





```

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 1
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 V03-002 HH0051 Hai Huang 21-Aug-1984
0000 49 Call check_dismount before posting I/O completion.
0000 50
0000 51 V03-001 ACG0320 Andrew C. Goldstein, 22-Mar-1983 12:25
0000 52 Change byte count handling to track IOPOST
0000 53
0000 54 V02-001 LJK0076 Lawrence J. Kenah 3-Nov-1981
0000 55 Remove check for "queue previously not empty" when making
0000 56 software interrupt request. The request is always made.
0000 57

```

```
0000 58 ;**
0000 59
0000 60 :
0000 61 : EQUATED SYMBOLS:
0000 62 :
00000004 0000 63 PACKET =4 ; ADDRESS OF I/O PACKET ARG
0000 64
0000 65 $ABDDEF ; DEFINE BUFFER PACKET OFFSETS
0000 66 $FIBDEF ; DEFINE FIB OFFSETS
0000 67 $IRPDEF ; DEFINE I/O PACKET OFFSETS
0000 68 $UCBDEF ; DEFINE UCB OFFSETS
0000 69 $VCBDEF ; DEFINE VCB OFFSETS
0000 70 $IPLDEF ; DEFINE IPL SYMBOLS
0000 71 $PRDEF ; DEFINE PRIORITY LEVELS
0000 72 $IODEF ; DEFINE I/O FUNCTION CODES
```

```

0000 74 :++
0000 75 :
0000 76 : FUNCTIONAL DESCRIPTION:
0000 77 :
0000 78 :     THIS ROUTINE POSTS I/O COMPLETION FOR THE INDICATED FCP REQUEST.
0000 79 :
0000 80 : CALLING SEQUENCE:
0000 81 :     CALL     IODONE (ARG1)
0000 82 :
0000 83 : INPUT PARAMETERS:
0000 84 :     ARG1: ADDRESS OF I/O PACKET
0000 85 :
0000 86 : IMPLICIT INPUTS:
0000 87 :     USER_STATUS: STATUS OF I/O REQUEST
0000 88 :
0000 89 : OUTPUT PARAMETERS:
0000 90 :     NONE
0000 91 :
0000 92 : IMPLICIT OUTPUTS:
0000 93 :     IOC$GL_PSBL: TAIL OF I/O POST QUEUE
0000 94 :
0000 95 : ROUTINE VALUE:
0000 96 :     NONE
0000 97 :
0000 98 : SIDE EFFECTS:
0000 99 :     I/O PACKET PLACED ON I/O POST QUEUE
0000 100 :     VOLUME CHECKED FOR DISMOUNT
0000 101 :
0000 102 : --
0000 103 :

```

00000000 104 .PSECT \$CODE\$,NOWRT, LONG

106 IO\_DONE::

```

00FC 0000 107 .WORD ^M<R2,R3,R4,R5,R6,R7> ; SAVE REGISTERS
38 A6 56 04 AC D0 0002 108 MOVL PACKET(AP),R6 ; GET PACKET ADDRESS
0000'CF 7D 0006 109 MOVQ W^USER_STATUS,IRPSL_MEDIA(R6) ; SET STATUS IN PACKET
06 00 EF 000C 110 EXTZV #IRPSV_FCODE,#IRPSS_FCODE,-
57 20 A6 000F 111 IRPSW_FUNC(R6),R7 ; GET FUNCTION CODE WITHOUT QUALIFIERS
0C 57 91 0012 112 CMPB R7,#IOS_READPBLK ; IF READ PHYSICAL
0B 57 91 0015 113 BEQL 10$
0B 57 91 0017 114 CMPB R7,#IOS_WRITEPBLK ; OR WRITE DO SPECIAL PROCESSING
35 13 001A 115 BEQL 10$
001C 116 :
001C 117 : POST PROCESSING FOR ALL ACP FUNCTIONS: BUMP DOWN THE VOLUME TRANSACTION
001C 118 : COUNT AND DO THE FIXUPS FOR THE BUFFER PACKET.
001C 119 :
54 0000'CF D0 001C 120 MOVL W^CURRENT_VCB,R4 ; GET VCB ADDRESS
0C A4 B7 0021 121 DECW VCB$W_TRANS(R4) ; DEDUCT THIS REQ FROM TRANS COUNT
2C 2A A6 03 E1 0024 122 BBC #IRPSV_COMPLX,IRPSW_STS(R6),30$ ; BRANCH IF NO BUFFER PACKET
54 2C B6 D0 0029 123 MOVL @IRPSL_SVAPTE(R6),R4 ; GET BUFFER DESCRIPTOR ADDRESS
12 A4 B4 002D 124 CLRW <ABD$C_NAME*ABD$C_LENGTH>+ABD$W_COUNT(R4)
0030 125 ; INHIBIT WRITE-BACK OF NAME STRING
52 08 A4 9E 0030 126 MOVAB <ABD$C_FIB*ABD$C_LENGTH>+ABD$W_TEXT(R4),R2
53 62 3C 0034 127 MOVZWL (R2),R3 ; GET OFFSET ADDRESS OF FIB IN PACKET
52 53 C0 0037 128 ADDL R3,R2 ; COMPUTE ABSOLUTE ADDRESS
00 0000'CF 0040 8F 2C 003A 129 MOVCS #FIB$C_LENGTH,W^LOCAL_FIB,#0,-
01 A2 0A A4 0042 130 <ABD$C_FIB*ABD$C_LENGTH>+ABD$W_COUNT(R4),1(R2)

