

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	UUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPP	

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


```

0000 64          $BEGIN RMSTEST2,004,__,RMSTEST,<BLOCK I/O TEST PROGRAM>,<GBL,LONG>
0000 65
0000 66 :
0000 67
0000 68          .ENABL  DBG
0000 69
0000 70 :
0000 71 : this program provides several tests of the block i/o facility
0000 72 : including mixed block and record operations
0000 73 :
0000 74 : currently tests sequential file org only.
0000 75 :
0000 76 :
0000 77
0000 78          $RMSDEF
0000 79          .NLIST  MEB
0000 80
0000 81 :
0000 82 : macros:
0000 83 :
0000 84
0000 85          .MACRO  BUFF  NAM,SIZE
0000 86 NAM'BUF ::
0000 87          .BLKB  SIZE
0000 88          NAM'BSZ==SIZE
0000 89          .ENDM  BUFF
0000 90
0000 91 :
0000 92 :
0000 93 :
0000 94
0000 95          .MACRO  TYPE  STRING, ?L
0000 96 STORE  <STRING>
0000 97 BLBC   VERBOSITY,L
0000 98 MOVL   $$$TMPX,CMDORAB+RAB$L,RBF
0000 99 MOVW   $$$TMPX1,CMDORAB+RAB$W,RSZ
0000 100 $PUT   RAB=CMDORAB,ERR=REPORT_ERROR
0000 101 BSBW   ERR
0000 102 L:
0000 103          .ENDM  TYPE
0000 104
0000 105 :
0000 106
0000 107          .MACRO  STORE  STRING,PRE
0000 108 .SAVE
0000 109 .PSECT  __$RMSNAM
0000 110 $$$TMPX=--
0000 111 PRE                                     ; store any carriage control info
0000 112 .ASCII  %STRING%
0000 113 $$$TMPX1=-.-$$.TMPX
0000 114 .RESTORE
0000 115          .ENDM  STORE
0000 116
0000 117 :
0000 118

```

```
0000 120 .MACRO BEGIN TSTNAM
0000 121 STORE <TSTNAM>
0000 122 MOVL #$$ .TMPX ,BEG_DESCR+4 ; addr
0000 123 MOVL #$$ .TMPX1 ,BEG_DESCR ; len
0000 124 BSBW BEGPOT
0000 125 .ENDM BEGIN
0000 126 .MACRO FINISH TSTNAM
0000 127 STORE <TSTNAM>
0000 128 MOVL #$$ .TMPX ,FIN_DESCR+4 ; addr
0000 129 MOVL #$$ .TMPX1 ,FIN_DESCR ; len
0000 130 BSBW FINPUT
0000 131 .ENDM FINISH
0000 132 .MACRO FIELD FLDNAM
0000 133 STORE <FLDNAM>
0000 134 MOVL #$$ .TMPX ,FLD_DESCR+4 ; addr
0000 135 MOVL #$$ .TMPX1 ,FLD_DESCR ; len
0000 136 BSBW FLDPUT
0000 137 .ENDM FIELD
0000 138 .MACRO MBPT ,?L
0000 139 BLBC VERBOSITY ,L
0000 140 BPT
0000 141 L:
0000 142 .ENDM MBPT
0000 143
0000 144 ;
0000 145
```

```

0000 147
0000 148 ;
0000 149
00000000 150 .PSECT  __RMSTEST,GBL,LONG
0000 151 T2START::
0000 152 .ALIGN  LONG
0000 153 T2FAB:: $FAB  FAC=<BIO,PUT>,FNM=<TST$DISK:T2FILE.DA1 1>,org=seq,rfm=fix,-
0000 154 RAT=BLK,MRS=128,NAM=NAMBLK,FOP=<SUP>,ALQ=10,DEQ=10
0050 155 T2RAB:: $RAB  FAB=T2FAB,UBF=CPYBUF,USZ=CPYBSZ,RBF=CPYBUF,MBC=4,MBF=2,-
0050 156 RSZ=512,ROP=<WBH>
0000009C 0094 157 RFASAV: .BLKW  4
009C 158
009C 159 :
009C 160 :
009C 161
009C 162 TST$BLOCKIO::
OFFC 009C 163 .WORD  ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
009E 164 BEGIN  <BLOCK I/O TESTS>
00B3 165 $CREATE FAB=T2FAB,ERR=REPORT_ERROR
5B  FF39' 30 00C4 166 BSBW  ERR
86 AF  DE 00C7 167 MOVAL  T2RAB,R11
00CB 168 $CONNECT RAB=R11,ERR=REPORT_ERROR
FF23' 30 00DA 169 BSBW  ERR
58 D4 00DD 170 CLRL  R8 ; vbn count
57 D4 0CDF 171 CLRL  R7
00E1 172
00E1 173 :
00E1 174 : fill block buffer
00E1 175 :
00E1 176
53 28 BB 9E 00E1 177 FILBUF: MOVAB @RAB$_RBF(R11),R3 ; get buff addr
57 57 D6 00E5 178 10$: INCL R7 ; next rec
63 0080 8F 57 6E 00 2C 00E7 179 MOVCS #0,(SP),R7,#128,(R3) ; build rec
57 03 D3 00EF 180 BITL #3,R7 ; 4th rec
F1 12 00F2 181 BNEQ 10$ ; branch if not
00F4 182 $WRITE RAB=R11,ERR=REPORT_ERROR
FEFA' 30 0103 183 BSBW  ERR
58 D6 0106 184 INCL R8 ; bump vbn
10 AB 58 D1 0108 185 CMPL R8,RAB$_RFA(R11) ; match?
1D 13 010C 186 BEQL RFAOK
010E 187 FIELD <RFA>
0123 188 MBPT
3F 58 D1 0128 189 RFAOK:
B1 1B 012E 190 CMPL R8,#63 ; end?
0130 191 BLEQU FILBUF ; not yet
0130 192
0130 193 :
0130 194
0130 195 $CLOSE FAB=T2FAB,ERR=REPORT_ERROR
FEBC' 30 0141 196 BSBW  ERR
02 AB B4 0144 197 CLRW RAB$_ISI(R11) ; allow rab reuse
00 11 0147 198 TYPE <FINISHED TEST 1 - BEGINNING TEST 2>
0176 199 BRB TEST2
0178 200
0178 201 :
0178 202

