


```

DDDDDDDD      IIIIII      SSSSSSSS  MM      MM      NN      NN      TTTTTTTTTT  DDDDDDDD      SSSSSSSS  PPPPPPPP
DDDDDDDD      IIIIII      SSSSSSSS  MM      MM      NN      NN      TTTTTTTTTT  DDDDDDDD      SSSSSSSS  PPPPPPPP
DD      DD      SS      MMMM  MMMM  NN      NN      TT      DD      DD      SS      P          PP
DD      DD      SS      MMMM  MMMM  NN      NN      TT      DD      DD      SS      P          PP
DD      DD      SS      MM  MM  NNNN  NN      NN      TT      DD      DD      SS      P          PP
DD      DD      SSSSSS  MM      MM  NN      NN  NN      NN      TT      DD      DD      SSSSSS  P          PP
DD      DD      SSSSSS  MM      MM  NN      NN  NN      NN      TT      DD      DD      SSSSSS  P          PP
DD      DD      SS      MM      MM  NN      NN  NNNN  NN      NN      TT      DD      DD      SS      P          PP
DD      DD      SS      MM      MM  NN      NN  NNNN  NN      NN      TT      DD      DD      SS      P          PP
DD      DD      SS      MM      MM  NN      NN      NN      NN      TT      DD      DD      SS      P          PP
DD      DD      SS      MM      MM  NN      NN      NN      NN      TT      DD      DD      SS      P          PP
DDDDDDDD      IIIIII      SSSSSSSS  MM      MM  NN      NN      TT      DD      DD      SSSSSSSS  P          PP
DDDDDDDD      IIIIII      SSSSSSSS  MM      MM  NN      NN      TT      DD      DD      SSSSSSSS  P          PP

```

```

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
LLLLLLLLLLLL IIIIII      SSSSSSSS
LLLLLLLLLLLL IIIIII      SSSSSSSS

```

(1)	106	Declarations and Equates
(1)	168	Transfer Vector and Service Definitions
(1)	205	Change Mode Dispatcher Vector Block
(1)	253	Kernel Mode Dispatcher
(1)	311	Executive Mode Dispatcher

