



```

LL      000000  CCCCCCCC  KK      KK
LL      000000  CCCCCCCC  KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KKKKKK
LL      00      00  CC      KKKKKK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LLLLLLLLLLLL 000000  CCCCCCCC  KK      KK
LLLLLLLLLLLL 000000  CCCCCCCC  KK      KK
          .....
          .....
          .....
          .....

```

```

LL      111111  SSSSSSSS
LL      111111  SSSSSSSS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SSSSSS
LL      11      SSSSSS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LLLLLLLLLLLL 111111  SSSSSSSS
LLLLLLLLLLLL 111111  SSSSSSSS

```

(1)	2	COPYRIGHT NOTICE
(1)	29	PROGRAM DESCRIPTION
(2)	77	DECLARATIONS
(3)	103	STORAGE DEFINITIONS
(5)	342	READ-ONLY DATA DEFINITIONS
(6)	465	SHOW_ALL_LOCKS - ACTION ROUTINE TO DISPLAY ALL LOCKS
(7)	499	SHOW_ONE_LOCK - ACTION ROUTINE TO DISPLAY ONE LOCK
(8)	529	SHOW_PROC_LOCK - SHOW LOCKS ASSOCIATED WITH A PROCESS
(9)	572	GET_CKB - GET LOCK BLOCK
(10)	618	SAVE_LOCK_DATA - RETRIEVE LOCK DATA FROM SYSTEM
(11)	658	DISPCAY_LOCK - CONTROL DISPLAY OF LOCK DATA
(12)	683	FORMAT_LOCK - FORMAT LOCK DATA
(13)	867	PRINT_LOCK - OUTPUT LOCK DATA
(14)	990	LOCK_COND_HAND - CONDITION HANDLER FOR SHOW ALL LOCKS
(15)	1029	SHOW_ALL_RES - ACTION ROUTINE TO DISPLAY ALL RESOURCES
(16)	1091	SHOW_ONE_RES - ACTION ROUTINE TO DISPLAY ONE RESOURCE
(17)	1124	SHOW_RSB - CONTROL FOR RSB DISPLAY
(18)	1147	SHOW_QUEUES - DISPLAY QUEUES FOR GIVEN RESOURCE
(19)	1180	FORMAT_RSB - FORMAT RSB DATA
(20)	1257	PRINT_RSB - OUTPUT RSB DATA
(21)	1358	PROCESS_QUEUE - TRAVERSE RESOURCE QUEUES
(22)	1428	PRINT_LINE - OUTPUT QUEUE DATA

```
0000 1 .TITLE LOCK LOCK AND RESOURCE FORMATTING ROUTINES
0000 2 .SBTTL COPYRIGHT NOTICE
0000 3 .IDENT 'V04-000'
0000 4 :
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1979, 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 29 .SBTTL PROGRAM DESCRIPTION
0000 30 :++
0000 31 FACILITY
0000 32
0000 33 SYSTEM DUMP ANALYZER
0000 34
0000 35 ABSTRACT
0000 36
0000 37 ROUTINES TO FORMAT LOCK AND RESOURCE INFORMATION
0000 38
0000 39 ENVIRONMENT
0000 40
0000 41 NATIVE MODE, USER MODE
0000 42
0000 43 AUTHOR
0000 44
0000 45 MARYANN HINDEN, JUNE 1982
0000 46
0000 47 MODIFIED BY
0000 48
0000 49 V03-007 SRB0109 Steve Beckhardt 2-Feb-1984
0000 50 Fixed formatting of Directory entry text in SHO RES line.
0000 51
0000 52 V03-006 SRB0107 Steve Beckhardt 7-Dec-1983
0000 53 Enhanced SHOW LOCK and SHOW RESOURCE to display
0000 54 distributed lock information.
0000 55
0000 56 V03-005 SRB0089 Steve Beckhardt 27-May-1983
0000 57 Fixed bug displaying resource name. Added new status bit
0000 58 definitions.
0000 59
0000 60 V03-004 SRB0071 Steve Beckhardt 16-Mar-1983
0000 61 Changed the way system resources are decoded in RSB
0000 62 to use zero group number instead of SYSNAM flag.
0000 63
0000 64 V03-003 MSH0003 Maryann Hinden 17-Dec-1982
0000 65 Fix broken assume's.
0000 66
0000 67 V03-002 MSH0002 Maryann Hinden 22-Oct-1982
0000 68 Determine empty lock queue correctly in SHOW_PROC_LOCK.
0000 69 Change lock display - output lock and parent id in hex;
0000 70 rearrange format.
0000 71
0000 72 V03-001 MSH0001 Maryann Hinden 16-Jul-1982
0000 73 Correct references to Lck$xxxx data structures; change
0000 74 condition handling.
0000 75 :--
```

000

```
0000 77 .SBTTL DECLARATIONS
0000 78 :
0000 79 : SYSTEM SYMBOL DEFINITIONS
0000 80 :
0000 81 :
0000 82 $CHFDEF
0000 83 $LCKDEF
0000 84 $LKBDEF
0000 85 $OPTDEF
0000 86 $PCBDEF
0000 87 $RSBDEF
0000 88 $STSDEF
0000 89 :
0000 90 :
0000 91 : LOCAL SYMBOL DEFINITIONS
0000 92 :
0000 93 :
FFFFFFFF 0000 94 GRANT = -1 ; SYMBOLIC DESIGNATIONS FOR QUEUE TYPES
00000000 0000 95 CONVERT = 0
00000001 0000 96 WAIT = 1
0000 97 :
00000000 0000 98 LKID_OFF = LKB$L_LKID-LKB$L_LKID ; OFFSETS INTO SAVED QUEUE DATA
00000005 0000 99 GRMD_OFF = LKB$B_GRMODE-LKB$L_LKID
00000004 0000 100 RQMD_OFF = LKB$B_RQMODE-LKB$L_LKID
0000 101
```

```

0000 103 .SBTTL STORAGE DEFINITIONS
00000000 104 .PSECT SDADATA,NOEXE,WRT,LONG
0000 105 :
0000 106 : Data area for SHOW LOCKS commands
0000 107 :
0000 108 :
0000 109 LOCK_COUNT: ; count of locks processed
00000000 0000 110 .LONG 0
00000000 0004 111 PARID_BFR: ; parent lock id
00000000 0004 112 .LONG 0
00000068 0008 113 LKB_BFR: ; lock block data
00000068 0008 114 .BLKB LKB$K_LENGTH
00000068 0068 115
0000006C 0068 116 RSBCSID_BFR: ; CSID in RSB
0000006C 0068 117 .BLKL 1
0000006E 006C 118 LKB_RSB_BFR: ; resource block data for given lock
0000006E 006C 119 GROOP_BFR: ; group number
0000006E 006C 120 .BLKW 1
0000006F 006E 121 RMOD_BFR: ; access mode
0000006F 006E 122 .BLKB 1
00000070 006F 123 RSNLEN_BFR: ; resource name length
00000070 006F 124 .BLKB 1
00000090 0070 125 RESNAM_BFR: ; resource name
00000090 0070 126 .BLKB RSB$K_MAXLEN+1
00000090 0090 127
00000094 0090 128 LOCKID:: ; lock ID currently being processed
00000094 0090 129 .BLKL 1
00000094 0094 130
00000094 0094 131 .ALIGN LONG
00000094 0094 132
00000094 0094 133 FAO_RMINFO_DSC:
63 6F 6C 20 66 6F 0000009C'010E0000' 0094 134 .ASCID "of lock !XL on system !XL"
73 79 73 20 6E 6F 20 4C 58 21 20 6B 00A2
4C 58 21 20 6D 65 74 00AE
00B5 135
00B5 136 .ALIGN LONG
00000000 00B8 137 RMINFO_LEN:
00000050 00B8 138 .LONG 0
000000C5' 00BC 139 RMINFO_DSC:
00C4 140 .LONG 80
00C4 141 .ADDRESS RMINFO_BFR
000000C5 00C4 142
00000115 00C4 143 RMINFO_CNT:
00000115 00C4 144 .BLKB 1
00000115 00C5 145 RMINFO_BFR:
0115 146 .BLKB 80
0115 147
0115 148 :
0115 149 : Data area for SHOW RESOURCES commands
0115 150 :
0115 151 :
00000000 0115 152 RES_COUNT: ; count of resources processed
00000000 0119 153 .LONG 0
00000000 0119 154 QUEUE_COUNT: ; count of queue elements processed
0000018C 011D 155 .LONG 0
0000018C 011D 156 RSB_BFR: ; resource block data
0000018C 011D 157 .BLKB RSB$K_MAXLEN+RSB$K_LENGTH

```

```

00000190 018C 158 HTBL_INDX: ; current index into hash table
018C 159 .BLKL 1
00000194 0190 160 HTBL_CNT: ; size of hash table
0190 161 .BLKL 1
00000198 0194 162 COUNT: ; # queue items to display
0194 163 .BLKL 1
0000019C 0198 164 QUEUE_TYPE: ; type of queue being processed
0198 165 .BLKL 1
0000019F 019C 166 GRMD_BFR: ; grant mode
019C 167 .BLKB 3
000001A2 019F 168 RQMD_BFR: ; requested mode
019F 169 .BLKB 3
000001AE 01A2 170 LKID_BFR: ; lock id
01A2 171 .BLKL 3
57 4F 33 21 000001B6'010E0000' 01AE 172 FAO_GROUP_DSC:
01AE 173 .ASCID "'!30W'"
01BA 174
000001BD 01BA 175 GROUP_BUF:
01BA 176 .BLKB 3
01BD 177
00000003 01BD 178 .ALIGN LONG
01C0 179 GROUP_BUF_DSC:
000001BA' 01C0 180 .LONG 3
01C4 181 .ADDRESS GROUP_BUF
01C8 182
6D 65 74 73 79 53 000001D0'010E0000' 01C8 183 SYS_DSC:
01C8 184 .ASCID "System"
01D6 185
0000000B 01D6 186 .ALIGN LONG
000001E0' 01D8 187 GROUP_DSC:
01D8 188 .LONG 11
01DC 189 .ADDRESS GROUP_TXT
01E0 190
20 20 20 70 75 6F 72 47 01E0 191 GROUP_TXT:
01E0 192 .ASCII "Group "
01E8 193 GROUP_NUM:
000001EB 01E8 194 .BLKB 3
01EB 195
01EB 196 :
01EB 197 : FAO data storage for RESOURCE display
01EB 198 :
01EB 199 :
01EB 200 : line 1
01EB 201
00000000 01EB 202 RSB: .LONG 0 ; Address of RSB
00000000 01EF 203 GGMOD: .LONG 0 ; Group grant mode (addr of ASCIC string)
01F3 204
01F3 205 : line 2
01F3 206
00000000 01F3 207 PRSB: .LONG 0 ; Address of parent RSB
00000000 01F7 208 CGMOD: .LONG 0 ; Conversion grant mode (addr of ASCIC string)
01FB 209
01FB 210 : line 3
01FB 211
00000000 01FB 212 SRSBCT: .LONG 0 ; Sub-RSB reference count
00000000 01FF 213 BAST: .LONG 0 ; Blocking AST count
0203 214

```



```

0203 215 ; line 4
0203 216
00000000 0203 217 VAL4: .LONG 0 ; Value block longword #4
00000000 0207 218 VAL3: .LONG 0 ; Value block longword #3
00000000 020B 219 VAL2: .LONG 0 ; Value block longword #2
00000000 020F 220 VAL1: .LONG 0 ; Value block longword #1
00000000 0213 221 SEQNUM: .LONG 0 ; Sequence number
00000000 0217 222 VALID: .LONG 0 ; Value block valid (addr. of .ASCII string)
021B 223
021B 224 ; line 5
021B 225
00000000 021B 226 RESN2: .LONG 0 ; Second longword of resource name
00000000 021F 227 RESN1: .LONG 0 ; First
00000008 0223 228 .LONG 8 ; Count of ASCII string
00000000 0227 229 BUF1: .LONG 0 ; Text of resource name (addr of ASCII string)
022B 230
022B 231 ; line 6
022B 232
00000000 022B 233 RESNLEN: .LONG 0 ; Resource name length
00000000 022F 234 RESN4: .LONG 0 ; Fourth longword of resource name
00000000 0233 235 RESN3: .LONG 0 ; Third
00000008 0237 236 .LONG 8 ; Count of ASCII string
00000000 023B 237 BUF2: .LONG 0 ; Text of resource name (addr of ASCII string)
00000000 023F 238 CSID: .LONG 0 ; RSB CSID
0243 239
0243 240 ; line 7
0243 241
00000000 0243 242 RACMOD: .LONG 0 ; Resource access mode (addr of ASCII string)
00000000 0247 243 RESN6: .LONG 0 ; Sixth longword of resource name
00000000 024B 244 RESN5: .LONG 0 ; Fifth
00000008 024F 245 .LONG 8 ; Count of ASCII string
00000000 0253 246 BUF3: .LONG 0 ; Text of resource name (addr of ASCII string)
0257 247 DIRENTRY:
00000000 0257 248 .LONG 0 ; Directory entry (addr. of .ASCII string)
025B 249
025B 250 ; line 8
025B 251
00000000 025B 252 RNSPACE: .LONG 0 ; Address of descriptor
00000000 025F 253 RESN8: .LONG 0 ; Eighth longword of resource name
00000000 0263 254 RESN7: .LONG 0 ; Seventh
00000008 0267 255 .LONG 8 ; Count of ASCII string
00000000 026B 256 BUF4: .LONG 0 ; Text of resource name (addr of ASCII string)
026F 257