```

```

0178 204
C178 205
0178 206 : re-read file created in test 1 and try random and sequential gets & reads
0178 207 :
0178 208 :
FE83 CF 01000000 8F C8 0178 209 TEST2: BISL2 #FABSM_NAM,T2FAB+FAB$L_FOP
FE8F CF 43 8F 90 0181 210 MOVB #FABSM_BRO!FABSM_GET!FABSM_PUT,T2FAB+FAB$B_FAC
FE74 CF 00300000 8F CA 0187 211 BICL #FABSM_CBT!FABSM_CTG,FAB$L_FOP+T2FAB; for ctg/cbt test
15 FE5B CF 14 E1 01A1 212 $OPEN FAB=T2FAB,ERR=REPORT_ERROR
15 FE5B CF 14 E1 01A4 213 BSBW ERR
15 FE40 CF 15 E1 01AA 214 BBC #FABSV_CTG,FAB$L_FOP+T2FAB,CTGOK
01BF 215 FIELD <CTG BIT SET, THEREFORE>
01BF 216 CTGOK: BBC #FABSV_CBT,FAB$L_FOP+T2FAB,BACK
01C5 217 FIELD <CBT BIT SET, THEREFORE>
01DA 218 BACK:
01DA 219
01DA 220
01DA 221 :
01DA 222 : movw #128,rab$w_rsz(r11) ;\needed?\
01DA 223 :
01DA 224 :
04 AB 00000200 8F D0 01DA 225 MOVL #RABSM_RAH,RAB$L_ROP(R11)
FE0C' 30 01E2 226 $CONNECT RAB=R11,ERR=REPORT_ERROR
01F1 227 BSBW ERR
01F4 228
01F4 229 :
01F4 230 : try doing gets on all records sequentially
01F4 231 :
01F4 232 :
57 D4 01F4 233 GETREC: CLRL R7 ; record count
01F6 234 $GET R11
01FF 235 BLBS R0,10$
0001827A 8F 12 50 E8 0202 236 CMPL R0,#RMS$_EOF
5A 5B D0 0209 237 BEQL T2EOF
FDEF' 30 020B 238 MOVL R11,R10
FDEC' 30 020E 239 BSBW REPORT_ERR
57 D6 0211 240 BSBW ERR
6E 00 57 28 BB 0080 8F 2D 0214 241 10$: INCL R7 ; bump rec count
15 13 0216 242 CMPCS #128,@RAB$L_RBF(R11),R7,#0,(SP); data ok?
021F 243 BEQL RECOK
0221 244 FIELD <RECORD CONTENTS>
0080 8F 22 AB B1 0236 245 RECOK: CMPW RAB$W_RSZ(R11),#128
1D 13 023C 246 BEQL RSZOK
023E 247 FIELD <RECORD SIZE>
0253 248 MBPT
99 11 025B 249 RSZOK: BRB GETREC
025D 250
025D 251
025D 252 :
025D 253 :
00000100 8F 57 D1 025D 254 T2EOF: CMPL R7,#256
0E 13 0264 255 BEQL TEST2A
5A 5B D0 0266 256 MOVL R11,R10
FD94' 30 0269 257 BSBW EOFPUT
026C 258 ; replaced type <bad eof>
0274 259 MBPT
260

