



```

IIIIII 000000 DDDDDDDD 000000 NN NN EEEEEEEEE
IIIIII 000000 DDDDDDDD 000000 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EEEEEEEEE
IIIIII 000000 DDDDDDDD 000000 NN NN EEEEEEEEE
IIIIII 000000 DDDDDDDD 000000 NN NN EEEEEEEEE

```

```

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
LLLLLLLL IIIIII SSSSSSSS
LLLLLLLL IIIIII SSSSSSSS

```

```

0000 1      .TITLE IODONE - POST REQUEST DONE TO USER
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: F11ACP STRUCTURE LEVEL 2
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 :     THIS ROUTINE POSTS I/O COMPLETION FOR THE INDICATED FCP REQUEST.
0000 35
0000 36 : ENVIRONMENT:
0000 37
0000 38 :     STARLET OPERATING SYSTEM, INCLUDING PRIVILEGED SYSTEM SERVICES
0000 39 :     AND INTERNAL EXEC ROUTINES. THIS ROUTINE MUST BE CALLED IN
0000 40 :     KERNEL MODE.
0000 41
0000 42 :--
0000 43
0000 44 : AUTHOR: ANDREW C. GOLDSTEIN, CREATION DATE: 20-DEC-1976 11:25
0000 45
0000 46 : MODIFIED BY:
0000 47
0000 48 :     V03-010 CDS0005      Christian D. Saether      21-Aug-1984
0000 49 :     Ca'l CHECK_DISMOUNT routine before posting i/o
0000 50 :     completion.
0000 51
0000 52 :     V03-009 DAS0001      David Solomon      02-May-1984
0000 53 :     Fix truncation error.
0000 54
0000 55 :     V03-008 ACG0408      Andrew C. Goldstein,  23-Mar-1984 10:58
0000 56 :     Make all of global storage based
0000 57

```

```

0000 58 : V03-007 WMC0001 Wayne Cardoza 14-Mar-1984
0000 59 : This routine must call POSTEF because of changes in IOPOST.
0000 60 :
0000 61 : V03-006 CDS0004 Christian D. Saether 15-Jan-1984
0000 62 : Call to IOC$_BUFPOST should be IOC$BUFPOST.
0000 63 :
0000 64 : V03-005 CDS0003 Christian D. Saether 8-Dec-1983
0000 65 : For buffered i/o completion, execute iopost as
0000 66 : jsb's to appropriate routines rather than actually
0000 67 : post a software interrupt.
0000 68 :
0000 69 : V03-004 CDS0002 Christian D. Saether 24-Apr-1983
0000 70 : Fix truncation error.
0000 71 :
0000 72 : V03-003 ACG0320 Andrew C. Goldstein, 22-Mar-1983 12:41
0000 73 : Change byte count handling to track IOPOST changes
0000 74 :
0000 75 : V03-002 RSH0013 R. Scott Hanna 17-Mar-1983
0000 76 : Fix truncation error.
0000 77 :
0000 78 : V03-001 CDS0001 C Saether 31-Jul-1982
0000 79 : Make reference to IOC$GL_PSBL pic.
0000 80 :
0000 81 : V02-001 LJK0076 Lawrence J. Kenah 3-Nov-1981
0000 82 : Remove check for 'queue previously not empty' when making
0000 83 : software interrupt request. The request is always made.
0000 84 :
0000 85 : **
0000 86 :
0000 87 :
0000 88 : EQUATED SYMBOLS:
0000 89 :
00000004 0000 90 PACKET =4 : ADDRESS OF I/O PACKET ARG
0000 91 :
0000 92 $ABDDEF : DEFINE BUFFER PACKET OFFSETS
0000 93 $ACBDEF : AST CONTROL BLOCK OFFSETS
0000 94 $FIBDEF : DEFINE FIB OFFSETS
0000 95 $IRPDEF : DEFINE I/O PACKET OFFSETS
0000 96 $UCBDEF : DEFINE UCB OFFSETS
0000 97 $VCBDEF : DEFINE VCB OFFSETS
0000 98 $IPLDEF : DEFINE IPL SYMBOLS
0000 99 $IODEF : DEFINE I/O FUNCTION CODES
0000 100 $PRDEF : DEFINE PROCESSOR REGISTERS