```

```

026F 259 :
026F 260 :      FAO data storage for LOCKS display
026F 261 :
026F 262 :
026F 263 ; LINE 1
026F 264 :
00000000 026F 265 LKID:  .LONG  0      ; Lock id (value)
00000000 0273 266 PID:  .LONG  0      ; PID (value)
00000000 0277 267 FLAGS1: .LONG  0      ; First line of flags (addr. of .ASCII string)
00000000 027B 268          .LONG  0      ;
00000000 027F 269          .LONG  0      ;      "      "      "      "
0283 270 :
0283 271 ; LINE 2
0283 272 :
00000000 0283 273 PARID: .LONG  0      ; Parent id (value)
00000000 0287 274 STATE1: .LONG  0      ; Lock state info (address of .ASCII string)
00000002 028B 275          .LONG  2      ; Lock mode (length of string)
00000000 028F 276 MODE1:  .LONG  0      ; Lock mode (address of .ASCII string)
00000000 0293 277 FLAGS2: .LONG  0      ; Second line of flags (addr. of .ASCII string)
00000000 0297 278          .LONG  0      ;
00000000 029B 279          .LONG  0      ;      "      "      "      "
029F 280 :
029F 281 ; LINE 3
029F 282 :
00000000 029F 283 SUBLKS: .LONG  0      ; Sublocks (value)
00000000 02A3 284 STATE2: .LONG  0      ; More lock state info (addr. of .ASCII string)
00000002 02A7 285          .LONG  2      ; More lock mode (length of string)
00000000 02AB 286 MODE2:  .LONG  0      ; More lock mode ( addr. of .ASCII string)
00000000 02AF 287 FLAGS3: .LONG  0      ; Third line of flags (addr. of .ASCII string)
00000000 02B3 288          .LONG  0      ;
00000000 02B7 289          .LONG  0      ;      "      "      "      "
02BB 290 :
02BB 291 ; LINE 3A
02BB 292 :
00000000 02BB 293 BLKAST: .LONG  0      ; Blocking AST (address of .ASCII string)
00000000 02BF 294 LKB:   .LONG  0      ; Address of LKB
02C3 295 :
02C3 296 ; LINE 4
02C3 297 :
00000000 02C3 298 RES2:  .LONG  0      ; Second longword of resource name (value)
00000000 02C7 299 RES1:  .LONG  0      ; First longword of resource name (value)
00000008 02CB 300          .LONG  8      ;
00000000 02CF 301 DMP1:  .LONG  0      ; First line of text name (.ASCII string)
00000000 02D3 302 STS1:  .LONG  0      ; 1st line of status (addr. of .ASCII strings)
00000000 02D7 303          .LONG  0      ;
00000000 02DB 304          .LONG  0      ;
02DF 305 :
02DF 306 ; LINE 5
02DF 307 :
00000000 02DF 308 RLEN:  .LONG  0      ; Resource name length (value)
00000000 02E3 309 RES4:  .LONG  0      ; Fourth longword of resource name (value)
00000000 02E7 310 RES3:  .LONG  0      ; Third longword of resource name (value)
00000008 02EB 311          .LONG  8      ;
00000000 02EF 312 DMP3:  .LONG  0      ; Second line of text name (.ASCII string)
00000000 02F3 313 STS2:  .LONG  0      ; 2nd line of status (addr. of .ASCII strings)
00000000 02F7 314          .LONG  0      ;
00000000 02FB 315          .LONG  0      ;

```

```
02FF 316  
02FF 317 ; LINE 6  
02FF 318  
00000000 02FF 319 ACMODE: .LONG 0 ; Access mode (Address of .ASCIC string)  
00000000 0303 320 RES6: .LONG 0 ; Sixth longword of resource name (value)  
00000000 0307 321 RES5: .LONG 0 ; Fifth longword of resource name (value)  
00000008 0308 322 .LONG 8  
00000000 030F 323 DMP5: .LONG 0 ; Third line of text name (.ASCII string)  
00000000 0313 324 STS3: .LONG 0 ; 3rd line of status (addr. of .ASCIC strings)  
00000000 0317 325 .LONG 0  
00000000 031B 326 .LONG 0  
031F 327  
031F 328 ; LINE 7  
031F 329  
00000000 031F 330 NSPACE: .LONG 0 ; Name space (group/system) (addr. of desc.)  
00000000 0323 331 RES8: .LONG 0 ; Eighth longword of resource name (value)  
00000000 0327 332 RES7: .LONG 0 ; Seventh longword of resource name (value)  
00000008 0328 333 .LONG 8  
00000000 032F 334 DMP7: .LONG 0 ; Fourth line of text name (.ASCII string)  
0333 335  
0333 336 ; LINE 8  
0333 337  
00000000 0333 338 TYPE: .LONG 0 ; Lock type (addr. of .ASCIC string)  
00000000 0337 339 RMINFO: .LONG 0 ; Remote info (addr. of .ASCIC string)  
0338 340
```

```

0000033B 342 .SBTTL READ-ONLY DATA DEFINITIONS
00000000 343 .PSECT LOCK,EXE,NOWRT,LONG
00000000 344
00000000 345 ; HEADERS
00000000 346
00000000 347 LOCK_HEAD: STRING <Lock database>
00150015 348 RES_READ: STRING <Resource database>
002E002E 349
002E002E 350 LOCK_STR_TBL: ; table of FAO strings for wait/grant queue display
00000000 002E 351 .LONG 0
0000005A' 0032 352 .LONG LOCKSTR1
0000006F' 0036 353 .LONG LOCKSTR2
00000093' 003A 354 .LONG LOCKSTR3
003E003E 355
003E003E 356 CONV_STR_TBL: ; table of FAO strings for convert queue display
00000000 003E 357 .LONG 0
000000C6' 0042 358 .LONG CONVSTR1
000000DF' 0046 359 .LONG CONVSTR2
0000010B' 004A 360 .LONG CONVSTR3
004E004E 361
004E004E 362 QUE_STR_TBL: ; table of headers for queue display
00000141' 004E 363 .LONG GRANTSTR
0000016B' 0052 364 .LONG CONVSTR
0000019B' 0056 365 .LONG WAITSTR
005A005A 366
21 20 20 20 20 20 00000062' 010E0000' 005A 367 LOCKSTR1: .ASCID # !XL !AC#
43 41 21 20 20 4C 58 0068
21 20 20 20 20 20 00000077' 010E0000' 006F 368 LOCKSTR2: .ASCID # !XL !AC !XL !AC#
20 20 20 20 20 43 41 21 20 20 4C 58 007D
43 41 21 20 20 4C 58 21 20 20 0089
21 20 20 20 20 20 0000009B' 010E0000' 0093 369 LOCKSTR3: .ASCID # !XL !AC !XL !AC !XL !AC#
20 20 20 20 20 43 41 21 20 20 4C 58 00A1
20 20 43 41 21 20 20 4C 58 21 20 20 00AD
41 21 20 20 4C 58 21 20 20 20 20 20 00B9
43 00C5
00C6 370
21 20 20 20 20 20 000000CE' 010E0000' 00C6 371 CONVSTR1: .ASCID # !XL !AC/!AC#
43 41 21 2F 43 41 21 20 20 4C 58 00D4
21 20 20 20 20 20 000000E7' 010E0000' 00DF 372 CONVSTR2: .ASCID # !XL !AC/!AC !XL !AC/!AC#
20 43 41 21 2F 43 41 21 20 20 4C 58 00ED
2F 43 41 21 20 20 4C 58 2' 20 20 20 00F9
43 41 21 0105
21 20 20 20 20 20 00000110' 010E0000' 0108 373 CONVSTR3: .ASCID # !XL !AC/!AC !XL !AC/!AC !XL !AC/!AC#
20 43 41 21 2F 43 41 21 20 20 4C 58 0116
2F 43 41 21 20 20 4C 58 21 20 20 20 0122
20 20 4C 58 21 20 20 20 20 43 41 21 012E
43 41 21 2F 43 41 21 013A
0141 374
65 74 6E 61 72 47 00000149' 010E0000' 0141 375 GRANTSTR: .ASCID #Granted queue (Lock ID / Gr mode):#
63 6F 4C 28 20 65 75 65 75 71 20 64 014F
6F 6D 20 72 47 20 2F 20 44 49 20 6B 015B
3A 29 65 64 0167
72 65 76 6E 6F 43 00000173' 010E0000' 0168 376 CONVSTR: .ASCID #Conversion queue (Lock ID / Gr/Rq mode):#
28 20 65 75 65 75 71 20 6E 6F 69 73 0179
72 47 20 2F 20 44 49 20 6B 63 6F 4C 0185
3A 29 65 64 6F 6D 20 71 52 2F 0191
6E 69 74 69 61 57 000001A3' 010E0000' 019B 377 WAITSTR: .ASCID #Waiting queue (Lock ID / Rq mode):#

```

```

63 6F 4C 28 20 65 75 65 75 71 20 67 01A9
6F 6D 20 71 52 20 2F 20 44 49 20 6B 01B5
      3A 29 65 64 01C1
      01C5 378
      01C5 379
      01C5 380
      01C5 381
      01C5 382 LOCKMODE_TBL:
      4C 4E 01C5 383 .ASCII 'NL'
      52 43 01C7 384 .ASCII 'CR'
      57 43 01C9 385 .ASCII 'CW'
      52 50 01CB 386 .ASCII 'PR'
      57 50 01CD 387 .ASCII 'PW'
      58 45 01CF 388 .ASCII 'EX'
      01D1 389
000001E9' 01D1 390 LKMODE_TBL: .ADDRESS NULL
000001EC' 01D5 391 .ADDRESS CREAD ; POINTERS TO MODE TEXT
0C0001EF' 01D9 392 .ADDRESS CWRITE
000001F2' 01DD 393 .ADDRESS PREAD
000001F5' 01E1 394 .ADDRESS PWRITE
000001F8' 01E5 395 .ADDRESS EX
4C 4E 00' 01E9 396 NULL: .ASCIC /NL/
      02 01E9
52 43 00' 01EC 397 CREAD: .ASCIC /CR/
      02 01EC
57 43 00' 01EF 398 CWRITE: .ASCIC /CW/
      02 01EF
52 50 00' 01F2 399 PREAD: .ASCIC /PR/
      02 01F2
57 50 00' 01F5 400 PWRITE: .ASCIC /PW/
      02 01F5
58 45 00' 01F8 401 EX: .ASCIC /EX/
      02 01F8
      01FB 402
      01FB 403 GR_STRING:
74 61 20 64 65 74 6E 61 72 47 00' 01FB 404 .ASCIC 'Granted at'
      0A 01FB
      0206 405 CV_STRING:
20 67 6E 69 74 72 65 76 6E 6F 43 00' 0206 406 .ASCIC 'Converting to'
      6F 74 0212
      0D 0206
      0214 407 WT_STRING:
72 6F 66 20 67 6E 69 74 69 61 57 00' 0214 408 .ASCIC 'Waiting for'
      0B 0214
      0220 409
      20 20 0220 410 BLANKS: .ASCII '' ''
      0222 411
      00' 0222 412 NULL_CSTRING:
      00 0222 413 .ASCIC ''''
      0223 414
      54 53 41 4B 4C 42 00' 0223 415 BL_STRING:
      06 0223 416 .ASCIC 'BLKAST'
      022A 417
      20 4B 4C 42 4C 41 56 00' 022A 418 FLAGS_TBL:
      022A 419 .ASCIC 'VALBLK ''

