

UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	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	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	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	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	

\_s  
Va  
--  
000  
000  
000  
7F1  
7F1  
7F1  
7F1  
7F1  
7F1  
7F1  
7F1

```

RRRRRRR      MM      MM      SSSSSSSS  TTTTTTTTTT  EEEEEEEEE  SSSSSSSS  TTTTTTTTTT  11
RRRRRRR      MM      MM      SSSSSSSS  TTTTTTTTTT  EEEEEEEEE  SSSSSSSS  TTTTTTTTTT  11
RR      RR    MMMM    MMMM  SS      TT      EE      SS      TT      1111
RR      RR    MMMM    MMMM  SS      TT      EE      SS      TT      1111
RR      RR    MM      MM    SS      TT      EE      SS      TT      11
RR      RR    MM      MM    SS      TT      EE      SS      TT      11
RRRRRRR      MM      MM      SSSSSS    TT      EEEEEEE  SSSSSS    TT      11
RRRRRRR      MM      MM      SSSSSS    TT      EEEEEEE  SSSSSS    TT      11
RR      RR    MM      MM      SS      TT      EE      SS      TT      11
RR      RR    MM      MM      SS      TT      EE      SS      TT      11
RR      RR    MM      MM      SS      TT      EE      SS      TT      11
RR      RR    MM      MM      SS      TT      EE      SS      TT      11
RR      RR    MM      MM      SSSSSSSS  TT      EEEEEEEEE  SSSSSSSS  TT      111111
RR      RR    MM      MM      SSSSSSSS  TT      EEEEEEEEE  SSSSSSSS  TT      111111

```

```

LL      IIIII  SSSSSSSS
LL      IIIII  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
LL      IIIII  SSSSSSSS
LLLLLLLLL IIIII  SSSSSSSS
LLLLLLLLL IIIII  SSSSSSSS

```

```

...
...
...

```

```

0000 64          $BEGIN RMSTEST1,009,_,_RMSTEST,<GENERAL RMS TEST PROGRAM>,<GBL, LONG>
0000 65
0000 66 ;
0000 67
0000 68          .ENABL  DBG
0000 69
0000 70
0000 71          : this program tests the sequential file org, for a disk device,
0000 72          : with vfc record format.
0000 73          :
0000 74          :
0000 75          test 1a:
0000 76          :
0000 77          : create known test file of 1000 records, where each record has
0000 78          : a 4-byte fixed control field giving the record # and a variable
0000 79          : length portion of n mod 100 bytes of the ascii character n mod 10
0000 80          : (where 'n' is the record number).
0000 81          :
0000 82          :
0000 83          $RMSDEF
0000 84          .NLIST  MEB
0000 85
0000 86          :
0000 87          macros:
0000 88          :
0000 89          :
0000 90          .MACRO  BUFF  NAM,SIZE
0000 91  NAM'BUF::
0000 92          .BLKB   SIZE
0000 93          NAM'BSZ==SIZE
0000 94          .ENDM   BUFF
0000 95
0000 96          :
0000 97          :
0000 98          :
0000 99          :
0000 100         .MACRO  TYPE  STRING, ?L
0000 101         STORE  <STRING>
0000 102         BLBC   VERBOSITY,L
0000 103         MOVL   $$$,TMPX,CMDORAB+RAB$RBF
0000 104         MOVW  $$$,TMPX1,CMDORAB+RAB$RSZ
0000 105         $PUT  RAB=CMDORAB,ERR=REPORT_ERROR
0000 106         BSBW  ERR
0000 107         L:
0000 108         .ENDM   TYPE
0000 109
0000 110          :
0000 111          :
0000 112         .MACRO  WTYPE  STRING
0000 113         $WAIT  RAB=CMDORAB
0000 114         TYPE  <STRING>
0000 115         .ENDM  WTYPE
0000 116
0000 117         .MACRO  WFIELD  STRING
0000 118         $WAIT  RAB=CMDORAB
0000 119         FIELD  <STRING>
0000 120         .ENDM

```

RMSTEST1  
V04-000

GENERAL RMS TEST PROGRAM ;

B 5

16-SEP-1984 01:45:37 VAX/VMS Macro V04-00  
5-SEP-1984 04:21:39 [UETP.SRC]RMSTEST1.MAR;1

Page 2  
(1)

