K 1

```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEE                   TTT        PPP         PPP
UUU         UUU  EEE                   TTT        PPP         PPP
UUU         UUU  EEE                   TTT        PPP         PPP
UUU         UUU  EEE                   TTT        PPP         PPP
UUU         UUU  EEE                   TTT        PPP         PPP
UUU         UUU  EEE                   TTT        PPP         PPP
UUU         UUU  EEEEEEEEEEEE          TTT        PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE          TTT        PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE          TTT        PPPPPPPPPPPP
UUU         UUU  EEE                   TTT        PPP
UUU         UUU  EEE                   TTT        PPP
UUU         UUU  EEE                   TTT        PPP
UUU         UUU  EEE                   TTT        PPP
UUU         UUU  EEE                   TTT        PPP
UUU         UUU  EEE                   TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE       TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE       TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE       TTT        PPP
```

```
_$


Va
---
000
000
000
7F
7F
7F
7F
7F
7F
7F
7F
7F
```

```
**FILE**ID**RMSTEST7
```

```
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT  77777777
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT  77777777
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT              77
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT              77
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT              77
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT              77
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT            77
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT            77
RR  RR      MM      MM          SS      TT      EE                  SS      TT          77
RR  RR      MM      MM          SS      TT      EE                  SS      TT          77
RR    RR    MM      MM          SS      TT      EE                  SS      TT        77            ....
RR    RR    MM      MM          SS      TT      EE                  SS      TT        77            ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT      77              ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT      77              ....

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
```

```
0000           $BEGIN  RMSTEST7,002,__TMSTEST,<FILE SHARING TEST PROGRAM>,<LONG,GBL>
0000           .ENABL  DBG
0000           .NLIST  MEB
0000
0000    ;
0000    ;      This test program tests the file sharing capability of rms.
0000    ;      Both sharing between processes and within a proc ss is tested.
0000    ;      Sharing between processes is accomplished by submitting a
0000    ;      command file which executes a program which synchronizes itself
0000    ;      with this program, through the use of group event flags.   The
0000    ;      command file which must be present is "detached.com", as well
0000    ;      as the file "detached.exe".
0000    ;
0000    ; Written by:   R. A. Newell    29 aug 78
0000    ;
0000    ; Modified By:
0000    ;
0000    ;      V02-002 REFORMAT        Maria del C. Nasr       24-Jul-1980
0000    ;
0000    ; Revision History:
0000    ;
0000    ;      D. M. Bousquet          1 nov 78
0000    ;      Created a cif race with the two processes and added a prompt
0000    ;      feature that asks for the r cord #, and step
0000    ;
0000    ;
0000    ;
0000    ;      macros
0000    ;
0000
0000           .MACRO  TYPE STRING,?L
0000           STORE   <STRING>
0000           BLBC    VERBOSITY,L
0000           MOVL    #$$.TMPX,CMDORAB+RAB$L_RBF
0000           MOVW    #$$.TMPX1,CMDORAB+RAB$W_RSZ
0000           $PUT    RAB=CMDORAB,ERR=REPORT_ERROR
0000           BSBW    ERR
0000    L:
0000           .ENDM   TYPE
0000
0000    ;
0000
0000           .MACRO  STORE STRING,PRE
0000           .SAVE
0000           .PSECT  __$RMSNAM
0000           $$.TMPX=.
0000           PRE
0000           .ASCII  %STRING%
0000           $$.TMPX1=.-$$.TMPX
0000           .RESTORE
0000           .ENDM   STORE
0000
0000    ;
0000    ;
0000
0000           .ALIGN  LONG
0000    SHRFAB::
```

```
                            0000            $FAB    FAC=<GET,PUT,UPD>,FNM=<TST$DISK:[RMS.TEST]SHARED.DAT>,ORG=IDX,-
                            0000            MRS=80,FOP=CIF,SHR=<PUT,GET>,RFM=VAR,XAB=SHRXAB0
                            0050
                            0050  SHRRAB::
                            0050            $RAB    UBF=RECBUF,USZ=80,FAB=SHRFAB,RBF=RECBUF,RSZ=80,RAC=KEY
                            0094  SHRXAB0::
                            0094            $XABKEY REF=0,POS=0,SIZ=4,DTP=BN4
                            00E0            .ALIGN  LONG
                            00E0  DPFAB:: $FAB    FAC=GET,FNM=<TST$DISK:[RMS.TEST]DETACHED.COM>,-
                            00E0            NAM=BCHBLK,FOP=SCF
                            0130  BCHBLK::
                            0130            $NAM    RSA=NAMBCH,RSS=48
                  000001C0  0190  NAMBCH: .BLKB   48
                            01C0  CLUSTER:
                  00000004' 01C0            .LONG   EFLNG
                  000001C8' 01C4            .LONG   EFNAM
               41 56 4F 4E  01C8  EFNAM:  .ASCII  /NOVA/
                  00000004  01CC            EFLNG=  .-EFNAM
                  0000021C  01CC  RECBUF: .BLKB   80
               54 53 45 54  021C  TSTWRD: .ASCII  /TEST/
                            0220  SH2FAB::
                            0220            $FAB    FAC=<GET,PUT>,FNM=<TST$DISK:[RMS.TEST]SHARED.DAT>,-
                            0220            MRS=80,SHR=<PUT,GET>,ORG=IDX,RFM=VAR,XAB=SHRXAB0
                            0270  SH2RAB::
                            0270            $RAB    UBF=RECBUF,USZ=80,FAB=SH2FAB,RBF=RECBUF,RSZ=80
                            02B4
                            02B4  ;
                            02B4  ;
                            02B4  ;
                            02B4
                            02B4  SHRSTRT::
                      0FFC  02B4            .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                            02B6
                            02B6            $ASCEFC_S       EFN=#64,NAME=CLUSTER; create cluster 2
                FD32'   30  02CB            BSBW    ERR                     ; check for error
                            02CE
                            02CE            $OPEN   FAB=DPFAB,-
                            02CE                    ERR=REPORT_ERROR        ; open detached.com file
                FD1E'   30  02DF            BSBW    ERR                     ; check for error
                            02E2
                            02E2            $CLOSE  FAB=DPFAB,-
                            02E2                    ERR=REPORT_ERROR        ; close and submit com file
                FD0A'   30  02F3            BSBW    ERR                     ; check for error
                            02F6
                            02F6            $CREATE FAB=SHRFAB,-
                            02F6                    ERR=REPORT_ERROR        ; open shared file
                FCF6'   30  0307            BSBW    ERR                     ; check for error
                            030A
             52    50   D0  030A            MOVL    R0,R2                   ; save r0, was it created?