```

54	52	45	56	4E	4F	43	00'	07	022A	420	.ASCIC	"CONVERT"	
45	55	45	55	51	4F	4E	00'	07	0232	421	.ASCIC	"NOQUEUE"	
53	54	53	43	4E	59	53	00'	07	023A	422	.ASCIC	"SYNCSTS"	
20	4D	45	54	53	59	53	00'	07	0242	423	.ASCIC	"SYSTEM "	
41	54	4F	55	51	4F	4E	00'	07	024A	424	.ASCIC	"NOQUOTA"	
20	53	59	53	54	56	43	00'	07	0252	425	.ASCIC	"CVTSYS "	
52	45	56	4F	43	45	52	00'	07	025A	426	.ASCIC	"RECOVER"	
54	43	45	54	4F	52	50	00'	07	0262	427	.ASCIC	"PROTECT"	
									026A	428			
54	53	41	4C	50	43	44	00'	07	0272	429	STATUS_TBL:	"DCPLAST"	
54	53	41	4B	4C	42	44	00'	07	027A	430	.ASCIC	"DBLKAST"	
20	20	43	4E	59	53	41	00'	07	027A	431	.ASCIC	"ASYNC "	
44	51	54	53	41	4C	42	00'	07	0282	432	.ASCIC	"BLASTQD"	
20	59	50	43	54	53	4D	00'	07	028A	433	.ASCIC	"MSTCPY "	
41	54	4F	55	51	4F	4E	00'	07	0292	434	.ASCIC	"NOQUOTA"	
51	54	55	4F	4D	49	54	00'	07	029A	435	.ASCIC	"TIMOUTQ"	
20	53	59	53	53	41	57	00'	07	02A2	436	.ASCIC	"WASSYS "	
53	59	53	4F	54	56	43	00'	07	02AA	437	.ASCIC	"CVTOSYS"	
									02B2	438			
									02BA	439			
	6C	61	63	6F	4C		00'	05	02BA	440	LOCAL:	"Local"	
									02C0	441	PROCESS:		
73	73	65	63	6F	72	50	00'	07	02C0	442	.ASCIC	"Process"	
									02C8	443	MASTER:	"Master:"	
									02CF	444			
							20	00'	02CF	445	SPACE:	" "	
									02D1	446			
									02D1	447	NOT_VALID:		
64	69	6C	61	76	20	74	6F	4E	00'	09	.ASCIC	"Not valid"	
									02DB	449	DIR_ENTRY:		
65	20	79	72	6F	74	63	65	72	69	44	00'	.ASCIC	"Directory entry"
								79	72	74	6E		
									02EB	450			
									02EB	451			

LOCK  
V04-000

LOCK AND RESOURCE FORMATTING ROUTINES  
READ-ONLY DATA DEFINITIONS

M 5

16-SEP-1984 01:31:38  
5-SEP-1984 03:32:46

VAX/VMS Macro V04-00  
[SDA.SRC]LOCK.MAR;1

Page 12  
(5)

LOC  
V04

65 64 6F 6D 20 6C 65 6E 72 65 4B 00'	02EB	452	KMODE:	.ASCIC	'Kernel mode'
0B	02EB				
65 64 6F 6D 20 2E 63 65 78 45 00'	02F7	453	EMODE:	.ASCIC	'Exec. mode'
0A	02F7				
65 64 6F 6D 20 2E 72 65 70 75 53 00'	0302	454	SMODE:	.ASCIC	'Super. mode'
0B	0302				
65 64 6F 6D 20 72 65 73 55 00'	030E	455	UMODE:	.ASCIC	'User mode'
09	030E				
	0318	456			
	0318	457			
	0318	458	.ALIGN	LONG	
	0318	459	ACMODE_TBL:		
000002EB'	0318	459	.ADDRESS		KMODE
000002F7'	031C	460	.ADDRESS		EMODE
00000302'	0320	461	.ADDRESS		SMODE
0000030E'	0324	462	.ADDRESS		UMODE
	0328	463			

```

0328 465 .SBTTL SHOW_ALL_LOCKS - ACTION ROUTINE TO DISPLAY ALL LOCKS
0328 466 :+++
0328 467 :
0328 468 SHOW_ALL_LOCKS
0328 469 :
0328 470 PURPOSE
0328 471 ACTION ROUTINE FOR THE "SHOW LOCKS/ALL" COMMAND
0328 472 :
0328 473 INPUT
0328 474 NONE
0328 475 :
0328 476 OUTPUT
0328 477 DISPLAYED DATA
0328 478 :
0328 479 :---
0000 0328 480 .ENTRY SHOW_ALL_LOCKS, ^M<>
032A 481
6D 000009E8'EF 9E 032A 482 MOVAB LOCK_COND_HAND, (FP) ; ESTABLISH CONDITION HANDLER
00000090'EF D4 0331 483 CLRL LOCKID ; INITIALIZE VARIABLE
FCC5 CF 9F 0337 484 PUSHAB LOCK_HEAD ; GET HEADER
00000000'EF 01 FB 033B 485 CALLS #1, SET_HEADING ; SET IT UP
00000000'EF D4 0342 486 CLRL LOCK_COUNT ; INIT COUNT OF LOCKS PROCESSED
0348 487
0348 488 :
0348 489 : STEP THROUGH LOCK ID TABLE, DISPLAYING ENTRIES WITH DATA
0348 490 :
00000090'EF D6 0348 491 10$: INCL LOCKID ; INCREMENT INDEX
00000443'EF 00 FB 034E 492 CALLS #0, GET_LKB ; GET ADDRESS OF ASSOCIATED LKB
FO 50 E9 0355 493 BLBC R0, 10$ ; IF LBC, NON-EXISTENT, TRY NEXT
00000000'EF D6 0358 494 INCL LOCK_COUNT ; GOT ONE
00000487'EF 00 FB 035E 495 CALLS #0, SAVE_LOCK_DATA ; GET ASSOCIATED DATA
0000051D'EF 00 FB 0365 496 CALLS #0, DISPLAY_LOCK ; DISPLAY DATA FOR THIS LOCKID
DA 11 036C 497 BRB 10$ ; LOOP (SIGNAL WILL EXIT FOR US)
    
```



```

036E 499 .SBTTL SHOW_ONE_LOCK - ACTION ROUTINE TO DISPLAY ONE LOCK
036E 500 :+++
036E 501 :
036E 502 SHOW_ONE_LOCK
036E 503 :
036E 504 PURPOSE
036E 505 ACTION ROUTINE TO PROCESS "SHOW LOCK lockid" COMMAND
036E 506 :
036E 507 INPUT
036E 508 LOCKID - INDEX INTO LOCK ID TABLE
036E 509 :
036E 510 OUTPUT
036E 511 DISPLAYED DATA
036E 512 :
036E 513 :---
0000 036E 514 .ENTRY SHOW_ONE_LOCK,^M<>
0370 515
00000443'EF 00 FB 0370 516 CALLS #0,GET_LKB ; GET LOCK BLOCK ADDRESS
21 50 E9 0377 517 BLBC RO,10$ ; IF LBC, NO LOCK BLOCK FOR ID
037A 518
000004B7'EF 00 FB 037A 519 CALLS #0,SAVE_LOCK_DATA ; SAVE LOCK DATA
FC7B CF 9F 0381 520 PUSHAB LOCK_HEAD ; SET UP LOCK HEADING
00000000'EF 01 FB 0385 521 CALLS #1,SET_HEADING
0000051D'EF 00 FB 038C 522 CALLS #0,DISPLAY_LOCK ; DISPLAY LOCK INFO
0393 523 STATUS SUCCESS ; SUCCESSFUL COMPLETION
04 039A 524 RET
039B 525
04 039B 526 10$: SIGNAL 0,NOLKB ; NO LOCK BLOCK FOR THIS ID
03AD 527 RET

```

```

03AE 529 .SBTTL SHOW_PROC_LOCK - SHOW LOCKS ASSOCIATED WITH A PROCESS
03AE 530 :+++
03AE 531 :
03AE 532 : SHOW_PROC_LOCK
03AE 533 :
03AE 534 : PURPOSE
03AE 535 : SHOW LOCKS ASSOCIATED WITH A GIVEN LOCK. CALLED FROM
03AE 536 : GENERIC CODE WHICH DISPLAYS PROCESS DATA.
03AE 537 :
03AE 538 : INPUT
03AE 539 : 4(AP) - ADDRESS OF PCB COPY IN SDA IMAGE
03AE 540 : 8(AP) - ADDRESS OF 'REAL' PCB IN SYSTEM
03AE 541 :
03AE 542 : OUTPUT
03AE 543 : DISPLAYED DATA
03AE 544 :---
000C 03AE 545 .ENTRY SHOW_PROC_LOCK,^M<R2,R3>
03B0 546
    53 04 AC D0 03B0 547 MOVL 4(AP),R3 ; GET ADDRESS OF PCB SAVE AREA
    53 0104 C3 D0 03B4 548 MOVL PCB$LOCKQFL(R3),R3 ; GET FORWARD LINK
    52 08 BC D0 03B9 549 MOVL @8(AP),R2 ; GET 'REAL' ADDRESS OF PCB
    52 0104 C2 DE 03BD 550 MOVAL PCB$LOCKQFL(R2),R2 ; GET 'REAL' ADDRESS OF HEAD OF QUEUE
    52 53 D1 03C2 551 CMPL R3,R2 ; ARE THERE ELEMENTS IN THIS QUEUE?
    69 13 03C5 552 BEQL NOLCK ; IF EQL, NO
03C7 553
000002BF'EF 53 00000040 8F C3 03C7 554 GETLCK: SUBL3 #LKB$LOWNQFL,R3,LKB ; ADJUST FOR START OF LKB
000004B7'EF 00 FB 03D3 555 CALLS #0,SAVE_LOCK_DATA ; SAVE LOCK DATA
03DA 556 SKIP PAGE
03E1 557 PRINT 0,<Lock data:>
00000556'EF 00 FB 03EE 558 CALLS #0,FORMAT_LOCK ; FORMAT LOCK DATA
03F5 559 ENSURE 9
040D 560 SKIP 1
00000856'EF 00 FB 0416 561 CALLS #0,PRINT_LOCK ; DISPLAY LOCK INFO
    52 53 D1 041D 562 REQMEM (R3),R3 ; GET NEXT QUEUE LINK
    14 13 0429 563 CMPL R3,R2 ; AT END OF QUEUE?
    97 11 042C 564 BEQL DONE ; IF EQL, DONE
    04 0442 565 BRB GETLCK ; LOOP
    0430 566
    0430 567 NOLCK: ; THIS PROCESS HAS NO LOCKS
    0430 568 SIGNAL 0,NOPRLOCK
    0442 569
    04 0442 570 DONE: RET ; SUCCESSFUL
    
```

```

0443 572 .SBTTL GET_LKB - GET LOCK BLOCK
0443 573 :+++
0443 574 :
0443 575 GET_LKB
0443 576 :
0443 577 PURPOSE
0443 578 GIVEN LOCK ID, GET ADDRESS OF LOCK BLOCK.
0443 579 :
0443 580 INPUT
0443 581 LOCKID - INDEX INTO LOCK ID TABLE
0443 582 :
0443 583 OUTPUT
0443 584 R0 - IF LBS, LKB CONTAINS ADDRESS OF LOCK BLOCK
0443 585 IF LBC, NO LOCK BLOCK FOR THIS ID
0443 586 :
0443 587 :---
000C 0443 588 .ENTRY GET_LKB,^M<R2,R3>
0445 589 :
0445 590 :
0445 591 VALIDATE INPUT
0445 592 :
52 00000090'EF 3C 0445 593 MOVZWL LOCKID,R2 ; MOVE TO REGISTER AND TEST
43 13 044C 594 BEQL 20$ ; IF EQL, NOT VALID
53 52 D1 044E 595 REQMEM @LCK$GL_MAXID,R3 ; GET MAX ID VALUE
41 1A 045E 596 CML R2,R3 ; CHECK IT
0461 597 BGTRU 30$ ; IF GTRU, TOO BIG
0463 598 :
0463 599 :
0463 600 GET ADDRESS OF LOCK BLOCK
0463 601 :
0463 602 :
53 6342 DE 0473 603 REQMEM @LCK$GL_IDTBL,R3 ; GET START OF TABLE
0477 604 MOVAL (R3)[R2],R3 ; CALC SLOT ADDRESS
0483 605 REQMEM (R3),R3 ; GET LKB ADDRESS
000002BF'EF 50 D4 0483 605 CLRL R0 ; ASSUME EMPTY SLOT
53 D0 0485 606 MOVL R3,LKB ; MOVE DATA AND TEST
02 18 048C 607 BGEQ 10$ ; IF GEQ, NO LKB FOR THIS ID
50 D6 048E 608 INCL R0 ; INDICATE SUCCESS
0490 609 :
04 0490 610 10$: RET ; DONE
0491 611 :
04 0491 612 20$: SIGNAL 0,LOCKIDZER ; LOCK ID 0
04A3 613 RET
04A4 614 :
04 04A4 615 30$: SIGNAL 0,OUTOFRANG ; LOCK ID GREATER THAN MAXID
0486 616 RET

