



```

RRRRRRRR      EEEEEEEEEEE MM      MM LL      000000      CCCCCCCC      KK      KK      DDDDDDDD      BBBB88888
RRRRRRRR      EEEEEEEEEEE MM      MM LL      000000      CCCCCCCC      KK      KK      DDDDDDDD      BBBB88888
RR      RR      EE      MMMM      MMMM LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MMMM      MM, ' LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MM      MM      MM LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MM      MM      MM LL      00      00      CC      KK      KK      DD      DD      BB      BB
RRRRRRRR      EEEEEEEEEEE MM      MM LL      00      00      CC      KKKKKK      DD      DD      BBBB88888
RRRRRRRR      EEEEEEEEEEE MM      MM LL      00      00      CC      KKKKKK      DD      DD      BBBB88888
RR      RR      EE      MM      MM      LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MM      MM      LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MM      MM      LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MM      MM      LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EE      MM      MM      LL      00      00      CC      KK      KK      DD      DD      BB      BB
RR      RR      EEEEEEEEEEE MM      MM LLLLLLLLLL      000000      CCCCCCCC      KK      KK      DDDDDDDD      BBBB88888
RR      RR      EEEEEEEEEEE MM      MM LLLLLLLLLL      000000      CCCCCCCC      KK      KK      DDDDDDDD      BBBB88888

```

```

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
LLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLL      IIIIII      SSSSSSSS

```



```

0000 1 .TITLE REMLOCKDB - LOCK AND UNLOCK I/O DATA BASE
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: REMACP - REMOTE I/O ACP
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 : THESE ROUTINES LOCK AND UNLOCK THE I/O DATA BASE MUTEX.
0000 35 : NEEDLESS TO SAY, THEY MUST BE CALLED IN KERNEL MODE.
0000 36
0000 37 : ENVIRONMENT:
0000 38
0000 39 : STARLET OPERATING SYSTEM, INCLUDING PRIVILEGED SYSTEM SERVICES
0000 40 : AND INTERNAL EXEC ROUTINE.
0000 41
0000 42 : AUTHOR: Scott G. Davis, CREATION DATE: 20-Jul-1979 15:31
0000 43
0000 44 : MODIFIED BY:
0000 45
0000 46 : V03-004 LMP0277 L. Mark Pilant, 12-Jul-1984 10:27
0000 47 : Copy protection from the template UCB to the cloned UCB.
0000 48 : (The protection in the cloned UCB is cleared in the
0000 49 : routine IOC$CLONE_UCB/IOC$COPY_UCB.)
0000 50
0000 51 : V03-003 ROW0127 Ralph O. Weber 5-OCT-1982
0000 52 : Make changes required to use new JCR creation and deletion
0000 53 : routines in UCBCREDEL. Rewrite REM$(CREATE_UCB adding
0000 54 : knowledge of CRBSW_REFC, number of UCB's on this CRB, and
0000 55 : restructuring to take maximum advantage of new routines.
0000 56 : Eliminate mailbox oriented parts of REM$KILL_UCB and change
0000 57 : IOC$DELMBX to IOC$DELETE_UCB.

```

PSE  
---

\$AB  
\$LO

Pha  
---

Ini  
Com  
Pas  
Sym  
Pas  
Sym  
Pse  
Cro  
Ass

The  
790  
The  
457  
36

Mac  
---  
-\$2  
-\$2  
-\$2  
TOT

172

The

MAC

REMLOCKDB  
V04-000

- LOCK AND UNLOCK I/O DATA BASE E 7

16-SEP-1984 02:10:10 VAX/VMS Macro V04-00  
5-SEP-1984 02:53:56 [REM.SRC]REMLOCKDB.MAR;1

Page 2  
(1)

\*\*F

0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :

V03-002 KDM0002  
Added \$PCBDEF.