```

```

0000 102 : **
0000 103 :
0000 104 : FUNCTIONAL DESCRIPTION:
0000 105 :
0000 106 :     THIS ROUTINE POSTS I/O COMPLETION FOR THE INDICATED FCP REQUEST.
0000 107 :
0000 108 : CALLING SEQUENCE:
0000 109 :     CALL     IODONE (ARG1)
0000 110 :
0000 111 : INPUT PARAMETERS:
0000 112 :     ARG1: ADDRESS OF I/O PACKET
0000 113 :
0000 114 : IMPLICIT INPUTS:
0000 115 :     USER_STATUS: STATUS OF I/O REQUEST
0000 116 :
0000 117 : OUTPUT PARAMETERS:
0000 118 :     NONE
0000 119 :
0000 120 : IMPLICIT OUTPUTS:
0000 121 :     IOC$GL_PSBL: TAIL OF I/O POST QUEUE
0000 122 :
0000 123 : ROUTINE VALUE:
0000 124 :     NONE
0000 125 :
0000 126 : SIDE EFFECTS:
0000 127 :     I/O PACKET PLACED ON I/O POST QUEUE
0000 128 :     VOLUME CHECKED FOR DISMOUNT
0000 129 :
0000 130 : --
0000 131 :
00000000 132 : .PSECT  $CODE$,NOWRT,LONG
0000 133 :
0000 134 IO_DONE::
0000 135 : .WORD  *M<R2,R3,R4,R5,R6,R7> ; SAVE REGISTERS
38 A5 55 04 AC 0000'CA 7D 0002 136 : MOVL  PACKET(AP),R5 ; GET PACKET ADDRESS
06 00 EF 000C 137 : MOVQ  W^USER_STATUS(R10),IRP$ MEDIA(R5) ; SET STATUS IN PACKET
57 20 A5 000F 138 : EXTZV #IRP$V_FCODE,#IRP$S_FCODE,-
0C 57 91 0012 139 : IRP$W_FUNC(R5),R7 ; GET FUNCTION CODE WITHOUT QUALIFIERS
0B 57 91 0015 140 : CMPB  R7,#IOS_READPBLK ; IF READ PHYSICAL
0B 57 91 0017 141 : BEQL  20$
0B 57 91 001A 142 : CMPB  R7,#IOS_WRITEPBLK ; OR WRITE DO SPECIAL PROCESSING
001C 143 : BEQL  20$
001C 144 :
001C 145 : POST PROCESSING FOR ALL ACP FUNCTIONS: BUMP DOWN THE VOLUME TRANSACTION
001C 146 : COUNT AND DO THE FIXUPS FOR THE BUFFER PACKET.
001C 147 :
56 1C A5 D0 001C 148 : MOVL  IRP$U_UCB(R5),R6 ; GET UCB ADDRESS
56 34 A6 D0 0020 149 : MOVL  UCBSL_VCB(R6),R6 ; TO GET VCB ADDRESS
2A 2A A5 03 E1 0024 150 : DECW  VCBSW_TRANS(R6) ; DEDUCT THIS REQ FROM TRANS COUNT
56 2C B5 D0 0027 151 : BBC   #IRP$V_COMPLX,IRP$W_STS(R5),10$ ; BRANCH IF NO BUFFER PACKET
56 12 A6 B4 002C 152 : MOVL  @IRP$S_SVAPTE(R5),R6 ; GET BUFFER DESCRIPTOR ADDRESS
52 08 A6 9E 0030 153 : CLRW  <ABD$C_NAME*ABD$C_LENGTH>+ABD$W_COUNT(R6)
53 62 3C 0033 154 : ; INHIBIT WRITE-BACK OF NAME STRING
52 08 A6 9E 0033 155 : MOVAB <ABD$C_FIB*ABD$C_LENGTH>+ABD$W_TEXT(R6),R2
53 53 C0 0037 156 : MOVZWL (R2),R3 ; GET OFFSET ADDRESS OF FIB IN PACKET
00 0000'CA 0040 8F 2C 003A 157 : ADDL  R3,R2 ; COMPUTE ABSOLUTE ADDRESS
0000 158 : MOVCS #FIB$C_LENGTH,W^LOCAL_FIB(R10),#0,-