```

```

04B7 618      .SBTTL SAVE_LOCK_DATA - RETRIEVE LOCK DATA FROM SYSTEM
04B7 619      :+++
04B7 620      :
04B7 621      :   SAVE_LOCK_DATA
04B7 622      :
04B7 623      :   PURPOSE
04B7 624      :   GIVEN ADDRESS OF LOCK BLOCK, RETRIEVE ALL DATA NEEDED TO
04B7 625      :   DISPLAY LOCK INFORMATION
04B7 626      :
04B7 627      :   INPUT
04B7 628      :   LKB - ADDRESS OF LOCK BLOCK
04B7 629      :
04B7 630      :   OUTPUT
04B7 631      :   LKB_BFR      - CONTENTS OF LOCK BLOCK
04B7 632      :   PARID_BFR     - ID OF PARENT LOCK
04B7 633      :   LKB_RSB_BFR  - DATA FROM ASSOCIATED RESOURCE
04B7 634      :
04B7 635      :---
000C 04B7 636      .ENTRY SAVE_LOCK_DATA,^M<R2,R3>
04B9 637
52   000002BF'EF  D0 04B9 638      MOVL   LKB,R2
04C0 639      REQMEM (R2),LKB_BFR,#LKBSK_LENGTH      ; STORE LKB DATA
53   00000050'EF  D0 04D5 640      MOVL   LKB_BFR+CKBSL_PARENT,R3      ; PARENT LKB ADDR
04DC 641      BEQL   10$      ; IF EQL, NO PARENT
04DE 642      REQMEM LKBSL_LKID(R3),R3      ; FETCH PARENT LOCK ID
04EB 643
00000004'EF  53  D0 04EB 644 10$: MOVL   R3,PARID_BFR      ; STORE IN BUFFER
53   00000058'EF  D0 04F2 645      MOVL   LKB_BFR+CKBSL_RSB,R3      ; GET ASSOCIATED RSB
04F9 646
04F9 647      ASSUME  RSB$B_RMOD-RSB$W_GROUP  EQ 2      ; MAKE SURE NOTHING CHANGES
04F9 648      ASSUME  RSB$B_RSNLEN-RSB$W_GROUP EQ 3
04F9 649      ASSUME  RSB$T_RESNAM-RSB$W_GROUP EQ 4
04F9 650      ASSUME  RSB$K_MAXLEN EQ <^X1F>
04F9 651
04F9 652      REQMEM  RSB$W_GROUP(R3),LKB_RSB_BFR,-      ; STORE RSB DATA
04F9 653      #RSB$K_MAXLEN+4
050B 654      REQMEM  RSB$S_CSID(R3),RSBCSID_BFR      ; GET CSID IN RSB
051C 655
04   051C 656      RET
    
```

```
051D 658 .SBTTL DISPLAY_LOCK - CONTROL DISPLAY OF LOCK DATA
051D 659 :+++
051D 660 :
051D 661 : DISPLAY_LOCK
051D 662 :
051D 663 : PURPOSE
051D 664 : CONTROLS FORMAT AND DISPLAY OF LOCK DATA
051D 665 :
051D 666 : INPUT
051D 667 : LOCK DATA AREAS (IMPLIED)
051D 668 :
051D 669 : OUTPUT
051D 670 : DISPLAYED LOCK INFORMATION
051D 671 :
051D 672 :---
0000 051D 673 .ENTRY DISPLAY_LOCK,^M<>
051F 674
00000556'EF 00 FB 051F 675 CALLS #0,FORMAT_LOCK ; FORMAT LOCK DATA
0526 676 SKIP PAGE
052D 677 ENSURE 10
0545 678 SKIP 1
00000856'EF 00 FB 054E 679 CALLS #0,PRINT_LOCK ; DISPLAY LOCK DATA
0555 680
04 0555 681 RET
```

```

0556 683 .SBTTL FORMAT_LOCK - FORMAT LOCK DATA
0556 684 :+++
0556 685 :
0556 686 :   FORMAT_LOCK
0556 687 :
0556 688 :   PURPOSE
0556 689 :   FORMAT RAW SYSTEM LOCK AND RESOURCE DATA. STORE
0556 690 :   FOR EASY ACCESS.
0556 691 :
0556 692 :   INPUT
0556 693 :   LKB_BFR - ADDRESS OF LOCK BLOCK DATA
0556 694 :   PARID_BFR - PARENT LOCK ID
0556 695 :   LKB_RSB_BFR - ASSOCIATED RESOURCE DATA
0556 696 :
0556 697 :   OUTPUT
0556 698 :   STORED DATA
0556 699 :
0556 700 :---
0556 701 :
007C 0556 702 .ENTRY FORMAT_LOCK,^M<R2,R3,R4,R5,R6>
0558 703
0558 704 ; Lock id., parent id., sublocks and PID
0558 705
0558 706 MOVAB LKB_BFR,R6
055F 707 MOVL LKB$$_LKID(R6),LKID
0567 708 MOVL PARID_BFR,PARID
0572 709 MOVZWL LKB$_REFCNT(R6),SUBLKS
057A 710 MOVL LKB$_PID(R6),PID
0582 711
0582 712 ; Lock state and mode(s)
0582 713
0582 714 TSTB LKB$_STATE(R6)
0585 715 BLSS 20$ ; Waiting
0587 716 BGTR 10$ ; Granted
0589 717 MOVAB CV_STRING,STATE1
0592 718 MOVAB GR_STRING,STATE2
0598 719 MOVZBL LKB$_GRMODE(R6),R0 ; Get granted mode
059F 720 MOVAB LOCKMODE_TBL[R0],MODE2 ; Store granted mode sting
05A9 721 MOVZBL LKB$_RQMODE(R6),R0 ; Get requested mode
05AD 722 BRB 30$
05AF 723 10$: MOVAB GR_STRING,STATE1
05B8 724 MOVZBL LKB$_GRMODE(R6),R0
05BC 725 BRB 25$
05BE 726 20$: MOVAB WT_STRING,STATE1
05C7 727 MOVZBL LKB$_RQMODE(R6),R0
05CB 728 25$: MOVAB NULL_CSTRING,STATE2 ; No state 2
05D4 729 MOVAB BLANKS,MODE2 ; No mode 2
05DD 730 30$: MOVAB LOCKMODE_TBL[R0],MODE1 ; Store mode 1
05E7 731
05E7 732 ; Blocking AST
05E7 733
05E7 734 MOVAB NULL_CSTRING,BLKAST ; Assume no blocking AST
05F0 735 TSTL LKB$_BLKASTADR(R6) ; Is there a blocking AST?
05F3 736 BEQL 32$
05F5 737 MOVAB BL_STRING,BLKAST ; Yes, store address of string
05FE 738
05FE 739 ; Flags

```

```

    53 FC20 CF 9E 05FE 740
00000277'EF 53 D0 05FE 741 32$: MOVAB NULL,CSTRING,R3 ; Initialize fields
0000027B'EF 53 D0 0603 742 MOVL R3,FLAGS1
0000027F'EF 53 D0 060A 743 MOVL R3,FLAGS1+4
00000293'EF 53 D0 0611 744 MOVL R3,FLAGS1+8
00000297'EF 53 D0 0618 745 MOVL R3,FLAGS2
0000029B'EF 53 D0 061F 746 MOVL R3,FLAGS2+4
000002AF'EF 53 D0 0626 747 MOVL R3,FLAGS2+8
000002B3'EF 53 D0 062D 748 MOVL R3,FLAGS3
000002B7'EF 53 D0 0634 749 MOVL R3,FLAGS3+4
000002B7'EF 53 D0 063B 750 MOVL R3,FLAGS3+8
0642 751
    55 28 A6 3C 0642 752 MOVZWL LKBSW_FLAGS(R6),R5 ; Pick up flags
54 00000277'EF DE 0646 753 MOVAL FLAGST,R4 ; Address of 1st arg. list
    52 D4 064D 754 CLRL R2
    53 D4 064F 755 CLRL R3
    64 25 55 53 E1 0651 756 35$: BBC R3,R5,60$ ; Br. if flag not set
    FBDO CF43 7E 0655 757 MOVAQ FLAGS_TBL[R3],(R4) ; Store appropriate string address
    54 54 04 C0 065B 758 ADDL #4,R4
    03 52 D6 065E 759 INCL R2 ; Incr. position counter
    09 12 D1 0660 760 CMPL R2,#3 ; Move to 2nd line?
54 00000293'EF DE 0663 761 BNEQ 40$ ; No
    OC 11 D1 0665 762 MOVAL FLAGS2,R4 ; Yes
    06 52 D1 0666 763 BRB 60$
54 000002AF'EF DE 066E 764 40$: CMPL R2,#6 ; Move to third line
    D3 53 09 12 0671 765 BNEQ 60$ ; No
    07 12 D1 0673 766 MOVAL FLAGS3,R4 ; Yes
    09 F2 067A 767 60$: AOBLSS #9,R3,35$ ; Repeat 9 times
    067E 768
    067E 769
    067E 770 PART2:
    067E 771 ; Status
    067E 772
    067E 773
    53 FBA0 CF 9E 067E 774 MOVAB NULL,CSTRING,R3 ; Initialize fields
000002D3'EF 53 D0 0683 775 MOVL R3,STS1
000002D7'EF 53 D0 068A 776 MOVL R3,STS1+4
000002DB'EF 53 D0 0691 777 MOVL R3,STS1+8
000002F3'EF 53 D0 0698 778 MOVL R3,STS2
000002F7'EF 53 D0 069F 779 MOVL R3,STS2+4
000002FB'EF 53 D0 06A6 780 MOVL R3,STS2+8
00000313'EF 53 D0 06AD 781 MOVL R3,STS3
00000317'EF 53 D0 06B4 782 MOVL R3,STS3+4
0000031B'EF 53 D0 06BB 783 MOVL R3,STS3+8
    06C2 784
    55 2A A6 3C 06C2 785 MOVZWL LKBSW_STATUS(R6),R5 ; Pick up status
54 000002D3'EF DE 06C6 786 MOVAL STS1,R4 ; Address of 1st arg. list
    52 D4 06CD 787 CLRL R2
    53 D4 06CF 788 CLRL R3
    64 25 55 53 E1 06D1 789 35$: BBC R3,R5,60$ ; Br. if flag not set
    FB98 CF43 7E 06D5 790 MOVAQ STATUS_TBL[R3],(R4) ; Store appropriate string address
    54 04 C0 06DB 791 ADDL #4,R4
    03 52 D6 06DE 792 INCL R2 ; Incr. position counter
    09 12 D1 06E0 793 CMPL R2,#3 ; Move to 2nd line?
54 000002F3'EF DE 06E3 794 BNEQ 40$ ; No
    OC 11 D1 06E5 795 MOVAL STS2,R4 ; Yes
    06EC 796 BRB 60$

```

```

      06 52 D1 06EE 797 40$:  CML  R2,#6           ; Move to third line
      07 12 06F1 798       BNEQ 60$           ; No
54 00000313'EF 07 DE 06F3 799       MOVAL STS3,R4       ; Yes
      D3 53 09 F2 06FA 800 60$: AOBLS #9,R3,35$    ; Repeat 9 times
      06FE 801
      06FE 802
      06FE 803
      06FE 804
      06FE 805
000002DF'EF 000006F'EF 9A 06FE 806       MOVZBL RSNLEN_BFR,RLEN
      51 000006F'EF 9A 0709 807       MOVZBL RSNLEN_BFR,R1
      50 0000070'EF 9E 0710 808       MOVAB  RESNAM_BFR,R0
      50 DD 0717 809       PUSHL  R0           ; Save R0
      60 20 00 60 51 2C 0719 810       MOVCS  R1,(R0),#0,#32,(R0) ; Zero out rest of buffer
      50 8ED0 071F 811       POPL   R0           ; Restore R0
      000002CF'EF 60 DE 0722 812       MOVAL  (R0),DMP1
      000002C7'EF 80 DO 0729 813       MOVL  (R0)+,RES1
      000002C3'EF 80 DO 0730 814       MOVL  (R0)+,RES2
      000002EF'EF 60 DE 0737 815       MOVAL  (R0),DMP3
      000002E7'EF 80 DO 073E 816       MOVL  (R0)+,RES3
      000002E3'EF 80 DO 0745 817       MOVL  (R0)+,RES4
      0000030F'EF 60 DE 074C 818       MOVAL  (R0),DMP5
      00000307'EF 80 DO 0753 819       MOVL  (R0)+,RES5
      00000303'EF 80 DO 075A 820       MOVL  (R0)+,RES6
      0000032F'EF 60 DE 0761 821       MOVAL  (R0),DMP7
      00000327'EF 80 DO 0768 822       MOVL  (R0)+,RES7
      00000323'EF 80 DO 076F 823       MOVL  (R0)+,RES8
      50 000006E'EF 9A 0776 824       MOVZBL RMOD_BFR,R0           ; Get access mode
000002FF'EF FB96 CF40 DO 077D 825       MOVL  ACMODE_TBL[R0],ACMODE
      000C01BA'EF B5 0787 826       TSTW  GROUP_BUF       ; System names have group = 0
      44 13 078D 827       BEQL  70$           ; Branch if system name
      078F 828
      078F 829
      078F 830
      078F 831
      03 50 E8 07AA 831       BLBS  R0,65$
      00A5 31 07AD 832       BRW   90$
000001EB'EF 000001BA'EF B0 07B0 833 65$:  MOVW  GROUP_BUF,GROUP_NUM
000001EA'EF 000001BC'EF 90 07BB 834       MOVB  GROUP_BUF+2,GROUP_NUM+2
0000031F'EF 000001D8'EF 7E 07C6 835       MOVAQ GROUP_DSC,NSPACE
      0B 11 07D1 836       BRB   80$
0000031F'EF 000001C8'EF 7E 07D3 837 70$:  MOVAQ SYS_DSC,NSPACE
      07DE 838
      07DE 839 80$: ; Format type of lock and remote lock info.
      07DE 840
      50 0000068'EF DO 07DE 841       MOVL  RSBCSID_BFR,R0       ; Get CSID
      19 12 07E5 842       BNEQ 82$           ; Must be process copy
      1F 2A A6 04 E0 07E7 843       BBS  #LKBSV_MSTCPY,LKBSW_STATUS(R6),84$ ; Branch if master copy
00000333'EF FACA CF 9E 07EC 844       MOVAB LOCAL_TYPE       ; Set type of lock
00000337'EF FAD6 CF 9E 07F5 845       MOVAB SPACE,RMINFO    ; Display no remote info
      52 11 07FE 846       BRB   88$
      0800 847
00000333'EF FABC CF 9E 0800 848 82$:  MOVAB PROCESS_TYPE     ; Set type of lock
      OD 11 0809 849       BRB   86$
      080B 850
00000333'EF FAB9 CF 9E 080B 851 84$:  MOVAB MASTER_TYPE     ; Set type of lock
      50 58 A6 DO 0814 852       MOVL  LKBSL_CSID(R6),R0 ; Get CSID
      0818 853