Kathleen D. Morse

28-Jun-1982

```
0000 63      .SBTTL  DECLARATIONS
0000 64
0000 65 :
0000 66 : INCLUDE FILES
0000 67 :
0000 68
0000 69      $AQBDEF
0000 70      $CCBDEF
0000 71      $CRBDEF
0000 72      $DEVDEF
0000 73      $IPLDEF
0000 74      $ORBDEF
0000 75      $PCBDEF
0000 76      $PRDEF
0000 77      $REMDEF
0000 78      $UCBDEF
0000 79      $VCBDEF
0000 80      $RTTUCBEXT
```

```

0000 82      .SBTTL  REM$LINK_AQB - Link aqb into the aqb list
0000 83      :++
0000 84      :
0000 85      : FUNCTIONAL DESCRIPTION:
0000 86      :
0000 87      :     This routine links an AQB into the AQB list.
0000 88      :
0000 89      : CALLING SEQUENCE:
0000 90      :     BSBW    REM$LINK_AQB
0000 91      :
0000 92      : INPUT PARAMETERS:
0000 93      :     R2 - AQB address
0000 94      :
0000 95      : SIDE EFFECTS:
0000 96      :     None
0000 97      :
00000000 98      .PSECT  $LOCKEDC1$,NOWRT
0000 99      :
0000 100     :--
0000 101     START_LOCK::                                ; LABEL FOR LKWSET SERVICE
0000 102     :
0000 103     REM$LINK_AQB::
58 52 DO 0000 104     MOVL  R2,R8                                ; Save the AQB address
005A 30 0003 105     BSBW  LOCK_IODB                            ; Lock the I/O database
0006 106     SETIPL #IPL$ SYNCH                               ; Up IPL
OC A8 60 A4 DO 0009 107     MOVL  PCB$ _PID(R4),AQB$ _ACPPID(R8) ; Stuff the PID
51 00000000'GF 9E 000E 108     MOVAB G^IOC$GL_AQB_LIST,R1          ; Get address of list head
10 A8 61 DO 0015 109     MOVL  (R1),AQB$ _LINK(R8)           ; Make forward link from AQB
61 58 DO 0019 110     MOVL  R8,(R1)                          ; Make this AQB head of list
004E 31 001C 111     BRW   UNLOCK_IODB                        ; Unlock the I/o database and return
  
```

```

001F 113 :++
001F 114 :
001F 115 : FUNCTIONAL DESCRIPTION:
001F 116 :
001F 117 :     This routine finds the UCB associated with a device name.
001F 118 :
001F 119 : CALLING SEQUENCE:
001F 120 :     BSBW     REMSFIND_UCB
001F 121 :
001F 122 : INPUT PARAMETERS:
001F 123 :     R1 - Address of device name descriptor
001F 124 :
001F 125 : OUTPUT PARAMETERS:
001F 126 :
001F 127 :     R0 - lbc=>error; lbs=>UCB found
001F 128 :     R1 - UCB address, if found
001F 129 :
0000001F 130 :     .PSECT $LOCKEDC1$,NOWRT
001F 131 :
001F 132 :--
001F 133 :
001F 134 REMSFIND_UCB::
001F 135     BSBW     LOCK_IODB           ; LOCK THE I/O DATA BASE
0021 136     JSB     G^IOC$SEARCHDEV  ; Find the UCB
7E 50 7D 0027 137     MOVQ    RO,-(SP)           ; Save return info
50 41 10 002A 138     BSBW     UNLOCK_IODB       ; Unlock the I/O data base
50 8E 7D 002C 139     MOVQ    (SP)+,RO     ; Restore return info
05 002F 140     RSB                    ; Return

