


```

RRRRRRR      EEEEEEEEE  MM      MM      SSSSSSSS  UU      UU  BBBB BBBB  RRRRRRR
RRRRRRR      EEEEEEEEE  MM      MM      SSSSSSSS  UU      UU  BBBB BBBB  RRRRRRR
RR      RR    EE          MMMM  MMMM  SS          UU      UU  BB      BB  RR      RR
RR      RR    EE          MMMM  MMMM  SS          UU      UU  BB      BB  RR      RR
RR      RR    EE          MM  MM  MM  SS          UU      UU  BB      BB  RR      RR
RR      RR    EE          MM  MM  MM  SS          UU      UU  BB      BB  RR      RR
RRRRRRR      EEEEEEEEE  MM      MM      SSSSSS   UU      UU  BBBB BBBB  RRRRRRR
RRRRRRR      EEEEEEEEE  MM      MM      SSSSSS   UU      UU  BBBB BBBB  RRRRRRR
RR      RR    EE          MM      MM      SS          UU      UU  BB      BB  RR      RR
RR      RR    EE          MM      MM      SS          UU      UU  BB      BB  RR      RR
RR      RR    EE          MM      MM      SS          UU      UU  BB      BB  RR      RR
RR      RR    EE          MM      MM      SS          UU      UU  BB      BB  RR      RR
RR      RR    EEEEEEEEE  MM      MM      SSSSSSSS  UU      UU  BBBB BBBB  RRRRRRR
RR      RR    EEEEEEEEE  MM      MM      SSSSSSSS  UU      UU  BBBB BBBB  RRRRRRR