```



```
51 54 A6 D0 0818 854 86$: MOVL LKBSL REMLKID(R6),R1 ; Get remote lock id.
081C 855 $FAO_S CTRSTR = FAO RMINFO_DSC,-
081C 856 OUTBUF = RMINFO_DSC,-
081C 857 OUTLEN = RMINFO_LEN,-
081C 858 P1 = R1,-
081C 859 P2 = R0,-
19 50 E9 0839 860 BLBC R0,90$
000000C4'EF 000000B8'EF 90 083C 861 MOVB RMINFO_LEN,RMINFO_CNT
00000337'EF 000000C4'EF 9E 0847 862 MOVAB RMINFO_CNT,RMINFO_CNT ; Store ptr to .ASCII string
0852 863
50 01 D0 0852 864 88$: MOVL #1,R0
04 0855 865 90$: RET
```

```

0856 867 .SBTTL PRINT_LOCK - OUTPUT LOCK DATA
0856 868 :+++
0856 869 :
0856 870 : PRINT_LOCK
0856 871 :
0856 872 : PURPOSE
0856 873 : OUTPUT FORMATTED LOCK DATA, LINE-BY-LINE.
0856 874 :
0856 875 : INPUT
0856 876 : FORMATTED LOCK DATA
0856 877 :
0856 878 : OUTPUT
0856 879 : DISPLAYED DATA
0856 880 :
0856 881 : ---
0000 0856 882 .ENTRY PRINT_LOCK,^M<>
0858 883 :
0858 884 :
0858 885 : LINE 1
0858 886 :
0000027F'EF DD 0858 887 PUSHL  FLAGS1+8
0000027B'EF DD 085E 888 PUSHL  FLAGS1+4
00000277'EF DD 0864 889 PUSHL  FLAGS1
00000273'EF DD 086A 890 PUSHL  PID
0000026F'EF DD 0870 891 PUSHL  LKID
0876 892 PRINT 5,<Lock id: !XL PID: !YL Flags: !3(8AC)>
0883 893 : LKID PID
0883 894 :
0883 895 :
0883 896 : LINE 2
0883 897 :
0000029B'EF DD 0883 898 PUSHL  FLAGS2+8
00000297'EF DD 0889 899 PUSHL  FLAGS2+4
00000293'EF DD 088F 900 PUSHL  FLAGS2
0000028F'EF DD 0895 901 PUSHL  MODE1
00000287'EF DD 089B 902 PUSHL  #2
00000283'EF DD 089D 903 PUSHL  STATE1
08A3 904 PUSHL  PARID
08A9 905 PRINT 7,<Par. id: !XL !13AC !AD!12* !3(8AC)>
08B6 906 : PARID STATE1 MODE1 FLAGS2
08B6 907 :
08B6 908 :
08B6 909 : LINE 3
08B6 910 :
000002B7'EF DD 08B6 911 PUSHL  FLAGS3+8
000002B3'EF DD 08BC 912 PUSHL  FLAGS3+4
000002AF'EF DD 08C2 913 PUSHL  FLAGS3
000002AB'EF DD 08C8 914 PUSHL  MODE2
000002A7'EF DD 08CE 915 PUSHL  #2
000002A3'EF DD 08D0 916 PUSHL  STATE2
0000029F'EF DD 08D6 917 PUSHL  SUBLKS
08DC 918 PRINT 7,<Sublocks: !7UL !13AC !AD!12* !3(8AC)>
08E9 919 : SUBLKS STATE2 MODE2 FLAGS3
08E9 920 :
08E9 921 :
08E9 922 : LINE 3A
08E9 923 :

```

```

000002BB'EF DD 08E9 924 PUSHL BLKAST
000002BF'EF DD 08EF 925 PUSHL LKB
08F5 926 PRINT 2,<LKB: !XL !9AC>
0902 927 ; LKB BLKAST
0902 928
0902 929
0902 930 : LINE 4
0902 931 :
000002DB'EF DD 0902 932 PUSHL STS1+8
000002D7'EF DD 0908 933 PUSHL STS1+4
000002D3'EF DD 090E 934 PUSHL STS1
000002CF'EF DD 0914 935 PUSHL DMP1
08 DD 091A 936 PUSHL #8
000002C7'EF DD 091C 937 PUSHL RES1
000002C3'EF DD 0922 938 PUSHL RES2
0928 939 PRINT 7,<Resource: !YL !XL !AF Status: !3(8AC)>
0935 940 ; RES2 RES1 DMP1 STS1
0935 941
0935 942 : LINE 5
0935 943 :
0935 944 :
000002FB'EF DD 0935 945 PUSHL STS2+8
000002F7'EF DD 093B 946 PUSHL STS2+4
000002F3'EF DD 0941 947 PUSHL STS2
000002EF'EF DD 0947 948 PUSHL DMP3
08 DD 094D 949 PUSHL #8
000002E7'EF DD 094F 950 PUSHL RES3
000002E3'EF DD 0955 951 PUSHL RES4
000002DF'EF DD 095B 952 PUSHL RLEN
0961 953 PRINT 8,< Length !27B !XL !XL !AF !3(8AC)>
096E 954 ; RLEN RES4 RES3 DMP3 STS?
096E 955
096E 956 : LINE 6
096E 957 :
096E 958 :
0000031B'EF DD 096E 959 PUSHL STS3+8
00000317'EF DD 0974 960 PUSHL STS3+4
00000313'EF DD 097A 961 PUSHL STS3
0000030F'EF DD 0980 962 PUSHL DMP5
08 DD 0986 963 PUSHL #8
00000307'EF DD 0988 964 PUSHL RES5
00000303'EF DD 098E 965 PUSHL RES6
000002FF'EF DD 0994 966 PUSHL ACMODE
099A 967 PRINT 8,< !11AC !XL !XL !AF !3(8AC)>
09A7 968 ; ACMODE RES6 RES5 DMP5 STS3
09A7 969
09A7 970 : LINE 7
09A7 971 :
09A7 972 :
0000032F'EF DD 09A7 973 PUSHL DMP7
08 DD 09AD 974 PUSHL #8
00000327'EF DD 09AF 975 PUSHL RES7
00000323'EF DD 09B5 976 PUSHL RES8
0000031F'EF DD 09BB 977 PUSHL NSPACE
09C1 978 PRINT 5,< !11AS !XL !XL !AF>
09CE 979 ; NSPACE RES8 RES7 DMP7
09CE 980

```

```
00000337'EF DD 09CE 981 :  
00000333'EF DD 09CE 982 : LINE 8  
09CE 983 :  
09CE 984 :  
DD 09D4 985 : PUSHL RMINFO  
09DA 986 : PUSHL TYPE  
09E7 987 : PRINT 2,<!AC copy !AC!/>  
04 09E7 988 : TYPE RMINFO  
RET
```

```

09E8 990      .SBTTL LOCK_COND_HAND - CONDITION HANDLER FOR SHOW_ALL_LOCKS
09E8 991      :+++
09E8 992      :
09E8 993      :      LOCK_COND_HAND
09E8 994      :
09E8 995      :      PURPOSE
09E8 996      :      PROVIDE EXIT PATH FOR SHOW_ALL_LOCKS WHEN THERE ARE NO MORE LOCKS
09E8 997      :      TO BE PROCESSED
09E8 998      :
09E8 999      :      INPUT
09E8 1000     :      4(AP) = POINTER TO SIGNAL ARGUMENTS
09E8 1001     :      8(AP) = POINTER TO MECHANISM ARGUMENTS
09E8 1002     :
09E8 1003     :      OUTPUT
09E8 1004     :      POSSIBLE MODIFICATION OF STATUS; POSSIBLE CHANGE IN FLOW
09E8 1005     :      OF CONTROL.
09E8 1006     :
09E8 1007     :      ---
001C 09E8 1008     .ENTRY LOCK_COND_HAND, ^M<R2,R3,R4>
09EA 1009
54   52 04 AC 7D 09EA 1010     MOVQ 4(AP),R2 ; GET ADDRESSES OF ARRAYS
54   54 04 A2 D0 09EE 1011     MOVL CHF$L_SIG_NAME(R2),R4 ; GET CONDITION NAME
54   00000000'8F D1 09F2 1012     CMPL #SS$_ONWIND,R4 ; ARE WE UNWINDING?
54   00000000'8F D1 09F9 1013     BEQL 20$ ; IF EQL, YES
54   00000000'8F D1 09FB 1014     CMPL #MSG$_OUTOFRANG,R4 ; DID WE GET THE SIGNAL WE'RE LOOKING FOR?
54   00000000'8F D1 0A02 1015     BNEQ 20$ ; IF NEQ, NO - PASS IT ON
54   00000000'8F D1 0A04 1016
54   00000000'EF D5 0A04 1017     TSTL LOCK_COUNT ; DID WE PROCESS LOCKS?
54   00000000'EF D5 0A0A 1018     BNEQ 10$ ; IF NEQ, YES
54   00000000'EF D5 0A0C 1019     SIGNAL 0,NOLOCKS ; DISPLAY INFO MESSAGE
54   00000000'EF D5 0A1E 1020
54   0C A3 50 D0 0A1E 1021 10$: MOVL R0,CHF$MCH_SAVR0(R3) ; INDICATE SUCCESS AFTER UNWIND
54   00000000'GF 02 7E 7C 0A22 1022     CLRQ -(SP) ; GO BACK TO ESTABLISHER
54   00000000'GF 02 FB 0A24 1023     CALLS #2,G^SYSSUNWIND ; UNWIND CALL FRAMES
54   00000000'GF 02 04 0A2B 1024     RET ; RETURN TO ESTABLISHER
54   00000000'GF 02 04 0A2C 1025
54   50 0000'8F 3C 0A2C 1026 20$: MOVZWL #SS$_RESIGNAL,R0 ; RESIGNAL CONDITION
54   00000000'8F 02 04 0A31 1027     RET
    
```

LOC  
Sym  
SST  
ACM  
ACM  
ARG  
BAS  
BLA  
BLK  
BL  
BUF  
BUF  
BUF  
BUF  
CGM  
CHF  
CHF  
CON  
CON  
CON  
CON  
CON  
CON  
COU  
CRE  
CSI  
CV  
CWR  
DIR  
DIR  
DIS  
DMP  
DMP  
DMP  
DON  
EMO  
EX  
EXI  
FAO  
FAO  
FLA  
FLA  
FLA  
FLA  
FOR  
FOR  
GET  
GET  
GGM  
GRA  
GRA  
GRM  
GRM  
GRO  
GRO  
GRO  
GRO

```

0A32 1029      .SBTTL SHOW_ALL_RES - ACTION ROUTINE TO DISPLAY ALL RESOURCES
0A32 1030      :+++
0A32 1031      :
0A32 1032      :   SHOW_ALL_RES
0A32 1033      :
0A32 1034      :   PURPOSE
0A32 1035      :   MAIN ROUTINE TO PROCESS THE "SHOW RES/ALL" COMMAND
0A32 1036      :
0A32 1037      :   INPUT
0A32 1038      :   NONE
0A32 1039      :
0A32 1040      :   OUTPUT
0A32 1041      :   DISPLAYED DATA
0A32 1042      :
0A32 1043      :---
001C 0A32 1044      .ENTRY SHOW_ALL_RES,^M<R2,R3,R4>
0A34 1045
0A34 1046      PUSHAB RES HEAD          ; SET UP HEADING
0A38 1047      CALLS #1,SET HEADING
0A3F 1048      CLRL RES_COUNT        ; INIT COUNT
0A45 1049      CLRL R4              ; INIT HASH TABLE INDEX
0A47 1050      REQMEM @LCK$GL HTBLCNT,R2 ; GET COUNT OF ELEMENTS IN HASH TABLE
0A57 1051      ASHL R2,#1,HTBL CNT
0A5F 1052      MOVL LCK$GL HASRTBL,R3 ; GET ADDRESS OF START OF HASH TABLE
0A66 1053      REQMEM (R3),R3       ; GET FIRST ELEMENT
0A72 1054
0A72 1055      :
0A72 1056      :   INDEX THROUGH HASH TABLE, LOOKING FOR HASH CHAINS
0A72 1057      :
0A72 1058      10$:
0A72 1059      MOVAL (R3)[R4],R2     ; GET HEAD OF CHAIN
0A76 1060      REQMEM (R2),RSB      ; GET FIRST RSB ADDRESS
0A86 1061      TSTL RSB             ; IF ZERO, EMPTY CHAIN
0A8C 1062      BEQL 20$
0A8E 1063
0A8E 1064      :
0A8E 1065      :   SEQUENCE THROUGH HASH CHAIN
0A8E 1066      :
0A8E 1067      15$:
0A8E 1068      INCL RES_COUNT        ; INCREMENT COUNT
0A94 1069      REQMEM @RSB,RSB_BFR,#RSB$K_MAXLEN+RSB$K_LENGTH ;SAVE RSB DATA
0AAD 1070      CALLS #0,SHOW_RSB    ; DISPLAY RSB
0AB4 1071      CALLS #0,SHOW_QUEUES ; DISPLAY QUEUES
0ABB 1072      MOVL RSB_BFR,RSB     ; GET NEXT RSB
0AC6 1073      BLSS 15$
0AC8 1074
0AC8 1075      :
0AC8 1076      :   OUTER LOOP CONTROL
0AC8 1077      :
0AC8 1078      20$:
0AC8 1079      AOBSS HTBL_CNT,R4,10$ ; INCREMENT INDEX, LOOP IF LSS
0AD0 1080
0AD0 1081      TSTL RES_COUNT        ; DID WE PROCESS ANY RESOURCES?
0AD6 1082      BEQL 30$            ; IF EQL, NO
0AD8 1083
0AD8 1084      STATUS SUCCESS       ; SUCCESS RETURN
0ADF 1085      RET

