

[illegible]

[illegible]

(2)	54
(3)	74
(4)	154
(5)	181

DECLARATIONS
CHANGE THE PROTECTION ON THE CLI PAGES
EXECUTE IMAGE, ACTIVATE AN IMAGE
LGISCMSOPR - Change Mode to Supervisor

.....

```

0000 1      .TITLE  PROTCLI - SET PROTECTION ON CLI PAGES
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:      LOGINOUT
0000 31
0000 32 ABSTRACT:      INITIALIZE CLI CODE SECTION
0000 33
0000 34 ENVIRONMENT:   NATIVE MODE PRIVILEGED PROCEDURE
0000 35
0000 36 AUTHOR:       LEN KAWELL, 20-MAR-1980
0000 37
0000 38 MODIFIED BY:
0000 39
0000 40 V03-003 LJK0262      Lawrence J. Kenah      15-Feb-1984
0000 41 Add LGI$CMSUPR to leave exec stack empty when getting into
0000 42 supervisor mode. Move code that cancels a CLI's exit handler
0000 43 from this module to LOGIN.B32.
0000 44
0000 45 V03-002 ACG0376      Andrew C. Goldstein,  28-Nov-1983  15:57
0000 46 Fix page fault window in scanning page table.
0000 47 Add LGI$CANCEL_CLI routine.
0000 48
0000 49 V03-001 KDM0002      Kathleen D. Morse      28-Jun-1982
0000 50 Added $PSLDEF and $VADEF.
0000 51
0000 52 :---

```



```

0000 54 .SBTTL DECLARATIONS
0000 55 :
0000 56 : SYMBOLIC CONSTANTS
0000 57 :
00000004 0000 58 ROUTIN = 4
00000008 0000 59 ARGST = 8
0000 60 :
0000 61 : MACROS
0000 62 :
0000 63 $IPLDEF : INTERRUPT PRIORITY LEVELS
0000 64 $PCBDEF : PROCESS CONTROL BLOCK
0000 65 $PRDEF : PROCESSOR REGISTERS
0000 66 $PRTDEF : PAGE PROTECTION CODES
0000 67 $PSLDEF : PROGRAM STATUS LONGWORD
0000 68 $PTEDEF : PAGE TABLE ENTRY
0000 69 $SFDEF : OFFSETS INTO CALL FRAME
0000 70 $VADEF : VIRTUAL ADDRESS FIELDS
0000 71
00000000 72 .PSECT $CODE$,NOWRT,EXE,WORD

```

```

0000 74 .SBTTL CHANGE THE PROTECTION ON THE CLI PAGES
0000 75 :++
0000 76 : LGI$PROTECT_CLI - CHANGE THE PROTECTION ON THE CLI PAGES
0000 77 :
0000 78 : THIS ROUTINE IS CALLED TO CHANGE THE OWNER AND PROTECTION OF THE
0000 79 : MAPPED CLI PAGES. THEY ARE CREATED OWNED BY USER MODE, AND THIS
0000 80 : ROUTINE CHANGES THE OWNER TO SUPERVISOR MODE; IF ANY PAGES ARE
0000 81 : WRITEABLE, THE PROTECTION IS CHANGED TO DISALLOW USER MODE WRITING.
0000 82 :
0000 83 : INPUTS:
0000 84 :
0000 85 : 4(AP) = ADDRESS OF CLI $IMGACT RETURN ADDRESS ARRAY
0000 86 :
0000 87 : MODE = KERNEL
0000 88 :
0000 89 : OUTPUTS:
0000 90 :
0000 91 : CTL$AG_IMAGE = SAVED CLI ADDRESS ARRAY
0000 92 :
0000 93 : PTE$V_OWN OF ALL CLI PTE'S CHANGED TO PSL$C SUPER
0000 94 : PTE$V_PROT OF ALL WRITEABLE CLI PTE'S CHANGED TO PRT$C_URSW
0000 95 : --
0000 96 :
0000 97 : .ENABLE LSB
0000 98 :
0000 99 LGI$PROTECT_CLI::
0000 100 .WORD ^M<R2,R3,R4,R5,R6>
0002 101 $LKWSET_S INADR=PROT_ROUTINE
0013 102 BLBC R0,20$
0016 103 MOVL 4(AP),R1
001A 104 MOVQ (R1),R5
001D 105 CMLP R5,R6
0020 106 BGTRU 10$
0022 107 MOVL R6,R5
0025 108 MOVL (R1),R6
0028 109 10$: BICW #VASM_BYTE,R5
002D 110 BICW #VASM_BYTE,R6
0032 111 MOVL R5,R2
0035 112 DSBINT #IPL$ ASTDEL
003B 113 MOVL G^SCH$GL_CURPCB,R4
0042 114 MOVL G^CTL$GL_PHD,R5
0049 115 JSB G^MMG$PTEINDX
004F 116 BLBS R0,30$
0052 117 ENBINT
0055 118 20$: RET
0056 119
0056 120 30$: MOVL G^MMG$GL_SPTBASE,R1
005D 121 PROT_LOOP:
005D 122 TSTL @PCB$[PHD(R4)[R3]]
0061 123 DSBINT #IPL$ SYNCH
0067 124 MOVAL @PCB$[PHD(R4)[R3],R5
006C 125 EXTZV #VASV_VPN,#VASS_VPN,R5,R0
0071 126 TSTL (R1)[R0]
0074 127 BGTR 60$
0076 128 MOVL (R5),R0
0079 129 CMPZV #PTE$V_OWN,#PTE$S_OWN,-
007C 130 R0,#PSL$C_USER