.....

```
0000 1 .TITLE DISMNTDSP - DISMOUNT system service dispatcher
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: MOUNT/DISMOUNT System Services
0000 29 ++
0000 30 Abstract:
0000 31 This module contains the dispatcher for the MOUNT and DISMOUNT
0000 32 system services, named $MOUNT and $DISMOU respectively. Both
0000 33 mount and dismount are implemented as privileged shareable images.
0000 34
0000 35 Overview:
0000 36 The $MOUNT and $DISMOU system services are contained in privileged
0000 37 shareable images that are linked into user program images in exactly
0000 38 the same fashion as any shareable image. The creation and installation
0000 39 of a privileged, shareable image is slightly different from that
0000 40 of an ordinary shareable image. These differences are:
0000 41
0000 42 1. A vector defining the entry points and providing other
0000 43 control information to the image activator. This vector
0000 44 is a the lowest address in an image section with the VEC
0000 45 attribute.
0000 46
0000 47 2. The shareable image is linked with the /PROTECT option
0000 48 that marks all of the image sections so that they will
0000 49 protected and given EXEC mode ownership by the image
0000 50 activator.
0000 51
0000 52 3. The shareable image MUST be installed /SHARE /PROTECT
0000 53 with the INSTALL utility in order for the image activator
0000 54 to connect the privileged shareable image to the change mode
0000 55 dispatchers.
0000 56
0000 57 A privileged shareable image implementing system services is comprised
```

```
0000 58 : of the following major components:
0000 59 :
0000 60 : 1. A transfer vector containing all of the entry points and
0000 61 : collecting them at the lowest virtual address in the shareable
0000 62 : image. This formalism enables revision of the shareable
0000 63 : image without necessitating the relinking of images that
0000 64 : use it.
0000 65 :
0000 66 : 2. A Privileged Library Vector in a PSECT with the VEC attribute
0000 67 : that describes the entry points for dispatching EXEC and
0000 68 : KERNEL mode services along with validation information.
0000 69 :
0000 70 : 3. A dispatcher for kernel mode services. This code will
0000 71 : be called by the VMS change mode dispatcher when it
0000 72 : fails to recognize a kernel mode service request.
0000 73 :
0000 74 : 4. A dispatcher for executive mode services. This code will
0000 75 : be called by the VMS change mode dispatcher when it fails
0000 76 : to recognize an executive mode service request.
0000 77 :
0000 78 : 5. Service routines to perform the various services.
0000 79 :
0000 80 :
0000 81 : Environment:
0000 82 :
0000 83 : VAX/VMS operating system, installed as a privileged shareable image.
0000 84 :
0000 85 : --
0000 86 :
0000 87 : Author:
0000 88 :
0000 89 : Steven T. Jeffreys
0000 90 :
0000 91 : Modified by:
0000 92 :
0000 93 : V03-001 STJ3060 Steven T. Jeffreys, 24-Feb-1983
0000 94 : Set SP equal to FP before dispatching to SYSSDISMOU.
0000 95 : This is VERY important, as high-level languages assume this
0000 96 : is true upon entry to a routine.
0000 97 :
0000 98 : V02-002 STJ0203 Steven T. Jeffreys, 07-Feb-1982
0000 99 : Add R4 to the $DISMOU register save mask.
0000 100 :
0000 101 : V02-001 STJ0106 Steven T. Jeffreys, 18-Aug-1981
0000 102 : Removed .LIBRARY reference to SYSSLIBRARY:LIB.MLB
0000 103 :
0000 104 : --
```



```
0000 163 ; number of required arguments.  
00000000 164 .PSECT EXEC_NARG,BYTE,NOWRT,EXEC,PIC  
0000 165 EXEC_NARG: ; Base of byte table containing the  
0000 166 ; number of required arguments.
```

```
0000 168 .SBTTL Transfer Vector and Service Definitions
0000 169 :++
0000 170 : The use of transfer vectors to effect entry to the system services
0000 171 : enables some updating of the shareable image containing them without necessitating
0000 172 : a re-link of all programs that call them. The PSECT containing the transfer
0000 173 : vector will be positioned at the lowest virtual address in the shareable image
0000 174 : and so long as the transfer vector is not re-ordered, programs linked with
0000 175 : one version of the shareable image will continue to work with the next.
0000 176 :
0000 177 : Thus as additional services are added to a privileged shareable image, their
0000 178 : definitions should be added to the end of the following list to ensure that
0000 179 : programs using previous versions of it will not need to be re-linked.
0000 180 : To completely avoid relinking existing programs the size of the privileged
0000 181 : shareable image must not change so some padding will be required to provide
0000 182 : the opportunity for future growth.
0000 183 :--
0000 184
0000 185 DEFINE_SERVICE SYSSDISMOU,2,KERNEL ; $DISMOU system service
0002 186
0002 187 :
0002 188 : The base values used to generate the dispatching codes should be negative for
0002 189 : user services and must be chosen to avoid overlap with any other privileged
0002 190 : shareable images that will be used concurrently. Their definition is
0002 191 : deferred to this point in the assembly to cause their use in the preceding
0002 192 : macro calls to be forward references that guarantee the size of the change
0002 193 : mode instructions to be four bytes. This satisfies an assumption that is
0002 194 : made by for services that have to wait and be retried. The PC for retrying
0002 195 : the change mode instruction that invokes the service is assumed to be 4 bytes
0002 196 : less than that saved in the change mode exception frame. Of course, the particula
0002 197 : service routine determines whether this is possible.
0002 198 :
0002 199 : The change-mode codes available to $MOUNT and $DISMOU are the decimal values
0002 200 : from 16520 to 16527.
0002 201 :
00004088 0002 202 KCODE_BASE=16520 ; Base CHMK code value for these services
00000000 0002 203 ECODE_BASE=0 ; Base CHME code value for these services
```



```

0002 205 .SBTTL Change Mode Dispatcher Vector Block
0002 206 :+
0002 207 : This vector is used by the image activator to connect the privileged shareable
0002 208 : image to the VMS change mode dispatcher. The offsets in the vector are self-
0002 209 : relative to enable the construction of position independent images. The system
0002 210 : version number will be used by the image activator to verify that this shareable
0002 211 : image was linked with the symbol table for the current system.
0002 212 :
0002 213 :
0002 214 :
0002 215 :
0002 216 :
0002 217 :
0002 218 :
0002 219 :
0002 220 :
0002 221 :
0002 222 :
0002 223 :
0002 224 :
0002 225 :
0002 226 :
0002 227 :
0002 228 :
0002 229 :
0002 230 :
0002 231 :
0002 232 :
0002 233 :
0002 234 :
0002 235 :
0002 236 :
0002 237 :
0002 238 :
0002 239 :
0002 240 :
0002 241 :
00000000 242 :
0000 243 :
00000001 0000 244 :
00000000' 0004 245 :
00000003' 0008 246 :
FFFFFFFF' 000C 247 :
00000000 0010 248 :
00000000 0014 249 :
00000000 0018 250 :
00000000 001C 251 :