```

```

          FSDD CF 9F
00000000'EF 01 FB
          00000115'EF D4
          54 D4
00000190'EF 01 52 78
53 00000000'EF D0
          52 6344 DE
000001EB'EF D5
          3A 13
          00000115'EF D6
00000B62'EF 00 FB
00000B92'EF 00 FB
000001EB'EF 0000011D'EF D0
          C6 19
A2 54 00000190'EF F2
          00000115'EF D5
          08 13
          04

```



```

OAF3 1090
OAF3 1091 .SBTTL SHOW_ONE_RES - ACTION ROUTINE TO DISPLAY ONE RESOURCE
OAF3 1092 :+++
OAF3 1093 :
OAF3 1094 : SHOW_ONE_RES
OAF3 1095 :
OAF3 1096 : PURPOSE
OAF3 1097 : MAIN ROUTINE TO PROCESS THE "SHOW RES/LOCK=lockid" COMMAND.
OAF3 1098 :
OAF3 1099 : INPUT
OAF3 1100 : LOCKID - ID OF LOCK WHOSE ASSOCIATED RESOURCE IS TO BE DISPLAYED.
OAF3 1101 :
OAF3 1102 : OUTPUT
OAF3 1103 : DISPLAYED DATA.
OAF3 1104 :
OAF3 1105 :---
0004 OAF3 1106 .ENTRY SHOW_ONE_RES,^M<R2>
OAF5 1107
F949 CF 00 FB OAF5 1108 CALLS #0,GET_LKB ; GET LKB ADDRESS
52 50 E9 OAF5 1109 BLBC R0,10$ ; IF LBC, NONE EXISTS
52 000002BF'EF D0 OAFD 1110 MOVL LKB,R2
OAF5 1111 REQMEM LKB$L,RSB(R2),RSB ; GET ADDRESS OF RSB
OAF5 1112 REQMEM @RSB,RSB_BFR,#RSB$K_MAXLEN+RSB$K_LENGTH ; GET RSB DATA
F4E3 CF 9F OB2E 1113 PUSHAB RES HEAD ; SET RSB HEADING
00000000'EF 01 FB OB32 1114 CALLS #1,SET HEADING
00000B62'EF 00 FB OB39 1115 CALLS #0,SHOW_RSB ; DISPLAY RSB
00000B92'EF 00 FB OB40 1116 CALLS #0,S JW_QUEUES ; DISPLAY QUEUES
OB47 1117
OB47 1118 STATUS SUCCESS
04 OB4E 1119 RET
OB4F 1120
04 OB4F 1121 10$: SIGNAL 0,NOLKB ; LOCKID NON-EXISTANT
OB61 1122 RET

```



```
OB62 1124 .SBTTL SHOW_RSB - CONTROL FOR RSB DISPLAY
OB62 1125 :+++
OB62 1126 :
OB62 1127 : SHOW_RSB
OB62 1128 :
OB62 1129 : PURPOSE
OB62 1130 : CONTROL THE FORMAT AND DISPLAY OF RSB DATA
OB62 1131 :
OB62 1132 : INPUT
OB62 1133 : RSB DATA STRUCTURES (IMPLIED)
OB62 1134 :
OB62 1135 : OUTPUT
OB62 1136 : DISPLAYED DATA
OB62 1137 : ---
0000 OB62 1138 .ENTRY SHOW_RSB,^M<>
0000 OB64 1139
0000 OB64 1140 CALLS #0,FORMAT_RSB
0000 OB6B 1141 SKIP PAGE
0000 OB72 1142 ENSURE 9
0000 OB8A 1143 CALLS #0,PRINT_RSB
0000 OB91 1144
04 OB91 1145 RET
```

```

OB92 1147 .SBTTL SHOW_QUEUES - DISPLAY QUEUES FOR GIVEN RESOURCE
OB92 1148 :+++
OB92 1149 :
OB92 1150 SHOW_QUEUES
OB92 1151 :
OB92 1152 PURPOSE
OB92 1153 DISPLAY, SEQUENTIALLY, THE GRANTED, CONVERSION AND WAIT
OB92 1154 QUEUES ASSOCIATED WITH A RESOURCE.
OB92 1155 :
OB92 1156 INPUT
OB92 1157 RSB - BUFFER CONTAINING RSB DATA
OB92 1158 :
OB92 1159 OUTPUT
OB92 1160 DISPLAYED QUEUES.
OB92 1161 :
OB92 1162 :---
0000 OB92 1163 .ENTRY SHOW_QUEUES,^M<>
OB94 1164
OB94 1165
7E 10 00001EB'EF C1 OB94 1166 ADDL3 RSB,#RSB$L_GRQFL,-(SP) ; GET ADDRESS OF GRANT QUE FLINK
      FFFFFFFF 8F DD OB9C 1167 PUSHL #GRANT ; IND GRANT QUE
      0000E55'EF 02 FB OBA2 1168 CALLS #2,PROCESS_QUEUE
      OBA9 1169
7E 18 00001EB'EF C1 OBA9 1170 ADDL3 RSB,#RSB$L_CVTQFL,-(SP) ; GET ADDRESS OF CONVERT QUE FLINK
      00 DD OBB1 1171 PUSHL #CONVERT ; IND CONVERT QUE
      0000E55'EF 02 FB OBB3 1172 CALLS #2,PROCESS_QUEUE
      OBBA 1173
7E 20 00001EB'EF C1 OBBA 1174 ADDL3 RSB,#RSB$L_WTQFL,-(SP) ; GET ADDRESS OF WAIT QUE FLINK
      01 DD OBC2 1175 PUSHL #WAIT ; IND WAIT QUE
      0000E55'EF 02 FB OBC4 1176 CALLS #2,PROCESS_QUEUE
      OBCB 1177
      04 OBCB 1178 RET
    
```

```

OBCC 1180 .SBTTL FORMAT_RSB - FORMAT RSB DATA
OBCC 1181 :+++
OBCC 1182 :
OBCC 1183 :   FORMAT_RSB
OBCC 1184 :
OBCC 1185 :   PURPOSE
OBCC 1186 :   FORMAT RAW RSB DATA, STORE FOR EASY ACCESS LATER ON.
OBCC 1187 :
OBCC 1188 :   INPUT
OBCC 1189 :   RSB_BFR - BUFFER CONTAINING RSB DATA
OBCC 1190 :
OBCC 1191 :   OUTPUT
OBCC 1192 :   STORED DATA.
OBCC 1193 :
OBCC 1194 :---
O0FC OBCC 1195 .ENTRY FORMAT_RSB,^M<R2,R3,R4,R5,R6,R7>
56 0000011D'EF 9E OBCE 1196   MOVAB RSB_BFR,R6 ; GET ADDRESS OF DATA BUFFER
OBCE 1197
OBD5 1198
OBD5 1199 :
OBD5 1200 :   FORMAT RESOURCE NAME
OBD5 1201 :
67 20 00 57 50 A6 9E OBD5 1202   MOVAB RSB$T_RESNAM(R6),R7 ; LET R7 POINT TO RESOURCE NAME
50 4F A6 9A OBD9 1203   MOVZBL RSB$B_RSLEN(R6),R0 ; ZERO OUT UNUSED PORTION OF BUFFER
00000227'EF 67 50 2C OBD8 1204   MOVCS R0,(R7),#0,#32,(R7)
0000021F'EF 87 DE OBE3 1205   MOVAL (R7),BUF1
0000021B'EF 87 DO OBEA 1206   MOVL (R7)+,RESN1
0000023B'EF 67 DE OBF1 1207   MOVL (R7)+,RESN2
00000233'EF 87 DO OBF8 1208   MOVAL (R7),BUF2
0000022F'EF 87 DO OBF8 1209   MOVL (R7)+,RESN3
00000253'EF 67 DE CC06 1210   MOVL (R7)+,RESN4
0000024B'EF 87 DE OC0D 1211   MOVAL (R7),BUF3
00000247'EF 87 DO OC14 1212   MOVL (R7)+,RESN5
0000026B'EF 67 DE OC1B 1213   MOVL (R7)+,RESN6
00000263'EF 87 DO OC22 1214   MOVAL (R7),BUF4
0000025F'EF 87 DO OC29 1215   MOVL (R7)+,RESN7
0000025F'EF 87 DO OC30 1216   MOVL (R7)+,RESN8
OC37 1217
OC37 1218 :
OC37 1219 :   FORMAT RSB
OC37 1220 :
000001F3'EF 48 A6 DO OC37 1221   MOVL RSB$L_PARENT(R6),PRSB
000001FB'EF 40 A6 3C OC3F 1222   MOVZWL RSB$W_REFCNT(R6),SRSBCT
000001FF'EF 42 A6 3C OC47 1223   MOVZWL RSB$W_BLKASTCNT(R6),BAST
0000020F'EF 28 A6 DO OC4F 1224   MOVL RSB$Q_VALBLK(R6),VAL1
0000020B'EF 2C A6 DO OC57 1225   MOVL RSB$Q_VALBLK+4(R6),VAL2
00000207'EF 30 A6 DO OC5F 1226   MOVL RSB$Q_VALBLK+8(R6),VAL3
00000203'EF 34 A6 DO OC67 1227   MOVL RSB$Q_VALBLK+12(R6),VAL4
00000213'EF 3C A6 DO OC6F 1228   MOVL RSB$L_VALSEQNUM(R6),SEQNUM
0000023F'EF 38 A6 DO OC77 1229   MOVL RSB$L_CSID(R6),CSID
00000217'EF F64C CF 9E OC7F 1230   MOVAB SPACE,VALID
09 OE A6 01 E1 OC88 1231   BBC #RSB$V_VALINVLID,RSB$W_STATUS(R6),15$
00000217'EF F640 CF 9E OC8D 1232   MOVAB NOT_VALID,VALID
00000257'EF F635 CF 9E OC96 1233 15$: MOVAB SPACE,DIRENTRY
09 OE A6 00 E1 OC9F 1234   BBC #RSB$V_DIRENTRY,RSB$W_STATUS(R6),17$
00000257'EF F633 CF 9E OCA4 1235   MOVAB DIR_ENTRY,DIRENTRY
51 OC A6 9A OCAD 1236 17$: MOVZBL RSB$B_GMODE(R6),R1

```

```

000001EF'EF  F51B CF41  D0  OCB1  1237  MOVL  LKMODE_TBL[R1],GGMOD
                51  0D A6  9A  OCB8  1238  MOVZBL RSB$B_CGMODE(R6),R1
000001F7'EF  F50D CF41  D0  OCBF  1239  MOVL  LKMODE_TBL[R1],CGMOD
                51  4E A6  90  OCC9  1240  MOVB  RSB$B_RMOD(R6),R1
00000243'EF  F646 CF41  D0  OCCD  1241  MOVL  ACMODE_TBL[R1],RACMOD
0000022B'EF  4F A6  9A  OCD7  1242  MOVZBL RSB$B_RSNLEN(R6),RESNLEN
0000025B'EF  000001C8'EF  7E  OCDF  1243  MOVAQ  SYS_DSC,RNSPACE
0000006C'EF  4C A6  B0  OCEA  1244  MOVW  RSB$W_GROUP(R6),GROUP_BFR
                3F  13  OCF2  1245  BEQL  90$
                OCF4  1246  $FAO_S  CTRSTR = FAO_GROUP_DSC,-
                OCF4  1247  OUTBUF = GROUP_BUF_DSC,-
                OCF4  1248  P1 = GROUP_BFR
000001E8'EF  21 50  E9  OD0F  1249  BLBC  R0,90$
000001EA'EF  000001BA'EF  B0  OD12  1250  MOVW  GROUP_BUF,GROUP_NUM
000001EB'EF  000001BC'EF  90  OD1D  1251  MOVB  GROUP_BUF+2,GROUP_NUM+2
0000025B'EF  000001DB'EF  7E  OD28  1252  MOVAQ  GROUP_DSC,RNSPACE
                OD33  1253
                OD33  1254 90$:
04  OD33  1255  RET