```

```

01 A2 UA AA      0045 159      <ABDSC_FIB*ABDSC_LENGTH>*ABDSW_COUNT(R6),1(R2)
                   0049 160      : COPY_LOCAL FIB BACK INTO PACKET
04 55 04 AL      DC 0049 161      MOVL  PACKET(AP), R5      : RESTORE IRP ADDR TO R5
   2A A5 01      E2 004D 162      BBSS  #IRPSV_FUNC,IRPSW_STS(R5),10$ : IF READ BIT IS SET, KEEP
   32 A5 05      B0 0052 163      MOVW  #ABDSC_ATTRIB,IRPSW_BCNT(R5) ; ELSE DUMP ATTRIBUTE TEXT
                   0056 164
                   0056 165 :
                   0056 166 : POST COMPLETION WITHOUT ACTUALLY POSTING AN IOPOST SOFTWARE INTERRUPT.
                   0056 167 :
                   0056 168
44 2A A5 00      E1 0056 169 10$: BBC  #IRPSV_BUFIO, IRPSW_STS(R5), 50$ ; THIS HAD BETTER BE A
00000000'EF 00    FB 005B 170      : BUFFERED I/O.
                   005B 171      CALLS #U,L^CHECK DISMOUNT : CHECK THE VOLUME FOR DISMOUNT
                   0062 172      TSTL  IRPSL_PID (R5)      : IF NEGATIVE, THIS DOES SPECIAL
                   0065 173      BLSS  30$              : IOPOST PROCESSING.
                   0067 174
54 00000000'GF  D0 0067 175      MOVL  G^CTL$GL_PCB, R4      : OUR PCB ADDRESS INTO R4.
                   006E 176
                   006E 177      SETIPL #IPL$ ASTDEL      : BLOCK AST DELIVERY
                   0071 178      JSB   G^IOC$BUFPOST      : DO FIRST PART OF IOPOST
                   51 0C A5  D0 0077 179      MOVL  IRPSL_PID(R5),R1      : PROCESS IDENTIFICATION
                   53 22 A5  9A 007B 180      MOVZBL IRPSB_EFN(R5),R3     : GET EVENT FLAG NUMBER
00000000'GF 16 007F 181      JSB   G^SCH$POSTEF      : AND POST IT
                   18 B5 16 0085 182      JSB   @ACBSL_KAST (R5)     : DO SPECIAL KERNEL AST PART OF IOPOST.
                   0088 183      SETIPL #0              : BACK TO IPL 0.
                   04 008B 184      RET                               : AND EXIT.
                   008C 185
                   008C 186 :
                   008C 187 : FOR READ/WRITE PHYSICAL, KNOCK DOWN THE VIRTUAL BIT IN THE PACKET. ONLY
                   008C 188 : ERRORS COME THROUGH HERE, AND WE DON'T WANT TO SEE THEM AGAIN (I/O POST
                   008C 189 : RECYCLES VIRTUAL I/O ERRORS FOR ACP ERROR PROCESSING).
                   008C 190 :
                   008C 191
                   2A A5 10 8A 008C 192 20$: ASSUME IRPSV VIRTUAL LE 7
                   0090 193      BICB  #IRPSM_VIRTUAL,IRPSW_STS(R5) ; CLEAR THE VIRTUAL BIT
50 00000000'GF 9E 0090 194 30$: MOVAB  G^IOC$GL_PSBL, R0      : GET ADDRESS OF BACK LINK
   00 B0 65    OE 0097 195      INSQUE (R5), @ (R0)      : INSERT PACKET INTO QUEUE
                   009B 196      SOFTINT #IPL$_IOPOST : SIGNAL I/O POST INTERRUPT
                   04 009E 197      RET
                   009F 198
                   009F 199 50$: BUG_CHECK      XQPERR      : BUFFERED I/O WAS EXPECTED.
                   00A3 200
                   00A3 201      .END

```

IODONE  
Symbol table

- POST REQUEST DONE TO USER

D 2

15-SEP-1984 23:43:02 VAX/VMS Macro V04-00  
5-SEP-1984 01:13:24 [F11X.SRC]IODONE.MAR;1

Page 5  
(2)