```

RMSTEST2
V04-000

BLOCK I/O TEST PROGRAM ;

N 6

16-SEP-1984 01:46:27 VAX/VMS Macro V04-00
5-SEP-1984 04:21:43 [UETP.SRC]RMSTEST2.MAR;1

Page 5
(6)

RMS
000

0274 261 ;
0274 262

```

0274 264
0274 265 :
0274 266 : now try sequential reads
0274 267 :
0274 268
0274 269 TEST2A:
0274 270 TYPE <FINISHED TEST 2 - BEGINNING TEST 2A>
38 AB 01 D0 02A3 271 MOVL #1,RAB$BKT(R11)
58 D4 02A7 272 CLRL R8 ; vbn
0001827A 8F 0C 50 E8 02A9 273 READVB: $READ R11
50 D1 02B2 274 BLBS R0,10$
OD 13 02B5 275 CMPL R0,#RMS$_EOF
FD3F' 30 02BC 276 BEQL T2AEOF
58 D6 02BE 277 BSBW ERR
02C0 30 02C1 278 10$: INCL R8
38 AB D4 02C3 279 BSBW CHKBLK ; verify contents
DE 11 02C6 280 CLRL RAB$BKT(R11) ; get sequential operation
02CB 281 BRB READVB
02CB 282
02CB 283 ;
00000040 8F 58 D1 02CB 284 T2AEOF: CMPL R8,#64
OE 13 02D2 286 BEQL TEST2B
5A 5B D0 02D4 287 MOVL R11,R10
FD26' 30 02D7 288 BSBW EOFPUT ; replaced type <bad eof>
02DA 289 MBPT

```



```

02E2 291
02E2 292 :
02E2 293 : now try mixed reads, writes, gets, and puts
02E2 294 :
02E2 295 :
02E2 296 TEST2B:
02E2 297 TYPE <FINISHED TEST 2A - BEGINNING TEST 2B>
0311 298
0311 299 :
0311 300 : multiple block test
0311 301 :
0311 302 :
20 AB 0204 8F B0 0311 303 MOVW #516,RAB$W USZ(R11)
38 AB 09 D0 0317 304 MOVL #9,RAB$L BKT(R11)
FC D3' 30 031B 305 $READ RAB=R11,ERR=REPORT_ERROR
58 09 D0 032A 306 BSBW ERR
0253 30 032D 307 MOVL #9,R8
38 AB D4 0330 308 BSBW CHKBLK
FC B8' 30 0333 309 CLRL RAB$L BKT(R11)
58 0B D0 0336 310 $READ RAB=R11,ERR=REPORT_ERROR
0238 30 0345 311 BSBW ERR
2D2D2D2D 8F 00000200'EF D1 0348 312 MOVL #11,R8
1D 13 034B 313 BSBW CHKBLK
034E 314 CMLL CPYBUF+512,#^X2D2D2D2D
0359 315 BEQL TEST2B2
035B 316 FIELD <DATA>
0370 317 MBPT
0378 318
0378 319 :
0378 320 : test short block write and auto extend of file
0378 321 :
0378 322 :
0378 323 TEST2B2:
38 AB 00000000'EF D4 0378 324 CLRL CPYBUF
22 AB 04 B0 037E 325 MOVW #4,RAB$W RSZ(R11)
00000050 8F D0 0382 326 MOVL #80,RAB$L BKT(R11) ; cause file extend
FC64' 30 038A 327 $WRITE RAB=R11,ERR=REPORT_ERROR
22 AB B4 0399 328 BSBW ERR
20 AB 0200 8F B0 039C 329 CLRL RAB$W RSZ(R11)
FC49' 30 039F 330 MOVW #512,RAB$W USZ(R11)
03A5 331 $READ RAB=R11,ERR=REPORT_ERROR
03B4 332 BSBW ERR
03B7 333 :
03B7 334 :
03B7 335 : SINCE BKT IS STILL 80, READ THAT BLK
03B7 336 : USED TO EXPECT EOF ERROR, NO LONGER
03B7 337 :
04 22 AB B1 03B7 338
1D 13 03B7 339 CMPW RAB$W RSZ(R11),#4
03BB 340 BEQL RSZOKT
03BD 341 FIELD <RECORD SIZE>
03D2 342 MBPT
03DA 343 RSZOK1:
00000000'EF D5 03DA 344 TSTL CPYBUF ; test first four bytes of block
1D 13 03E0 345 BEQL TEST2B3 ; (since that's all we wrote)
03E0 346 BEQL TEST2B3
03E2 347 FIELD <DATA>