```

; Branch if system name

```

0D34 1257 .SBTTL PRINT_RSB - OUTPUT RSB DATA
0D34 1258 :+++
0D34 1259 :
0D34 1260 PRINT_RSB
0D34 1261 :
0D34 1262 PURPOSE
0D34 1263 OUTPUT RSB DATA, LINE-BY-LINE.
0D34 1264 :
0D34 1265 INPUT
0D34 1266 FORMATTED DATA.
0D34 1267 :
0D34 1268 OUTPUT
0D34 1269 DISPLAYED DATA.
0D34 1270 :
0D34 1271 :---
0000 0D34 1272 :
0D34 1273 .ENTRY PRINT_RSB,^M<>
0D36 1274 :
0D36 1275 :
0D36 1276 LINE 1
0D36 1277 :
000001EF'EF DD 0D36 1278 PUSHL GGMOD
000001EB'EF DD 0D3C 1279 PUSHL RSB
0D42 1280 PRINT 2,<Address of RSB: !XL Group grant mode: !AC >
0D4F 1281 : RSB GGMOD
0D4F 1282 :
0D4F 1283 :
0D4F 1284 LINE 2
0D4F 1285 :
000001F7'EF DD 0D4F 1286 PUSHL CGMOD
000001F3'EF DD 0D55 1287 PUSHL PRSB
0D5B 1288 PRINT 2,<Parent RSB: !XL Conversion grant mode: !AC >
0D68 1289 : PRSB CGMOD
0D68 1290 :
0D68 1291 :
0D68 1292 LINE 3
0D68 1293 :
000001FF'EF DD 0D68 1294 PUSHL BAST
000001FB'EF DD 0D6E 1295 PUSHL SRSBCT
0D74 1296 PRINT 2,<Sub-RSB count: !8UL BLKAST count: !8UL>
0D81 1297 : SRSBCT BAST
0D81 1298 :
0D81 1299 :
0D81 1300 LINE 4
0D81 1301 :
00000217'EF DD 0D81 1302 PUSHL VALID
00000213'EF DD 0D87 1303 PUSHL SEQNUM
0000020F'EF DD 0D8D 1304 PUSHL VAL1
0000020B'EF DD 0D93 1305 PUSHL VAL2
00000207'EF DD 0D99 1306 PUSHL VAL3
00000203'EF DD 0D9F 1307 PUSHL VAL4
0DA5 1308 PRINT 6,<Value block: !XL !XL !XL !XL Seq. #: !XL !AC>
0DB2 1309 : VAL4 VAL3 VAL2 VAL1 SEQNUM VALID
0DB2 1310 :
0DB2 1311 :
0DB2 1312 LINE 5
0DB2 1313 :

```

```

00000227'EF DD ODB2 1314 PUSHL BUF1
00000228'EF DD ODB8 1315 PUSHL #8
0000021F'EF DD ODBA 1316 PUSHL RESN1
0000021B'EF DD ODC0 1317 PUSHL RESN2
ODC6 1318 PRINT 4,<Resource: !XL !XL !AF >
ODD3 1319 : RESN2 RESN1 BUF1
ODD3 1320 :
ODD3 1321 :
ODD3 1322 : LINE 6
ODD3 1323 :
0000023F'EF DD ODD3 1324 PUSHL ^SID
0000023B'EF DD ODD9 1325 PUSHL BUF2
00000238'EF DD ODDF 1326 PUSHL #8
00000233'EF DD ODE1 1327 PUSHL RESN3
0000022F'EF DD ODE7 1328 PUSHL RESN4
0000022B'EF DD ODED 1329 PUSHL RESNLEN
ODEF 1330 PRINT 6,< Length !2UL !XL !XL !AF CSID: !XL>
OE00 1331 : RESNLEN RESN4 RESN3 BUF2 CSID
OE00 1332 :
OE00 1333 : LINE 7
OE00 1334 :
OE00 1335 :
00000257'EF DD OE00 1336 PUSHL DIRENTRY
00000253'EF DD OE06 1337 PUSHL BUF3
00000250'EF DD OE0C 1338 PUSHL #8
0000024B'EF DD OE0E 1339 PUSHL RESN5
00000247'EF DD OE14 1340 PUSHL RESN6
00000243'EF DD OE1A 1341 PUSHL RACMOD
OE20 1342 PRINT 6,< !11AC !XL !XL !AF !AC>
OE2D 1343 : RACMOD RESN6 RESN5 BUF3 DIRENTRY
OE2D 1344 :
OE2D 1345 : LINE 8
OE2D 1346 :
OE2D 1347 :
0000026B'EF DD OE2D 1348 PUSHL BUF4
00000268'EF DD OE33 1349 PUSHL #8
00000263'EF DD OE35 1350 PUSHL RESN7
0000025F'EF DD OE3B 1351 PUSHL RESN8
0000025B'EF DD OE41 1352 PUSHL RNSPACE
OE47 1353 PRINT 5,< !11AS !XL !XL !AF>
OE54 1354 : RNSPACE RESN8 RESN7 BUF4
OE54 1355 :
04 OE54 1356 RET

```

```

OE55 1358 .SBTTL PROCESS_QUEUE - TRAVERSE RESOURCE QUEUES
OE55 1359 :+++
OE55 1360 :
OE55 1361 :   PROCESS_QUEUE
OE55 1362 :
OE55 1363 :   PURPOSE
OE55 1364 :   TRAVERSE AN INDIVIDUAL RESOURCE QUEUE, FORMATTING AND
OE55 1365 :   DISPLAYING ITS CONTENTS.
OE55 1366 :
OE55 1367 :   INPUT
OE55 1368 :   4(AP) - QUEUE TYPE, WHERE:  -1: GRANT
OE55 1369 :                               0: CONVERSION
OE55 1370 :                               1: WAIT
OE55 1371 :   8(AP) - ADDRESS OF HEAD OF QUEUE
OE55 1372 :
OE55 1373 :   OUTPUT
OE55 1374 :   DISPLAYED CONTENTS OF QUEUE
OE55 1375 :
OE55 1376 :---
03FC OE55 1377 .ENTRY PROCESS_QUEUE,^M<R2,R3,R4,R5,R6,R7,R8,R9>
OE57 1378
54 000001A2'EF 9E OE57 1379 MOVAB LKID_BFR,R4 ; GET ADDRESS OF LOCKID STORAGE BUFFER
55 0000019C'EF 9E OE5E 1380 MOVAB GRMD_BFR,R5 ; GET ADDRESS OF GRANT MODE BUFFER
56 0000019F'EF 9E OE65 1381 MOVAB RQMD_BFR,R6 ; GET ADDRESS OF REQUEST MODE BUFFER
   52 08 AC DO OE6C 1382 MOVL 8(AP),R2 ; GET ADDRESS OF HEAD OF QUEUE
   59 52 DO OE70 1383 MOVL R2,R9
00000119'EF D4 OE73 1384 CLRL QUEUE_COUNT ; INIT COUNT OF QUEUE ITEMS PROCESSED
00000198'EF 04 AC DO OE79 1385 MOVL 4(AP),QUEUE_TYPE ; GET TYPE OF QUEUE
   58 01 CE OE81 1386 MNEGL #1,R8
57 00000198'EF 01 C1 OE84 1387 ADDL3 #1,QUEUE_TYPE,R7
   57 F1BD CF47 DO OE8C 1388 MOVL QUE_STR_TBL[R7],R7 ; GET HEADER FOR THIS TYPE OF QUEUE
OE92 1389 ENSURE 3
OEAA 1390 SKIP 1
OEB3 1391 PRINTD R8,(R7) ; DISPLAY IT
OEBE 1392
OEBE 1393 :
OEBE 1394 :   INIT FOR SCAN OF QUEUE
OEBE 1395 :
53 D4 OEBE 1396 10$: CLRL R3
OECO 1397
OECO 1398 :
OECO 1399 :   GET QUEUE ELEMENT
OECO 1400 :
20$: OECO 1401 REQMEM (R2),R2 ; GET QUEUE ELEMENT
OEC 1402 Cmpl R9,R2 ; AT END OF QUEUE?
52 59 D1 OECF 1403 BEQL 30$ ; IF EQL, YES
3D 13 OED1 1404
OED1 1405 ASSUME LKBSL_SQFL-LKBSL_LKID GT 0
OED1 1406 ASSUME LKBSB_RQMODE-LKBSL_LKID EQ 4
OED1 1407 ASSUME LKBSB_GRMODE-LKBSL_LKID EQ 5
OED1 1408
00000119'EF D6 OED1 1409 INCL QUEUE_COUNT ; INCREMENT COUNT
58 52 08 C3 OED7 1410 SUBL3 #<LKBSL_SQFL-LKBSL_LKID>,R2,R8 ; GET START OF RELEVANT DATA
OEDB 1411 REQMEM (R8),LKB_BFR,#6 ; SAVE ID, GRANT & REQUEST MODE
57 00000008'EF 9E OEEC 1412 MOVAB LKB_BFR,R7
   6443 67 DO OEF3 1413 MOVL LKID_OFF(R7),(R4)[R3] ; MOVE INTO STORAGE BUFFERS
   6543 05 A7 90 OEF7 1414 MOVB GRMD_OFF(R7),(R5)[R3]

```

```
6643 04 A7 90 0EFC 1415      MOVB  RQMD_OFF(R7),(R6)[R3]
          0F01 1416
          BB 53 03 F2 0F01 1417      AOBLS #3,R3,20$      ; IF WE'VE SAVED 3 ITEMS, PRINT THEM
00000F2B'EF 00 FB 0F05 1418      CALLS #0,PRINT_LINE
          B0 11 0F0C 1419      BRB 10$      ; GET MORE ITEMS
          0F0E 1420
00000F2B'EF 00 FB 0F0E 1421 30$: CALLS #0,PRINT_LINE      ; PRINT ANY REMAINING ITEMS
          00000119'EF 05 0F15 1422      TSTL QUEUE_COUNT      ; WERE ANY ITEMS PROCESSED?
          OD 12 0F1B 1423      BNEQ EXIT      ; IF NEQ, YES
          0F1D 1424      PRINT 0,< *** EMPTY QUEUE ***>
          0F2A 1425
          04 0F2A 1426 EXIT: RET
```



```

OF2B 1428 .SBTTL PRINT_LINE - OUTPUT QUEUE DATA
OF2B 1429 :+++
OF2B 1430 :
OF2B 1431 PRINT_LINE
OF2B 1432 :
OF2B 1433 PURPOSE
OF2B 1434 GIVEN LOCK DATA FOR ONE LINE (0 - 3 ITEMS), FORMAT AND
OF2B 1435 DISPLAY THIS DATA
OF2B 1436 :
OF2B 1437 INPUT
OF2B 1438 R3 - COUNT OF ITEMS TO DISPLAY
OF2B 1439 R4 - ADDRESS OF BUFFER CONTAINING LOCK IDS
OF2B 1440 R5 - ADDRESS OF BUFFER CONTAINING GRANT-MODES
OF2B 1441 R6 - ADDRESS OF BUFFER CONTAINING REQUESTED-MODES
OF2B 1442 GJEUQ_TYPE - TYPE OF QUEUE WE ARE WORKING ON
OF2B 1443 :
OF2B 1444 OUTPUT
OF2B 1445 ONE DISPLAYED LINE OF QUEUE LOCK DATA.
OF2B 1446 :
OF2B 1447 :---
OF2B 1448
0180 OF2B 1449 .ENTRY PRINT_LINE,^M<R7,R8>
OF2D 1450
OF2D 1451 :
OF2D 1452 INITIALIZE
OF2D 1453 :
57 00000198'EF D0 OF2D 1454 MOVL QUEUE_TYPE,R7 ; SAVE QUEUE TYPE
00000194'EF 53 D0 OF34 1455 MOVL R3,COUNT ; SAVE COUNT
01 53 F4 OF3B 1456 SOBGEQ R3,10$ ; CONVERT TO ARRAY INDEX
04 OF3E 1457 RET ; IF COUNT 0, NO ITEMS TO DISPLAY
OF3F 1458
OF3F 1459 :
OF3F 1460 FORMAT DATA
OF3F 1461 :
57 D5 OF3F 1462 10$: TSTL R7 ; ONLY FORMAT RQ MODE IF WAIT OR CONVERT
09 19 OF41 1463 BLSS 15$ ; QUEUE
58 6643 9A OF43 1464 MOVZBL (R6)[R3],R8
F285 CF48 DD OF47 1465 PUSHL LKMODE_TBL[R8]
OF4C 1466
57 D5 OF4C 1467 15$: TSTL R7 ; ONLY FORMAT GR MODE IF GRANT OR CONVERT
09 14 OF4E 1468 BGTR 20$ ; QUEUE
58 6543 9A OF50 1469 MOVZBL (R5)[R3],R8
F278 CF48 DD OF54 1470 PUSHL LKMODE_TBL[R8]
OF59 1471
6443 DD OF59 1472 20$: PUSHL (R4)[R3] ; PROCESS LOCK ID
E0 53 F4 OF5C 1473 SOBGEQ R3,10$ ; IF NOT YET DONE, LOOP
OF5F 1474
58 F0DB CF 9E OF5F 1475 MOVAB CONV_STR_TBL,R8 ; GET ADDRESS OF CORRECT FAO STRING TABLE
57 D5 OF64 1476 TSTL R7
05 13 OF66 1477 BEQL 25$ ; IF EQL, CONVERT
58 F0C2 CF 9E OF68 1478 MOVAB LOCK_STR_TBL,R8 ; ELSE WAIT/GRANT
OF6D 1479
53 00000194'EF D0 OF6D 1480 25$: MOVL COUNT,R3 ; PICK UP CORRECT STRING FOR NUMBER OF
58 6843 DG OF74 1481 MOVL (R8)[R3],R8 ; ITEMS
53 53 C0 OF78 1482 ADDL R3,R3 ; CALCULATE CORRECT FAO ARG COUNT
57 D5 OF7B 1483 TSTL R7
07 12 OF7D 1484 BNEQ 30$