```

LOCK ROUTINE IN WORKING SET
BR IF FAILURE
GET ADDRESS OF CLI ADDRESS ARRAY
GET CLI ADDRESS ARRAY
FIRST ADDRESS GTRU LAST?
BR IF YES - ORDER IS OK
SWITCH ORDER OF ADDRESSES
CLEAR BYTE OFFSETS
GET FIRST ADDRESS OF CLI
DISABLE AST DELIVERY
GET ADDRESS OF PCB
GET ADDRESS OF PHD WINDOW
CONVERT VA TO PTE INDEX
BRANCH IF SUCCESS
RE-ENABLE AST'S
RETURN FAILURE
GET ADDRESS OF SYSTEM PAGE TABLE
FAULT IN PAGE TABLE PAGE
DISABLE SWAPPING
VA OF PAGE TABLE ENTRY
GET SYSTEM VPN OF PTE
CHECK IF SYSTEM PTE IS STILL VALID
BRANCH IF NOT - RETRY
GET PAGE TABLE ENTRY
IS PAGE CURRENTLY OWNED BY USER MODE?


```

50 02 18 12 007E 131 BNEQ 50$ ; BR IF NOT
02 F0 0080 132 INSV #PSL$C_SUPER,- ; MAKE THE OWNER SUPERVISOR MODE
17 0082 133 #PTESV-OWN,#PTESS-OWN,R0
0C F0 0085 134 IFNOWRT #1,(R2),40$ ; BR IF PAGE NOT WRITEABLE
50 04 1B 135 INSV #PRT$C_URSW,- ; MAKE PAGE WRITEABLE ONLY BY SUPER
65 50 D0 008D 136 #PTESV-PROT,#PTESS-PROT,R0
03 14 0090 137 40$: MOVL R0,(R5) ; SAVE NEW PTE
0093 138 BGTR 50$ ; BR IF NOT VALID
0095 139 INVALID R2 ; INVALIDATE TRANSLATION BUFFER
52 FE00 C2 9E 0098 140 50$: DECL R3 ; DECREMENT PTE INDEX
009A 141 MOVAB -512(R2),R2 ; DECREMENT VA
009F 142 60$: ENBINT ; RE-ENABLE SWAPPING AND PAGEFAULTS
56 52 D1 00A2 143 CMPL R2,R6 ; ALL PAGES DONE?
B6 1E 00A5 144 BGEQU PROT_LOOP ; BR IF NO - CONTINUE
00A7 145 ENBINT ; RE-ENABLE AST'S
50 01 D0 00AA 146 MOVL #1,R0 ; SET SUCCESS
04 00AD 147 RET
00AE 148
00AE 149 PROT_ROUTINE: ; PROTECTION ROUTINE DESCRIPTOR
000000AE'00000000' 00AE 150 .LONG LGI$PROTECT_CLI,PROT_ROUTINE
00B6 151
00B6 152 .DISABLE LSB

