

SSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSS
SSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSS
SSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSS
SSS	YYY	YYY	SSS
SSS	YYY	YYY	SSS
SSS	YYY	YYY	SSS
SSS	YYY	YYY	SSS
SSS	YYY	YYY	SSS
SSS	YYY	YYY	SSS
SSSSSSSSSS	YYY	YYY	SSSSSSSSSS
SSSSSSSSSS	YYY	YYY	SSSSSSSSSS
SSSSSSSSSS	YYY	YYY	SSSSSSSSSS
	YYY	YYY	SSS
	YYY	YYY	SSS
	YYY	YYY	SSS
	YYY	YYY	SSS
	YYY	YYY	SSS
	YYY	YYY	SSS
	YYY	YYY	SSS
	YYY	YYY	SSS
SSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSS
SSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSS
SSSSSSSSSSSS	YYY	YYY	SSSSSSSSSSSS

_S

Ps

--

YZ

ZS

ZS

ZS

ZS

ZS

ZS

ZS

ZS

ZS

ZS

ZS

```

BBBBBBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEEEE  RRRRRRRR  CCCCCCCC  TTTTITTTTT  LL
BBBBBBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEEEE  RRRRRRRR  CCCCCCCC  TTTTITTTTT  LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BBBBBBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEEEE  RRRRRRRR  CCCCCCCC  TT          LL
BBBBBBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEEEE  RRRRRRRR  CCCCCCCC  TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BB      BB  UU      UU  FF          FF          EE          RR      RR  CC          TT          LL
BBBBBBBB  UUUUUUUUU  FF          FF          EEEEEEEEE  RR      RR  CCCCCCCC  TT          LL
BBBBBBBB  UUUUUUUUU  FF          FF          EEEEEEEEE  RR      RR  CCCCCCCC  TT          LL

```

```

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
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

BUFFERCTL
Table of contents

- I/O BUFFER CONTROL

F 2

15-SEP-1984 23:52:45 VAX/VMS Macro V04-00

Page 0

BU
VO

(1)	53	GET ONE BYTE OF DATA FROM USER BUFFER
(1)	89	PUT ONE BYTE OF DATA INTO USER'S BUFFER
(1)	123	INITIALIZE FOR SINGLE BYTE TRANSFERS
(1)	145	MOVE FROM USER BUFFER
(1)	178	MOVE TO USER BUFFER
(1)	211	FILL SYSTEM PTE WITH TRANSFER PTE

```

0000 1
0000 2 .TITLE BUFFERCTL - I/O BUFFER CONTROL
0000 3 .IDENT 'V04-000'
0000 4
0000 5
0000 6 :*****
0000 7 :*
0000 8 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 9 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 10 :* ALL RIGHTS RESERVED. *
0000 11 :*
0000 12 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 13 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 14 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 15 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 16 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 17 :* TRANSFERRED. *
0000 18 :*
0000 19 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 20 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 21 :* CORPORATION. *
0000 22 :*
0000 23 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 24 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 25 :*
0000 26 :*
0000 27 :*****
0000 28
0000 29 D. N. CUTLER 9-AUG-76
0000 30
0000 31 MODIFIED BY:
0000 32
0000 33 04 STJ0002 S. JEFFREYS 29-FEB-1980
0000 34 ADD ALTERNATE ENTRY POINTS FOR IOC$MOVTOUSER AND IOC$MOVFRUSER.
0000 35
0000 36 03 STJ0001 S. JEFFREYS 26-SEP-1979
0000 37 MODIFY IOC$PUTBYTE AND IOC$GETBYTE TO WORK CORRECTLY FOR
0000 38 BUFFERS THAT ARE PAGE ALIGNED.
0000 39
0000 40 02 CAM001 C. MONIA 15-FEB-1979
0000 41 ADD IOC$PUTBYTE AND IOC$GETBYTE ROUTINES FOR TU-58 SUPPORT
0000 42
0000 43
0000 44 I/O BUFFER CONTROL ROUTINES
0000 45
0000 46 MACRO LIBRARY CALLS
0000 47
0000 48
0000 49 $PRDEF ;DEFINE PROCESSOR REGISTERS
0000 50 $PTEDEF ;PAGE TABLE ENTRY DEFINITIONS
0000 51 $UCBDEF ;DEFINE UCB OFFSETS

```

52
20

43
47

47

48
4C

20
54

41
2C
00

45

