 112 :   $$$_NONEXPR - NON-EXISTENT PROCESS SPECIFIED
0000 113 :
0000 114 : SIDE EFFECTS:
0000 115 :   A RESCHEDULING INTERRUPT MAY BE REQUESTED IF THE TARGET PROCESS
0000 116 :   IS RAISED TO A PRIORITY GREATER THAN THAT OF THE CURRENT PROCESS.
0000 117 :
0000 118 :--
0000 119 :
0000 120 : Put error exit paths before entry mask so that byte displacements reach
0000 121 :
50  0C  3C 0000 122 ACCVIO: MOVZWL #$$$_ACCVIO,R0      ; SET ACCESS VIOLATION
0003 123 EXITN: SETIPL #0                    ; ENABLE INTERRUPTS
0006 124      RET                          ; AND RETURN TO CALLER
0007 125
0007 126      .ENTRY EXE$SETPRI,^M<R2,R3,R4,R5> ; SET PRIORITY SYSTEM SERVICE
55  10  AC  D0 0009 127      MOVL PRVPRI(AP),R5      ; SAVE PREVIOUS PRIORITY
0006 128      BEQL 10$                          ; NONE SUPPLIED
000F 129      IFNOWRT #4,(R5),ACCVIO          ; ACCESS VIOLATION IF NO WRT
7E  0C  AC  05  00  EF 0015 130 10$: EXTZV #0,#5,PRI(AP),-(SP) ; SAVE DESIRED PRIORITY
0007 131      BSBW EXE$NAMPID                    ; LOOK UP PID/PNAME
000E 132      BLBC R0,EXITN                      ; EXIT IF ERROR (RET WILL CLEAN STACK)
53  1F  8E  C3 0021 133      SUBL3 (SP)+,#31,R3 ; CONVERT DESIRED PRIORITY TO INTERNAL FORM
7E  29  A4  9A 0025 134      MOVZBL PC$B_PRI$SAV(R4),-(SP) ; SAVE CURRENT PRIORITY
52  0000'CF D0 0029 135      MOVL W^SCH$GL_CURPCB,R2 ; GET POINTER TO CURRENT PCB
002E 136      IFPRIV SETPRI,20$,R2             ; GENERAL SETPRI PRIVILEGE?
51  2B  A4  9A 0034 137      MOVZBL PC$B_AUTHPRI(R4),R1 ; GET AUTHORIZED PRIORITY OF TARGET
51  2F  A4  91 0038 138      CMPB PC$B_PRI$B(R4),R1 ; COMPARE WITH BASE PRIORITY
003C 139      BGEQU 15$                          ; AND CHOOSE 'LARGER' FOR NEXT STEP
51  2F  A4  9A 003E 140      MOVZBL PC$B_PRI$B(R4),R1 ; BASE PRIORITY IS 'LARGER'

```

```

53 51 91 0042 141 15$: CMPB R1,R3 ; COMPARE DESIRED VALUE WITH TARGET
    03 15 0045 142 BLEQ 20$ ; WITHIN RANGE, ALLOW AS IS
53 51 9A 0047 143 MOVZBL R1,R3 ; MINIMIZE WITH BASE OF TARGET
    004A 144
    004A 145 ; R2 - PCB address of $SETPRI caller
    004A 146 ; R3 - New priority of target process
    C04A 147 ; R4 - PCB address of target process
    004A 148
28 A4 53 90 004A 149 20$: MOVB R3,PCBSB_PRISAV(R4) ; SET SAVED CURRENT PRIORITY
29 A4 53 90 004E 150 MOVB R3,PCBSB_PRIBSAV(R4) ; SET SAVED BASE PRIORITY
    OE A4 B5 0052 151 TSTW PCBSW_MTXCNT(R4) ; DOES THIS PROCESS HOLD A MUTEX?
    2E 12 0055 152 BNEQ 40$ ; YES
2F A4 53 90 0057 153 MOVB R3,PCBSB_PRI(R4) ; SET BASE PRIORITY
    005B 154
    005B 155 ; Note that the following test must be based on process state to allow for
    005B 156 ; multiple current processes on multiprocessor systems.
    005B 157
2C A4 OE B1 005B 158 CMPW #SCH$C_CUR,PCBSW_STATE(R4) ; TARGET PROCESS CURRENTLY EXECUTING?
    1D 12 005F 159 BNEQ 30$ ; BR IF NOT
OB A4 53 90 0061 160 MOVB R3,PCBSB_PRI(R4) ; SET CURRENT PRIORITY
    0065 161
    0065 162 ; If the caller of $SETPRI is lowering its own priority, a check must be made
    0065 163 ; to insure that the caller is still the highest priority computable process. If
    0065 164 ; some other COM process has a higher priority, a rescheduling request is made.
    0065 165
52 54 D1 0065 166 CMPL R4,R2 ; TARGET PROCESS SAME AS CALLER?
    14 12 0068 167 BNEQ 30$ ; SKIP RESCHEDULE CHECK IF NOT
52 0000'CF 53 90 006A 168 MOVB R3,W^SCH$GB_PRI ; RECORD NEW 'TOP' PRIORITY
    20 00 EA 006F 169 FFS #0,#32,W^SCH$GL_COMQS,R2 ; FIND PRIORITY OF NEXT COMPUTABLE PROCESS
52 53 91 0076 170 CMPB R3,R2 ; CHECK FOR DELAYED PREEMPTION
    03 1B 0079 171 BLEQU 30$ ; NO, CONTINUE
    007B 172 SOFTINT #IPL$_SCHED ; ELSE RESCHEDULE WHEN IPL DROPS
52 01 9A 007E 173 30$: MOVZBL #PRI$_IOCOM,R2 ; GIVE BENEFIT OF DOUBT
    0081 174 RPTEVT SETPRI ; REPORT EVENT
50 01 3C 0085 175 40$: MOVZWL #SS$_NORMAL,R0 ; SET SUCCESS CODE
    0088 176 SETIPL #0 ; DROP IPL
    55 D5 008B 177 TSTL R5 ; RETURN PREVIOUS PRIORITY?
    04 13 008D 178 BEQL 50$ ; NO
65 1F 8E C3 008F 179 SUBL3 (SP)+,#31,(R5) ; STORE PREVIOUS PRIORITY IN EXTERNAL FORM
    04 0093 180 50$: RET ; AND RETURN
    0094 181
    0094 182 .END

```


43598 bytes (86 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 859 non-local and 6 local symbols.
182 source lines were read in Pass 1, producing 15 object records in Pass 2.
20 pages of virtual memory were used to define 19 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	10
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	16

969 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSSETPRI/OBJ=OBJ\$:SYSSETPRI MSRC\$:SYSSETPRI/UPDATE=(ENH\$:SYSSETPRI)+EXECML\$/LIB

SYS
Pse

PSE

\$AE
YSE
\$MP

Pha

Ini
Com
Pas
Syn
Pas
Syn
Pse
Crc
Ass

The
505
The
412
22

Mac

-\$2
-\$2
TOT
104
The
MAC