```

Change Mode Vector Format

Vector Type Code (PLVSC_TYP_CM0D)	PLVSL_TYPE
System Version Number (SYSSK_VERSION)	PLVSL_VERSION
Kernel Mode Dispatcher Offset	PLVSL_KERNEL
Exec Mode Entry Offset	PLVSL_EXEC
Reserved	
Reserved	
RMS Dispatcher Offset	PLVSL_RMS
Address Check	PLVSL_CHECK

```

.PSECT DISMNTDSP,PAGE,VEC,PIC,NOVRT,EXE
.LONG PLVSC_TYP_CM0D ; Set type of vector to change mode dispatch
.LONG SYSSK_VERSION ; Identify system version
.LONG KERNEC_DISPATCH- ; Offset to kernel mode dispatcher
.LONG EXEC_DISPATCH- ; Offset to executive mode dispatcher
.LONG 0 ; Reserved.
.LONG 0 ; Reserved.
.LONG 0 ; No RMS dispatcher
.LONG 0 ; Address check - PIC image

```

```

0020 253      .SBTTL  Kernel Mode Dispatcher
0020 254      :++
0020 255      : Input Parameters:
0020 256      :
0020 257      : (SP) - Return address if bad change mode value
0020 258      :
0020 259      : R0 - Change mode argument value.
0020 260      :
0020 261      : R4 - Current PCB Address. (Therefore R4 must be specified in all
0020 262      :       register save masks for kernel routines.)
0020 263      :
0020 264      : AP - Argument pointer existing when the change
0020 265      :       mode instruction was executed.
0020 266      :
0020 267      : FP - Address of minimal call frame to exit
0020 268      :       the change mode dispatcher and return to
0020 269      :       the original mode.
0020 270      :--
00000000 271      .PSECT  DISMNTDSP_KERNEL_DISP0, BYTE, NOWRT, EXE, PIC
0000 272      KACCVIO:      ; Kernel access violation
50 0C 3C 0000 273      MOVZWL  #SS$_ACCVIO, R0      ; Set access violation status code
04 0003 274      RET      ; and return
0004 275      KINSFARG:   ; Kernel insufficient arguments.
50 0114 8F 3C 0004 276      MOVZWL  #SS$_INSFARG, R0      ; Set status code and
04 0009 277      RET      ; return
05 000A 278      KNOTME:  RSB      ; RSB to forward request
000B 279
000B 280      KERNEL_DISPATCH: ; Entry to dispatcher
51 BF78 C0 9E 000B 281      MOVAB   W^<KCODE_BASE(R0), R1      ; Normalize dispatch code value
01 51 B1 0010 282      BLSS   KNOTME      ; Branch if code value too low
01 51 B1 0012 283      CMPW   R1, #KERNEL_COUNTER      ; Check high limit
01 51 B1 0015 284      BGEQU  KNOTME      ; Branch if out of range
0017 285      :
0017 286      : The dispatch code has now been verified as being handled by this dispatcher,
0017 287      : now the argument list will be probed and the required number of arguments
0017 288      : verified.
0017 289      :
0017 290      MOVZBL  W^KERNEL_NARG[R1], R1      ; Get required argument count
51 00000004 9A 001D 291      MOVAL   @#4[R1], R1      ; Compute byte count including arg count
0025 292      IFNORD  R1, (AP), KACCVIO      ; Branch if arglist not readable
BF78'CF40 6C 91 002B 293      CMPB   (AP), W^<KERNEL_NARG-KCODE_BASE>[R0] ; Check for required number
01 0031 294      BLSSU  KINSFARG      ; of arguments
5E 5D D0 0033 295      MOVL   FP, SP      ; Set SP equal to FP
01 50 AF 0036 296      CASEW   R0, -      ; Case on change mode
0038 297      -      ; argument value
0038 298      #KCODE BASE, -      ; Base value
00 4088 8F 0038 299      #<KERNEL_COUNTER-1>      ; Limit value (number of entries)