```

```

00B6 154 .SBTTL EXECUTE_IMAGE, ACTIVATE AN IMAGE
00B6 155 :---
00B6 156 :
00B6 157 : THIS ROUTINE IS COPIED INTO P1 SPACE (AND THEREFORE MUST BE PIC),
00B6 158 : AND WHEN CALLED, WILL RUNDOWN THE LOGIN IMAGE AND ACTIVATE ANOTHER
00B6 159 : IMAGE (ESSENTIALLY A CHAIN). AFTER THE IMAGE COMPLETES, THE PROCESS
00B6 160 : WILL BE TERMINATED.
00B6 161 :
00B6 162 : INPUTS:
00B6 163 :
00B6 164 : ACCESS MODE IS EXECUTIVE.
00B6 165 :
00B6 166 : MMG$IMGHDRBUF = IMAGE HEADER BUFFER, THE FIRST QUADWORD CONTAINS
00B6 167 : A DESCRIPTOR OF THE IMAGE FILE SPECIFICATION.
00B6 168 :
00B6 169 : OUTPUTS:
00B6 170 :
00B6 171 : NONE, CONTROL NEVER RETURNS TO THE CALLER.
00B6 172 :---
00B6 173 :
0000 00B6 174 EXECUTE_IMAGE::
00B6 175 .WORD 0
00B8 176
00B8 177 $RUNDWN_S ACMODE=#PSL$C_USER ; RUNDOWN LOGIN IMAGE
5C 00000000'GF 9E 00C1 178 MOVAB G^MMG$IMGHDRBUF,AP ; POINT TO IMGHDR BUFFER/FILESPEC
00000000'GF 17 00C8 179 JMP G^EXE$PROCIMGACT ; ACTIVATE THE REQUESTED IMAGE

```



```

00CE 181      .SUBTITLE      LGI$CMSUPR - Change Mode to Supervisor
00CE 182      ;+
00CE 183      ; Functional Description:
00CE 184      ;
00CE 185      ;     This procedure allows a process with CMEXEC or CMKRNL privilege to
00CE 186      ;     call an arbitrary procedure in supervisor mode. The code drops from
00CE 187      ;     exec mode to supervisor mode in such a way that the exec stack is
00CE 188      ;     empty when supervisor mode is entered.
00CE 189      ;
00CE 190      ; Calling Sequence:
00CE 191      ;
00CE 192      ;     $CMEXEC (routin = LGI$CMSUPR)
00CE 193      ;
00CE 194      ; Input Parameters:
00CE 195      ;
00CE 196      ;     04(AP) - Address of procedure to be executed in supervisor mode
00CE 197      ;     08(AP) - Address of argument list to pass to said procedure.
00CE 198      ;
00CE 199      ; Implicit Input:
00CE 200      ;
00CE 201      ;     It is assumed that this routine is entered via a $CMEXEC system
00CE 202      ;     service so that there are only two call frames on the exec stack.
00CE 203      ;     The current frame is the result of the CALLG instruction inside
00CE 204      ;     the $CMEXEC system service. The previous frame is created by the
00CE 205      ;     change mode dispatcher.
00CE 206      ;
00CE 207      ; Output Parameters:
00CE 208      ;
00CE 209      ;     There are no explicit output parameters from this procedure.
00CE 210      ;
00CE 211      ; Implicit Output:
00CE 212      ;
00CE 213      ;     The frame built by the change mode dispatcher is moved from the
00CE 214      ;     exec stack to the supervisor stack. The procedure specified by
00CE 215      ;     the first parameter is called in supervisor mode.
00CE 216      ;
00CE 217      ;     The exec stack is empty when supervisor mode is entered.
00CE 218      ;
00CE 219      ; Side Effects:
00CE 220      ;
00CE 221      ;     The procedure, although entered in exec mode, exits in supervisor mode.
00CE 222      ;
00CE 223      ; Status:
00CE 224      ;
00CE 225      ;     This routine simply passes back whatever status is returned by
00CE 226      ;     the caller's supervisor procedure.
00CE 227      ;
00CE 228      ;     If the $ADJSTK call fails, this procedure passes back the error
00CE 229      ;     status without attempting to enter supervisor mode.
00CE 230      ;
00CE 231      ;     If this procedure is called (via $CMEXEC or $CMKRNL) from exec
00CE 232      ;     or kernel mode, the failure status SS$_IVSSRQ is returned.
00CE 233      ;
00CE 234      ; Note:
00CE 235      ;
00CE 236      ;     Modification of a call frame is a flagrant violation of the VAX-11
00CE 237      ;     Calling Standard. Such is life.