```



```

0030 142 :++
0030 143 :
0030 144 : FUNCTIONAL DESCRIPTION:
0030 145 :
0030 146 :     This routine clones a UCB .
0030 147 :
0030 148 : CALLING SEQUENCE:
0030 149 :     BSBW     REM$CREATE_UCB
0030 150 :
0030 151 : INPUT PARAMETERS:
0030 152 :     R5 - address of UCB to be cloned
0030 153 :
0030 154 : OUTPUT PARAMETERS:
0030 155 :
0030 156 :     R0 - lbc=>error; lbs=>UCB found
0030 157 :     R2 - Cloned UCB address
0030 158 :
00000030 159 :     .PSECT $LOCKEDC1$,NOWRT
0030 160 :
0030 161 :--
0030 162 :
0030 163 REM$CREATE_UCB::
0030 164     CLRL     R0                ; Assume too many RT devices already.
0032 165     MOVL   UCBS$_CRB(R5), R1    ; Get CRB address.
0036 166     CMPW   CRBSW-REFC(R1), -    ; Have the maximum number of links
003C 167         W*REMSGB_MAXLINKS      ; already been established?
003C 168     BGEQU   90$                ; Branch, error exit, if no free links.
003E 169     BSBB   LOCK_IODB          ; Lock I/O database for write access.
0040 170     CLRW   UCBS$_UNIT SEED(R5) ; Limit unit numbers to MAXLINKS.
0042 171     JSB    G*IOC$CLONE_UCB     ; Clone the UCB.
0048 172     BLBC   R0, 80$            ; Branch, error exit, if clone failed.
004B 173     CLRL   UCBS$_PID(R2)      ; Insure no owner of cloned UCB.
004E 174     BBCC   #DEV$V ALL, -     ; Mark cloned UCB unallocated.
0053 175         UCBS$_DEVCHAR(R2), 80$
0053 176 80$:   PUSHL  R0                ; Save our two output registers.
0055 177         PUSHL  R2
0057 178         BSBB   UNLOCK_IODB     ; Unlock the I/O database.
0059 179         POPL  R2                ; Restore saved output registers.
005C 180         POPL  R0
005F 181 90$:   RSB                    ; Return to caller.

```

```
0060 183 .SBTTL LOCK_IODB - Lock the IO database
0060 184 :++
0060 185 :
0060 186 : FUNCTIONAL DESCRIPTION:
0060 187 :
0060 188 : THIS ROUTINE LOCKS THE I/O DATA BASE MUTEX.
0060 189 :
0060 190 : CALLING SEQUENCE:
0060 191 : BSB LOCK_IODb
0060 192 :
0060 193 : INPUT PARAMETERS:
0060 194 : NONE
0060 195 :
0060 196 : IMPLICIT INPUTS:
0060 197 : NONE
0060 198 :
0060 199 : OUTPUT PARAMETERS:
0060 200 : R4 - My PCB address
0060 201 :
0060 202 : IMPLICIT OUTPUTS:
0060 203 : NONE
0060 204 :
0060 205 : ROUTINE VALUE:
0060 206 : NONE
0060 207 :
0060 208 : SIDE EFFECTS:
0060 209 : I/O DATA BASE MUTEX LOCKED
0060 210 :
0060 211 :--
00000060 212
00000060 213 .PSECT $LOCKEDC1$,NOWRT
0060 214
54 00000000'GF D0 0060 215 LOCK_IODB:
00000000'GF 17 0067 216 MOVL G^SCH$GL_CURPCB,R4 ; GET OWN PCB ADDRESS
217 JMP G^SCH$IO[OCKW ; Lock the database and return
```

```

006D 219      .SBTTL UNLOCK_IODB - Unlock the io database
006D 220      :++
006D 221      :
006D 222      : FUNCTIONAL DESCRIPTION:
006D 223      :
006D 224      :     THIS ROUTINE UNLOCKS THE I/O DATA BASE MUTEX.
006D 225      :
006D 226      : CALLING SEQUENCE:
006D 227      :     BSB UNLOCK_IODB
006D 228      :
006D 229      : INPUT PARAMETERS:
006D 230      :     NONE
006D 231      :
006D 232      : IMPLICIT INPUTS:
006D 233      :     NONE
006D 234      :
006D 235      : OUTPUT PARAMETERS:
006D 236      :
006D 237      :     R4 - My PCB address
006D 238      :
006D 239      : IMPLICIT OUTPUTS:
006D 240      :     NONE
006D 241      :
006D 242      : ROUTINE VALUE:
006D 243      :     NONE
006D 244      :
006D 245      : SIDE EFFECTS:
006D 246      :     I/O DATA BASE MUTEX UNLOCKED
006D 247      :     IPL Set to 0
006D 248      :
006D 249      :--
0000006D 250
0000006D 251      .PSECT $LOCKEDC1$,NOWRT
006D 252
006D 253 UNLOCK_IODB:
54 00000000'GF D0 006D 254      MOVL  G^SCH$G' CURPCB,R4      ; Get PCB address
   00000000'GF 16 0074 255      JSB   G^SCH$IOUNLOCK      ; Unlock I/O database
   007A 256      SETIPL #0      ; Bring down the IPL
05 007D 257      RSB