```

				03F7	348	MBPT	
				03FF	349	TEST2B3:	
	1E AB	02	90	03FF	350	MOVB	#RAB\$C RFA,RAB\$B RAC(R11); get rfa access
10	AB	00000050	8F	D0	0403	MOVL	#80,RAB\$W RFA(R1T) ; give the rfa
		14	AB	B4	040B	CLRW	RAB\$W RFA+4(R11)
	00000000	'EF	01	D0	040E	MOVL	#1,CPYBUF
					0415	\$GET	RAB=R11,ERR=REPORT_ERROR
		FBD9'		30	0424	BSBW	ERR
00	00000000	'EF	0080	8F	2D	CMPCS	#128,CPYBUF,#0,#0,(SP)
		6E	00		0431		
		1D		13	0433	BEQL	TEST2B4
					0435	FIELD	<DATA>
					044A	MBPT	

```

00000000'EF      1E AB  00  90 0452 361
                  000012AF 8F  D0 0452 362
                  FB8D'  30 0452 363 : check that we can put to eof
FC1A CF      10 AB  06  28 0452 364 :
                  38 AB  01  D0 0452 365
                  FB70'  30 0452 366 TEST2B4:
                  58  01  D0 0452 367   MOVB   #RAB$C_SEQ,RAB$B_RAC(R11); sequential mode
                  00F0  30 0456 368   MOVL   #4783,CPYBUF; funny data
10 AB      1E AB  02  90 0461 369   $PUT   RAB=R11,ERR=REPORT_ERROR
                  FBF5 CF  06  28 0470 370   BSBW   ERR
                  FB4D'  30 0473 371   MOV3   #6,RAB$W_RFA(R11),RFASAV
000012AF 8F      00000000'EF  D1 047A 372   MOVL   #1,RAB$L_BKT(R11); read blk 1 in between
                  1D  13 047E 373   $READ  RAB=R11,ERR=REPORT_ERROR
                  FB4D'  30 048D 374   BSBW   ERR
                  00000000'EF  D1 0490 375   MOVL   #1,R8
                  1D  13 0493 376   BSBW   CHKBLK
                  00000000'EF  D1 0496 377   MOV3   #RAB$C_RFA,RAB$B_RAC(R11)
                  1D  13 049A 378   MOV3   #6,RFASAV,RAB$W_RFA(R11)
                  00000000'EF  D1 04A1 379   $GET   RAB=R11,ERR=REPORT_ERROR
                  1D  13 04B0 380   BSBW   ERR
                  00000000'EF  D1 04B3 381   CMPL   CPYBUF,#4783; funny data back?
                  1D  13 04BE 382   BEQL   TESTEND
                  00000000'EF  D1 04C0 383   FIELD  <DATA>
                  1D  13 04D5 384   MBPT
                  00000000'EF  D1 04DD 385
                  1D  13 04DD 386 :
                  00000000'EF  D1 04DD 387
                  1D  13 04DD 388 TESTEND:
                  00000000'EF  D1 04DD 389   $CLOSE  FAB=T2FAB,ERR=REPORT_ERROR
                  00000000'EF  D1 04EE 390   BSBW   ERR
                  00000000'EF  D1 04F1 391   FINISH <BLOCK I/O TESTS>
                  00000000'EF  D1 0506 392   CLRW   RAB$W_ISI(R11); make test restartable
                  00000000'EF  D1 0509 393   MOVL   #FAB$M_SUP,T2FAB+FAB$L_FOP
                  00000000'EF  D1 050E 394   MOV3   #FAB$M_PUT,FAB$M_BIO,T2FAB+FAB$B_FAC
04 AB      00000400 8F  D0 0513 395   MOVL   #RAB$M_WBH,RAB$L_ROP(R11)
22 AB      0200 8F  B0 051B 396   MOVW   #512,RAB$W_RSZ(R11)
                  38 AB  D4 0521 397   CLRL   RAB$L_BKT(R11)
                  1E AB  00  90 0524 398   MOV3   #RAB$C_SEQ,RAB$B_RAC(R11)
                  00000000'EF  D1 0528 399
                  00000000'EF  D1 0528 400 :
                  00000000'EF  D1 0528 401 : but first erase the file by namblk
                  00000000'EF  D1 0528 402 :
                  00000000'EF  D1 0528 403
FAD3 CF      01000000 8F  C8 0528 404   BISL2  #FAB$M_NAM,FAB$L_FOP+T2FAB
                  FAF7 CF  D4 0531 405   CLRL   FAB$L_FNA+T2FAB
                  FAFB CF  94 0535 406   CLRB   FAB$B_FNS+T2FAB
                  FAB3'  30 0539 407   $ERASE FAB=T2FAB,ERR=REPORT_ERROR
                  01000000 8F  CA 054A 408   BSBW   ERR
                  01000000 8F  CA 054D 409   TYPE  <T2FILE.DAT HAS BEEN ERASED BY NAMBLK>
FA7F CF      01000000 8F  CA 057C 410   BICL   #FAB$M_NAM,FAB$L_FOP+T2FAB; for re-entry
                  04  0585 411   RET