003C 300      KCASE_BASE:      ; Case table base address for DEFINE_SERVICE
003C 301      :
003C 302      : Case table entries are made in the PSECT DISMNTDSP_KERNEL_DISP1 by
003C 303      : invocations of the DEFINE_SERVICE macro. The three PSECTS,
003C 304      : DISMNTDSP_KERNEL_DISP0, 1, 2 will be abutted in lexical order at link-time.
003C 305      :
00000000 306      .PSECT  DISMNTDSP_KERNEL_DISP2, BYTE, NOWRT, EXE, PIC
0000 307      BUG_CHECK IVSSRVRQST, FATAL      ; Invalid system service request
05 0004 308      RSB      ; Return to reject out of
0005 309      ; range value

```

```

0005 311 .SBTTL Executive Mode Dispatcher
0005 312 :++
0005 313 : Input Parameters:
0005 314 :
0005 315 : (SP) - Return address if bad change mode value
0005 316 :
0005 317 : RO - Change mode argument value.
0005 318 :
0005 319 : AP - Argument pointer existing when the change
0005 320 : mode instruction was executed.
0005 321 :
0005 322 : FP - Address of minimal call frame to exit
0005 323 : the change mode dispatcher and return to
0005 324 : the original mode.
0005 325 :--
00000000 326 .PSECT DISMNTDSP_EXEC_DISPO, BYTE, NOWRT, EXE, PIC
50 0C 3C 0000 327 EACCVIO: ; Exec access violation
04 0003 328 MOVZWL #SS$_ACCVIO, R0 ; Set access violation status code
0004 329 RET ; and return
50 0114 8F 3C 0004 330 EINSFARG: ; Exec insufficient arguments.
04 0009 331 MOVZWL #SS$_INSFARG, R0 ; Set status code and
05 000A 332 RET ; return
000B 333 ENOTME: RSB ; RSB to forward request
000B 334
51 0000 C0 9E 000B 335 EXEC_DISPATCH:: ; Entry to dispatcher
00 F8 19 0010 336 MOVAB W^_ECODE_BASE(R0), R1 ; Normalize dispatch code value
00 51 B1 0012 337 BLSS ENOTME ; Branch if code value too low
00 F3 1E 0015 338 CMPW R1, #EXEC_COUNTER ; Check high limit
0017 339 BGEQU ENOTME ; Branch if out of range
0017 340 :
0017 341 : The dispatch code has now been verified as being handled by this dispatcher,
0017 342 : now the argument list will be probed and the required number of arguments
0017 343 : verified.
0017 344 :
51 51 0000'CF41 9A 0017 345 MOVZBL W^EXEC_NARG[R1], R1 ; Get required argument count
51 00000004 9F41 DE 001D 346 MOVAL @#4[R1], R1 ; Compute byte count including arg count
0000'CF40 6C 91 0025 347 IFNORD R1, (AP), EACCVIO ; Branch if arglist not readable
D1 1F 0031 348 CMPB (AP), W^<EXEC_NARG-ECODE_BASE>[R0] ; Check for required number
SE 5D D0 0033 349 BLSSU EINSFARG ; of arguments
50 50 AF 0036 350 MOVL FP, SP ; Set SP equal to FP
0038 351 CASEW R0, - ; Case on change mode
0038 352 - ; argument value
0038 353 #ECODE_BASE, - ; Base value
FFFF 8F 00 0038 354 #<EXEC_COUNTER-1> ; Limit value (number of entries)
0C3C 355 ECASE_BASE: ; Case table base address for DEFINE_SERVICE
003C 356 :
003C 357 : Case table entries are made in the PSECT DISMNTDSP_EXEC_DISP1 by
003C 358 : invocations of the DEFINE_SERVICE macro. The three PSECTS,
003C 359 : DISMNTDSP_EXEC_DISPO, 1, 2 will be abutted in lexical order at link-time.
00000000 360 :
05 0C00 361 .PSECT DISMNTDSP_EXEC_DISP2, BYTE, NOWRT, EXE, PIC
0004 362 BUG_CHECK IVSSRVRQST, FATAL ; Invalid system service request
0005 363 RSB ; Return to reject out of
0005 364 ; range value
0005 365
0005 366 .END