```

```

0A 2A A6 01 E2 0046 131 ; COPY LOCAL FIB BACK INTO PACKET
  32 A6 05 B0 0046 132 BBSS #IRPSV_FUNC,IRPSW_STS(R6),30$ ; IF READ BIT IS SET, KEEP
  04 11 004B 133 MOVW #ABDSC_ATTRIB,IRPSW_BCNT(R6) ; ELSE DUMP ATTRIBUTE TEXT
  0051 134 BRB 30$
  0051 135 :
  0051 136 : FOR READ/WRITE PHYSICAL, KNOCK DOWN THE VIRTUAL BIT IN THE PACKET. ONLY
  0051 137 : ERRORS COME THROUGH HERE, AND WE DON'T WANT TO SEE THEM AGAIN (I/O POST
  0051 138 : RECYCLES VIRTUAL I/O ERRORS FOR ACP ERROR PROCESSING).
  0051 139 :
  2A A6 10 8A 0051 140 ASSUME IRPSV_VIRTUAL LE 7
  0055 141 10$: BICB #IRPSM_VIRTUAL,IRPSW_STS(R6) ; CLEAR THE VIRTUAL BIT
  0000'CF 00 FB 0055 142
  0055 143 30$: CALLS #0,W^CHECK_DISMOUNT ; CHECK THE VOLUME FOR DISMOUNT
  005A 144
  00000000'FF 66 0E 005A 145 INSQUE (R6),@IIOC$GL_PSBL ; INSERT PACKET INTO QUEUE
  0061 146 SOFTINT #IPL$_IOPOST ; SIGNAL I/O POST INTERRUPT
  04 0064 147 RET
  0065 148
  0065 149
  0065 150
  0065 151 .END

```

LOI  
Syr  
AQE  
BIT  
DIF  
FCE  
HE/  
INC  
LCC  
LCC  
LD/  
LD/  
LOC  
LOC  
LOC  
MVI  
RVI  
SYI  
VCE  
WCE

PSI  
---

\$L  
\$L  
\$L  
\$L  
\$L  
\$C

Phi  
---

In  
Com  
Par  
Syr  
Par  
Syr  
Psi  
Cre  
As

Th  
25  
Th  
22  
6

IODONE  
Symbol table

- POST REQUEST DONE TO USER

I 15

16-SEP-1984 00:42:32 VAX/VMS Macro V04-00  
5-SEP-1984 01:08:00 [F11A.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
AQB_TYPE         = 00000005
BITMAP_TYPE      = 00000001
CHECK_DISMOUNT   ***** X 02
CURRENT_VCB      ***** X 02
DIRECTORY_TYPE   = 00000002
FCB_TYPE         = 00000000
FIBSC_LENGTH     = 00000040
HEADER_TYPE      = 00000000
INDEX_TYPE       = 00000003
IOS_READPBLK     = 0000000C
IOS_WRITEPBLK    = 0000000B
IOCSGL_PSBL      ***** X 02
IO_DONE          00000000 RG 02
IPCS_IOPOST      = 00000004
IRPSC_MEDIA      = 00000038
IRPSL_SVAPE     = 0000002C
IRPSM_VIRTUAL    = 00000010
IRPSS_FCODE      = 00000006
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
PRS_SIRR         = 00000014
RVT_TYPE         = 00000003
USER_STATUS      ***** X 02
VCBSQ_TRANS      = 0000000C
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
\$CODE\$	00000065 ( 101.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

LO  
VA

Ma  
-S  
-S  
TO  
14  
Th  
MA

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.09	00:00:00.52
Command processing	148	00:00:00.76	00:00:04.45
Pass 1	340	00:00:11.62	00:00:29.59
Symbol table sort	0	00:00:02.09	00:00:03.19
Pass 2	45	00:00:01.88	00:00:04.49
Symbol table output	6	00:00:00.09	00:00:00.13
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	575	00:00:16.55	00:00:42.46

The working set limit was 1200 pages.  
64141 bytes (126 pages) of virtual memory were used to buffer the intermediate code.  
There were 70 pages of symbol table space allocated to hold 1336 non-local and 2 local symbols.  
254 source lines were read in Pass 1, producing 13 object records in Pass 2.  
21 pages of virtual memory were used to define 20 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	6
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	13

1415 GETS were required to define 13 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