ABDSC_ATTRIB	=	00000005		
ABDSC_FIB	=	00000001		
ABDSC_LENGTH	=	00000008		
ABDSC_NAME	=	00000002		
ABDSW_COUNT	=	00000002		
ABDSW_TEXT	=	00000000		
ACBSL_KAST	=	00000018		
ACL_TYPE	=	00000007		
AQB_TYPE	=	00000005		
BITMAP_TYPE	=	00000001		
BUGS_XOPERR	=	*****	X	02
CACHE_TYPE	=	00000006		
CHECK_DISMOUNT	=	*****	X	02
CHIP_TYPE	=	00000008		
CTLSGL_PCB	=	*****	X	02
DATA_TYPE	=	00000004		
DIRECTORY_TYPE	=	00000002		
FCB_TYPE	=	00000000		
FIBSC_LENGTH	=	00000040		
HEADER_TYPE	=	00000000		
INDEX_TYPE	=	00000003		
IOS_READPBLK	=	0000000C		
IOS_WRITEPBLK	=	0000000B		
IOCSBUFPOST	=	*****	X	02
IOC^GL_PSB	=	*****	X	02
IO_DONE	=	00000000	RG	02
IPLS_ASTDEL	=	00000002		
IPLS_IOPOST	=	00000004		
IRPSB_EFN	=	00000022		
IRPSL_MEDIA	=	00000038		
IRPSL_PID	=	0000000C		
IRPSL_SVAPE	=	0000002C		
IRPSL_UCB	=	0000001C		
IRPSM_VIRTUAL	=	00000010		
IRPSV_FCODE	=	00000006		
IRPSV_BUFIO	=	00000000		
IRPSV_COMPLX	=	00000003		
IRPSV_FCODE	=	00000000		
IRPSV_FUNC	=	00000001		
IRPSV_VIRTUAL	=	00000004		
IRPSW_BCNT	=	00000032		
IRPSW_FUNC	=	00000020		
IRPSW_STS	=	0000002A		
LOCAL_FIB	=	*****	X	02
MVL_TYPE	=	00000004		
PACKET	=	00000004		
PRB_IPL	=	00000012		
PRB_SIRR	=	00000014		
QUOTA_TYPE	=	00000005		
RVT_TYPE	=	00000003		
SCHSPOSTEF	=	*****	X	02
UCBSL_VCB	=	00000034		
USER_STATUS	=	*****	X	02
VCBSQ_TRANS	=	0000000C		
VCB_TYPE	=	00000002		
WCB_TYPE	=	00000001		

LOCK  
Symb

SST1  
ACL\_  
AOB\_  
BITM  
CACH  
CHIP  
DATA  
DIRE  
FCB\_  
HEAD  
INCE  
LAST  
LCOD  
LCOD  
LDAT  
LDAT  
LOCK  
LOCK  
LOCK  
MVL  
QUOT  
RVT\_  
SET\_  
SYSS  
SYSS  
SYSS  
VCB\_  
WCB\_  
WORK

PSEC  
----  
A  
\$LOC  
\$LOC  
\$LOC  
\$LOC  
\$LOC  
\$LOC

Phas  
----  
Init  
Comm  
Pass  
Symb  
Pass  
Symb  
Psec

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$A95\$	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$CODE\$	000000A3 ( 163.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.63
Command processing	118	00:00:00.66	00:00:04.63
Pass 1	355	00:00:11.68	00:00:25.70
Symbol table sort	0	00:00:02.16	00:00:03.54
Pass 2	55	00:00:02.10	00:00:04.91
Symbol table output	7	00:00:00.10	00:00:00.25
Psect synopsis output	2	00:00:00.03	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	569	00:00:16.82	00:00:39.71

The working set limit was 1350 pages.  
66502 bytes (130 pages) of virtual memory were used to buffer the intermediate code.  
There were 80 pages of symbol table space allocated to hold 1366 non-local and 4 local symbols.  
302 source lines were read in Pass 1, producing 13 object records in Pass 2.  
22 pages of virtual memory were used to define 21 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	9
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	16