```

```

0586 413
0586 414 :
0586 415 :   chkblk subroutine to verify we read the right block
0586 416 :
0586 417 :
10 AB 58 D1 0586 418 CHKBLK: CMPL R8,RABSW_RFA(R11)
1D 13 058A 419 BEQL CHKB1
058C 420 FIELD <RFA>
05A1 421 MBPT
20 AB 22 AB B1 05A9 422 CHKB1: CMPW RABSW_RSZ(R11),RABSW_USZ(R11)
1D 13 05AE 423 BEQL CHKB2
05B0 424 FIELD <RSZ>
05C5 425 MBPT
56 58 02 9C 05CD 426 CHKB2: ROTL #2,R8,R6
56 03 C2 05D1 427 SUBL2 #3,R6
6E 00 56 51 28 BB 9E 05D4 428 MOVAB @RABSL,RBF(R11),R1 ; get starting rec #
61 0080 8F 2D 05D8 429 RIGHT: CMPCS #128,(R1),R6,#0,(SP) ; right data?
1D 13 05E0 430 BEQL AOK
05E2 431 FIELD <DATA>
05F7 432 MBPT
01 56 02 56 D6 05FF 433 AOK: INCL R6 ; next record
00 ED 0601 434 CMPZV #0,#2,R6,#1 ; done with block?
D0 12 0606 435 BNEQ RIGHT
05 0608 436 RSB
0609 437
0609 438 :
0609 439
0609 440 .END

```

```

$$PSECT_EP      = 00000000
$$TAB           = 00000050 R D 01
$$TABEND       = 00000094 R D 01
$$TMP          = 00000400
$$TMP1         = 00000002
$$TMP2         = 000000EF
$$TMPX         = 00000115 R D 04
$$TMPX1        = 00000004
$$RMSTEST      = 0000001E
$$RMS_PBUGCHK  = 00000010
$$RMS_TBUGCHK  = 00000008
$$RMS_UMODE    = 00000004
AOK            = 00G005FF R D 01
BACK          = 000001DA R D 01
BEGPUT        = ***** X 01
BEG_DESCR     = ***** X 01
CHKB1         = 000005A9 R D 01
CHKB2         = 000005CD R D 01
CHKBLK        = 00000586 R D 01
CMDORAB       = ***** X 01
CPYBSZ        = ***** X 01
CPYBUF        = ***** X 01
CTGOK         = 000001BF R D 01
EOFPUT        = ***** X 01
ERR           = ***** X 01
FABS$B_FAC    = 00000016
FABS$B_FNS    = 00000034
FABS$C_BID    = 00000003
FABS$C_BLN    = 00000050
FABS$C_FIX    = 00000001
FABS$C_SEQ    = 00000000
FABS$L_ALQ    = 00000010
FABS$L_FNA    = 0000002C
FABS$L_FOP    = 00000004
FABS$M_BIO    = 00000020
FABS$M_BRO    = 00000040
FABS$M_CBT    = 00200000
FABS$M_CTG    = 00100000
FABS$M_GET    = 00000002
FABS$M_NAM    = 01000000
FABS$M_PUT    = 00000001
FABS$M_SUP    = 00000004
FABS$V_BIO    = 00000005
FABS$V_BLK    = 00000003
FABS$V_CBT    = 00000015
FABS$V_CHAN_MODE = 00000002
FABS$V_CTG    = 00000014
FABS$V_FILE_MODE = 00000004
FABS$V_LNM_MODE = 00000000
FABS$V_PUT    = 00000000
FABS$V_SUP    = 00000002
FABS$W_GBC    = 00000048
FILBUF        = 000000E1 R D 01
FINPUT        = ***** X 01
FIN_DESCR     = ***** X 01
FLDPUT        = ***** X 01
FLD_DESCR     = ***** X 01