```

BUGS_IVSSRVRQST	*****	X	08
EACCVIO	00000000	R	09
ECASE_BASE	0000003C	R	09
ECODE_BASE	= 00000000		
EINSFARG	00C00004	R	09
ENOTME	0000000A	R	09
EXEC_COUNTER	= 00000000		
EXEC_DISPATCH	0000000B	RG	09
EXEC_NARG	00000000	R	03
KACCVIO	00000000	R	07
KCASE_BASE	0000003C	R	07
KCODE_BASE	= 00004088		
KERNEL_COUNTER	= 00000001		
KERNEL_DISPATCH	0000000B	RG	07
KERNEL_NARG	00000000	R	02
KINSFARG	00000004	R	07
KNOTME	0000000A	R	07
PLVSC_TYP_CMOD	= 00000001		
SSS_ACCVIO	= 0000000C		
SSS_INSFARG	= 00000114		
SYSDISMOU	*****	X	04
SYSSK_VERSION	*****	X	06

↑-----↑
! 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			
KERNEL_NARG	00000001 (1.)	02 (2.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			
EXEC_NARG	00000000 (0.)	03 (3.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			
SSSTRANSFER_VECTOR	00000007 (7.)	04 (4.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	PAGE			
DISMNTDSP_KERNEL_DISP1	00000002 (2.)	05 (5.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			
DISMNTDSP_KERNEL_DISP0	00000020 (32.)	06 (6.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	VEC	PAGE			
DISMNTDSP_KERNEL_DISP2	0000003C (60.)	07 (7.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			
DISMNTDSP_EXEC_DISP2	00000005 (5.)	08 (8.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			
DISMNTDSP_EXEC_DISP0	0000003C (60.)	09 (9.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			
DISMNTDSP_EXEC_DISP1	00000005 (5.)	0A (10.)	PIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE			

↑-----↑
! Performance indicators !
↑-----↑

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:01.27
Command processing	114	00:00:00.61	00:00:03.71
Pass 1	175	00:00:04.48	00:00:13.05
Symbol table sort	0	00:00:00.66	00:00:01.30
Pass 2	72	00:00:01.14	00:00:05.65
Symbol table output	4	00:00:00.03	00:00:00.22
Psect synopsis output	3	00:00:00.07	00:00:00.21
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	397	00:00:07.06	00:00:25.56

The working set limit was 1500 pages.
23738 bytes (47 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 437 non-local and 0 local symbols.
366 source lines were read in Pass 1, producing 29 object records in Pass 2.
14 pages of virtual memory were used to define 11 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

493 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

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