```

LOCK  
V04-000

LOCK AND RESOURCE FORMATTING ROUTINES N 7  
PRINT\_LINE - OUTPUT QUEUE DATA

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00  
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

Page 39  
(22)

MAI  
V04

```
53 00000194'EF  C0  OF7F 1485      ADDL  COUNT,R3
                   OF86 1486
                   OF86 1487 30$:  PRINTD r3,(r8)      ; DISPLAY LINE
                   OF91 1488
                   04  OF91 1489      RET
                   OF92 1490
                   OF92 1491      .END
```

LOCK  
Symbol table

LOCK AND RESOURCE FORMATTING ROUTINES

B 8

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00  
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

\$ST2	= 00000004		GROUP_TXT	000001E0	R	02
ACMODE	000002FF	R 02	GR_STRING	000001FB	R	03
ACMODE_TBL	00000318	R 03	HTBL_CNT	00000190	R	02
ARGS	= 00000003		HTBL_INDY	0000018C	R	02
BAST	000001FF	R 02	KMODE	000002EB	R	02
BLANKS	00000220	R 03	LCK\$GL_HASHTBL	*****	X	03
BLKAST	000002BB	R 02	LCK\$GL_HTBLCNT	*****	X	03
BL_STRING	00000223	R 03	LCK\$GL_IDTBL	*****	X	03
BUF1	00000227	R 02	LCK\$GL_MAXID	*****	X	03
BUF2	0000023B	R 02	LIB\$SIGNAL	*****	X	03
BUF3	00000253	R 02	LINE_COUNT	*****	X	03
BUF4	00000268	R 02	LKB	000002BF	R	02
CGMOD	000001F7	R 02	LKBSB_GRMODE	= 00000035		
CHF\$MCH_SAVRO	= 0000000C		LKBSB_RQMODE	= 00000034		
CHF\$SIG_NAME	= 00000004		LKBSB_STATE	= 00000036		
CONVERT	= 00000000		LKBSK_LENGTH	= 00000060		
CONVSTR	0000016B	R 03	LKBSL_BLKASTADR	= 00000020		
CONVSTR1	000000C6	R 03	LKBSL_CSID	= 00000058		
CONVSTR2	000000DF	R 03	LKBSL_LKID	= 00000030		
CONVSTR3	00000108	R 03	LKBSL_OWNOFL	= 00000040		
CONV_STR_TBL	0000003E	R 03	LKBSL_PARENT	= 00000048		
COUNT	00000194	R 02	LKBSL_PID	= 0000000C		
CREAD	000001EC	R 03	LKBSL_REMLKID	= 00000054		
CSID	0000023F	R 02	LKBSL_RSB	= 00000050		
CV_STRING	00000206	R 03	LKBSL_SQFL	= 00000038		
CWRITE	000001EF	R 03	LKBSV_MSTCPY	= 00000004		
DIRENTRY	00000257	R 02	LKBSW_FLAGS	= 00000028		
DIR_ENTRY	000002DB	R 03	LKBSW_REFcnt	= 0000004C		
DISPLAY_LOCK	0000051D	RG 03	LKBSW_STATUS	= 0000002A		
DMP1	000002CF	R 02	LKB_BFR	00000008	R	02
DMP3	000002EF	R 02	LKB_RSB_BFR	0000006C	R	02
DMP5	0000030F	R 02	LKID	0000026F	R	02
DMP7	0000032F	R 02	LKID_BFR	000001A2	R	02
DONE	00000442	R 03	LKID_OFF	= 00000000		
EMODE	000002F7	R 03	LKMODE_TBL	000001D1	R	03
EX	000001F8	R 03	LOCAL	000002BA	R	03
EXIT	00000F2A	R 03	LOCKID	00000090	RG	02
FAO_GROUP_DSC	000001AE	R 02	LOCKMODE_TBL	000001C5	R	03
FAO_RMINFO_DSC	00000094	R 02	LOCKSTR1	0000005A	R	03
FLAGS1	00000277	R 02	LOCKSTR	0000006F	R	03
FLAGS2	00000293	R 02	LOCKSTR	00000093	R	03
FLAGS3	000002AF	R 02	LOCKSTR	00000098	RG	03
FLAGS_TBL	0000022A	R 03	LOCK_COND_HAND	00000000	R	02
FORMAT_LOCK	00000556	RG 03	LOCK_COUNT	00000000	R	03
FORMAT_RSB	00000BCC	RG 03	LOCK_HEAD	0000002E	R	03
GETLCK	000003C7	R 03	LOCK_STR_TBL	000002C8	R	03
GET_LKB	00000443	RG 03	MASTER	0000028F	R	02
GGMOD	000001EF	R 02	MODE1	000002AB	R	02
GRANT	= FFFFFFFF		MODE2	*****	X	03
GRANTSTR	00000141	R 03	MSG\$_LOCKIDZER	*****	X	03
GRMD_BFR	0000019C	R 02	MSG\$_NOLKB	*****	X	03
GRMD_OFF	= 00000005		MSG\$_NOLOCKS	*****	X	03
GROUP_BFR	0000006C	R 02	MSG\$_NOPRLOCK	*****	X	03
GROUP_BUF	000001BA	R 02	MSG\$_NORESOURC	*****	X	03
GROUP_BUF_DSC	000001C0	R 02	MSG\$_OUTOFRANG	*****	X	03
GROUP_DSC	000001D8	R 02	MSG\$_SUCCESS	*****	X	03
GROUP_NUM	000001E8	R 02	NEW_PAGE	*****	X	03
			NOLCK	00000430	R	03

LOCK  
Symbol table

LOCK AND RESOURCE FORMATTING ROUTINES

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00  
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

NOT\_VALID 000002D1 R 03  
 NSPACE 0000031F R 02  
 NULL 000001E9 R 03  
 NULL\_CSTRING 00000222 R 03  
 PAGE\_SIZE \*\*\*\*\* X 03  
 PARID 00000283 R 02  
 PARID\_BFR 00000004 R 02  
 PART2 0000067E R 03  
 PCB\$LOCKQFL = 00000104  
 PID 00000273 R 02  
 PREAD 000001F2 R 03  
 PRINT \*\*\*\*\* X 03  
 PRINT\_LINE 00000F2B RG 03  
 PRINT\_LOCK 00000856 RG 03  
 PRINT\_RSB 00000D34 RG 03  
 PROCESS 000002C0 R 03  
 PROCESS\_QUEUE 00000E55 RG 03  
 PRSB 000001F3 R 02  
 PWRITE 000001F5 R 03  
 QUEUE\_COUNT 00000119 R 02  
 QUEUE\_TYPE 00000198 R 02  
 QUE\_STR\_TBL 0000004E R 03  
 RACMOD 00000243 R 02  
 REQMEM \*\*\*\*\* X 03  
 RES1 000002C7 R 02  
 RES2 000002C3 R 02  
 RES3 000002E7 R 02  
 RES4 000002E3 R 02  
 RES5 00000307 R 02  
 RES6 00000303 R 02  
 RES7 00000327 R 02  
 RES8 00000323 R 02  
 RESN1 0000021F R 02  
 RESN2 0000021B R 02  
 RESN3 00000233 R 02  
 RESN4 0000022F R 02  
 RESN5 0000024B R 02  
 RESN6 00000247 R 02  
 RESN7 00000263 R 02  
 RESN8 0000025F R 02  
 RESNAM\_BFR 00000070 R 02  
 RESLEN 0000022B R 02  
 RES\_COUNT 00000115 R 02  
 RES\_HEAD 00000015 R 03  
 RLEN 000002DF R 02  
 RMINFO 00000337 R 02  
 RMINFO\_BFR 000000C5 R 02  
 RMINFO\_CNT 000000C4 R 02  
 RMINFO\_DSC 0000008C R 02  
 RMINFO\_LEN 00000088 R 02  
 RMOD\_BFR 0000006E R 02  
 RNSPACE 0000025B R 02  
 RQMD\_BFR = 0000019F R 02  
 RQMD\_OFF = 00000004  
 RSB = 000001EB R 02  
 RSB\$B\_CGMODE = 0000000D  
 RSB\$B\_GGMODE - 0000000C

RSB\$B\_RMOD = 0000004E  
 RSB\$B\_RSLEN = 0000004F  
 RSB\$K\_LENGTH = 00000050  
 RSB\$K\_MAXLEN = 0000001F  
 RSB\$L\_CSID = 00000038  
 RSB\$L\_CVTQFL = 00000018  
 RSB\$L\_GRQFL = 00000010  
 RSB\$L\_PARENT = 00000048  
 RSB\$L\_VALSEQNUM = 0000003C  
 RSB\$L\_WTQFL = 00000020  
 RSB\$Q\_VALBLK = 00000028  
 RSB\$T\_RESNAM = 00000050  
 RSB\$V\_DIRENTRY = 00000000  
 RSB\$V\_VALINVL = 00000001  
 RSB\$W\_BLKASTCNT = 00000042  
 RSB\$W\_GROUP = 0000004C  
 RSB\$W\_REFCNT = 00000040  
 RSB\$W\_STATUS = 0000000E  
 RSBCSID\_BFR 00000068 R 02  
 RSB\_BFR 0000011D R 02  
 RSNLEN\_BFR 0000006F R 02  
 SAVE\_LOCK\_DATA 000004B7 RG 03  
 SEQNUM 00000213 R 02  
 SET\_HEADING \*\*\*\*\* X 03  
 SHOW\_ALL\_LOCKS 00000328 RG 03  
 SHOW\_ALL\_RES 00000A32 RG 03  
 SHOW\_ONE\_LOCK 0000036E RG 03  
 SHOW\_ONE\_RES 00000AF3 RG 03  
 SHOW\_PROC\_LOCK 000003AE RG 03  
 SHOW\_QUEUES 00000B92 RG 03  
 SHOW\_RSB 00000B62 RG 03  
 SKIP\_LINES \*\*\*\*\* X 03  
 SMODE 00000302 R 03  
 SPACE 000002CF R 03  
 SRSBCT 000001FB R 02  
 SSS\_RESIGNAL \*\*\*\*\* X 03  
 SSS\_UNWIND \*\*\*\*\* X 03  
 STATE1 00000287 R 02  
 STATE2 000002A3 R 02  
 STATUS\_TBL 00000272 R 03  
 STS1 000002D3 R 02  
 STS2 000002F3 R 02  
 STS3 00000313 R 02  
 SUBLKS 0000029F R 02  
 SYSSFAO \*\*\*\*\* X 03  
 SYSSUNWIND \*\*\*\*\* X 03  
 SYS\_DSC 000001C8 R 02  
 TYPE 00000333 R 02  
 UMODE 0000030E R 03  
 VAL1 0000020F R 02  
 VAL2 00000208 R 02  
 VAL3 00000207 R 02  
 VAL4 00000203 R 02  
 VALID 00000217 R 02  
 WAIT = 00000001  
 WAITSTR 0000019B R 03  
 WT\_STRING 00000214 R 03

-----  
! 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
SDADATA	0000033B ( 827.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
LOCK	00000F92 ( 3986.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
LITERALS	0000034C ( 844.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.24
Command processing	107	00:00:00.45	00:00:03.92
Pass 1	310	00:00:06.76	00:00:26.79
Symbol table sort	0	00:00:00.69	00:00:01.98
Pass 2	264	00:00:02.72	00:00:10.19
Symbol table output	26	00:00:00.14	00:00:00.30
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	741	00:00:10.83	00:00:44.44

The working set limit was 1650 pages.  
66653 bytes (131 pages) of virtual memory were used to buffer the intermediate code.  
There were 40 pages of symbol table space allocated to hold 660 non-local and 73 local symbols.  
1491 source lines were read in Pass 1, producing 79 object records in Pass 2.  
26 pages of virtual memory were used to define 24 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1	9
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	21

660 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

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

A grid of 15 columns and 10 rows of small, faded technical diagrams and charts. The diagrams contain various data points, labels, and structural elements. Some larger, more legible labels are scattered throughout the grid, including:

- HANDLER LIS
- DUMP LIS
- MAIN LIS
- EXAMPS LIS
- MMG LIS
- INDEX LIS
- LOCK LIS
- PARSE LIS