                            030D            $CONNECT        RAB=SHRRAB,-
                            030D                    ERR=REPORT_ERROR        ; connect to shared file
                FCDF'   30  031E            BSBW    ERR                     ; check for error
                            0321
   00000000'8F     52   D1  0321            CMPL    R2,#RMS$_CREATED        ; opened or created?
                   17   12  0328            BNEQ    OK0                     ; opened - synchronized w/d.p.
             52    01   D0  032A            MOVL    #1,R2                   ;  initialize reg 2
        FE9A CF    52   D0  032D  OK:     MOVL    R2,RECBUF               ;  set up record buffer
```

```
                           0332         $PUT    RAB=SHRRAB               ;  put the record
          EC 52    05  F2  033D         AOBLSS  #5,R2,OK                 ;  put 4 records
                           0341  OK0:   $WAITFR_S      EFN=#64           ; wait for sync with d.p.
                           034E         $CLREF_S       EFN=#64           ; clear ef #64
                           035B         TYPE    <*** WE ARE NOW SYNCHRONIZED WITH THE DETACHED PROCESS ***>
                           038A         $SETEF_S       EFN=#65           ; turn on d.p.
                           0397         $WAITFR_S      EFN=#64           ; wait for d.p. for restart
                           03A4
                           03A4  ;
                           03A4  ;      The detached process is now going to read record 1 from the shared
                           03A4  ;      file.  This controlling program will then attempt to read record 2
                           03A4  ;      which is in the same bucket as record 1.  This read should be
                           03A4  ;      successful.  it will then try to read record 1 which should be
                           03A4  ;      automatically locked by the detached process.  Therefore, a record
                           03A4  ;      locked error should be returned to this program.
                           03A4  ;
                           03A4
                           03A4         $CLREF_S       EFN=#64           ; clear ef #64
       FE16 CF    02   D0  03B1         MOVL    #2,RECBUF                ; set key=rec 2
                           03B6         TYPE    < AUTOMATIC LOCKING TEST >
                           03E5         $GET    RAB=SHRRAB,ERR=REPORT_ERROR; get rec 2
             2F 50     E8  03F6         BLBS    R0,OK1                   ; branch if rec read ok
                           03F9         TYPE    <*** UNEXPECTED BUCKET LOCKED STATUS ***>
                           0428                                          ; rec shouldn't be locked
                           0428
       FD9F CF    01   D0  0428  OK1:   MOVL    #1,RECBUF                ; set key=rec 1
                           042D         $GET    RAB=SHRRAB,ERR=REPORT_ERROR; get rec 1
             2F 50     E9  043E         BLBC    R0,OK2                   ; branch if rec locked
                           0441         TYPE    <*** AUTO LOCK FAILURE ON REC 1 *** >
                           0470                                          ; rec should be locked
                           0470
                           0470  OK2:   $SETEF_S       EFN=#65           ; restart d.p.
                           047D         $WAITFR_S      EFN=#64           ; wait for d.p. restart
                           048A
                           048A  ;
                           048A  ;      The detached process will now read record 2.  This should release the
                           048A  ;      automatic lock on record 1, allowing this controlling process to read
                           048A  ;      it.  This program will then attempt to modify record 1 and store it
                           048A  ;      out to the shared file.  If this succeeds, we will read it back and
                           048A  ;      make sure that the update did indeed succeed.  Finally, we will also
                           048A  ;      try to read record 2 as a double-check on the previous test.