```



```

00CE 238 :-
00CE 239
00CE 240 LGISCMSUPR::
50 0000 00CE 241 .WORD 0 ; Make this part simple
DC 00D0 242 MOVPSL R0 ; Get current PSL
00D2 243
00D2 244 ; This procedure could cause ugly things to happen (an REI instruction will
00D2 245 ; cause a reserved operand exception) if called from exec or kernel mode and
00D2 246 ; supervisor mode were blindly entered.
00D2 247
00D2 248 BBS #PSL$V,PRVMOD+1,- ; Branch if previous mode is
00D4 249 R0,10$ ; supervisor or user.
50 0000'8F 3C 00D6 250 MOVZWL #SS$_IVSSRQ, R0 ; Otherwise, report an error
04 00DB 251 RET
00DC 252
00DC 253 ; We want to evaporate the frame put here by EXE$CMKRNL. The argument list
00DC 254 ; passed to this procedure must be preserved.
00DC 255
08 AD 5C D0 00DC 256 10$: MOVL AP, SF$L_SAVE_AP(FP) ; Alter the saved AP
10 AD E6'AF 9E 00E0 257 MOVAB B^20$, SF$L_SAVE_PC(FP) ; Alter the return PC
04 00E5 258 RET ; Transfer control to 20$
00E6 259
00E6 260 ; We arrive here with the exec stack in the following shape.
00E6 261 :
00E6 262 : 00(SP) = 0 \ Frame
00E6 263 : 04(SP) = 0 \ built by
00E6 264 : 08(SP) = Saved AP \ system's
00E6 265 : 12(SP) = Saved FP > change mode
00E6 266 : 16(SP) = Address of SRVEXIT in CMODSSDSP / dispatcher
00E6 267 : 20(SP) = Saved PC of CHME exception
00E6 268 : 24(SP) = Saved PSL of CHME exception
00E6 269
7E D4 00E6 270 20$: CLRL -(SP) ; Indicate simple SP adjustment
00E8 271 $ADJSTK_S - ; Expand supervisor stack
00E8 272 ACMODE = #PSL$C SUPER,- ; by seven longwords
00E8 273 ADJUST = #-<EXE$C_CMSTKSZ+8>,-
00E8 274 NEWADR = (SP) ; New SSP returned in (SP)
25 50 E9 00F8 275 BLBC R0,40$ ; Quit if error occurs
00FB 276
00FB 277 ; At this point, the frame created by CMODSSDSP as well as the PC,PSL
00FB 278 ; exception pair generated by the CHMK must be moved to the supervisor
00FB 279 ; stack so that the returns all work correctly.
00FB 280
3F BB 00FB 281 PUSHR #^M<R0,R1,R2,R3,R4,R5> ; Save MOVC registers
00FD 282
00FD 283 ; This adds 24 bytes to the stack so the operands to MOVC3 must be
00FD 284 ; adjusted accordingly. (The NEWSP longword is also on the stack.)
00FD 285
0008'8F 28 00FD 286 MOVC3 #<EXE$C_CMSTKSZ+8>,- ; Move frame and PC/PSL pair
1C AE 0101 287 <<6+1>*4>(SP),- ; Former 4(SP). Address of first
18 BE 0103 288 - ; longword that must be moved
0105 289 @<6*4>(SP) ; Former top of stack. Contains
0105 290 ; adjusted value of supervisor
0105 291 ; stack pointer
3F BA 0105 292 POPR #^M<R0,R1,R2,R3,R4,R5> ; Restore MOVC registers
0107 293
5D 8E D0 0107 294 MOVL (SP)+, FP ; Point FP to supervisor stack

