


```
1 0001 0
2 0002 0 MODULE STRVIO (LANGUAGE (BLISS32) ,
3 0003 0 IDENT = 'V04-000' ,
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1 ++
31 0031 1
32 0032 1 FACILITY: MTAACP
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1 This module requeues block virtual IO on the current unit
36 0036 1
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 STARLET operating system, including privileged system services
41 0041 1 and internal exec routines.
42 0042 1
43 0043 1 --
44 0044 1
45 0045 1
46 0046 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 30-AUG-1977
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1 V03-002 MMD0002 Meg Dumont, 9-Feb-1983 15:45
51 0051 1 Clean up of START_VIO, so that no locations in paged
52 0052 1 memory are accessed while we are at FIPL. Also
53 0053 1 fix to always set the UCB address before calling
54 0054 1 EXESINSIOQ.
55 0055 1
56 0056 1 V03-001 MMD0001 Meg Dumont, 3-Jan-1983 16:18
57 0057 1 Added routine that is called when the MTAACP has read a trailer
```

```

: 58      0058 1 |
: 59      0059 1 |          label. Its purpose is to stop accidental reading or writing
: 60      0060 1 |          of the trailer labels.
: 61      0061 1 |          V02-002 REFORMAT          Maria del C. Nasr          30-Jun-1980
: 62      0062 1 |
: 63      0063 1 |          **
: 64      0064 1 |
: 65      0065 1 | LIBRARY 'SYSSLIBRARY:LIB.L32';
: 66      0066 1 |
: 67      0067 1 | REQUIRE 'SRCS:MTADEF.B32';
: 68      0451 1 |
: 69      0452 1 | LINKAGE
: 70      0453 1 |     INS_QUE      = JSB (REGISTER = 3, REGISTER = 5) : NOPRESERVE (1, 2, 4);
: 71      0454 1 |
: 72      0455 1 | EXTERNAL
: 73      0456 1 |     CURRENT_UCB : REF BBLOCK,          ! address of current unit control block
: 74      0457 1 |     CURRENT_WCB : REF BBLOCK,          ! address of current window control block
: 75      0458 1 |     QUEUE_HEAD  : REF BBLOCK;         ! address of ACP input queue head
: 76      0459 1 |
: 77      0460 1 | EXTERNAL ROUTINE
: 78      0461 1 |     EXE$INSIOQ  : INS_QUE ADDRESSING_MODE (ABSOLUTE);
: 79      0462 1 |
: 80      0463 1 | LOCK_CODE;
: 81      0464 1 |

```

```

83 0465 1 GLOBAL ROUTINE START_VIO : COMMON_CALL NOVALUE =
84 0466 1
85 0467 1 ++
86 0468 1
87 0469 1 FUNCTIONAL DESCRIPTION:
88 0470 1     This routine queues blocked virtual IO to the current unit
89 0471 1
90 0472 1 CALLING SEQUENCE:
91 0473 1     START_VIO(), called in kernel mode
92 0474 1
93 0475 1 INPUT PARAMETERS:
94 0476 1     none
95 0477 1
96 0478 1 IMPLICIT INPUTS:
97 0479 1     CURRENT_UCB      - address of current unit control block to which all
98 0480 1                   blocked virtual IO is to be queued
99 0481 1     CURRENT_VCB      - address of current volume control block which contains
100 0482 1                   the blocked virtual IO list head
101 0483 1     CURRENT_WCB      - address of current window control block which is currently
102 0484 1                   not mapping virtual IO
103 0485 1     QUEUE_HEAD       - address of ACP input queue head to which mapping errors
104 0486 1                   for this volume are queued
105 0487 1
106 0488 1 OUTPUT PARAMETERS:
107 0489 1     none
108 0490 1
109 0491 1 IMPLICIT OUTPUTS:
110 0492 1     window restored mapping to current unit
111 0493 1     virtual IO requeued to that unit
112 0494 1
113 0495 1 ROUTINE VALUE:
114 0496 1     none
115 0497 1
116 0498 1 SIDE EFFECTS:
117 0499 1     this routine runs at fork level in order to synchronize with the mapping
118 0500 1     portion of QIO processing and the magnetic tape driver
119 0501 1
120 0502 1 --
121 0503 1
122 0504 2 BEGIN
123 0505 2
124 0506 2 EXTERNAL REGISTER
125 0507 2 COMMON_REG;
126 0508 2
127 0509 2 LOCAL
128 0510 2 ENTRY      : REF BBLOCK,  ! address of ACP queue entry
129 0511 2 FUNCTION,   ! IO function code
130 0512 2 PACKET     : REF BBLOCK,  ! address of IO request packet which is to be requeued
131 0513 2 UCB        : REF BBLOCK,  ! address of current UCB
132 0514 2 VCB        : REF BBLOCK,  ! address of current volume control block
133 0515 2 WCB        : REF BBLOCK,  ! address of the window control block
134 0516 2 QUEUE_ENTRY : REF BBLOCK, ! address of ACP input queue
135 0517 2 UCB_SAVE   : REF BBLOCK;  ! address of saved UCB
136 0518 2
137 0519 2 ! All data structures used in these routine must be accessible from
138 0520 2 ! local storage. Because we raise to fork IPL we can not afford to
139 0521 2 ! cause a page fault.

```

```

140 0522 2
141 0523 2
142 0524 2
143 0525 2
144 0526 2
145 0527 2
146 0528 2
147 0529 2
148 0530 2
149 0531 2
150 0532 2
151 0533 2
152 0534 2
153 0535 2
154 0536 2
155 0537 2
156 0538 2
157 0539 2
158 0540 2
159 0541 2
160 0542 2
161 0543 2
162 0544 2
163 0545 2
164 0546 2
165 0547 2
166 0548 2
167 0549 2
168 0550 2
169 0551 2
170 0552 2
171 0553 2
172 0554 2
173 0555 2
174 0556 2
175 0557 2
176 0558 2
177 0559 2
178 0560 2
179 0561 2
180 0562 2
181 0563 2
182 0564 2
183 0565 2
184 0566 2
185 0567 2
186 0568 2
187 0569 2
188 0570 2
189 0571 2
190 0572 2
191 0573 2
192 0574 2
193 0575 2
194 0576 2
195 0577 2
196 0578 2

```

```

UCB = .CURRENT_UCB;          ! address of UCB to which io is to be queue
VCB = .CURRENT_VCB;          ! address of current VCB
WCB = .CURRENT_WCB;          ! address of current WCB
QUEUE_ENTRY = .QUEUE_HEAD;  ! address of ACP input queue
UCB_SAVE = .CURRENT_UCB;     ! Address of current UCB
SET_IPL(.UCB[UCB$B_FIPL]);    ! raise to fork level

! fix up map pointer with UCB which is to receive virtual IO
!
WCB[WCB$W_NMAP] = 1;
(WCB[WCB$W_P1_COUNT])<0, 32> = .UCB;

! requeue all blocked io to current unit
!
WHILE 1
DO
  BEGIN
    IF REMQUE(.VCB[VCB$B_BLOCKFL], PACKET)
    THEN
      EXITLOOP;

    ! may have been cleared when error was processed
    !
    PACKET[IRP$V_VIRTUAL] = 1;

    ! this is here because INSIOQ does not preserve R5
    !
    UCB = .UCB_SAVE;
    EXE$INSIOQ(.PACKET, .UCB);
    END;

    ! Scan input queue for any mapping errors that belong to this volume
    ! put them at the tail of blocked IO list. Also the MSCP tape drives
    ! outstanding I/O's will be found in this queue under most circumstances.

    ENTRY = .QUEUE_ENTRY[AQB$B_ACPQFL];

    WHILE .ENTRY NEQA .QUEUE_ENTRY
    DO
      BEGIN
        FUNCTION = .ENTRY[IRP$V_FCODE];

        IF .FUNCTION EQL IOS_READPBLK
        OR
        .FUNCTION EQL IOS_WRITEPBLK
        THEN
          BEGIN
            BEGIN
              IF .BBLOCK[.ENTRY[IRP$L_UCB], UCB$L_VCB] EQLA .VCB
              THEN
                BEGIN
                  ENTRY = .ENTRY[IRP$L_IOQBL];
                  REMQUE(.ENTRY[IRP$L_IOQFL], PACKET);
                END;
              END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```

: 197 0579 S
: 198 0580 S
: 199 0581 S
: 200 0582 S
: 201 0583 S
: 202 0584 S
: 203 0585 S
: 204 0586 S
: 205 0587 S
: 206 0588 S
: 207 0589 S
: 208 0590 S
: 209 0591 S
: 210 0592 S

```

```

! may have been cleared when error was processed

PACKET[IRPSV_VIRTUAL] = 1;
UCB = .UCB_SAVE;
EXESINS!OQT.PACKET, .UCB);
END;

END;

ENTRY = .ENTRY[IRPSL_IOQFL];
END;

SET_IPL(0);
END;

```

```

.TITLE STRVIO
.IDENT \V04-000\

.EXTRN CURRENT_UCB, CURRENT_WCB
.EXTRN QUEUE_HEAD, EXESINSIOQ

.PSECT $LOCKEDC1$,NOWRT,2

.ENTRY START VIO, Save R2,R3,R4,R5,R6,R7,R8,R9,R10 : 0465
MOVL CURRENT_UCB, UCB : 0523
MOVL CURRENT_VCB, VCB : 0524
MOVL CURRENT_WCB, WCB : 0525
MOVL QUEUE_HEAD, QUEUE_ENTRY : 0526
MOVL CURRENT_UCB, UCB_SAVE : 0527
MOVZBL 11(UCB), R1 : 0528
MTPR R1, #18
MOVW #1, 22(WCB) : 0532
MOVL UCB, 48(WCB) : 0533
REMQUE @0(VCB), PACKET : 0542
BVS 2$
BISB2 #16, 42(PACKET) : 0548
MOVL UCB_SAVE, UCB : 0552
JSB @#EXESINSIOQ : 0553
BRB 1$ : 0538
MOVL (QUEUE_ENTRY), ENTRY : 0560
CPL ENTRY, -QUEUE_ENTRY : 0562
BEQL 6$
EXTZV #0, #6, 32(ENTRY), FUNCTION : 0565
CPL FUNCTION, #12 : 0567
BEQL 4$
CPL FUNCTION, #11 : 0569
BNEQ 5$
MOVL 28(ENTRY), R0 : 0573
CPL 52(R0), VCB
BNEQ 5$
MOVL 4(ENTRY), ENTRY : 0576
REMQUE @0(ENTRY), PACKET : 0577
BISB2 #16, 42(PACKET) : 0581
MOVL UCB_SAVE, UCB : 0582
JSB @#EXESINSIOQ : 0583
MOVL (ENTRY), ENTRY : 0588