```

```

007E 259 .SBTTL REMSKILL_UCB - Deallocate a ucb
007E 260 :++
007E 261 :
007E 262 : REMSKILL_UCB - Deallocate the UCB, maybe, and deassign the I/O channel
007E 263 :
007E 264 : INPUTS:
007E 265 :
007E 266 : R11 - device index
007E 267 :
007E 268 :--
007E 269 :
007E 270 REMSKILL_UCB::
55 0000'DF4B D0 007E 271 MOVL @W^REMSGL_UCBVEC[R11],R5 ; Get the UCB address
26 13 0084 272 BEQL KILL_NET_CHAN ; Branch if no UCB.
D8 10 0086 273 BSBB LOCK_IODB ; Lock I/O database for write access.
38 A5 00080000 8F CA 0088 274 BICL #DEVSM_MNT, - ; Reset driver interlock.
0090 275 UCBSL_DEVCHAR(R5)
5C A5 B5 0090 276 TSTW UCBSW_REFC(R5) ; Is the UCB ready to go bye bye.
D8 12 0093 277 BNEQ UNLOCK_IODB ; If not, just give up (for now).
50 34 A5 D0 0095 278 MOVL UCBSL_VCB(R5), R0 ; Get Volume Control Block address.
4C A0 B7 0099 279 DECW VCB$W_MCOUNT(R0) ; Decrement its mounted volume count.
64 A5 00010000 8F C8 009C 280 BISL #UCBSM_DELETEUCB, - ; Mark the UCB for deletion.
00A4 281 UCBSL_STS(R5)
00000000'GF 16 00A4 282 JSB G^IOCSDELETE_UCB ; Blow the UCB away.
C1 10 00AA 283 BSBB UNLOCK_IODB ; Unlock the I/O database.
00AC 284
00AC 285 KILL_NET_CHAN:
0000'DF4B D4 00AC 286 CLRL @W^REMSGL_UCBVEC[R11] ; UCB is gone
00B1 287
00B1 288 $DASSGN_S @W^REMSGL_CHANVEC[R11] ; Deassign the channel
0000'DF4B B4 00BE 289 CLRW @W^REMSGL_CHANVEC[R11] ; Clear the channel vector slot
05 00C3 290 RSB ; Done