```

```

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

```

REMSUBR
Table of contents

- SUBROUTINES FOR REMOTE I/O

N 10

16-SEP-1984 02:12:32 VAX/VMS Macro V04-00

Page 0

R
V

(3) 39
(4) 54
(5) 82
(6) 102
(7) 135

HISTORY
DECLARATIONS
REMSCLEAN UP - Clean up io in progress
REMSATTNMSG - Send an attention message
REMSQUEUE - Put element in work queue

```
0000 1      .TITLE  REMSUBR - SUBROUTINES FOR REMOTE I/O
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 : FACILITY:      REMOTE I/O ACP
0000 30 :
0000 31 : ABSTRACT:
0000 32 :      This module contains miscellaneous routines
0000 33 :
0000 34 : ENVIRONMENT:
0000 35 :      MODE = KERNEL
0000 36 :--
```

```
0000 38
0000 39
0000 40 ; .SBTTL HISTORY
0000 41 ; AUTHOR: SCOTT G. DAVIS, CREATION DATE: 17-JUL-79
0000 42 ;
0000 43 ; MODIFIED BY:
0000 44 ;
0000 45 ; V02-004 DJD2002 Darrell Duffy 28-July-1981
0000 46 ; Fix to use DDT entries as absolute addresses.
0000 47 ;
0000 48 ; V02-003 DJD2001 Darrell Duffy 5-Mar-1981
0000 49 ; RTTDRIVER calls NETDRIVER directly for IO.
0000 50 ; Pass real buffered IO packets to RTTDRIVER.
0000 51 ;
0000 52 ; V02-002 SGD2001 Scott G. Davis 23-Sep-1980
Remove multiple devices, ignore unknown messages
```

```
0000 54      .SBTTL  DECLARATIONS
0000 55      :
0000 56      : INCLUDE FILES:
0000 57      :
0000 58
0000 59      $CCBDEF
0000 60      $CRBDEF
0000 61      $DDBDEF
0000 62      $DDTDEF
0000 63      $DYNDEF
0000 64      $IRPDEF
0000 65      $RBFDEF
0000 66      $REMDEF
0000 67      $UCBDEF
0000 68
0000 69      $RTTUCBEXT                ; RTT extensions to UCB
0000 70      :
0000 71      : MACROS:
0000 72      :
0000 73      :
0000 74      :
0000 75      : EQUATED SYMBOLS:
0000 76      :
0000 77      :
0000 78      :
0000 79      : OWN STORAGE:
0000 80      :
```

```
0000 82      .SBTTL  RFM$CLEAN_UP - Clean up io in progress
0000 83
00000000 84      .PSECT  REM_CODE,NOWRT
0000 85      :++
0000 86      :
0000 87      : REM$CLEAN_UP - Clean up the I/O in progress
0000 88      :
0000 89      : INPUTS:
0000 90      :
0000 91      :      R11 - device index
0000 92      :
0000 93      :--
0000 94
0000 95  REM$CLEAN_UP::
55 0000'DF4B  D0 0000 96      MOVL   @W^REMS$GL_UCBVEC[R11],R5 ; Get the associated UCB address
    01      12 0006 97      BNEQ   5$      ; If NEQ proceed
    52 01    05 0008 98      RSB    ; UCB is gone
    0009 99 5$: MOVL   #RBF$C_TT_HANGUP, R2 ; The code for a hangup.
000C 100 ;      **** Fall through ****
```

```

000C 102      .SBTTL  REM$ATTNMSG - Send an attention message
000C 103      :++
000C 104      : REM$ATTNMSG - Send an attention message to the driver
000C 105      :
000C 106      : An attention message is formatted for the driver given a code
000C 107      : as input to this routine.
000C 108      :
000C 109      : inputs:
000C 110      : r2 = modifier for attention message
000C 111      : r5 = rtt ucb
000C 112      : outputs:
000C 113      : r0 - r4, r6 clobbered
000C 114      :--
000C 115      REM$ATTNMSG::
56 52 D0 000C 116      MOVL  R2, R6      ; Save the modifier out of harms way
   FFEE' 30 000F 117      BSBW  REM$ALLOC_IRP ; Get a block
53 52 D0 0012 118      MOVL  R2, R3      ; Get address where it belongs
   13 90 0015 119      MOVB  #DYN$C_BUFIO, - ; Set the block type
   OA A2 0017 120      RBF$B_TYPE(R2)
   OE A2 9E 0019 121      MOVAB RBF$W_OPCODE(R2), - ; Set the data address
   62 001C 122      RBF$L_MSGDAT(R2)
   04 A2 D4 001D 123      CLRL  RBF$L_USRBFRR(R2) ; Set the user buffer address
   01 AE 0020 124      0020 124
   OE A2 0022 125      MNEGW #-RBF$C_ATTN, - ; This is an attention packet
10 A2 56 B0 0024 126      RBF$W_OPCODE(R2)
52 28 A5 D0 0028 127      MOVW  R6, RBF$W_MOD(R2) ; Modifier into message
52 0C A2 D0 002C 128      MOVL  UCB$L_DDBTR5), R2 ; Get the DDB address
52 04 A2 D0 0030 129      MOVL  DDB$L_DDT(R2), R2 ; Get the DDT address
   16 0034 130      MOVL  DDT$L_UN$OLINT(R2), R2 ; Get the UNSOL address
   05 0036 131      JSB   (R2) ; Pass the message and return
   0037 132      RSB
   0037 133