```

0000 53      .SBTTL  GET ONE BYTE OF DATA FROM USER BUFFER
0000 54      :+
0000 55      : IOC$GETBYTE - GET ONE BYTE OF DATA FROM USER'S BUFFER
0000 56      :
0000 57      : THIS ROUTINE IS CALLED BY AN I/O DRIVER TO GET A SINGLE BYTE FROM THE
0000 58      : USER'S BUFFER.
0000 59      :
0000 60      : PRIOR TO CALLING THIS ROUTINE, A CALL TO IOC$INITBUFWINDOW MUST BE MADE TO
0000 61      : MAP THE SYSTEM PAGE TABLE ENTRY INTO THE USER'S BUFFER
0000 62      :
0000 63      : INPUTS:
0000 64      :
0000 65      :     R0 = SYSTEM VIRTUAL ADDRESS OF ONE-PAGE WINDOW INTO USER'S BUFFER.
0000 66      :     R5 = ADDRESS OF UCB.
0000 67      :
0000 68      : OUTPUTS:
0000 69      :
0000 70      :     R0 = UPDATED SYSTEM VIRTUAL ADDRESS
0000 71      :     R1 = ONE BYTE OF DATA (ZERO EXTENDED)
0000 72      :
0000 73      :     UCB$L_SVAPTE IS UPDATED WHENEVER A PAGE BOUNDARY IS CROSSED
0000 74      :
0000 75      : THE DRIVER IS EXPECTED TO SAVE THE VALUE OF R0 FOR SUBSEQUENT CALLS.
0000 76      :
0000 77      : -
0000 78      :
00000000 79      .PSECT  WIONONPAGED
0000 80
0000 81      IOC$GETBYTE::
50  51  80  90 0000 82      MOVB      (R0)+,R1      ;GET BYTE FROM USER'S BUFFER
      01FF 8F 83 0003 83      BITW      #^X01FF,R0      ;OVERFLOW PAGE BOUNDARY?
      06 12 0008 84      BNEQ      10$      ;IF NEQ NO
78  A5  04  C0 000A 85      ADDL      #4,UCB$L_SVAPTE(R5) ;UPDATE ADDRESS OF PROCESS PTE
      49 10 000E 86      BSBB      IOC$FILSPT      ;FILL SPT, COMPUTE SYSTEM ADDRESS OF PAGE
05  0010 87 10$:  RSB      ;RETURN

```

```

0011 89 .SBTTL PUT ONE BYTE OF DATA INTO USER'S BUFFER
0011 90 :+
0011 91 : IOC$PUTBYTE - PUT ONE BYTE OF DATA IN USER'S BUFFER
0011 92 :
0011 93 : THIS ROUTINE IS CALLED BY AN I/O DRIVER TO PUT A SINGLE BYTE OF DATA
0011 94 : INTO THE USER'S BUFFER.
0011 95 :
0011 96 : PRIOR TO CALLING THIS ROUTINE, A CALL TO IOC$INITBUFWINDOW MUST BE MADE TO
0011 97 : MAP THE SYSTEM PAGE TABLE ENTRY INTO THE USER'S BUFFER.
0011 98 :
0011 99 : INPUTS:
0011 100 :
0011 101 : RO = SYSTEM VIRTUAL ADDRESS OF ONE-PAGE WINDOW INTO USER'S BUFFER
0011 102 : R1 LOW BYTE = DATA TO BE TRANSFERRED TO USER
0011 103 : R5 = ADDRESS OF UCB
0011 104 :
0011 105 : OUTPUTS:
0011 106 :
0011 107 : RO = UPDATED SYSTEM VIRTUAL ADDRESS OF BUFFER WINDOW
0011 108 :
0011 109 : UCB$$_SVAPTE IS UPDATED WHENEVER A PAGE BOUNDARY IS CROSSED
0011 110 :
0011 111 : THE DRIVER IS EXPECTED TO SAVE THE VALUE OF RO FOR SUBSEQUENT CALLS.
0011 112 :
0011 113 : -
0011 114 :
0011 115 IOC$PUTBYTE::
50 80 51 90 0011 116 M 'B R1,(R0)+ ;PUT BYTE INTO USER'S BUFFER
01FF 8F B3 0014 117 BITW #^X01FF,R0 ;OVERFLOW PAGE BOUNDARY?
06 12 0019 118 BNEQ 10$ ;IF NEQ NO
78 A5 04 C0 001B 119 ADDL #4,UCB$$_SVAPTE(R5) ;UPDATE ADDRESS OF PROCESS PTE
38 10 001F 120 BSBB IOC$FILSPT ;FILL SPT, COMPUTE SYSTEM ADDRESS OF PAGE
. 05 0021 121 10$: RSB ;RETURN

```