                           048A  ;
                           048A
                           048A         $CLREF_S       EFN=#64           ; clear ef #64
       FD30 CF    01   D0  0497         MOVL    #1,RECBUF                ; set key=rec 1
                           049C         $GET    RAB=SHRRAB,-
                           049C                  ERR=REPORT_ERROR        ; get rec 1
             32 50     E8  04AD         BLBS    R0,OK3                   ; branch if rec read ok
                           04B0         TYPE    <*** AUTO UNLOCK FAILURE ***>
                           04DF                                          ; record still locked
              00F8     31  04DF         BRW     OK4                      ; bypass update test
                           04E2
 FCE3 CF    FD36 CF    D0  04E2  OK3:   MOVL    TSTWRD,RECBUF            ; update record 1
                           04E9         TYPE    < UPDATE SHARING TEST >
                           0518         $UPDATE RAB=SHRRAB,-
                           0518                  ERR=REPORT_ERROR        ; put the updated rec
             31 50     E8  0529         BLBS    R0,OK3A                  ; says ok, check it
```

```
                             052C          TYPE      <*** UPDATE ON REC 1 FAILED ***>
                             055B                                                ; record 1 wasn't updated
               7D      11    055B          BRB       OK4                         ; bypass check of update
                             055D
                             055D   OK3A:  $GET      RAB=SHRRAB,-
                             055D                     ERR=REPORT_ERROR           ; get the updated record
            31 50      E8    056E          BLBS      R0,OK3B                     ; branch if rec read ok
                             0571          TYPE      <*** READ OF UPDATED RECORD FAILED ***>
                             05A0                                                ; error on read of rec
               38      11    05A0          BRB       OK4                         ; bypass rest of check
                             05A2
   FC23 CF   FC76 CF   D1    05A2   OK3B:  CMPL      TSTWRD,RECBUF               ; check the update
               2F      13    05A9          BEQL      OK4                         ; branch if update succeeded
                             05AB          TYPE      <*** UPDATE OF RECORD DIDN'T MAKE IT ***>
                             05DA                                                ; comparison failed
                             05DA
        FBED CF      02  D0  05DA   OK4:   MOVL      #2,RECBUF                   ; set key=rec 2
                             05DF          $GET      RAB=SHRRAB,-
                             05DF                     ERR=REPORT_ERROR           ; attempt to read rec 2
            2F 50      E9    05F0          BLBC      R0,OK5                      ; branch if rec not read
                             05F3          TYPE      <*** AUTO LOCK FAILURE ON REC 2 ***>
                             0622                                                ; rec not locked
                             0622
                             0622   OK5:   $SETEF_S          EFN=#65             ; restart d.p.
                             062F          $WAITFR_S         EFN=#64             ; wait for d.p. restart
                             063C
                             063C   ;
                             063C   ;      The detached process will now read record 3 with a manual lock.
                             063C   ;      This program will then attempt to read record 3 and expects to get
                             063C   ;      a locked status returned.  The detached process will then be
                             063C   ;      restarted and record 4 will be read with a manual lock.  This
                             063C   ;      program will then attempt to read both records 3 and 4, and expects
                             063C   ;      to get locked status back on each of the read attempts.
                             063C   ;
                             063C
                             063C          $CLREF_S          EFN=#64             ; clear ef #64
                             0649          TYPE      < MANUAL LOCKING TEST >
        FB4F CF      03  D0  0678          MOVL      #3,RECBUF                   ; set key=rec 3
                             067D          $GET      RAB=SHRRAB,-
                             067D                     ERR=REPORT_ERROR           ; attempt to get rec 3
            2F 50      E9    068E          BLBC      R0,OK6                      ; branch if error on get
                             0691          TYPE      <*** MANUAL LOCK FAILURE ON REC 3 ***>
                             06C0                                                ; rec not locked
                             06C0
                             06C0   OK6:   $SETEF_S          EFN=#65             ; restart d.p.
                             06CD          $WAITFR_S         EFN=#64             ; wait for d.p. restart
                             06DA          $CLREF_S          EFN=#64             ; clear ef #64
        FAE0 CF      04  D0  06E7          MOVL      #4,RECBUF                   ; set key=rec 4
                             06EC          $GET      RAB=SHRRAB,-
                             06EC                     ERR=REPORT_ERROR           ; attempt to get rec 4
            2F 50      E9    06FD          BLBC      R0,OK7                      ; branch if locked
                             0700          TYPE      <*** MANUAL LOCK FAILURE ON REC 4 ***>
                             072F                                                ; should be locked
                             072F
        FA98 CF      03  D0  072F   OK7:   MOVL      #3,RECBUF                   ; set key=rec 3
                             0734          $GET      RAB=SHRRAB,-
                             0734                     ERR=REPORT_ERROR           ; attempt to get rec 3
            2F 50     E9  0745             BLBC     R0,OK8                 ; branch if locked
                          0748             TYPE     <*** PREMATURE MANUAL UNLOCK ON REC 3 ***>
                          0777                                             ; should still be locked
                          0777        OK8: $SETEF_S        EFN=#65         ; restart d.p.
                          0784             $WAITFR_S       EFN=#64         ; wait for d.p. restart
                          0791
                          0791       ;
                          0791       ;     The detached process will now release the manual lock on record 3,
                          0791       ;     but will not unlock record 4.  This program will then try to read
                          0791       ;     both records 3 and 4 again, expecting to be able to read record 3
                          0791       ;     but not record 4.
                          0791       ;
                          0791
                          0791             $CLREF_S        EFN=#64         ; clear ef #64
    FA29 CF    03     D0  079E             MOVL     #3,RECBUF              ; set key=rec 3
                          07A3             $GET     RAB=SHRRAB,-
                          07A3               ERR=REPORT_ERROR              ; get record 3
            2F 50     E8  07B4             BLBS     R0,OK9                 ; branch if rec 3 read ok
                          07B7             TYPE     <*** MANUAL UNLOCK FAILURE ON REC 3 ***>
                          07E6                                             ; should be unlocked
                          07E6
    F9E1 CF    04     D0  07E6        OK9: MOVL     #4,RECBUF              ; set key=rec 4
                          07EB             $GET     RAB=SHRRAB,-
                          07EB               ERR=REPORT_ERROR              ; get record 4
            2F 50     E9  07FC             BLBC     R0,OK10                ; branch if rec not read
                          07FF             TYPE     <*** PREMATURE MANUAL UNLOCK ON REC 4 ***>
                          082E                                             ; should still be locked
                          082E
                          082E       OK10: $SETEF_S        EFN=#65         ; restart d.p.