```



```

0037 135      .SBTTL REMSENQUEUE - Put element in work queue
0037 136      :++
0037 137      :
0037 138      : REMSENQUEUE - Put an element into the work queue
0037 139      :
0037 140      : INPUTS:
0037 141      :
0037 142      :
0037 143      : R10 - Low byte has work type
0037 144      : R11 - Device index - saved, but not always relevant
0037 145      : OUTPUTS:
0037 146      :
0037 147      : R0-R3 destroyed
0037 148      :
0037 149      :--
0037 150
0037 151 REMSENQUEUE::
53   FFC6' 30 0037 152      BSBW      REM$ALLOC IRP      : Get storage for a queue element
      OA A2 9E 003A 153      MOVAB     IRP$B_TYPE(R2),R3  : Point to area to put info
      83 83 94 003E 154      CLRB      (R3)+          : Mark as a work element
      83 5A 90 0040 155      MOVB      R10,(R3)+       : Store the work type
      63 5B D0 0043 156      MOVL      R11,(R3)         : Store the device index (maybe)
53   0000'CF D0 0046 157      MOVL      W^REM$GL Q HEAD,R3  : Get the ACP queue head
      04 B3 62 0E 004B 158      INSQUE   (R2),a4(R3)      : Put element in the queue
      0B 12 004F 159      BNEQ      10$              : If NEQ this is not the first element
      0051 160      $WAKE_S      : Cause processing to occur
      005L 161 10$:
      05 005C 162      RSB              : Done
      005D 163
      005D 164      .END

```

```

DDBSL_DDT          = 0000000C
DDTSL_UNSOINT     = 00000004
DYNSC_BUFIO      = 00000013
IRPSB_TYPE       = 0000000A
RFBFB_TYPE       = 0000000A
RBFSC_ATTN       = FFFFFFFF
RBFSC_TT_HANGUP  = 00000001
RBFSL_MSGDAT     = 00000000
RBFSL_USRBF      = 00000004
RBFSL_MOD        = 00000010
RBFSL_OPCODE     = 0000000E
REMSA[LOC_IRP    = ***** X 02
REMSATTNMSG      = 0000000C RG 02
REMSCLEANUP      = 00000000 RG 02
REMSC_CURECO     = 00000001
REMSC_CURVRS     = 00000001
REMSC_LNK_READ   = 00000002
REMSC_MAXDEVS    = 0000000A
REMSC_MAXLINKS   = 00000010
REMSC_MAXUNITS   = 00000010
REMSC_MBX_READ   = 00000001
REMSC_ST_ATTRIB  = 00000002
REMSC_ST_CONFIG  = 00000001
REMSQUEUE       = 00000037 RG 02
REMSGL_Q_HEAD    = ***** X 02
REMSGL_UCBVEC    = ***** X 02
SYSSWARE        = ***** GX 02
UCBSK_RTT_LEN    = 00000138
UCBSK_RTT_LENGTH = 00000138
UCBSL_DDB        = 00000028
UCBSL_DEVDEPND2 = 00000048
UCBSL_RTT_CTRLC  = 00000094
UCBSL_RTT_CTRLY  = 00000090
UCBSL_RTT_DEVDEP2 = 00000048
UCBSL_TL_BANDQUE = 0000009C
UCBSL_TL_CTRLC   = 00000094
UCBSL_TL_CTRLY   = 00000090
UCBSL_TL_OUTBAND = 00000098
    
```

! 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
REM_CODE	0000005D (93.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.10	00:00:00.49
Command processing	133	00:00:00.61	00:00:02.28

Pass 1	360	00:00:12.47	00:00:26.18
Symbol table sort	0	00:00:02.14	00:00:04.36
Pass 2	45	00:00:01.91	00:00:03.47
Symbol table output	5	00:00:00.07	00:00:00.07
Psect synopsis output	1	00:00:00.01	00:00:00.20
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	583	00:00:17.32	00:00:37.05

The working set limit was 1500 pages.
66261 bytes (130 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1368 non-local and 2 local symbols.
164 source lines were read in Pass 1, producing 13 object records in Pass 2.
29 pages of virtual memory were used to define 28 macros.

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

Macro library name	Macros defined
-----	-----
-\$255\$DUA28:[REM.OBJ]REM.MLB;1	2
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	17
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	24

1524 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:REMSUBR/OBJ=OBJ\$:REMSUBR MSRC\$:REMSUBR/UPDATE=(ENH\$:REMSUBR)+EXECML\$/LIB+LIB\$:REM/LIB

RI
P

P
S
R
R
R
R

P
I
C
P
S
P
S
P
C
A

T
B
T
4
4

M
I
T
T
M