RM  
VO

0000 121  
0000 122 ;  
0000 123

```
0000 125      .MACRO STORE STRING,PRE
0000 126      .SAVE
0000 127      .PSECT  _$_RMSNAM
0000 128      $$TMPX=.
0000 129      PRE                                ; store any carriage control info
0000 130      .ASCII %STRING%
0000 131      $$TMPX1=-$$TMPX
0000 132      .RESTORE
0000 133      .ENDM  STORE
0000 134
0000 135      :
0000 136
0000 137      .MACRO BEGIN TSTNAM
0000 138      STORE <TSTNAM>
0000 139      MOVL  #$$TMPX,BEG_DESCR+4      ; addr
0000 140      MOVL  #$$TMPX1,BEG_DESCR      ; len
0000 141      BSBW  BEGPUT
0000 142      .ENDM  BEGIN
0000 143      .MACRO FINISH TSTNAM
0000 144      STORE <TSTNAM>
0000 145      MOVL  #$$TMPX,FIN_DESCR+4    ; addr
0000 146      MOVL  #$$TMPX1,FIN_DESCR    ; len
0000 147      BSBW  FINPUT
0000 148      .ENDM  FINISH
0000 149      .MACRO FIELD FLDNAM
0000 150      STORE <FLDNAM>
0000 151      MOVL  #$$TMPX,FLD_DESCR+4    ; addr
0000 152      MOVL  #$$TMPX1,FLD_DESCR    ; len
0000 153      BSBW  FLDPUT
0000 154      .ENDM  FIELD
0000 155      .MACRO MBPT, ?L
0000 156      BLBC  VERBOSITY,L
0000 157      BPT
0000 158      L:
0000 159      .ENDM  MBPT
0000 160
0000 161      :
0000 162
```

```

00000000 164 .PSECT RMSTEST,GBL,LONG
0000 165 .ALIGN LONG
0000 166 T1START::
0000 167 WTRAB:: $RAB
00000364 0044 168 RFATBL: .BLKQ 100
0364 169 T1FAB:: $FAB FAC=PUT,FNM=<TST$DISK:T1FILE.DAT;1>,org=seq,rfm=vfc,-
0364 170 RAT=CR,FSZ=4,MRS=100,NAM=NAMBLK,FOP=<SUP,CTG>,-
0364 171 ALQ=48,DEQ=12,SHR=<PUT,GET,UPI>
03B4 172 T1RAB:: $RAB FAB=T1FAB,UBF=CPYBUF,USZ=CPYBSZ,RBF=CPYBUF,MBC=4,MBF=2,-
03B4 173 ROP=<WBH>,RHB=RECCNT,KBF=RECCNT
03F8 174 RECCNT::
00000000 00000000 03F8 175 .LONG 0,0
0400 176
0400 177 ;
0400 178
00000408'00000025' 0400 179 T1STR: .LONG T1L,T1S
44 52 4F 43 45 52 20 2E 4C 55 34 21 0408 180 T1S: .ASCII '!4UL. RECORDS WRITTEN. RFA = !XL,!XW'
20 20 2E 4E 45 54 54 49 52 57 20 53 0414
58 21 2C 4C 58 21 20 3D 20 41 46 52 0420
57 042C
00000025 042D 181 T1L=-T1S
042D 182
042D 183 ;
042D 184
00000435'0000005F' 042D 185 T2STR: .LONG T2L,T2S
20 3D 20 23 20 44 52 4F 43 45 52 20 0435 186 T2S: .ASCII 'RECORD # = !4UL, RHB = !4UL!/'
20 3D 20 42 48 52 20 2C 4C 55 34 21 0441
2F 21 4C 55 34 21 044D
4C 55 33 21 20 3D 20 45 5A 49 53 20 0453 187 .ASCII 'SIZE = !3UL, RSZ = !UW!/'
21 57 55 21 20 3D 20 5A 53 52 20 2C 045F
2F 046B
3D 20 41 46 52 20 44 45 56 41 53 20 046C 188 .ASCII 'SAVED RFA = !XL,!XW, FILE RFA = !XL,!XW'
49 46 20 2C 57 58 21 2C 4C 58 21 20 0478
4C 58 21 20 3D 20 41 46 52 20 45 4C 0484
57 58 21 2C 0490
0000005F 0494 189 T2L=-T2S
0494 190
0494 191 ;
0494 192
0000049C'0000000F' 0494 193 T3STR: .LONG T3L,T3S
46 41 21 27 3D 20 44 52 4F 43 45 52 049C 194 T3S: .ASCII 'RECORD = '!AF'!/'
2F 21 27 04A8
0000000F 04AB 195 T3L=-T3S
01 04AB 196 RHBSW: .BYTE 1 ; switch for modifying rhb contents

```

```

04AC 198 RMT$TEST_1A:
OFFC 04AC 199      _WORD      ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
      04AE 200      BEGIN      <SEQUENTIAL TESTS>
FB36 CF 00000002'EF BO 04C3 201      MOVW      CMDORAB+RAB$W_ISI,WTRAB+RAB$W_ISI; copy output isi for
      04CC 202
      04CC 203
      04CC 204 :DIFFERENT RAB $WAIT TEST
      04CC 205
      04CC 206
FE93 CF 00100000 8F CA 04CC 207      BICL2     #FAB$M_CTG,T1FAB+FAB$L_FOP; don't want ctg set
      00000004'EF 01 C8 04D5 208      BISL2     #RAB$M_ASY,CMDORAB+RAB$L_ROP
      FF17 CF 01 DO 04DC 209      MOVL     #1,RECCNT
      58 FB5F CF DE 04E1 210      MOVAL   RFATBL,R8
      FB06' 30 04E6 211      $CREATE  FAB=T1FAB,ERR=REPORT_ERROR
      58 FEB6 CF DE 04F7 212      BSBW    ERR
      FAEF' 30 04FA 213      MOVAL   T1RAB,R11
      04FF 214      $CONNECT RAB=R11,ERR=REPORT_ERROR
      050E 215      BSBW    ERR
      0511 216
      0511 217 ;
      0511 218
52 FEDE CF 00000064 8F 7B 0511 219 NXTREC: EDIV #100,RECCNT,R2,R6 ; compute record length
      56 051B
57 52 FED7 CF 0A 7B 051C 220      EDIV    #10,RECCNT,R2,R7 ; compute character for record
      57 30 80 0523 221      ADDB   #48,R7 ; make it ascii
      22 AB 56 80 0526 222      MOVW   R6,RAB$W_RSZ(R11) ; tell rms the record size
56 57 6E 00 2C 052A 223      MOVCS  #0,(SP),R7,R6,CPYBUF ; fill record with character
      00000000'EF 052F
      FABA' 30 0534 224      $PUT    RAB=R11,ERR=REPORT_ERROR
      30 57 D1 0543 225      BSBW   ERR
      08 12 0546 226      CMPL   R7,#48 ; is this 10th record?
      88 10 AB 7D 0549 227      BNEQ   T1CNT ; branch if not
      56 D5 054B 228      MOVQ   RAB$W_RFA(R11),(R8)+ ; save rfa in table
      03 12 054F 229      TSTL   R6 ; is this 100th record?
      093F 30 0551 230      BNEQ   T1CNT
      0553 231      BSBW   TYPRFA ; type out the rfa
      0556 232
      0556 233 ;
      0556 234
FFB1 FE9A CF 01 03E8 8F 3D 0556 235 T1CNT: ACBW #1000,#1,RECCNT,NXTREC
      FABC' 30 0560 236      $CLOSE  FAB=T1FAB,ERR=REPORT_ERROR
      FE3E CF B4 0571 237      BSBW   ERR
      0574 238      CLRW   T1RAB+RAB$W_ISI ; allow rab to be re-used
      0578 239      $WAIT  CMDORAB
      0585 240      TYPE  <FINISHED TEST 1A - CREATED TEST FILE>