```
0022 123 .SBTTL INITIALIZE FOR SINGLE BYTE TRANSFERS
0022 124 :
0022 125 :+ IOC$INITBUFWIND - INITIALIZE ONE-PAGE WINDOW INTO USER'S BUFFER
0022 126 :
0022 127 : THIS ROUTINE MUST BE CALLED BY A DRIVER TO SETUP THE INITIAL ONE-
0022 128 : PAGE WINDOW INTO A USER'S BUFFER BEFORE CALLING IOC$GETBYTE OR
0022 129 : IOC$PUTBYTE.
0022 130 :
0022 131 : INPUTS:
0022 132 :
0022 133 : R5 = ADDRESS OF UCB
0022 134 :
0022 135 : OUTPUTS:
0022 136 :
0022 137 : R0 = SYSTEM VIRTUAL ADDRESS OF WINDOW INTO USER'S BUFFER
0022 138 : -
0022 139 :
0022 140 IOC$INITBUFWIND::
50 7C 35 10 0022 141 BSBB IOC$FILSPT : FILL SPT, COMPUTE VIRTUAL ADDRESS OF PAGE
0024 142 BISW UCBSW_BOFF(R5),R0 : MERGE BYTE OFFSET INTO ADDRESS
0028 143 RSB :
```

```

0029 145 .SBTTL MOVE FROM USER BUFFER
0029 146 :+
0029 147 : IOC$MOVFRUSER - MOVE FROM USER BUFFER
0029 148 :
0029 149 : THIS ROUTINE IS CALLED BY AN I/O DRIVER TO MOVE A STRING FROM A USER
0029 150 : BUFFER TO AN INTERNAL BUFFER.
0029 151 :
0029 152 : INPUTS:
0029 153 :
0029 154 : R1 = ADDRESS OF INTERNAL BUFFER.
0029 155 : R2 = NUMBER OF BYTES TO BE MOVED.
0029 156 : R5 = UCB ADDRESS OF DEVICE UNIT.
0029 157 :
0029 158 : OUTPUTS:
0029 159 :
0029 160 : ***TBS***
0029 161 :-
0029 162 :
0029 163 .ENABLE LSB
0029 164 IOC$MOVFRUSER:: :MOVE FROM USER BUFFER
F7 10 0029 165 BSBB IOC$INITBUFWIND :SETUP WINDOW INTO BUFFER
OD 11 002B 166 BRB 20$ :
50 01FF 8F B3 002D 167 IOC$MOVFRUSER2:: :
78 A5 04 C0 002D 168 10$: BITW #X01FF,R0 :OVERFLOW PAGE BOUNDRY?
06 12 0032 169 BNEQ 20$ :IF NEQ NO
1F 10 0034 170 ADDL #4,UCB$L SVAPTE(R5) :UPDATE ADDRESS OF USER PTE
81 80 90 0038 171 BSBB IOC$FILSPT :FILL SYSTEM PTE WITH PROPER RELOCATION
ED 52 F5 003A 172 IOC$MOVFRUSER1:: :
05 003A 173 20$: MOVB (R0)+,(R1)+ :MOVE BYTE TO INTERNAL BUFFER
0040 174 SOBGTR R2,10$ :ANY MORE BYTES TO MOVE?
0041 175 RSB :
0041 176 .DSABL LSB

```



```

0041 178      .SBTTL MOVE TO USER BUFFER
0041 179      :+
0041 180      : IOC$MOVTOUSER - MOVE TO USER BUFFER
0041 181      :
0041 182      : THIS ROUTINE IS CALLED BY AN I/O DRIVER TO MOVE A STRING FROM AN INTERNAL
0041 183      : BUFFER TO A USER BUFFER.
0041 184      :
0041 185      : INPUTS:
0041 186      :
0041 187      :     R1 = ADDRESS OF INTERNAL BUFFER.
0041 188      :     R2 = NUMBER OF BYTES TO BE MOVED.
0041 189      :     R5 = UCB ADDRESS OF DEVICE UNIT.
0041 190      :
0041 191      : OUTPUTS:
0041 192      :
0041 193      : ***TBS***
0041 194      :-
0041 195
0041 196      .ENABLE LSB
0041 197 IOC$MOVTOUSER::      :MOVE TO USER BUFFE
0041 198      BSBB      IOC$INITBUFWIND      :INITIALIZE WINDOW INTO BUFFER
0043 199      BRB      20$      :
0045 200 IOC$MOVTOUSER2::
50 01FF 8F B3 0045 201 10$: BITW      #^X01FF,R0      :OVERFLOW PAGE BOUNDRY?
    06 12 004A 202      BNEQ      20$      :IF NEQ NO
78 A5 04 C0 004C 203      ADDL      #4,UCB$L SVAPTE(R5)      :UPDATE ADDRESS OF USER PTE
    07 10 0050 204      BSBB      IOC$FILSPT      :FILL SYSTEM PTE WITH PROPER RELOCATION
    0052 205 IOC$MOVTOUSER1::
    80 81 90 0052 206 20$: MOVB      (R1)+,(R0)+      :MOVE BYTE TO USER BUFFER
    ED 52 F5 0055 207      SOBGTR   R2,10$      :ANY MORE BYTES TO MOVE?
    05 0058 208      RSB
    0059 209      .DSABL   LSB

```

