



```

LL          000000  CCCCCCCC  KK      KK  DDDDDDDD  88888888
LL          000000  CCCCCCCC  KK      KK  DDDDDDDD  88888888
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KKKKKK  DD      DD  88888888
LL          00      00  CC      CC  KKKKKK  DD      DD  88888888
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LL          00      00  CC      CC  KK      KK  DD      DD  88      88
LLLLLLLLLLL 000000  CCCCCCCC  KK      KK  DDDDDDDD  88888888
LLLLLLLLLLL 000000  CCCCCCCC  KK      KK  DDDDDDDD  88888888

```

```

....
....
....
....

```

```

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
LLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLL IIIIII  SSSSSSSS

```





```

0000 53 :++
0000 54 :
0000 55 : FUNCTIONAL DESCRIPTION:
0000 56 :
0000 57 :     THIS ROUTINE LOCKS THE I/O DATA BASE MUTEX.
0000 58 :
0000 59 : CALLING SEQUENCE:
0000 60 :     CALL LOCK_IODB ()
0000 61 :
0000 62 : INPUT PARAMETERS:
0000 63 :     NONE
0000 64 :
0000 65 : IMPLICIT INPUTS:
0000 66 :     NONE
0000 67 :
0000 68 : OUTPUT PARAMETERS:
0000 69 :     NONE
0000 70 :
0000 71 : IMPLICIT OUTPUTS:
0000 72 :     NONE
0000 73 :
0000 74 : ROUTINE VALUE:
0000 75 :     NONE
0000 76 :
0000 77 : SIDE EFFECTS:
0000 78 :     I/O DATA BASE MUTEX LOCKED
0000 79 :
0000 80 :--
0000 81 :
00000000 82 : .PSECT $LOCKEDC1$,NOWRT
0000 83 :
0000 84 LOCK_IODB::
50 00000000'9F 003C 0000 85 : .WORD ^M<R2,R3,R4,R5> ; SAVE REGISTERS
54 00000000'9F DE 0002 86 : MOVAL @#IOC$GL_Mutex,R0 ; GET I/O DATA BASE MUTEX
00000000'9F D0 0009 87 : MOVL @#SCH$GL_CURPCB,R4 ; GET OWN PCB ADDRESS
00000000'9F 16 0010 88 : JSB @#SCH$LOCKW ; AND LOCK IT
04 0016 89 : RET

```

```

0017 91 :++
0017 92
0017 93 : FUNCTIONAL DESCRIPTION:
0017 94
0017 95 :     THIS ROUTINE UNLOCKS THE I/O DATA BASE MUTEX.
0017 96
0017 97 : CALLING SEQUENCE:
0017 98 :     CALL UNLOCK_IODB ( )
0017 99
0017 100 : INPUT PARAMETERS:
0017 101 :     NONE
0017 102
0017 103 : IMPLICIT INPUTS:
0017 104 :     NONE
0017 105
0017 106 : OUTPUT PARAMETERS:
0017 107 :     NONE
0017 108
0017 109 : IMPLICIT OUTPUTS:
0017 110 :     NONE
0017 111
0017 112 : ROUTINE VALUE:
0017 113 :     NONE
0017 114
0017 115 : SIDE EFFECTS:
0017 116 :     I/O DATA BASE MUTEX UNLOCKED
0017 117 :     IPL LOWERED TO 0
0017 118
0017 119 :--
0017 120
0017 121 UNLOCK_IODB::
0017 122 .WORD ^M<R2,R3,R4,R5> ; SAVE REGISTERS
50 00000000'9F DE 0019 123 MOVAL @#IOC$GL_Mutex,R0 ; GET I/O DATA BASE MUTEX
54 00000000'9F DO 0020 124 MOVL @#SCH$GL_CURPCB,R4 ; AND OWN PCB ADDRESS
00000000'9F 16 0027 125 JSB @#SCH$UNLOCK ; AND UNLOCK IT
002D 126 SETIPL #0 ; ALSO LOWER IPL
04 0030 127 RET
0031 128
0031 129
0031 130
0031 131 .END

```

00000000'9F DE 0019 123 MOVAL @#IOC\$GL\_Mutex,R0 ; GET I/O DATA BASE MUTEX

```

AOB_TYPE      = 00000005
BITMAP_TYPE   = 00000001
DIRECTORY_TYPE = 00000002
FCB_TYPE      = 00000000
HEADER_TYPE   = 00000000
INDEX_TYPE    = 00000003
IOCSGC_MUTEX  ***** X 02
LOCK_IODB     00000000 RG 02
MVL_TYPE      = 00000004
PRS_IPL       = 00000012
RVT_TYPE      = 00000003
SCH$GL_CURPCB ***** X 02
SCH$LOCKW     ***** X 02
SCH$UNLOCK    ***** X 02
UNLOCK_IODB   00000017 RG 02
VCB_TYPE      = 00000002
WCB_TYPE      = 00000001
    
```

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$LOCKEDC1\$	00000031 ( 49.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	44	00:00:00.12	00:00:00.51
Command processing	169	00:00:00.78	00:00:05.10
Pass 1	138	00:00:01.48	00:00:07.87
Symbol table sort	0	00:00:00.08	00:00:00.11
Pass 2	39	00:00:00.58	00:00:02.58
Symbol table output	2	00:00:00.03	00:00:00.03
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	396	00:00:03.10	00:00:16.23

The working set limit was 1050 pages.  
6296 bytes (13 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 96 non-local and 0 local symbols.  
234 source lines were read in Pass 1, producing 13 object records in Pass 2.  
13 pages of virtual memory were used to define 12 macros.

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

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	5

146 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

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



0165 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

EXTFCB LIS									
DELJL LIS	DIRGET LIS								
						EXTIOX LIS		IODONE LIS	LOCKDN LIS
				ENTER LIS					
							GETREQ LIS	GETTIM LIS	
			DISPAT LIS					INIFCP LIS	
	DIRFCB LIS					EXTHDR LIS			
		DIRSCN LIS							LOGDEL LIS
									LOCKDB LIS
						FIND LIS	GETFIB LIS		
DIRACC LIS								INIFCB LIS	
				EXTDIR LIS	EXTEND LIS				