```





```

06F1 297
06F1 298
06F1 299 : test 1c
06F1 300
06F1 301 : use update to modify records created in test 1a
06F1 302
06F1 303
06F1 304
FC48 CF 08 88 072D 305 WTTYPE <START TEST 1C - $UPDATE>
      F8BA' 30 0732 306 BISB2 #FABS$M_UPD,T1FAB+FABS$B_FAC
      02 AB B4 0743 307 $OPEN FAB=T1FAB,ERR=REPORT_ERROR
      F8A5' 30 0746 308 BSBW ERR
      1E AB 02 90 0749 309 CLRW RABS$W_ISI(R11)
      59 0A D0 0758 310 $CONNECT RAB=R11,ERR=REPORT_ERROR
      58 F8DE CF DE 075B 311 BSBW ERR
      57 D4 075F 312 MOVB #RABS$C_RFA,RABS$B_RAC(R11)
      0762 313 MOVL #10,R9 ; start at record 10
      0767 314 MOVAL RFATBL,R8
      0769 315 CLRL R7 ; get single record
      0769 316
      0769 317
      0769 318 T1BLOOP4:
      FD34 CF 059A 30 0769 319 BSBW GETANDCHK
      FC88 CF D6 076C 320 INCL RECCNT ; modify rhb
      FD37 CF 8E 0770 321 MNEGB RHBSW,RHBSW ; toggle rhb modify flag
      03 19 0777 322 BLSS 10$
      2C AB D4 0779 323 CLRL RABS$L_RHB(R11) ; don't modify rhb (default)
      55 28 AB D0 077C 324 10$: MOVL RABS$L_RBF(R11),R5 ; get record addr
      54 22 AB 3C 0780 325 MOVL RABS$W_RSZ(R11),R4 ; and len
      85 96 0784 326 20$: INCB (R5)+ ; modify record contents
      FB 54 F5 0786 327 SOBGTR R4,20$
      F865' 30 0789 328 $UPDATE RAB=R11,ERR=REPORT_ERROR
      2C AB FC59 CF DE 0798 329 BSBW ERR
      58 08 C0 079B 330 MOVAL RECCNT,RABS$L_RHB(R11) ; restore rhb addr
      FFBD 59 0A 03E8 8F 3D 07A1 331 ADDL #8,R8 ; bump to next rfa entry
      07A4 332 ACBW #1000,#10,R9,T1BLOOP4 ; modify every 10th record
      07AC 333
      07AC 334 :
      07AC 335 : now reread the modified file and check that every 10th (and only
      07AC 336 : every 10th) record has been correctly modified
      07AC 337 :
      07AC 338
      58 F894 CF DE 07AC 339 MOVAL RFATBL,R8
      59 D4 07B1 340 CLRL R9 ; reset record #
      1E AB 00 90 07B3 341 MOVB #RABS$C_SEQ,RABS$B_RAC(R11)
      07B7 342
      07B7 343 :
      07B7 344 : $disconnect r11
      07B7 345 : bsbw err
      07B7 346 : $connect r11 ; do an effective rewind
      07B7 347 : bsbw err
      07B7 348 :
      07B7 349 :
      F837' 30 07B7 350 $REWIND RAB=R11,ERR=REPORT_ERROR; so do a rewind
      07C6 351 BSBW ERR
      07C9 352
      07C9 353 :

```

```

07C9 354
07C9 355 T1BLOOP5:
57 D6 07C9 356 INCL R7 ; set switch for no rfa compare
09 DD 07CB 357 PUSHL #9 ; # gets of unmodified recs
59 D6 07CD 358 10$: INCL R9 ; bump rec #
F81F' 30 07CF 359 $GET RAB=R11,ERR=REPORT_ERROR
057A 30 07DE 360 BSBW ERR
E6 6E FS 07E1 361 BSBW CHKREC
8E D5 07E4 362 SOBGTR (SP),10$
59 D6 07E7 363 TSTL (SP)+
07E9 364 INCL R9 ; bump rec #
07EB 365 $GET RAB=R11,ERR=REPORT_ERROR
F803' 30 07FA 366 BSBW ERR
FCAS CF FCAB CF 8E 07FD 367 CLRL R7 ; specify rfa to be checked
04 19 0806 369 BLSS 15$ ; toggle rhb modified switch
50 FBE7 CF 59 C3 0808 370 INCL RECCNT
50 D7 0812 372 15$: SUBL3 R9,RECCNT,R0 ; rhb contents = rec # + 1?
03 13 0814 373 DECL R0
0590 30 0816 374 BEQL 20$ ; branch if yes
0549 30 0819 375 BSBW BADRHB ; report error
58 08 C0 081C 376 BSBW CHKRC1 ; check record len, rfa, and contents
03E8 8F 59 B1 081F 377 ADDL #8,R8 ; bump rfa table addr
A3 12 0824 378 CMPW R9,#1000 ; done?
0826 379 BNEQ T1BLOOP5 ; branch if not
F7C6' 30 0837 380 $CLOSE FAB=T1FAB,ERR=REPORT_ERROR
BSBW ERR