```

```

GETREC
NAMBLK
RABS$B_RAC    = 0000001E
RABS$C_BID    = 00000001
RABS$C_BLN    = 00000044
RABS$C_RFA    = 00000002
RABS$C_SEQ    = 00000000
RABS$L_BKT    = 00000038
RABS$L_CTX    = 00000018
RABS$L_RBF    = 00000028
RABS$L_ROP    = 00000004
RABS$M_RAH    = 00000200
RABS$M_WBH    = 00000400
RABS$V_WBH    = 0000000A
RABS$W_ISI    = 00000002
RABS$W_RFA    = 00000010
RABS$W_RSZ    = 00000022
RABS$W_USZ    = 00000020
READVB
RECOK
REPORT_ERR
REPORT_ERROR
RFAOK
RFASAV
RIGHT
RMSS_EOF
RSZOR
RSZOK1
SYSS$CLOSE
SYSS$CONNECT
SYSS$CREATE
SYSS$ERASE
SYSS$GET
SYSS$OPEN
SYSS$PUT
SYSS$READ
SYSS$WRITE
T2AEOF
T2EOF
T2FAB
T2RAB
T2START
TEST2
TEST2A
TEST2B
TEST2B2
TEST2B3
TEST2B4
TESTEND
TST$BLOCKIO
VERBOSITY

```

```

000001F6 R D 01
***** X 01
= 0000001E D
= 00000001 D
= 00000044 D
= 00000002 D
= 00000000 D
= 00000038 D
= 00000018 D
= 00000028 D
= 00000004 D
= 00000200 D
= 00000400 D
= 0000000A D
= 00000002 D
= 00000010 D
= 00000022 D
= 00000020 D
000002A9 R D 01
00000236 R D 01
***** X 01
***** X 01
0000012B R D 01
00000094 R D 01
000005D8 R D 01
= 0001827A D
0000025B R D 01
000003DA R D 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
***** GX 01
000002CB R D 01
0000025D R D 01
00000000 RG D 01
00000050 RG D 01
00000000 RG D 01
00000178 R D 01
00000274 R D 01
000002E2 R D 01
00000378 R D 01
000003FF R D 01
00000452 R D 01
000004DD R D 01
0000009C RG D 01
***** X 01

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMSTEST	00000609 (1545.)	01 (1.)	NOPIC USR CON REL GBL NOSHR EXE RD WRT NOVEC LONG
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$RMSNAM	00000015 (21.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
__SRMSNAM	00000119 (281.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.68
Command processing	106	00:00:00.66	00:00:03.74
Pass 1	271	00:00:08.75	00:00:19.14
Symbol table sort	0	00:00:00.65	00:00:01.17
Pass 2	105	00:00:02.26	00:00:05.29
Symbol table output	13	00:00:00.09	00:00:00.26
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	529	00:00:12.51	00:00:30.32

The working set limit was 1350 pages.
47237 bytes (93 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 583 non-local and 19 local symbols.
440 source lines were read in Pass 1, producing 50 object records in Pass 2.
38 pages of virtual memory were used to define 32 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	20
TOTALS (all libraries)	20

798 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSTEST2/OBJ=OBJ\$:RMSTEST2 MSRC\$:RMSTEST2/UPDATE=(ENH\$:RMSTEST2)+EXECMLS/LIB

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