1453 GETS were required to define 16 macros.

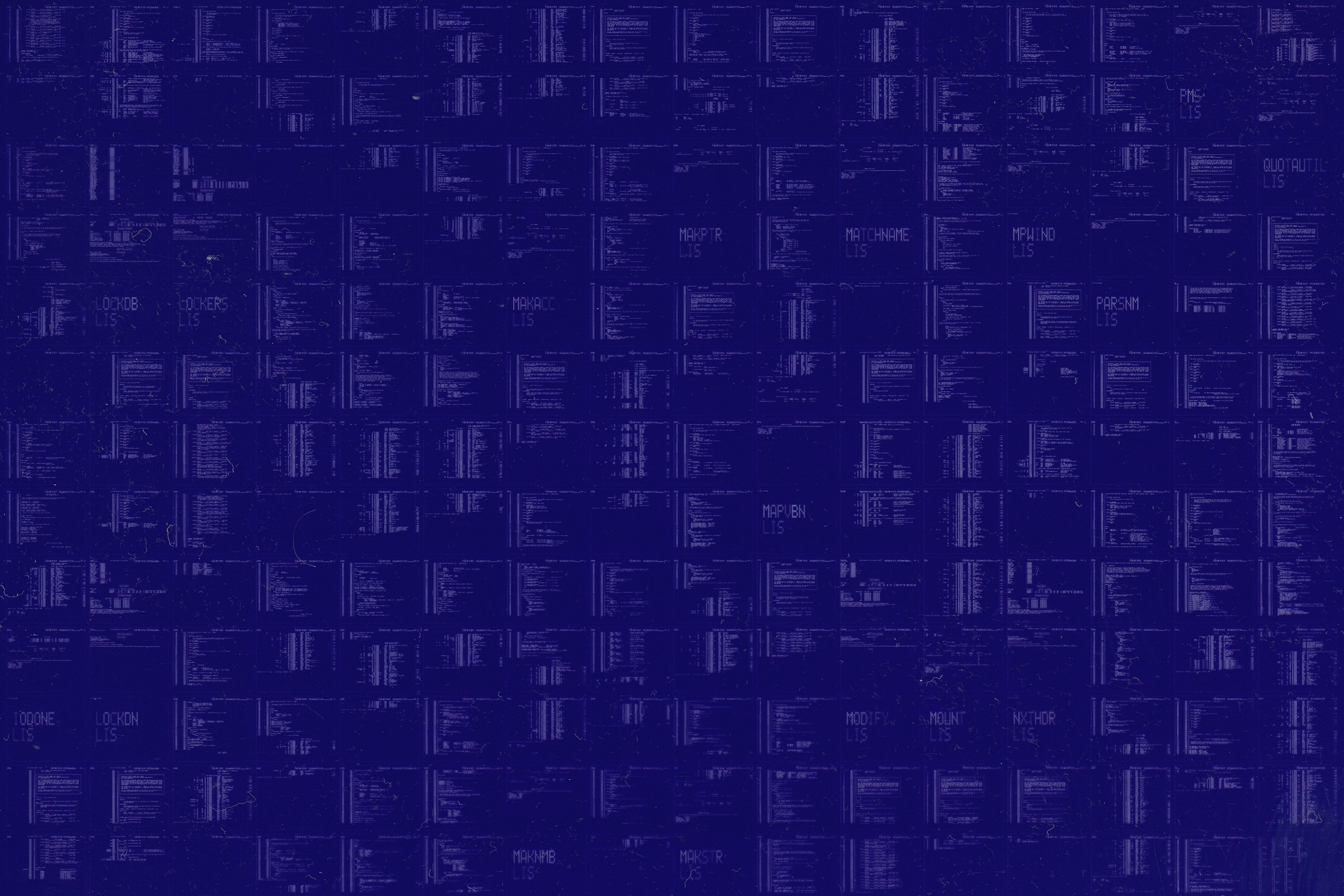
There were no errors, warnings or information messages.

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



0171 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY



PMS  
LIS

QUOTAUTIL  
LIS

MAKPTR  
LIS

MATCHNAME  
LIS

MPWIND  
LIS

LOCKDB  
LIS

LOCKERS  
LIS

MAKACC  
LIS

PARSNM  
LIS

MAPVBN  
LIS

TODONE  
LIS

LOCKDN  
LIS

MODIFY  
LIS

MOUNT  
LIS

MYTHOR  
LIS

MAKNMB  
LIS

MAKSTR  
LIS