```

```

083A 382
083A 383
083A 384 : test 2 - random i/o test for sequential file org
083A 385
083A 386 : test 2a
083A 387
083A 388 : create a known test file of 1000 records of fixed length = 49. bytes,
083A 389 : 1st longword has the record # (n) followed by 45 bytes of the ascii
083A 390 : character (n mod 42) + 48.
083A 391
083A 392
083A 393
083A 394 T2SETUP: WTTYPE <START TEST 2A - RANDOM SEQ I/O>
0876 394
0876 395 CLRW RAB$W ISI(R11)
0879 396 MOVAL CPYBUF,RAB$R,RBF(R11)
0881 397 MOVZBW #49,RAB$W_RSZ(R11)
0885 398 MOVAL T1FAB,R10
088A 399 MOVZBL #48,FAB$R_ALQ(R10)
088E 400 BICL2 #FAB$M_NAM,FAB$R_FOP(R10)
0896 401 CLRB FAB$R_FAC(R10) ; check for put default
0899 402 MOVB #FAB$C_FIX,FAB$R_RFM(R10)
089D 403 MOVW #49,FAB$W_MRS(R10) ; rec len
08A1 404 $RAB STORE RAB=(R11),ROP=<LOC,UIF>
08A9 405 $CREATE FAB=R10,ERR=REPORT_ERROR
08B8 406 BSBW ERR
08BB 407 Cmpl #RMS$_SUPERSEDE,T1FAB+FAB$R_STS
08C4 408 BEQL SUPOK
08C6 409 FIELD <STATUS WORD IS NOT SUPERSEDE, THEREFORE IT>
08DB 410 SUPOK: $CONNECT RAB=R11,ERR=REPORT_ERROR
08EA 411 BSBW ERR
08ED 412
08ED 413 :
08ED 414 : pre-extend file on 1st pass, put sequentially on second pass
08ED 415 :
08ED 416
08ED 417 BBS #FAB$V_BLK,FAB$R_RAT(R10),10$; branch if pass 2
FAFF 03E8 8F 3C 08F2 418 MOVZWL #1000,RECCNT
1E AB 01 90 08F9 419 MOVB #RAB$C_KEY,RAB$R_RAC(R11)
F6F1' 30 090C 420 $PUT RAB=R1T,ERR=REPORT_ERROR
04 11 090F 421 BSBW ERR
1E AB 00 90 0911 422 BRB 20$
FADE CF 01 3C 0915 423 10$: MOVB #RAB$C_SEQ,RAB$R_RAC(R11)
091A 424 20$: MOVZWL #1,RECCNT
091A 425
091A 426 :
091A 427
57 52 FAD9 CF 2A 7B 091A 428 NXTRC2: EDIV #42,RECCNT,R2,R7 ; compute char for record
00000000'EF FAD0 CF D0 0921 429 ADDB2 #48,R7 ; make it ascii
2D 57 6E 00 2C 0924 430 MOVL RECCNT,CPYBUF ; insert rec #
00000004'EF 092D 431 MOVCS #0,(SP),R7,#45,CPYBUF+4 ; fill rec with char
0932
0937 432 $PUT RAB=R11,ERR=REPORT_ERROR
F6B7' 30 0946 433 BSBW ERR
05A1 30 0949 434 BSBW CHKRFA
094C 435
094C 436
094C 437 : print message every 100 records

```

```

52  FAA3 CF  00000064 8F  7B  094C  438 ;
      57      094C  439
      57      094C  440      EDIV  #100,RECCNT,R2,R7
      03      0956
      0537    05  0957  441      TSTL  R7
      12      0959  442      BNEQ  T2CNT
      30      095B  443      BSBW  TYPRFA
      095E  444
      095E  445 ;
FFB2 FA92 CF  01  03E8 8F  3D  095E  446 T2CNT:  ACBW  #1000,#1,RECCNT,NXTRC2
      F686'  30  095E  447      $CLOSE FAB=R10,ERR=REPORT_ERROR
      02 AB  B4  0968  448      BSBW  ERR
      0977  449      CLRW  RAB$W,ISI(R11)
      097A  450      WTTYPE <END STEP 2A - START STEP 2B>
      097D  451

```

```

09B9 453
09B9 454 :
09B9 455 : test 2b
09B9 456 :
09B9 457 : reread file created in step 2a and try random and sequential
09B9 458 : access via $get
09B9 459 :
09B9 460
04 AA 16 AA 09 90 09B9 461 MOVB #FAB$M_PUT!FAB$M_UPD,FAB$B_FAC(R10); upd implies get access
01000000 8F C8 09BD 462 BISL2 #FAB$M_NAM,FAB$L_FOP(R10)
F629' 30 09C5 463 $OPEN FAB=R10,ERR=REPORT_ERROR
09D4 464 BSBW ERR
09D7 465
09D7 466 :
09D7 467 : movl #rab$m_loc,rab$l_rop(r11)
09D7 468 :
09D7 469
1E AB 01 90 09D7 470 $RAB_STORE RAB=(R11),ROP=<LOC,UIF>
F9FC CF 03E8 8F 30 09DF 471 MOVB #RAB$C_KEY,RAB$B_RAC(R11)
F60B' 30 09E3 472 $CONNECT RAB=R11,ERR=REPORT_ERROR
09F5 473 BSBW ERR
09FC 474 MOVZWL #1000,RECCNT
09FC 475
09FC 476 :
09FC 477 : get all records in reverse order
09FC 478 :
09FC 479
FFF3 F9F1 CF FFFF 8F 054D 30 09FC 480 10$: BSBW GTCHK2
02 3D 09FF 481 ACBW #2,#-1,RECCNT,10$
0A09 482
0A09 483 :
0A09 484 : now get them all forward
0A09 485 :
0A09 486
FFF3 F9E4 CF 01 03E8 8F 0540 30 0A09 487 20$: BSBW GTCHK2
03E8 8F 3D 0A0C 488 ACBW #1000,#1,RECCNT,20$
0A16 489
0A16 490 :
0A16 491 : now get every 10th record in reverse order followed by the
0A16 492 : next 10 in sequential order
0A16 493 :
0A16 494
F9DB CF 03DE 8F B0 0A16 495 MOVW #990,RECCNT
052C 30 0A1D 496 30$: BSBW GTCHK2
1E AB 00 90 0A20 497 MOVB #RAB$C_SEQ,RAB$B_RAC(R11)
58 0A D0 0A24 498 MOVL #10,R8
F9CD CF D6 0A27 499 25$: INCL RECCNT
051E 30 0A2B 500 BSBW GTCHK2
F6 58 F5 0A2E 501 SOBGTR R8,25$
1E AB 01 90 0A31 502 MOVB #RAB$C_KEY,RAB$B_RAC(R11)
FFDE F9BB CF FFEC 8F 01 3D 0A35 503 ACBW #1,#-20,RECCNT,30$
0A3F 504
0A3F 505 :
0A3F 506
0A3F 507 WTTYE <PASS 1 O.K.>