```

RE  
SY  
  
IO  
IP  
PR  
RE  
UC  
  
PS  
--  
SA  
RE  
  
Ph  
--  
In  
Co  
Pa  
Sy  
Pa  
Sy  
Ps  
Cr  
As  
  
Th  
36  
Th  
13  
13  
  
Ma  
--  
--  
--  
--  
TO  
84  
Th  
MA

```

00C4 292      .SBTTL  REM$SET_UP_IN - Create a ucb and setup the net channel
00C4 293      :++
00C4 294      :
00C4 295      : REM$SET_UP_IN - Set up the I/O data base for inbound connects
00C4 296      :
00C4 297      : INPUTS:
00C4 298      :
00C4 299      :     NONE
00C4 300      :
00C4 301      : OUTPUTS:
00C4 302      :
00C4 303      :     R0 - LBC => couldn't do it; LBS => set up successful
00C4 304      :     R7 - channel number
00C4 305      :     R11 - device index
00C4 306      :
00C4 307      :--
00C4 308      :
00C4 309      REM$SET_UP_IN::
55  0000'CF  D0 00C4 310      MOVL  W^REM$GL_TEMPLATE,R5      ; Get address of UCB template
      FF64    30 00C9 311      BSBW  REM$CREATE_UCB          ; Make a copy
      6B 50   E9 00CC 312      BLBC  R0,SET_UP_DONE          ; If LBC no go
      FF2E'   30 00CF 313      BSBW  REM$EXPORT_CHAN         ; Get an I/O channel
      65 50   E9 00D2 314      BLBC  R0,SET_UP_DONE          ; If LBC error
      5C A2   B4 00D5 315      CLRW  UCBSW_REFCT(R2)        ; Reset the ref count - no channel yet
38 A2 00080000 8F C8 00D8 316      BISL  #DEV$M_MNT,UCBSL_DEVCHAR(R2) ; Mark device mounted, to get
      00E0 317      :
      00E0 318      :
      00E0 319      :
0080 C2  5B  90 00E0 320      MOVVB  R11,UCBSB_ERTCNT(R2)    ; Save the index for later
      50  1C A5 D0 00E5 321      MOVL  UCBSL_ORB(R5),R0        ; Get template device ORB
      51  1C A2 D0 00E9 322      MOVL  UCBSL_ORB(R2),R1        ; Get cloned device ORB
      00ED 323      :
      00ED 324      ASSUME  ORBSL_OWN_PROT EQ ORBSL_SYS_PROT+4
      00ED 325      ASSUME  ORBSL_WOR_PROT EQ ORBSL_GRP_PROT+4
      00ED 326      :
      18 A1  18 A0 7D 00ED 327      MOVQ  ORBSL_SYS_PROT(R0),ORBSL_SYS_PROT(R1) ; Copy device
      20 A1  20 A0 7D 00F2 328      MOVQ  ORBSL_GRP_PROT(R0),ORBSL_GRP_PROT(R1) ; Protection
      00F7 329      :
0000'DF4B  52 D0 00F7 330      MOVL  R2,@W^REM$GL_UCBVEC[R11] ; Save the UCB address
57  0000'DF4B B0 00FD 331      MOVW  @W^REM$GL_CHANVEC[R11],R7 ; Get the I/O channel
      55  52   D0 0103 332      MOVL  R2,R5                    ; Move UCB address for call
      50  57   3C 0106 333      MOVZWL R7,R0                  ; Move the channel
      00000000'GF 16 0109 334      JSB   G^IOC$VERIFYCHAN        ; Obtain the CCB address
      61  61   D0 010F 335      MOVL  CCB$SL_UCB(R1),-        ; Obtain the UCB of the network
      00B0 C5   0111 336      UCBSL_RTT_NETUCB(R5)        ; and save in the RTT UCB
      00B8 C5   DE 0114 337      MOVAL UCBSL_RTT_IRPFL(R5),-    ; Init the IRP queue head
      00B8 C5   0118 338      UCBSL_RTT_IRPFL(R5),-
      00B8 C5   DE 011B 339      MOVAL UCBSL_RTT_IRPFL(R5),-
      00BC C5   011F 340      UCBSL_RTT_IRPBL(R5),-
      00C0 C5   D4 0122 341      CLRL  UCBSL_RTT_NETIRP(R5)   ; No IRP for Net receive yet
      00B4 C5   D4 0126 342      CLRL  UCBSL_RTT_NETWIND(R5)  ; No window yet
      012A 343      DSBINT #IPL$-SYNCH        ; Disable interrupts
      51  34 A5 D0 0130 344      MOVL  UCBSL_VCB(R5),R1        ; VCB address
      4C A1   B6 0134 345      INCW  VCB$W_MCOUNT(R1)     ; One more device mounted
      0137 346      ENBINT        ; Drop the ipl
      013A 347      SET_UP_DONE:
05  013A 348      RSB          ; Done