```

0059 211 .SBTTL FILL SYSTEM PTE WITH TRANSFER PTE
0059 212 :+
0059 213 : IOC$FILSPT - FILL SYSTEM PTE WITH TRANSFER PTE
0059 214 :
0059 215 : THIS ROUTINE IS CALLED TO FILL A SYSTEM PTE WITH A TRANSFER PTE OF A
0059 216 : LOCKED DOWN BUFFER SO THAT IT MAY BE DIRECTLY ADDRESS.
0059 217 :
0059 218 : INPUTS:
0059 219 :
0059 220 : R5 = DEVICE UNIT UCB ADDRESS.
0059 221 :
0059 222 : OUTPUTS:
0059 223 :
0059 224 : R0 = SYSTEM VIRTUAL ADDRESS OF START OF PAGE CONTAINING THE BUFFER.
0059 225 :
0059 226 : REGISTERS R1, R2, AND R3 ARE PRESERVED ACROSS CALL.
0059 227 :-
0059 228 :
0059 229 IOC$FILSPT:: :FILL SYSTEM PTE WITH TRANSFER PTE
50 74 A5 53 DD 0059 230 PUSHL R3 :SAVE REGISTERS
53 78 B5 D0 0058 231 MULL3 #4,UCB$L_SVPN(R5),R0 :CALCULATE BYTE OFFSET IN SYSTEM PAGE TABLE
03 19 0064 232 MOVL @UCB$L_SVAPTE(R5),R3 :GET CONTENTS OF TRANSFER PTE
FF97' 30 0066 233 BLSS 10$ :IF LSS ENTRY IS VALID
00000000'FF40 15 00 53 F0 0069 234 BSBW IOC$PTETOPFN :GET PFN FOR INVALID PTE
50 50 07 78 0073 235 10$: INSV R3,#PTESV_PFN,#PTESS_PFN,@MMG$GL_SPTBASE[R0] :SET SYSTEM PTE ENTRY PFN
00 50 1F E2 0077 236 :SET SYSTEM PTE ENTRY PFN
0073 237 ASHL #7,R0,R0 :CONVERT SVPN TO SYSTEM VIRTUAL ADDRESS
0077 238 BBSS #31,R0,20$ :
0078 239 20$: INVALID R0 :INVALIDATE TRANSLATION BUFFER
007E 240 POPR #^M<R3> :RESTORE REGISTERS
0080 241 RSB :
0081 242 :
0081 243 .FND

```

BUFFERCTL
Symbol table

- I/O BUFFER CONTROL

N 2

15-SEP-1984 23:52:45 VAX/VMS Macro V04-00
5-SEP-1984 03:40:11 [SYS.SRC]BUFFERCTL.MAR;1

Page 8
(1)

BU
VO

IOCSFILSPT	00000059	RG	02
IOCSGETBYTE	00000000	RG	02
IOCSINITBUFWIND	00000022	RG	02
IOCSMOVFRUSER	00000029	RG	02
IOCSMOVFRUSER1	0000003A	RG	02
IOCSMOVFRUSER2	0000002D	RG	02
IOCSMOVTOUSER	00000041	RG	02
IOCSMOVTOUSER1	00000052	RG	02
IOCSMOVTOUSER2	00000045	RG	02
IOCSPTETOPFN	*****	X	02
IOCSPUTBYTE	00000011	RG	02
MMG\$GL_SPTBASE	*****	X	02
PR\$ TBIS	= 0000003A		
PTE\$S_PFN	= 00000015		
PTE\$V_PFN	= 00000000		
UCBSL_SVAPTE	= 00000078		
UCBSL_SVFN	= 00000074		
UCBSW_BOFF	= 0000007C		

! 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	
WIONONPAGED	00000081 (129.)	02 (2.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE	

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:02.08
Command processing	108	00:00:00.53	00:00:03.44
Pass 1	222	00:00:04.52	00:00:16.39
Symbol table sort	0	00:00:00.71	00:00:01.92
Pass 2	60	00:00:00.97	00:00:04.09
Symbol table output	4	00:00:00.04	00:00:00.05
Psect synopsis output	1	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	426	00:00:06.89	00:00:28.00

The working set limit was 1050 pages.
 25635 bytes (51 pages) of virtual memory were used to buffer the intermediate code.
 There were 30 pages of symbol table space allocated to hold 490 non-local and 8 local symbols.
 243 source lines were read in Pass 1, producing 13 object records in Pass 2.
 11 pages of virtual memory were used to define 10 macros.

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

Macro library name	Macros defined
-----	-----
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	4
TOTALS (all libraries)	7

557 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:BUFFERCTL/OBJ=OBJ\$:BUFFERCTL MSRCS:BUFFERCTL/UPDATE=(ENHS:BUFFERCTL)+EXECMLS/LIB