```



```

SE  00000008'8F  C0  010A  295      ADDL2  #<EXESC_CMSTKSZ+8>, SP ; Remove rest of stuff from exec stack
                        0111  296
                        0111  297 ; At this point, the exec stack is empty. The change mode dispatcher's frame
                        0111  298 ; and the PC/PSL pair from the CHME exception are located at the top of the
                        0111  299 ; supervisor stack.
                        0111  300
02800000 8F  DD  0111  301      PUSHL  #<<PSL$C_SUPER@PSL$V_PRVMOD>+- ; Fabricate PSL to get into
                        0117  302      <PSL$C_SUPER@PSL$V_CURMOD>> ; supervisor mode
1B'AF  9F  0117  303      PUSHAB  B^30$ ; We will begin executing at 30$
02  011A  304      REI ; Change mode to supervisor
                        011B  305
                        011B  306 ; We are now in supervisor mode. The stack contents are exactly the same
                        011B  307 ; as pictured above.
                        011B  308
08 BC  FA  011B  309 30$: CALLG @ARGLIST(AP),- ; Call specified procedure
04 BC  011E  310      @ROUTIN(AP) ; with specified argument list
04  0120  311 40$: RET ; Return to caller
                        0121  312
                        0121  313      .END

```

PROTCLI
Symbol table

- SET PROTECTION ON CLI PAGES

G 4

16-SEP-1984 02:04:56 VAX/VMS Macro V04-00
5-SEP-1984 01:46:30 [LOGIN.SRC]PROTCLI.MAR;1

Page 9
(5)

ARGLST	= 00000008		
CTL\$GL PHD	*****	X	02
EXESC CMSTKSZ	*****	X	02
EXESPROCIMGACT	*****	X	02
EXECUTE IMAGE	000000B6	RG	02
IPL\$ASTDEL	= 00000002		
IPL\$SYNCH	= 00000008		
LGISCMSUPR	000000CE	RG	02
LGISPROTECT CLI	00000000	RG	02
MMG\$GL SPTBASE	*****	X	02
MMG\$IMGHDRBUF	*****	X	02
MMG\$PTEINDEX	*****	X	02
PCBSL PHD	= 0000006C		
PR\$ IPL	= 00000012		
PR\$TBIS	= 0000003A		
PROT_LOOP	0000005D	R	02
PROT_ROUTINE	000000AE	R	02
PRT\$C_URSW	= 0000000C		
PSL\$C_SUPER	= 00000002		
PSL\$C_USER	= 00000003		
PSL\$V_CURMOD	= 00000018		
PSL\$V_PRIVMOD	= 00000016		
PTES\$OWN	= 00000002		
PTES\$PROT	= 00000004		
PTES\$V_OWN	= 00000017		
PTES\$V_PROT	= 0000001B		
ROUTIN	= 00000004		
SCH\$GL_CURPCB	*****	X	02
SF\$SL_SAVE_AP	= 00000008		
SF\$SL_SAVE_PC	= 00000010		
SS\$ IVSSRO	*****	X	02
SYSSADJSTK	*****	GX	02
SYSSLKWSET	*****	GX	02
SYSSRUNDOWN	*****	GX	02
VASM_BYTE	= 000001FF		
VASS_VPN	= 00000015		
VASV_VPN	= 00000009		

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPI C USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPI C USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$CODE\$	00000121 (289.)	02 (2.)	NOPI C USR CON REL LCL NOSHR EXE RD NOWRT NOVEC WORD

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.04	00:00:00.88
Command processing	139	00:00:00.45	00:00:02.07
Pass 1	203	00:00:03.17	00:00:15.47

Symbol table sort	0	00:00:00.37	00:00:00.89
Pass 2	71	00:00:00.75	00:00:03.33
Symbol table output	6	00:00:00.03	00:00:00.03
Psect synopsis output	0	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	453	00:00:04.83	00:00:22.69

The working set limit was 1350 pages.

25168 bytes (50 pages) of virtual memory were used to buffer the intermediate code.

There were 20 pages of symbol table space allocated to hold 395 non-local and 10 local symbols.

313 source lines were read in Pass 1, producing 14 object records in Pass 2.

23 pages of virtual memory were used to define 22 macros.

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

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[LOGIN.OBJ]LOGIN.MLB;1	0
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	8
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	11
TOTALS (all libraries)	19

525 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

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

0223 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY