


```

IIIIII 000000 DDDDDDDD 000000 NN NN EEEEEEEEE
IIIIII 000000 DDDDDDDD 000000 NN NN EEEEEEEEE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
II      00      00 DD      DD 00      00 NN NN EE
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
0000 31 Facility: f11acp structure level 1
0000 32
0000 33 Abstract:
0000 34
0000 35 this routine posts i/o completion for the indicated fcp request.
0000 36
0000 37 Environment:
0000 38
0000 39 starlet operating system, including privileged system services
0000 40 and internal exec routines. this routine must be called in
0000 41 kernel mode.
0000 42
0000 43 Author: andrew c. goldstein, Creation Date: 20-DEC-1976 11:25
0000 44
0000 45 Modified By:
0000 46
0000 47 V03-001 MMD0130 Meg Dumont, 8-Apr-1983 14:22
0000 48 Change the call to check dismount to happen before
0000 49 returning the IO to the requesting process.
0000 50
0000 51 V02-003 DMW00049 David Michael Walp 6-Nov-1981
0000 52 Lawrence J. Kenah fix to posting I/O done hang.
0000 53 Removed check for "que previously not empty" when making
0000 54 software interrupt request. The request is always made.
0000 55
0000 56 V02-002 REFORMAT Keith B. Thompson 30-Jul-1980
0000 57

```

```

0000 58 :      A0101  DGH0002      D. H. Gillespie      15-NOV-1978 17:00
0000 59 :      fix race condition between post io and check dismount
0000 60 :--
0000 61 :
0000 62 :
0000 63 : Equated Symbols:
0000 64 :
0000 65 :
00000004 0000 66      PACKET =4      ; address of i/o packet arg
0000 67
0000 68      $ABDDEF      ; define buffer packet offsets
0000 69      $DEVDEF      ; define device characteristic bits
0000 70      $FIBDEF      ; define fib offsets
0000 71      $IRPDEF      ; define i/o packet offsets
0000 72      $UCBDEF      ; define ucb offsets
0000 73      $VCBDEF      ; define vcb offsets
0000 74      $IPLDEF      ; define ipl symbols
0000 75      $PRDEF      ; define priority levels
0000 76      $IODEF      ; define i/o function codes

```



```

0000 78
0000 79 :++
0000 80 :
0000 81 : IOSDONE - this routine posts i/o completion for the indicated
0000 82 : fcp request.
0000 83 :
0000 84 : Calling sequence:
0000 85 : call iodone (arg1)
0000 86 :
0000 87 : Input Parameters:
0000 88 : arg1: address of i/o packet
0000 89 :
0000 90 : Implicit Inputs:
0000 91 : USER_STATUS: status of i/o request
0000 92 :
0000 93 : Output Parameters:
0000 94 : none
0000 95 :
0000 96 : Implicit Outputs:
0000 97 : IOC$GL_PSBL: tail of i/o post queue
0000 98 :
0000 99 : Routine Value:
0000 100 : none
0000 101 :
0000 102 : Side Effects:
0000 103 : i/o packet placed on i/o post queue
0000 104 : operation counted in ucb
0000 105 : volume checked for dismount
0000 106 :
0000 107 :--
0000 108 :
00000000 109 .PSECT $CODE$,NOWRT, LONG
0000 110
0000 111 IO_DONE::
0000 112 .WORD ^M<R2,R3,R4,R5,R6,R7,R8> ; save registers
38 A6 56 04 AC 01FC 0002 113 MOVL PACKET(AP),R6 ; get packet address
58 1C A6 7D 0006 114 MOVQ W^USER_STATUS,IRP$L_MEDIA(R6) ; set status in packet
54 34 A8 D0 000C 115 MOVL IRP$L_UCB(R6),R8 ; get ucb address
70 A8 D6 0014 116 MOVL UCB$L_VCB(R8),R4 ; get vcb address
06 00 EF 0017 117 INCL UCB$L_OPCNT(R8) ; count completed operation
57 20 A6 001A 118 EXTZV #IRP$L_FCODE,#IRP$L_FCODE,- ; get function code without qualifie
0C 57 91 001D 120 CMPB R7,#IOS_READPBLK ; if read physical
0B 35 13 0020 121 BEQL 10$
0B 57 91 0022 122 CMPB R7,#IOS_WRITEPBLK ; or write do special processing
30 13 0025 123 BEQL 10$
0027 124
0027 125 :
0027 126 : post processing for all acp functions: bump down the volume transaction
0027 127 : count and do the fixups for the buffer packet.
0027 128 :
0027 129 :
0C A4 B7 0027 130 DECW VCB$W_TRANS(R4) ; deduct this req from trans
002A 131 ; count
37 2A A6 03 E1 002A 132 BBC #IRP$L_COMPLX,IRP$W_STS(R6),30$ ; branch if no buffer packet
54 2C B6 D0 002F 133 MOVL @IRP$L_SVAPTE(R6),R4 ; get buffer descriptor address
12 A4 B4 0033 134 CLRW <ABD$C_NAME+ABD$C_LENGTH>+ABD$W_COUNT(R4) ; inhibit write

```

```

0036 135
0036 136 ; back of name
52 08 A4 9E 0036 137 ; string
53 62 3C 003A 138
003D 139
00 0000*CF 52 53 C0 003D 140
0040 8F 2C 0040 141
0A A4 0048 142
01 A2 004A 143
15 2A A6 01 E2 004C 144
32 A6 05 B0 0051 145
OF 11 0055 146
0057 147
0057 148
0057 149 ; for read/write physical, knock down the virtual bit in the packet. only
0057 150 ; errors come through here, and we don't want to see them again (i/o post
0057 151 ; recycles virtual i/o errors for acp error processing). we also restore
0057 152 ; the original svapte and byte count from where they were saved in the packet
0057 153 ; so that the pages can be unlocked.
0057 154
0057 155
00 2A 04 E5 0057 156 10$: BBCC #IRPSV_VIRTUAL,- ; clear the virtual bit
4C A6 0059 157 IRPSW_STS(R6),20$
2C A6 D0 005C 158 20$: MOVL IRPSL_DIAGBUF(R6),- ; restore original page address
32 A6 44 A6 B0 005F 159 IRPSL_SVAPTE(R6)
0061 160 MOVW IRPSW_OBCNT(R6),IRPSW_BCNT(R6) ; and byte count for page unlock
0066 161
0066 162 ; The call to check dismount must happen before the IO is returned to the user.
0066 163 ; This is true to avoid the race condition that occurs when a process with
0066 164 ; a priority higher then the MTAACP allocates and mounts a tape drive,
0066 165 ; then logs out without dismounting and deallocating the drive. Because
0066 166 ; deallocates sees that the drive is still owned by a process the deallocate
0066 167 ; fails. However, process deletion still occurs and the tape is eventually
0066 168 ; dismounted, thus leaving the drive allocated to a nonexistant process.
0066 169
00000000*EF 58 DD 0066 170 30$: PUSHL R8
00000000*FF 01 FB 0068 171 CALLS #1,CHECK_DISMOUNT ; put check here to stop race
0E 006F 172 INSQUE (R6),@IOC$GL_PSBL ; insert packet into queue
0076 173 SOFTINT #IPL$_IOPOST ; signal i/o post interrupt
0079 174 ; with user deallocate
04 0079 175 RET
007A 176
007A 177
007A 178
007A 179 .END

```


IODONE
Symbol table

- POST REQUEST DONE TO USER

M 3

16-SEP-1984 02:04:11
5-SEP-1984 02:11:55

VAX/VMS Macro V04-00
[MTAACP.SRC]IODONE.MAR;1

Page 5
(3)

```

ABD$C_ATTRIB      = 00000005
ABD$C_FIB         = 00000001
ABD$C_LENGTH     = 00000008
ABD$C_NAME       = 00000002
ABD$W_COUNT      = 00000002
ABD$W_TEXT       = 00000000
AQB_TYPE         = 00000005
CHECK_DISMOUNT   = ***** X 02
FCB_TYPE         = 00000000
FIB$C_LENGTH     = 00000040
IOS_READPBLK    = 0000000C
IOS_WRITEPBLK   = 0000000B
IOC$GL_PSBL     = ***** X 02
IO_DONE         = 00000000 RG 02
IP$S_IOPOST     = 00000004
IRP$C_DIAGBUF   = 0000004C
IRP$S_MEDIA     = 00000038
IRP$S_SVAPTE    = 0000002C
IRP$S_UCB       = 0000001C
IRP$S_FCODE     = 00000006
IRP$V_COMPLX    = 00000003
IRP$V_FCODE     = 00000000
IRP$V_FUNC      = 00000001
IRP$V_VIRTUAL   = 00000004
IRP$W_BCNT      = 00000032
IRP$W_FUNC      = 00000020
IRP$W_OBCNT     = 00000044
IRP$W_STS       = 0000002A
LOCAL_FIB       = ***** X 02
MVL_TYPE        = 00000004
PACKET          = 00000004
PR$S_SIRR       = 00000014
RVT_TYPE        = 00000003
UCB$S_OPCNT     = 00000070
UCB$S_VCB       = 00000034
USER_STATUS     = ***** X 02
VCB$S_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
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$CODE\$	0000007A (122.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

LOC
Sym

\$\$T
AQB
FCB
LAS
LCO
LCO
LDA
LDA
LOCI
LOCI
LOCI
MVL
RVT
SET
SYS
SYS
SYS
VCB
WCB
WOR

PSE

\$.LO
\$.LO
\$.LO
\$.LO
\$.LO
\$.CO

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
406
The
344
12

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.34
Command processing	108	00:00:00.63	00:00:04.63
Pass 1	404	00:00:12.11	00:00:37.36
Symbol table sort	11	00:00:02.17	00:00:05.43
Pass 2	81	00:00:02.14	00:00:08.29
Symbol table output	6	00:00:00.07	00:00:00.07
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	644	00:00:17.21	00:00:57.17

The working set limit was 1200 pages.
68512 bytes (134 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1408 non-local and 3 local symbols.
362 source lines were read in Pass 1, producing 13 object records in Pass 2.
24 pages of virtual memory were used to define 22 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

1490 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:IODONE/OBJ=OBJ\$:IODONE MSRC\$:MTADEF1/UPDATE=(ENH\$:MTADEF1)+MSRC\$:IODONE/UPDATE=(ENH\$:IODONE)+EXECMLS/LIB