```

```

0A7B 509
0A7B 510 ;
0A7B 511 ; do 10 random puts, changing record contents
0A7B 512 ;
0A7B 513 ;
          56 30 90 0A7B 514      MOVB    #^A/O/,R6          ; updating character
F974 CF 5B 8F 9A 0A7E 515      MOVZBL  #91,RECCNT
00000000'EF F970 CF D0 0A84 516 40$:  MOVL   RECCNT,CPYBUF
          2D 56 6E 00 2C 0A8D 517      MOVCS   #0,(SP),R6,#45,CPYBUF+4 ; change the record
          00000004'EF
          0A92
          0A97 518      $PUT   RAB=R11,ERR=REPORT_ERROR
          F557' 30 0AA6 519      BSBW   ERR
          0441 30 0AA9 520      BSBW   CHKRFA
CC F942 CF 0065 8F 03E8 8F 3D 0AAC 521      ACBW   #1000,#101,RECCNT,40$
          FF 0AB6
          04EF 30 0AB8 522      BSBW   CHKMOD          ; go verify changes
          0ABB 523
          0ABB 524 ;
          0ABB 525 ; do 10 random get/update pairs
          0ABB 526 ;
          0ABB 527 ;
          56 31 90 0ABB 528      MOVB    #^A/I/,R6          ; updating character
F934 CF 5B 8F 9A 0ABE 529      MOVZBL  #91,RECCNT          ; starting rec #
          F52A' 30 0AC4 530 50$:  $GET   RAB=R11,ERR=REPORT_ERROR
          58 00000004'EF DE 0AD3 531      BSBW   ERR
          59 2D D0 0AD6 532      MOVAL  CPYBUF+4,R8          ; get addr of record char
          88 46 D0 0ADD 533      MOVL   #45,R9
          FB 59 F5 0AEO 534 55$:  INCB   (R8)+          ; bump contents
          F509' 30 0AE2 535      SOBGTR R9,55$
          03F3 30 0AE5 536      $UPDATE RAB=R11,ERR=REPORT_ERROR
          04A1 30 0AF4 537      BSBW   ERR
          BE F8F4 CF 0065 8F 03E8 8F 3D 0AF7 538      BSBW   CHKRFA
          FF 0B04 539      ACBW   #1000,#101,RECCNT,50$
          04A1 30 0B06 540      BSBW   CHKMOD          ; go verify changes
          0B09 541
          0B09 542 ;
          0B09 543 ;
          0B09 544      WTTYPE  <PASS 2 O.K.>
          0B45 545      $CLOSE  FAB=R10,ERR=REPORT_ERROR
          3F 1E AA F4A9' 30 0B54 546      BSBW   ERR
          03 E2 0B57 547      BBSS   #FABS$V_BLK,FABS$B_RAT(R10),DONE
          FCDB 31 0B5C 548      WTTYPE  <DUPLICATE TEST WITH RECORDS NOT CROSSING BLOCK BOUNDARIES>
          0B98 549      BRW    T2SETUP
          0B98 550
          0B98 551 ;
          0B98 552 ;
          16 AA 01 90 0B98 553 DONE:  MOVB   #FABS$M_PUT,FABS$B_FAC(R10)
          1E AB 00 90 0B9F 554      MOVB   #RABS$C_SEQ,RABS$B_RAC(R11)
          00000004'EF 00000400 8F D0 0BA3 555      $WAIT  CMDORAB
          F7A4 CF 01000000 8F CA 0BB0 556      MOVL   #RABS$M_WBH,RABS$L_ROP+CMDORAB
          0BBB 557      BICL2  #FABS$M_NAM,T1FAB$FABS$L_FOP
          0BC4 558
          0BC4 559
          0BC4 560 ;
          0BC4 561 ;BUT FIRST -- TEST 3
          0BC4 562 ;

```

```

OBC4 563 ; test truncate
OBC4 564 ;
OBC4 565 ;
OBC4 566 $FAB_STORE FAB=R10,SHR=#0,FAC=<PUT,GET,TRN>; set up for trn
OBCB 567 $OPEN FAB=R10,ERR=REPORT_ERROR
F423' 30 OBDA 568 BSBW ERR
OBDD 569 $CONNECT RAB=R11,ERR=REPORT_ERROR
F411' 30 OBEC 570 BSBW ERR
OBEF 571 $GET RAB=R11,ERR=REPORT_ERROR; get 1st record, setting up cp
F3FF' 30 OBFE 572 BSBW ERR
OC01 573 $TRUNCATE RAB=R11,ERR=REPORT_ERROR; truncate all
F3ED' 30 OC10 574 BSBW ERR
OC13 575 $GET RAB=R11 ; should get eof
0001827A 8F 50 D1 OC1C 576 CMPL R0,#RMS$_EOF
OC23 577 BEQL TROK
5A 5B DO OC25 578 MOVL R11,R10
F3D5' 30 OC28 579 BSBW EOFPUT ; too bad
OC2B 580 TROK: TYPE <TRUNCATE SUCCEEDED!>
OC5A 581 $DISCONNECT RAB=R11,ERR=REPORT_ERROR; clean up
F394' 30 OC69 582 BSBW ERR
OC6C 583 $CLOSE FAB=R10,ERR=REPORT_ERROR
F382' 30 OC7B 584 BSBW ERR
OC7E 585 $FAB_STORE FAB=R10,SHR=<PUT,GET,UPI>,FAC=PUT
1F AA 03 90 OC87 586 MOVB #FAB$_VFC,FAB$_RFM(R10)
36 AA 64 8F 98 OC88 587 MOVZBW #100,FAB$_MRS(R10)
1E AA 02 90 OC90 588 MOVB #FAB$_CR,FAB$_RAT(R10)
10 AA 30 DO OC94 589 MOVL #48,FAB$_ALQ(R10)
3F AA 04 90 OC98 590 MOVB #4,FAB$_FSZ(R10)
28 AB 00000000'EF DE OC9C 591 MOVAL CPYBUF,RAB$_RBF(R11)
OCA4 592
OCA4 593 ;
OCA4 594 ;
OCA4 595 ;
OCA4 596 ;
OCA4 597
F348' 30 OCB5 598 $ERASE FAB=T1FAB,ERR=REPORT_ERROR
BSBW ERR
OCB8 599 TYPE <T1FILE.DAT HAS BEEN ERASED>
F678 CF 00100000 8F C8 OCE7 600 BISL2 #FAB$_CTG,FAB$_FOP+T1FAB; for 'f' test
OCF0 601 FINISH <SEQUENTIAL TESTS>
04 OD05 602 RET