```

```

07FC 00000
55 0000G CF D0 00002
5A 5B D0 00007
50 0000G CF D0 0000A
59 0000G CF D0 0000F
58 0000G CF D0 00014
51 0B A5 9A 00019
12 51 DA 0001D
16 A0 01 B0 00020
30 A0 55 D0 00024
53 00 BA 0F 00028 1$:
0F 1D 0002C
2A A3 10 88 0002E
55 58 D0 00032
00000000G 9F 16 00035
EB 11 00038
56 69 D0 0003D 2$:
59 56 D1 00040 3$:
34 13 00043
06 00 EF 00045
0C 57 D1 00048
05 13 0004E
0B 57 D1 00050
1F 12 00053
50 1C A6 D0 00055 4$:
5A 34 A0 D1 00059
15 12 0005D
56 04 A6 D0 0005F
53 00 B6 0F 00063
2A A3 10 88 00067
55 58 D0 0006B
00000000G 9F 16 0006E
56 66 D0 00074 5$:

```

12 C7 11 00077 BRB 3\$
00 DA 00079 6\$: MTPR #0, #18
04 0007C RET

: 0562
: 0591
: 0592

: Routine Size: 125 bytes, Routine Base: \$LOCKEDC1\$ + 0000

```

: 211 0593 1
: 212 0594 1 GLOBAL ROUTINE STOP_VIO : COMMON_CALL NOVALUE =
: 213 0595 1 +-
: 214 0596 1
: 215 0597 1 FUNCTIONAL DESCRIPTION:
: 216 0598 1 This routine blocks virtual IO from happening over the current WCB
: 217 0599 1
: 218 0600 1 CALLING SEQUENCE:
: 219 0601 1 STOP_VIO(), called in kernel mode
: 220 0602 1
: 221 0603 1 INPUT PARAMETERS:
: 222 0604 1 none
: 223 0605 1
: 224 0606 1 IMPLICIT INPUTS:
: 225 0607 1 CURRENT_WCB - address of current window control block which is currently
: 226 0608 1 not mapping virtual IO
: 227 0609 1
: 228 0610 1 OUTPUT PARAMETERS:
: 229 0611 1 none
: 230 0612 1
: 231 0613 1 IMPLICIT OUTPUTS:
: 232 0614 1 window mapping to current unit stopped
: 233 0615 1
: 234 0616 1 ROUTINE VALUE:
: 235 0617 1 none
: 236 0618 1
: 237 0619 1 SIDE EFFECTS:
: 238 0620 1 this routine runs at fork level in order to synchronize with the mapping
: 239 0621 1 portion of QIO processing and the magnetic tape driver
: 240 0622 1
: 241 0623 1 --
: 242 0624 1
: 243 0625 2 BEGIN
: 244 0626 2
: 245 0627 2 EXTERNAL REGISTER
: 246 0628 2 COMMON_REG;
: 247 0629 2
: 248 0630 2 CURRENT_WCB[WCB$W_NMAP] = 0;
: 249 0631 1 END;

```

```

50 0000G CF D0 0000 .ENTRY STOP_VIO, Save nothing
16 A0 B4 00007 MOVL CURRENT_WCB, R0
04 0000A CLRW 22(R0)
RET
```

: 0594
: 0630
: 0631

: Routine Size: 11 bytes, Routine Base: \$LOCKEDC1\$ + 0070


```
: 250      0632  1  
: 251      0633  1 END  
: 252      0634  1  
: 253      0635  0 ELUDOM
```

PSECT SUMMARY

```
: Name          Bytes          Attributes  
: $LOCKEDC1$    136 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
```

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	17 0	1000	00:01.8

COMMAND QUALIFIERS

```
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:STRVIO/OBJ=OBJ$:STRVIO MSRCS:STRVIO/UPDATE=(ENHS:STRVIO)
```

```
: Size:          136 code + 0 data bytes  
: Run Time:      00:08.0  
: Elapsed Time: 00:17.8  
: Lines/CPU Min: 4756  
: Lexemes/CPU-Min: 19707  
: Memory Used:  99 pages  
: Compilation Complete
```