```

```

0138 350 .SBTTL REM$CHK_ACPDONE - See if the acp can exit
0138 351 .SBTTL REM$GO_AWAY - Terminate the acp
0138 352 :++
0138 353 : FUNCTIONAL DESCRIPTION:
0138 354 :
0138 355 : REM$CHK_ACPDONE - See if the ACP can cease operation
0138 356 : REM$GO_AWAY - Terminate operation. Called if ACP can't really get started.
0138 357 :
0138 358 : CALLING SEQUENCE:
0138 359 :
0138 360 : BSB REM$CHK_ACPDONE
0138 361 :
0138 362 : INPUT PARAMETERS:
0138 363 :
0138 364 : R2 - AQB ADDRESS
0138 365 :
0138 366 : SIDE EFFECTS:
0138 367 :
0138 368 : ACP MAY TERMINATE
0138 369 :
0138 370 :--
0138 371 : .ENABLE LSB
0138 372 :
0138 373 REM$CHK_ACPDONE::
0138 374 :
0138 375 : The following code gets rid of all channels over which names were declared
0138 376 : and forces closeouts (e.g., logouts) on all active remote devices.
0138 377 : The cleanup is asynchronous, so a number of passes may be made through
0138 378 : this code.
0138 379 :
0138 380 CLRL R6 ; Reset active channel counter
5B 0000'CF D4 0138 381 MOVZBL W^REM$GB_MAXLINKS,R11 ; Set up to loop
0138 382 10$:
0138 383 TSTW @W^REM$GL_CHANVEC[R11] ; Check the next channel
0138 384 BEQL 30$ ; If EQL not active
05 0000'DF4B B5 0142 385 BLBC @W^REM$GL_UCBVEC[R11],20$ ; If LBC remote device
0138 386 BSBW KILL_NET_CHAN ; Get rid of this comm channel
0138 387 BRB 30$ ; Loop
0138 388 20$:
0138 389 :*****
0138 390 : If the following instruction is included, REMACP will blow away
0138 391 : all remote terminal connections. When it is not included, REMACP
0138 392 : will not evaporate until all remote terminals are gone.
0138 393 :
0138 394 : BSBW REM$CLEAN_UP ; Pretend there was a disconnect
0138 395 :
0138 396 :*****
0138 397 INCL R6 ; This was an active channel
0138 398 30$:
0138 399 SOBGTR R11,10$ ; Loop
0138 400 TSTL R6 ; Were there any active channels?
0138 401 BNEQ 50$ ; If NEQ yes - wait
0138 402 :
0138 403 : Now see if there are any UCB's still active
0138 404 :
5B 0000'CF 9A 015D 405 MOVZBL W^REM$GB_MAXLINKS,R11 ; Init count
0162 406 40$:

```

```

0000'DF4B D5 0162 407 TSTL @W^REMSGL_UCBVEC[R11] ; In use?
      75 12 0167 408 50$: BNEQ 110$ ; If NEQ yes - go away
      F6 5B F5 0169 409 SOBGTR R11,40$ ; Loop
      016C 410 $DASSGN_S W^REMSGW_MBX_CHAN ; Kill the mailbox
      0178 411
      0178 412 REMSGO_AWAY::
      0178 413 :
      0178 414 : Everybody went away - now deallocate VCB
      0178 415 :
54 0000'CF D0 0178 416 MOVL W^REMSGL_TEMPLATE,R4 ; Get UCB address
      OF 13 017D 417 BEQL 60$ ; If EQL done
      50 34 A4 D0 017F 418 MOVL UCBSL_VCB(R4),R0 ; Get the VCB address
      09 13 0183 419 BEQL 60$ ; If EQL none was allocated
00000000'GF 16 0185 420 JSB G^EXE$DEANONPAGED ; Deallocate it
      34 A4 D4 018B 421 CLRL UCBSL_VCB(R4) ; No VCB any more
      018E 422 60$:
58 0000'CF D0 018E 423 MOVL W^REMSGL_Q_HEAD,R8 ; Get the AQB address
      FECA 30 0193 424 BSBW LOCK_IODB ; LOCK THE I/O DATA BASE
      0196 425 SETIPL #IPL$_SYNCH ; Up IPL
      58 68 D1 0199 426 CMPL AQB$LINK(R8),R8 ; ANY I/O IN QUEUE?
      3D 12 019C 427 BNEQ 100$ ; IF NEQ YES - DON'T TERMINATE
      019E 428 :
      019E 429 : NO MORE I/O - NOW UNHOOK THE AQB FROM THE SYSTEM AQB LIST
      019E 430 :
51 00000000'GF 9E 019E 431 MOVAB G^IOCSGL_AQBLIST,R1 ; GET THE LIST ADDRESS
      50 61 D0 01A5 432 MOVL (R1),R0 ; GET FIRST AQB POINTER
      50 58 D1 01A8 433 CMPL R8,R0 ; IS IT THE FIRST?
      06 12 01AB 434 BNEQ 70$ ; IF NEQ NO
      61 10 A8 D0 01AD 435 MOVL AQB$LINK(R8),(R1) ; LINK IT IN
      11 11 01B1 436 BRB 90$ ; DONE
      01B3 437 70$:
      58 10 A0 D1 01B3 438 CMPL AQB$LINK(R0),R8 ; IS THIS IT?
      06 13 01B7 439 BEQL 80$ ; IF EQL YES
      50 10 A0 D0 01B9 440 MOVL AQB$LINK(R0),R0 ; GET LINK
      F4 11 01BD 441 BRB 70$ ; LOOP
      01BF 442 80$:
10 A0 10 A8 D0 01BF 443 MOVL AQB$LINK(R8),AQB$LINK(R0) ; RELINK
      01C4 444 90$:
      50 58 D0 01C4 445 MOVL R8,R0 ; SET TO DEALLOCATE THE AQB
00000000'GF 16 01C7 446 JSB G^EXE$DEANONPAGED ; DO IT
      FE9D 30 01CD 447 BSBW UNLOCK_IODB ; NOW UNHOOK THE I/O DATABASE
      01D0 448 $DELPRC_S ; SAYONARA!
      01DB 449 100$:
      FE8F 31 01DB 450 BRW UNLOCK_IODB ; Now unlock the I/O database and return
      05 01DE 451 110$:
      01DF 452
      01DF 453 .DISABLE LSB
      01DF 454
      01DF 455 END_LOCK::
      01DF 456
      01DF 457 .END

```

```

AQB$L_ACPID      = 0000000C
AQB$L_ACPQFL     = 00000000
AQB$L_LINK       = 00000010
CCB$S_UCB        = 00000000
CRB$W_REF_C      = 0000000C
DEV$M_MNT        = 00080000
DEV$V_ALL        = 00000017
END_LOCK         = 000001DF  RG    02
EXE$DEANONPAGED *****  X    02
IOC$CLONE_UCB    *****  X    02
IOC$DELETE_UCB  *****  X    02
IOC$GL_AQB$IST  *****  X    02
IOC$SEARCHDEV   *****  X    02
IOC$VERIFYCHAN  *****  X    02
IPL$SYNCH       = 00000008
KILL_NET_CHAN   = 000000AC  R    02
LOCK_IODB       = 00000060  R    02
ORB$C_GRP_PROT  = 00000020
ORB$S_OWN_PROT  = 0000001C
ORB$S_SYS_PROT  = 00000018
ORB$S_WOR_PROT  = 00000024
PCB$S_PID       = 00000060
PR$ IPL         = 00000012
REM$CHK_ACPDONE = 0000013B  RG    02
REM$CREATE_UCB  = 00000030  RG    02
REM$C_CURECO    = 00000001
REM$C_CURVRS    = 00000001
REM$C_LNK_READ  = 00000002
REM$C_MAXDEVS   = 0000000A
REM$C_MAXLINKS  = 00000010
REM$C_MAXUNITS  = 00000010
REM$C_MBX_READ  = 00000001
REM$C_ST_ATTRIB = 00000002
REM$C_ST_CONFIG = 00000001
REMS$IND_UCB    = 0000001F  RG    02
REMS$GB_MAXLINKS *****  X    02
REMS$GL_CHANVEC *****  X    02
REMS$GL_Q_HEAD  *****  X    02
REMS$GL_TEMPLATE *****  X    02
REMS$GL_UCBVEC  *****  X    02
REMS$GO_AWAY    = 00000178  RG    02
REMS$GW_MBX_CHAN *****  X    02
REMS$KICL_UCB   = 0000007E  RG    02
REMS$LINK_AQB   = 00000000  RG    02
REMS$SET_OP_IN  = 000000C4  RG    02
REMS$EXPORT_CHAN *****  X    02
SCH$SGL_CURPCB  *****  X    02
SCH$SIOLOCKW    *****  X    02
SCH$SIOUNLOCK   *****  X    02
SET_UP_DONE     = 0000013A  R    02
START_LOCK      = 00000000  RG    02
SYSS$DASSGN     *****  GX    02
SYSS$DELPRC     *****  GX    02
UCB$B_ERTCNT    = 00000080
UCB$K_RTT_LEN   = 00000138
UCB$K_RTT_LENGTH = 00000138
UCB$S_CRB       = 00000024