```

```

        OD06 604
        OD06 605 ;
        OD06 606 ; subroutine to read in a record
        OD06 607 ;
        OD06 608
        OD06 609 GETANDCHK:
10 AB 68 06 28 OD06 610 MOV C3 #6,(R8),RAB$W_RFA(R11) ; rfa to rab
      15 57 E8 OD0B 611 BLBS R7,PASS2 ; branch if pass 2
      F2E0' 30 OD0E 612 $GET RAB=R11,ERR=REPORT_ERROR; get via rfa
      3C 10 OD1D 613 BSBW ERR
      05 OD20 614 BSBB CHKREC
      OD22 615 RSB
      OD23 616 PASS2:
1E AB 02 90 OD23 617 MOV B #RAB$C_RFA,RAB$B_RAC(R11)
      F2C7' 30 OD27 618 $FIND RAB=R1T,ERR=REPORT_ERROR
1E AB 00 90 OD36 619 BSBW ERR
      0A DD OD39 620 MOV B #RAB$C_SEQ,RAB$B_RAC(R11); switch back to sequential
      F2AF' 30 OD3D 621 PUSHL #10 ; loop count
      0B 10 OD3F 622 T1BLOOP3:
      59 D6 OD3F 623 $GET RAB=R11,ERR=REPORT_ERROR
      E7 6E F5 OD4E 624 BSBW ERR
      59 0A C2 OD51 625 BSBB CHKREC
      8E D5 OD53 626 INCL R9 ; bump record count
      05 OD55 627 SOGTR (SP),T1BLOOP3
      OD58 628 SUBL2 #10,R9 ; restore record count
      OD5B 629 TSTL (SP)+ ; clean up stack
      OD5D 630 RSB

```



```

OD5E 632
OD5E 633 ;
OD5E 634 ; subroutine to check that record read is really the right record
OD5E 635 ;
OD5E 636
OD5E 637 CHKREC:
56 52 59 F695 CF 59 D1 OD5E 638 CMPL R9,RECCNT ; fixed header = record #?
      22 AB 56 B1 OD63 639 BNEQ BADRHB
      55 52 F67F CF 0A 7B OD65 640 CHKRC1: EDIV #100,R9,R2,R6 ; compute record length
      10 AB 68 07 57 E8 OD6E 641 CMPW R6,RAB$W_RSZ(R11) ; = rms record len?
      01 05 OD72 642 BNEQ BADRSZ
      OD74 643 EDIV #10,RECCNT,R2,R5 ; compute character
      OD7B 644 ADDB #48,R5 ; make ascii
      OD7E 645 CMPC5 R6,@RAB$L_RBF(R11),R5,#0,(SP); match the record?
      OD85 646 BNEQ BADRBF
      OD87 647 BLBS R7,10$ ; branch if pass 2
      OD8A 648 CMPC3 #6,(R8),RAB$W_RFA(R11) ; rfa the same?
      OD8F 649 BNEQ BADRFA
      OD91 650 10$: RSB

```

```

OD92 652
OD92 653 :
OD92 654 : handle errors
OD92 655 :
OD92 656 :
46 11 ODA7 657 BADRFA. FIELD <RFA>
OD92 658 BRB ERROR
2F 11 ODA9 659 BADRHB: FIELD <RHB>
ODBE 660 BRB ERROR
18 11 ODC0 661 BADRSZ: FIELD <RSZ>
ODD5 662 BRB ERROR
01 11 ODD7 663 BADRBF: FIELD <RBF>
ODEC 664 BRB ERROR
ODEE 665
ODEE 666 :
ODEE 667
05 ODEE 668 DONT: RSB
ODEF 669 ERROR:
FB 00000000'EF E9 ODEF 670 BLBC VERBOSITY,DONT
00000004'EF 01 CA ODF6 671 $WAIT RAB=CMDORAB
OE03 672 BICL2 #RAB$M_ASY,RAB$L_RDP+CMDORAB
OE0A 673 $FAO S T2STR, CMDORAB+RAB$W_RSZ,FAOBUF,-
OE0A 674 R9,RECCNT,R6,RAB$W_RSZ(R11),-
OE0A 675 (R8),4(R8),RAB$W_RFA(R11),RAB$W_RFA+4(R11)
00000028'EF FF 50 E9 OE37 676 BLBC RO,.
00000000'EF 9E OE3A 677 MOVAB CMDBUF, CMDORAB+RAB$L_RBF
FF 50 F1A5' 30 OE58 678 $PUT RAB=CMDORAB,ERR=REPORT_ERROR
50 22 AB 3C OE5B 679 BSBW ERR
OE5F 680 MOVZWL RAB$W_RSZ(R11),RO
OE5F 681 $FAO S T3STR, CMDORAB+RAB$W_RSZ,FAOBUF,-
FF 50 OE5F 682 RO,RAB$L_RBF(R11)
OE7B 683 BLBC RO,.
F16C' 30 OE7E 684 $PUT RAB=CMDORAB,ERR=REPORT_ERROR
05 OE91 685 BSBW ERR
OE94 686 RSB
OE95 687
OE95 688 :
OE95 689 : output a message
OE95 690 :
50 00000000'EF E9 OE95 691
OE95 692 TYPRFA: BLBC VERBOSITY,NO
OE9C 693 $WAIT RAB=WTRAB ; wait on different rab for i/o to complete
OE A7 694 $FAO S T1STR, CMDORAB+RAB$W_RSZ,FAOBUF,RECCNT,RAB$W_RFA(R11),-
OE A7 695 RAB$W_RFA+4(R11)
00000028'EF FF 50 E9 OEC8 696 BLBC RO,.
00000000'EF 9E OECB 697 MOVAB CMDBUF, CMDORAB+RAB$L_RBF
F114' 30 OED6 698 $PUT RAB=CMDORAB,ERR=REPORT_ERROR
05 OEE9 699 BSBW ERR
OEEC 700 NO: RSB