                          083B             $WAITFR_S       EFN=#64         ; wait for d.p. restart
                          0848
                          0848       ;
                          0848       ;     The detached process will now read record 1 without manual locking.
                          0848       ;     This program will then try to read both records 1 and 4, expecting
                          0848       ;     to fail on both.  The detached process will then read record 2,
                          0848       ;     automatically unlocking record 1.  This program will then again try
                          0848       ;     to read records 1 and 4, this time expecting to be able to read
                          0848       ;     record 1, but still not record 4 as it should still be manually
                          0848       ;     locked.  Finally, the detached process will release record 4 and
                          0848       ;     we will try to read it from this program, expecting to be able to.
                          0848       ;
                          0848
                          0848             $CLREF_S        EFN=#64         ; clear ef #64
    F972 CF    01     D0  0855             MOVL     #1,RECBUF              ; set key=rec 1
                          085A             $GET     RAB=SHRRAB,-
                          085A               ERR=REPORT_ERROR              ; attempt to read rec 1
            2F 50     E9  086B             BLBC     R0,OK11                ; branch if not read
                          086E             TYPE     <*** AUTO LOCK 2 FAILURE ON REC 1 ***>
                          089D                                             ; should be locked
    F92A CF    04     D0  089D       OK11: MOVL     #4,RECBUF              ; set key=rec 4
                          08A2             $GET     RAB=SHRRAB,-
                          08A2               ERR=REPORT_ERROR              ; attempt to read rec 4
            2F 50     E9  08B3             BLBC     R0,OK12                ; branch if not read
                          08B6             TYPE     <*** PREMATURE MANUAL UNLOCK ON REC 4 ***>
                          08E5                                             ; should still be locked
                          08E5
                          08E5       OK12: $SETEF_S        EFN=#65         ; restart d.p.
```

```
                        08F2                $WAITFR_S       EFN=#64         ; wait for d.p. restart
                        08FF                $CLREF_S        EFN=#64         ; clear ef #64
F8BB CF    01     D0    090C                MOVL    #1,RECBUF               ; set key=rec 1
                        0911                $GET    RAB=SHRRAB,-
                        0911                         ERR=REPORT_ERROR       ; attempt to read rec 1
         2F 50    E8    0922                BLBS    R0,OK13                 ; branch if able to read
                        0925                TYPE    <*** AUTO UNLOCK 2 FAILURE ON REC 1 ***>
                        0954                                                ; should be unlocked
                        0954
F873 CF    04     D0    0954      OK13:     MOVL    #4,RECBUF               ; set key=rec 4
                        0959                $GET    RAB=SHRRAB,-
                        0959                         ERR=REPORT_ERROR       ; attempt to read rec 4
         2F 50    E9    096A                BLBC    R0,OK14                 ; branch if unable to read
                        096D                TYPE    <*** PREMATURE UNLOCK 2 FAILURE ON REC 4 ***>
                        099C                                                ; should be locked
                        099C
                        099C      OK14:     $SETEF_S        EFN=#65         ; restart d.p.
                        09A9                $WAITFR_S       EFN=#64         ; wait for d.p. restart
                        09B6                $CLREF_S        EFN=#64         ; clear ef #64
F804 CF    04     D0    09C3                MOVL    #4,RECBUF               ; set key=rec 4
                        09C8                $GET    RAB=SHRRAB,-
                        09C8                         ERR=REPORT_ERROR       ; attempt to read rec 4
         2F 50    E8    09D9                BLBS    R0,OK15                 ; branch if able to read
                        09DC                TYPE    <*** MANUAL UNLOCK 2 FAILURE ON REC 4 ***>
                        0A0B                                                ; should be unlocked
                        0A0B
                        0A0B      OK15:     $RELEASE RAB=SHRRAB,-
                        0A0B                          ERR=REPORT_ERROR      ; release record 4
         2F 50    E8    0A1C                BLBS    R0,OK16                 ; branch if successful
                        0A1F                TYPE    <*** RELEASE FAILURE ON REC 4 ***>
                        0A4E                                                ; release failed??
                        0A4E
                        0A4E      OK16:
                        0A4E
                        0A4E      ;
                        0A4E      ;         We have completed the file sharing tests for multi-process
                        0A4E      ;         sharing of a file.  Now we will test file sharing within one
                        0A4E      ;         process.  This is accomplished by multiple opens on the same
                        0A4E      ;         file.  The results should be identical to those experienced
                        0A4E      ;         by the multi-process file sharing tests.
                        0A4E      ;
                        0A4E      ;         We begin by closing the current shared file and the doing two
                        0A4E      ;         opens on it using different fabs and rabs.