```

```

UCB$S_DEVCHAR      = 00000038
UCB$S_DEVDEPND2   = 00000048
UCB$S_ORB          = 0000001C
UCB$S_PID          = 0000002C
UCB$S_RTT_CTRL_C  = 00000094
UCB$S_RTT_CTRL_Y  = 00000090
UCB$S_RTT_DEVDEPND2 = 00000048
UCB$S_RTT_IRPBL   = 000000BC
UCB$S_RTT_IRPFL   = 000000B8
UCB$S_RTT_NETIRP  = 000000C0
UCB$S_RTT_NETUCB  = 000000B0
UCB$S_RTT_NETWIND = 000000B4
UCB$S_STS         = 00000064
UCB$S_TL_BANDQUE  = 0000009C
UCB$S_TL_C_LLC    = 00000094
UCB$S_TL_CTRL_Y   = 00000090
UCB$S_TL_OUTBAND  = 00000098
UCB$S_VCB         = 00000034
UCB$M_DELETEUCB   = 00010000
UCB$W_REF_C       = 0000005C
UCB$W_UNIT_SEED   = 00000000
UNLOCK_IODB       = 0000006D  R    02
VCB$W_MCOUNT    = 0000004C

```



-----  
! Psect synopsis !  
-----

| PSECT name   | Allocation       | PSECT No. | Attributes  |
|--------------|------------------|-----------|---|
| . ABS .      | 00000000 ( 0.)   | 00 ( 0.)  | NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE |
| \$ABS\$      | 00000000 ( 0.)   | 01 ( 1.)  | NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE       |
| \$LOCKEDC1\$ | 000001DF ( 479.) | 02 ( 2.)  | NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE     |

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

| Phase                  | Page faults | CPU Time    | Elapsed Time |
|------------------------|-------------|-------------|--------------|
| Initialization         | 38          | 00:00:00.10 | 00:00:00.61  |
| Command processing     | 156         | 00:00:00.63 | 00:00:03.45  |
| Pass 1                 | 408         | 00:00:14.57 | 00:00:27.62  |
| Symbol table sort      | 0           | 00:00:02.37 | 00:00:03.33  |
| Pass 2                 | 91          | 00:00:02.58 | 00:00:04.00  |
| Symbol table output    | 10          | 00:00:00.14 | 00:00:00.87  |
| 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   | 707         | 00:00:20.43 | 00:00:39.90  |

The working set limit was 1650 pages.  
79097 bytes (155 pages) of virtual memory were used to buffer the intermediate code.  
There were 90 pages of symbol table space allocated to hold 1548 non-local and 13 local symbols.  
457 source lines were read in Pass 1, producing 16 object records in Pass 2.  
36 pages of virtual memory were used to define 35 macros.

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

| Macro library name                  | Macros defined |
|-------------------------------------|----------------|
| -\$255\$DUA28:[REM.OBJ]REM.MLB;1    | 2              |
| -\$255\$DUA28:[SYS.OBJ]LIB.MLB;1    | 20             |
| -\$255\$DUA28:[SYSLIB]STARLET.MLB;2 | 9              |
| TOTALS (all libraries)              | 31             |

1725 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

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