```

```

      OEEED 702
      OEEED 703 :
      OEEED 704 : subroutines for test 2
      OEEED 705 :
      OEEED 706 : subroutine to check rfa correctness
      OEEED 707 :
      OEEED 708
50      F506 CF 01 C3 OEEED 709 CHKRFA: SUBLT #1,RECCNT,R0 ; get rec # - 1
      10 1E AA 03 E0 OEEED 710 BBS #FABS$V_BLK,FABS$B_RAT(R10),10$
51      00 32 50 7A OEEED 711 EMUL R0,#50,#0,R1 ; get byte addr
52 51 51 00000200 8F 7B OEEED 712 EDIV #512,R1,R1,R2 ; compute vbn-1 & offset
      OA 11 OF06 713 BRB 20$
      OF08 714
      OF08 715 :
      OF08 716 : alternate calc for records not crossing blk boundaries
      OF08 717 :
      OF08 718
52      51 50 51 D4 OF08 719 10$: CLRL R1 ; zero extend record #
      0A 7B OF0A 720 EDIV #10,R0,R1,R2 ; compute vbn-1 & rec # in blk
      32 A4 OF0F 721 MULW2 #50,R2 ; compute offset
      51 D6 OF12 722 20$: INCL R1 ; vbn
      10 AB 51 D1 OF14 723 CMPL R1,RABS$W_RFA(R11)
      07 12 OF18 724 BNEQ ERRRFA
      14 AB 52 B1 OF1A 725 CMPW R2,RABS$W_RFA+4(R11)
      01 12 OF1E 726 BNEQ ERRRFA
      05 OF20 727 RSB
      OF21 728 ERRRFA: WFIELD <RFA>
      OF43 729 MBPT
      05 OF4B 730 RSB

```

```

OF4C 732
OF4C 733 :
OF4C 734 : subroutine to get and check a record for test 2
OF4C 735 :
OF4C 736 :
57 50 F4A7 CF 2A 7B OF4C 737 GTCHK2: EDIV #42,RECCNT,R0,R7 ; compute char
      57 30 80 OF53 738 ADDB #48,R7
      F098' 30 OF56 739 GTCHK2A: $GET RAB=R11,ERR=REPORT_ERROR
      FF82 30 OF65 741 BSBW ERR
      52 28 AB D0 OF68 742 BSBW CHKRFA
      F484 CF 82 D1 OF6B 743 MOVL RAB$RBF(R11),R2 ; get rec addr
      09 12 OF6F 744 CMPL (R2)+,RECCNT ; recnt o.k.?
6E 00 57 62 2D 2D OF74 745 BNEQ ERRREC
      01 12 OF76 746 CMPC5 #45,(R2),R7,#0,(SP) ; match?
      05 05 OF77 747 BNEQ ERRREC
      OF7E 748 RSB
      OF7F 749 ERRREC: WFIELD <RECORD CONTENTS>
      OFA1 750 MBPT
      05 OFA9 751 RSB
      OFAA 752 :
      OFAA 753 :
      OFAA 754 : subroutine to verify that updated records were changed but not
      OFAA 755 : the preceding or following records
      OFAA 756 :
      OFAA 757 :
      OFAA 758 CHKMOD:
      F448 CF 5A 8F 9A OFAA 759 MOVZBL #90,RECCNT ; starting rec #
      FF99 30 OFB0 760 CHKNXT: BSBW GTCHK2
      F441 CF D6 OFB3 761 INCL RECCNT
      57 56 D0 OFB7 762 MOVL R6,R7 ; check character
      FF99 30 OFBA 763 BSBW GTCHK2A
      F437 CF D6 OFBD 764 INCL RECCNT
      F433 CF B1 OFC1 765 CMPW RECCNT,#1001 ; all done?
      OC 13 OFC8 766 BEQL 10$
      FF7F 30 OFCA 767 BSBW GTCHK2
      F424 CF 0063 8F A0 OFCD 768 ADDW #99,RECCNT
      DA 11 OFD4 769 BRB CHKNXT
      0001827A 8F 50 D1 OFD6 770 10$: $GET R11
      01 12 OFDF 771 CMPL R0,#RMS$_EOF
      05 05 OFE6 772 BNEQ 20$
      OFE8 773 RSB
      OFE9 774 20$:
      OFE9 775 $WAIT RAB=CMDORAB
      SA DD OFF6 776 PUSHL R10 ; save it
      SA 5B D0 OFF8 777 MOVL R11,R10 ; bad structure!!!
      F002' 30 OFFB 778 BSBW EOFPUT
      0400 8F BA OFFE 779 POPR #^M<R10>
      1002 780 MBPT
      05 100A 781 RSB
      100B 782
      100B 783 .END

```

\$\$PSECT_EP	=	00000000				FABSM_CTG	=	00100000	D		
\$\$TAB	=	000003B4	R	D	01	FABSM_GET	=	00000002	D		
\$\$TABEND	=	000003F8	R	D	01	FABSM_NAM	=	01000000	D		
\$\$TMP	=	00000043		D		FABSM_PUT	=	00000001	D		
\$\$TMP1	=	00000001		D		FABSM_UPD	=	00000008	D		
\$\$TMP2	=	000000EF		D		FABSV_BLK	=	00000003	D		
\$\$TMPX	=	00000187	R	D	04	FABSV_CHAN_MODE	=	00000002	D		
\$\$TMPX1	=	0000000F		D		FABSV_CR	=	00000001	D		
\$\$RMSTEST	=	0000001E		D		FABSV_CTG	=	00000014	D		
\$\$RMS_PBUGCHK	=	00000010		D		FABSV_FILE_MODE	=	00000004	D		
\$\$RMS_TBUGCHK	=	00000008		D		FABSV_GET	=	00000001	D		
\$\$RMS_UMODE	=	00000004		D		FABSV_LNM_MODE	=	00000000	D		
\$\$T2	=	00000006		D		FABSV_PUT	=	00000000	D		
..AFLG	=	00000000		D		FABSV_SUP	=	00000002	D		
..FLG	=	00000002		D		FABSV_TRN	=	00000004	D		
..MOD	=	00000001		D		FABSV_UPI	=	00000006	D		
..N	=	00000003		D		FABSW_GBC	=	00000048	D		
..TYP	=	00000003		D		FABSW_MRS	=	00000036	D		
.LEN	=	00000001		D		FAOBUF	*****		X		01
BADRFB	=	00000DD7	R	D	01	FINPUT	*****		X		01
BADRFA	=	00000D92	R	D	01	FIN_DESCR	*****		X		01
BADRHB	=	00000DA9	R	D	01	FLDPUT	*****		X		01
BADRSZ	=	00000DC0	R	D	01	FLD_DESCR	*****		X		01
BEGPUT	*****			X	01	GETANDCHK	*****		X		01
BEG_DESCR	*****			X	01	GTCHK2	00000D06	R	D		01
CHKMOD	=	00000FAA	R	D	01	GTCHK2A	00000F4C	R	R	D	01
CHKNXT	=	00000FB0	R	D	01	NAMBLK	00000F56	R	R	D	01
CHKRC1	=	00000D65	R	D	01	NO	*****		X		01
CHKREC	=	00000D5E	R	D	01	NXTRC2	00000EEC	R	R	D	01
CHKRFA	=	00000EED	R	D	01	NXTREC	0000091A	R	R	D	01
CMDBUF	*****			X	01	PASS2	00000511	R	R	D	01
CMORAB	*****			X	01	RABSB_RAC	=	00000D23	R	D	01
CPYBSZ	*****			X	01	RABSC_BID	=	0000001E	D		
CPYBUF	*****			X	01	RABSC_BLN	=	00000001	D		
DONE	=	00000B9B	R	D	01	RABSC_KEY	=	00000044	D		
DONT	=	00000DEE	R	D	01	RABSC_RF	=	00000001	D		
EOFPUT	*****			X	01	RABSC_SEQ	=	00000002	D		
ERR	*****			X	01	RABSC_SEQ	=	00000000	D		
ERROR	=	00000DEF	R	D	01	RABSL_CTX	=	00000000	D		
ERRREC	=	00000F7F	R	D	01	RABSL_RBF	=	00000018	D		
ERRRFA	=	00000F21	R	D	01	RABSL_RHB	=	00000028	D		
FABSB_FAC	=	00000016		D		RABSL_RHP	=	0000002C	D		
FABSB_FNS	=	00000034		D		RABSL_ROP	=	00000004	D		
FABSB_FSZ	=	0000003F		D		RABSM_ASY	=	00000001	D		
FABSB_RAT	=	0000001E		D		RABSM_LOC	=	00010000	D		
FABSB_RFM	=	0000001F		D		RABSM_WBH	=	00000400	D		
FABSB_SHR	=	00000017		D		RABSV_LOC	=	00000010	D		
FABSC_BID	=	00000003		D		RABSV_UIF	=	00000004	D		
FABSC_BLN	=	00000050		D		RABSV_WBH	=	0000000A	D		
FABSC_FIX	=	00000001		D		RABSW_ISI	=	00000002	D		
FABSC_SEQ	=	00000000		D		RABSW_RFA	=	00000010	D		
FABSC_VFC	=	00000003		D		RABSW_RSZ	=	00000022	D		
FABSL_ALQ	=	00000010		D		RECCNT	000003F8	RG	D		01
FABSL_FNA	=	0000002C		D		REPORT_ERROR	*****		X		01
FABSL_FOP	=	00000004		D		RFATBL	00000044	R	D		01
FABSL_STS	=	00000008		D		RHBSW	000004AB	R	D		01
FABSM_CR	=	00000002		D		RMSS_EOF	=	0001827A	D		
						RMSS_SUPERSEDE	=	00010631	D		
						RMTSTEST_1A	000004AC	RG	D		01



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

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	38	00:00:00.07	00:00:00.55
Command processing	133	00:00:00.54	00:00:02.18
Pass 1	376	00:00:17.01	00:00:35.60
Symbol table sort	0	00:00:00.74	00:00:01.42
Pass 2	138	00:00:03.96	00:00:06.21
Symbol table output	18	00:00:00.13	00:00:00.17
Psect synopsis c .put	2	00:00:00.03	00:00:00.12
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	707	00:00:22.49	00:00:46.25

The working set limit was 1650 pages.  
86298 bytes (169 pages) of virtual memory were used to buffer the intermediate code.  
There were 30 pages of symbol table space allocated to hold 618 non-local and 34 local symbols.  
783 source lines were read in Pass 1, producing 58 object records in Pass 2.  
69 pages of virtual memory were used to define 50 macros.

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

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	36
TOTALS (all libraries)	36

1074 GETS were required to define 36 macros.

There were no errors, warnings or information messages.

MACRO/LIS=\_ISS:RMSTEST1/OBJ=OBJ\$:RMSTEST1 MSRC\$:RMSTEST1/UPDATE=(ENH\$:RMSTEST1)+EXECMLS/LIB

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

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