                        0A4E      ;
                        0A4E
                        0A4E                $DISCONNECT RAB=SHRRAB,-
                        0A4E                             ERR=REPORT_ERROR   ; disconnect from file
         F59E'    30    0A5F                BSBW    ERR                     ; check for error
                        0A62                $CLOSE  FAB=SHRFAB,-
                        0A62                         ERR=REPORT_ERROR       ; close the file
         F58A'    30    0A73                BSBW    ERR                     ; check for error
                        0A76
                        0A76      ;
                        0A76      ;         ok, we are ready to start the multi-open tests
                        0A76      ;
                        0A76
                        0A76                TYPE    < MULTI-OPEN SHARING TEST >
```

```
                     0AA5              $OPEN    FAB=SHRFAB,-
                     0AA5                        ERR=REPORT_ERROR        ; open the shared file
       F547'  30     0AB6              BSBW     ERR                     ; check for error
                     0AB9              $CONNECT RAB=SHRRAB,-
                     0AB9                        ERR=REPORT_ERROR        ; connect to the file
       F533'  30     0ACA              BSBW     ERR                     ; check for error
                     0ACD              $OPEN    FAB=SH2FAB,-
                     0ACD                        ERR=REPORT_ERROR        ; open it again
       F51F'  30     0ADE              BSBW     ERR                     ; check for error
                     0AE1              $CONNECT RAB=SHRRAB,-
                     0AE1                        ERR=REPORT_ERROR        ; connect to it again
       F50B'  30     0AF2              BSBW     ERR                     ; check for error
                     0AF5
                     0AF5        ;
                     0AF5        ;     RAB 1 will now read the first record of the file, hopefully
                     0AF5        ;     causing an automatic lock to come up on it.  RAB 2 will try to
                     0AF5        ;     read it, expecting to get an error.
                     0AF5        ;
                     0AF5
F6D2 CF    01    D0  0AF5              MOVL     #1,RECBUF               ; set key=rec 1
                     0AFA              $GET     RAB=SHRRAB,-
                     0AFA                        ERR=REPORT_ERROR        ; get record 1
       F4F2'  30     0B0B              BSBW     ERR                     ; check for error
                     0B0E              $GET     RAB=SH2RAB,-
                     0B0E                        ERR=REPORT_ERROR        ; attempt to get rec 1
      31 50   E9     0B1F              BLBC     R0,OK20                 ; branch if rec not read
                     0B22              TYPE     <*** MULTI-OPEN AUTO LOCK FAILURE ***>
                     0B51                                               ; rec should be locked
         61   11     0B51              BRB      OK21                    ; bypass auto lock test
                     0B53
                     0B53        OK20:
                     0B53
                     0B53        ;
                     0B53        ;     Now rab 1 will read record 2.  This should automatically unlock
                     0B53        ;     record 1 and rab 2 should be able to read it.
                     0B53        ;
                     0B53
F674 CF    02    D0  0B53              MOVL     #2,RECBUF               ; set key=rec 2
                     0B58              $GET     RAB=SHRRAB,-
                     0B58                        ERR=REPORT_ERROR        ; get record 2
       F494'  30     0B69              BSBW     ERR                     ; check for error
F65B CF    01    D0  0B6C              MOVL     #1,RECBUF               ; set key=rec 1
                     0B71              $GET     RAB=SH2RAB,-
                     0B71                        ERR=REPORT_ERROR        ; rab 2 tries to read rec 1
      2F 50   E8     0B82              BLBS     R0,OK21                 ; branch if rec 2 read ok
                     0B85              TYPE     <*** MULTI-OPEN AUTO UNLOCK FAILURE ***>
                     0BB4                                               ; rec should be unlocked
                     0BB4
                     0BB4        OK21:
                     0BB4
                     0BB4        ;
                     0BB4        ;     Continuing with the multi-open file sharing tests, we shall now
                     0BB4        ;     read record 3 (rab 1) with manual locking specified.  This should
                     0BB4        ;     unlock record 2 and lock record 3, as will be verified by read
                     0BB4        ;     attempts by rab 2.
                     0BB4        ;
                     0BB4
```

```
                      0BB4          $RAB_STORE RAB=SHRRAB,ROP=ULK    ; set manual lock bit
F606 CF    03    D0   0BC1          MOVL    #3,RECBUF                ; set key=rec 3
                      0BC6          $GET    RAB=SHRRAB,-
                      0BC6                   ERR=REPORT_ERROR        ; read record 3
         F426'   30   0BD7          BSBW    ERR                      ; check for error
F5ED CF    02    D0   0BDA          MOVL    #2,RECBUF                ; set key=rec 2
                      0BDF          $GET    RAB=SH2RAB,-
                      0BDF                   ERR=REPORT_ERROR        ; rab 2 read rec 2
        2F 50    E8   0BF0          BLBS    R0,OK22                  ; branch if rec read ok
                      0BF3          TYPE    <*** MULTI-OPEN AUTO UNLOCK 2 FAILURE ***>
                      0C22                                           ; rec should be unlocked
                      0C22
F5A5 CF    03    D0   0C22  OK22:   MOVL    #3,RECBUF                ; set key=rec 3
                      0C27          $GET    RAB=SH2RAB,-
                      0C27                   ERR=REPORT_ERROR        ; attempt to read rec 3
        2F 50    E9   0C38          BLBC    R0,OK23                  ; branch if rec not read
                      0C3B          TYPE    <*** MULTI-OPEN MANUAL LOCK FAILURE ***>; rec should be locked
                      0C6A
                      0C6A  OK23:
                      0C6A
                      0C6A  ;
                      0C6A  ;       Now we will read record 4 with manual locking.  This should not,
                      0C6A  ;       however, unlock the manually locked record 3.
                      0C6A  ;
                      0C6A
F55D CF    04    D0   0C6A          MOVL    #4,RECBUF                ; set key=rec 4
                      0C6F          $GET    RAB=SHRRAB,-
                      0C6F                   ERR=REPORT_ERROR        ; read record 4
         F37D'   30   0C80          BSBW    ERR                      ; check for error
F544 CF    03    D0   0C83          MOVL    #3,RECBUF                ; set key=rec 3
                      0C88          $GET    RAB=SH2RAB,-
                      0C88                   ERR=REPORT_ERROR        ; attempt to read rec 3
        2F 50    E9   0C99          BLBC    R0,OK24                  ; branch if rec not read
                      0C9C          TYPE    <*** MULTI-OPEN PREMATURE MANUAL UNLOCK ***>
                      0CCB                                           ; rec should be locked
                      0CCB
F4FC CF    04    D0   0CCB  OK24:   MOVL    #4,RECBUF                ; set key=rec 4
                      0CD0          $GET    RAB=SH2RAB,-
                      0CD0                   ERR=REPORT_ERROR        ; read record 4
        2F 50    E9   0CE1          BLBC    R0,OK25                  ; branch if rec not read
                      0CE4          TYPE    <*** MULTI-OPEN MANUAL LOCK 2 FAILURE ***>
                      0D13                                           ; rec should be locked
                      0D13
                      0D13  OK25:
                      0D13
                      0D13  ;
                      0D13  ;       The test program will not unlock record 3, but will maintain the
                      0D13  ;       lock on record 4.
                      0D13  ;
                      0D13
F4B4 CF    03    D0   0D13          MOVL    #3,RECBUF                ; set key=rec 4
                      0D18          $RAB_STORE RAB=SHRRAB,ROP=#0     ; clear manual lock bit
                      0D20          $GET    RAB=SHRRAB,-
                      0D20                   ERR=REPORT_ERROR        ; read record 3 again
         F2CC'   30   0D31          BSBW    ERR                      ; check for error
                      0D34          $RELEASE RAB=SHRRAB,-
                      0D34                     ERR=REPORT_ERROR      ; release record 3
         F2B8'  30  0D45              BSBW    ERR                           ; check for error
                    0D48              $GET    RAB=SH2RAB,-
                    0D48                       ERR=REPORT_ERROR             ; rab2 tries to read rec 3
          2F 50  E8  0D59              BLBS    R0,OK26                       ; branch if read ok
                    0D5C              TYPE    <*** MULTI-OPEN MANUAL UNLOCK FAILURE ***>
                    0D8B                                                    ; rec should be unlocked
                    0D8B
                    0D8B     OK26:
                    0D8B
                    0D8B     ;
                    0D8B     ;        Finally, the test program will read record 5 without manual
                    0D8B     ;        locking.  This should put an auto lock on rec 5 and also
                    0D8B     ;        maintain the manual lock on record 4.
                    0D8B     ;
                    0D8B
   F43C CF    05  D0  0D8B              MOVL    #5,RECBUF                     ; set key=rec 5
                    0D90              $GET    RAB=SHRRAB,-
                    0D90                       ERR=REPORT_ERROR             ; read record 5
         F25C'  30  0DA1              BSBW    ERR                           ; check for error
   F423 CF    04  D0  0DA4              MOVL    #4,RECBUF                     ; set key=rec 4
                    0DA9              $GET    RAB=SH2RAB,-
                    0DA9                       ERR=REPORT_ERROR             ; read rec 4
          2F 50  E9  0DBA              BLBC    R0,OK27                       ; branch if rec not read
                    0DBD              TYPE    <*** MULTI-OPEN PREMATURE MANUAL UNLOCK 2 ***>
                    0DEC                                                    ; rec should be locked
                    0DEC
                    0DEC     OK27:
                    0DEC
                    0DEC     ;
                    0DEC     ;        This completes the test section on multi-open file sharing.
                    0DEC     ;
                    0DEC
                    0DEC              $EXIT_S                               ; end
                    0DF5              .END
```

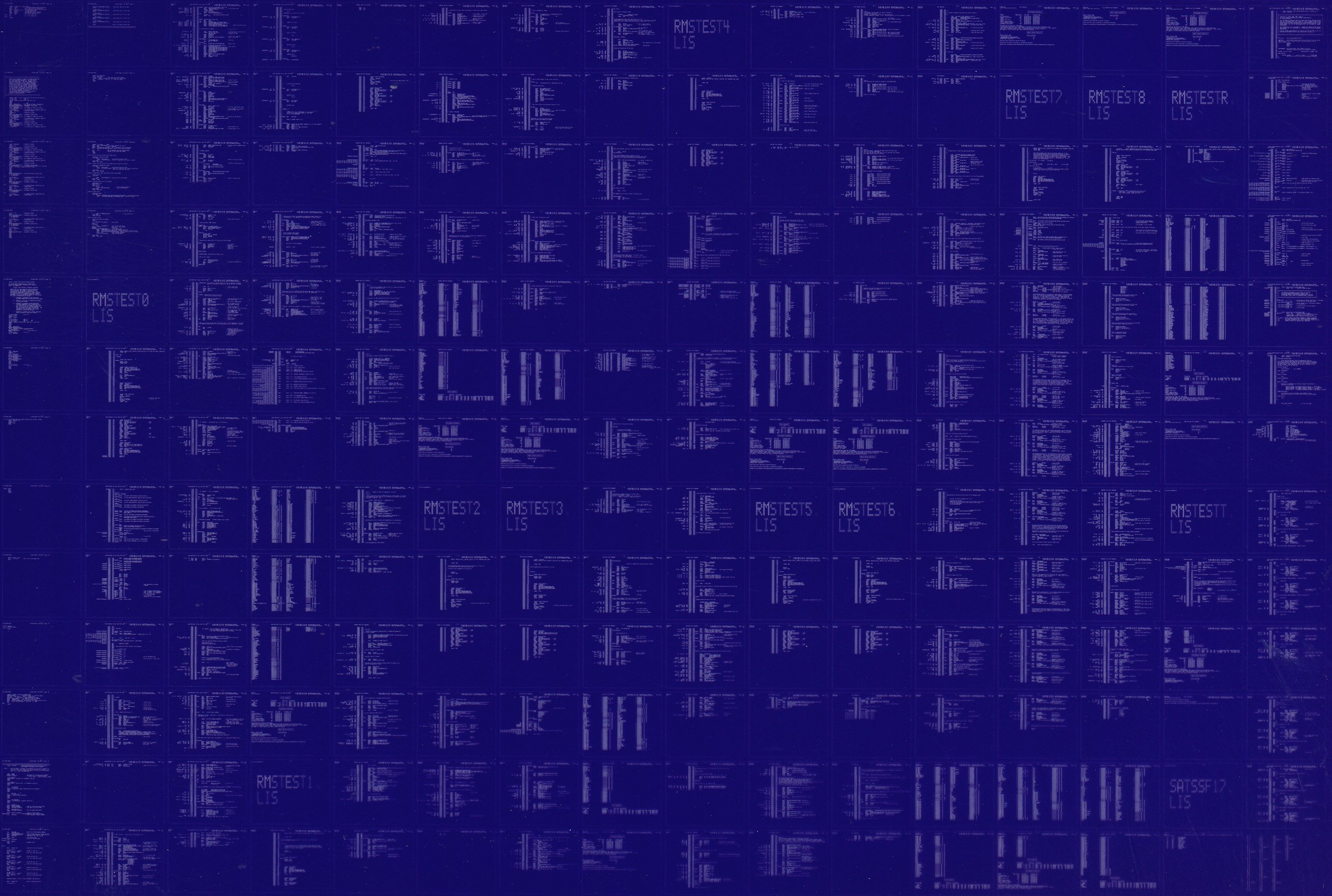
```
$$.DTPTMP                       = 00000004     D               OK1                                   00000428 R   D   01
$$.PSECT_EP                     = 00000000                     OK10                                  0000082E R   D   01
$$.SPTMP                        = 00000000     D               OK11                                  0000089D R   D   01
$$.STMP                         = 00000004     D               OK12                                  000008E5 R   D   01
$$.TAB                          = 00000270 R   D   01          OK13                                  00000954 R   D   01
$$.TABEND                       = 000002B4 R   D   01          OK14                                  0000099C R   D   01
$$.TMP                          = 00000000     D               OK15                                  00000A0B R   D   01
$$.TMP1                         = 00000002     D               OK16                                  00000A4E R   D   01
$$.TMP2                         = 000000EF                     OK2                                   00000470 R   D   01
$$.TMPX                         = 0000043A R   D   04          OK20                                  00000B53 R   D   01
$$.TMPX1                        = 0000002C     D               OK21                                  00000BB4 R   D   01
$$RMSTEST                       = 0000001E                     OK22                                  00000C22 R   D   01
$$RMS_PBUGCHK                   = 00000010                     OK23                                  00000C6A R   D   01
$$RMS_TBUGCHK                   = 00000008                     OK24                                  00000CCB R   D   01
$$RMS_UMODE                     = 00000004                     OK25                                  00000D13 R   D   01
$$T1                            = 00000000     D               OK26                                  00000D8B R   D   01
..AFLG                          = 00000000     D               OK27                                  00000DEC R   D   01
..FLG                           = 00000001     D               OK3                                   000004E2 R   D   01
..MOD                           = 00000000     D               OK3A                                  0000055D R   D   01
..N                             = 00000001                     OK3B                                  000005A2 R   D   01
..TYP                           = 00000000     D               OK4                                   000005DA R   D   01
.LEN                            = 00000004     D               OK5                                   00000622 R   D   01
BCHBLK                            00000130 RG  D   01          OK6                                   000006C0 R   D   01
CLUSTER                           000001C0 R   D   01          OK7                                   0000072F R   D   01
CMDORAB                           ********    X    01          OK8                                   00000777 R   D   01
DPFAB                             000000E0 RG  D   01          OK9                                   000007E6 R   D   01
EFLNG                           = 00000004     D               RAB$B_RAC                           = 0000001E     D
EFNAM                             000001C8 R   D   01          RAB$C_BID                           = 00000001     D
ERR                               ********    X    01          RAB$C_BLN                           = 00000044     D
FAB$B_FNS                       = 00000034     D               RAB$C_KEY                           = 00000001     D
FAB$C_BID                       = 00000003     D               RAB$C_SEQ                           = 00000000     D
FAB$C_BLN                       = 00000050     D               RAB$L_CTX                           = 00000018     D
FAB$C_IDX                       = 00000020     D               RAB$L_RBF                           = 00000028     D
FAB$C_SEQ                       = 00000000     D               RAB$L_ROP                           = 00000004     D
FAB$C_VAR                       = 00000002     D               RAB$V_ULK                           = 00000012     D
FAB$L_ALQ                       = 00000010     D               RAB$W_RSZ                           = 00000022     D
FAB$L_FNA                       = 0000002C     D               RECBUF                                000001CC R   D   01
FAB$L_FOP                       = 00000004     D               REPORT_ERROR                          ********    X    01
FAB$V_CHAN_MODE                 = 00000002     D               RMS$_CREATED                          ********    X    01
FAB$V_CIF                       = 00000019     D               SH2FAB                                00000220 RG  D   01
FAB$V_FILE_MODE                 = 00000004     D               SH2RAB                                00000270 RG  D   01
FAB$V_GET                       = 00000001     D               SHRFAB                                00000000 RG  D   01
FAB$V_LNM_MODE                  = 00000000     D               SHRRAB                                00000050 RG  D   01
FAB$V_PUT                       = 00000000     D               SHRSTRT                               000002B4 RG  D   01
FAB$V_SCF                       = 0000000E     D               SHRXABO                               00000094 RG  D   01
FAB$V_UPD                       = 00000003     D               SYS$ASCEFC                            ********   GX    01
FAB$W_GBC                       = 00000048     D               SYS$CLOSE                             ********   GX    01
NAM$B_ESS                       = 0000000A     D               SYS$CLREF                             ********   GX    01
NAM$B_NOP                       = 00000008     D               SYS$CONNECT                           ********   GX    01
NAM$B_RSS                       = 00000002     D               SYS$CREATE                            ********   GX    01
NAM$C_BID                       = 00000002     D               SYS$DISCONNECT                        ********   GX    01
NAM$C_BLN                       = 00000060     D               SYS$EXIT                              ********   GX    01
NAM$L_ESA                       = 0000000C     D               SYS$GET                               ********   GX    01
NAM$L_RSA                       = 00000004     D               SYS$OPEN                              ********   GX    01
NAMBCR                            00000190 R   D   01          SYS$PUT                               ********   GX    01
OK                                0000032D R   D   01          SYS$RELEASE                           ********   GX    01
OK0                               00000341 R   D   01          SYS$SETEF                             ********   GX    01
```

```
SYS$UPDATE                       ********  GX   01
SYS$WAITFR                       ********  GX   01
TSTWRD                           0000021C R  D  01
VERBOSITY                        ********   X   01
XAB$B_DAN                      = 0000000A    D
XAB$B_DTP                      = 00000013    D
XAB$B_FLG                      = 00000012    D
XAB$B_IAN                      = 00000008    D
XAB$B_LAN                      = 00000009    D
XAB$B_NUL                      = 00000015    D
XAB$B_PROLOG                   = 00000048    D
XAB$B_REF                      = 00000017    D
XAB$B_SIZ0                     = 0000002E    D
XAB$C_BN2                      = 00000002    D
XAB$C_BN4                      = 00000004    D
XAB$C_KEY                      = 00000015    D
XAB$C_KEYLEN                   = 0000004C    D
XAB$C_PAC                      = 00000005    D
XAB$L_KNM                      = 00000038    D
XAB$L_NXT                      = 00000004    D
XAB$W_DFL                      = 0000001C    D
XAB$W_IFL                      = 0000001A    D
XAB$W_POS0                     = 0000001E    D


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

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
__TMSTEST                        00000DF5  ( 3573.)  01 (  1.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD     WRT NOVEC LONG
$ABS$                            00000000  (    0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$RMSNAM                          00000059  (   89.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
__$RMSNAM                        00000466  ( 1126.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE


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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                       38     00:00:00.06    00:00:01.01
Command processing                  140     00:00:00.66    00:00:02.40
Pass 1                              380     00:00:17.31    00:00:47.43
Symbol table sort                     0     00:00:00.85    00:00:00.93
Pass 2                              124     00:00:03.78    00:00:09.86
Symbol table output                  17     00:00:00.15    00:00:00.51
Psect synopsis output                 2     00:00:00.03    00:00:00.03
Cross-reference output                0     00:00:00.00    00:00:00.00
Assembler run totals                703     00:00:22.85    00:01:02.18

The working set limit was 1650 pages.
87333 bytes (171 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 642 non-local and 31 local symbols.
532 source lines were read in Pass 1, producing 64 object records in Pass 2.
66 pages of virtual memory were used to define 49 macros.
```

```
                                                 +----------------------------+
                                                 ! Macro library statistics !
                                                 +----------------------------+

Macro library name                               Macros defined
------------------                               --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     42
TOTALS (all libraries)                                42

1202 GETS were required to define 42 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSTEST7/OBJ=OBJ$:RMSTEST7 MSRC$:RMSTEST7/UPDATE=(ENH$:RMSTEST7)+EXECML$/LIB
```

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



RMSTEST4
LIS

RMSTEST7
LIS

RMSTEST8
LIS

RMSTESTR
LIS

RMSTEST0
LIS

RMSTEST2
LIS

RMSTEST3
LIS

RMSTEST5
LIS

RMSTEST6
LIS

RMSTESTT
LIS

RMSTEST1
LIS

SATSSF17
LIS