

Table of contents

|     |    |                        |
|-----|----|------------------------|
| 2-  | 12 | DATA AREAS             |
| 3-  | 1  | MAIN SORT FLOW         |
| 4-  | 2  | RSTS oriented routines |
| 5-  | 1  | I/O ROUTINES           |
| 7-  | 1  | KEY COMPARE ROUTINES   |
| 11- | 1  | MULTIPLY & DIVIDE      |

```

1          .TITLE  SORTO  RTSORT  ---  ROOT  MODULE
2          .ENABL  LC
3          ;
4          ;
5          ;  Copyright (c) 1977, 1978, 1979, 1980.
6          ;  S&H Computer Systems Inc.
7          ;  Nashville, Tennessee
8          ;
9 000000          .CSECT  SORTO
10          POBASE = 000000'
11          .ENABL  LC
12          .SBTTL  DATA AREAS
13          ;
14          ;  ROOT MODULE OF RT-11/TSX SORT PROGRAM.
15          ;
16          ;  WRITTEN BY PHIL SHERROD.
17          ;
18          ;
19          ;  LINK COMMAND TO FORM OVERLAYED VERSION OF RTSORT:
20          ;
21          ;  .R LINK
22          ;  *RTSORT=SORTO/C
23          ;  *SORT1/O:1/C
24          ;  *SORT1/O:1/C
25          ;  *SORT2/O:1
26          ;
27          ;
28          ; #####
29          ;
30          ;  Modified in Feburary, 1980 by K. C. Lidster of DISC to also execute
31          ;  under the RT11 run-time for the RSTS operating system.  This mainly
32          ;  involved the handling of the extra file parameters pertaining to
33          ;  RSTS (PPN, clustersize, etc.).
34          ;
35          ; #####
36          ;
37          ;
38          ;  GLOBAL DEFINITIONS
39          ;
40          .GLOBL  @BLK, @LEN, MODE, TSXFL
41          .GLOBL  TOPMEM, NUMKY2, RECLN, ISMARS
42          .GLOBL  SKPCNT, MXREC1, MXREC2
43          .GLOBL  EOFCHR, EOFCOL
44          .GLOBL  INFLEN, KYUCA, EOFBUF
45          .GLOBL  BLKFC, OWCNT, OBUFSZ, START1
46          .GLOBL  STACK, P1TERM, P2TERM, KTYPE
47          .GLOBL  KAD, KSTRT, KLEN, OBUF1, OBUF2
48          .GLOBL  NXTBLK, OBFEND, ORPOS, FLBLK
49          .GLOBL  ITYPE, OTYPE, F$INPL
50          .GLOBL  RT$CBL, RT$FTN, RT$DBL, RT$TXT, RT$X
51          .GLOBL  RT$OS, RT$OR, RT$OX, RT$F11
52          .GLOBL  RT$RA, RT$RB, CURSAV, ENDSAV, SAVSPC, SKPVAL
53          .GLOBL  RT$VLN, RT$CR, MULTPY, DIVIDE, SRTCNT, TPCNT
54          .GLOBL  CISRT1, CISRT1, CISRT2
55          .GLOBL  NUMRUF, NUMCOL, NUMLEN
56          .GLOBL  IFPNT, CUROFL, LOWMEM, RWDLEN
57          .GLOBL  RCNT1, RCNT2, FLUSH, OUTWRT, MXFILS

```

```

58          . GLOBL  MXFIL2, COMPAR, ABORT, FILFL
59          . GLOBL  FILNB, FILRN, FILST, MSGNAM
60          . GLOBL  REC1, REC2, KYDIS
61          . GLOBL  T1SIZ, T2SIZ, DSKCVT
62          . GLOBL  JSW, OVLBIT, POTOP, POBASE
63          . GLOBL  UEOF, V3FLAG, ANDFLG, DRFLG
64          . GLOBL  KYCOMP, INRLEN, SELADR, FLDMAP, MAPEND
65          . GLOBL  F$BIN, F$CR, F$PAUS, F$VLEN
66          . GLOBL  KYBIN, KYLSS, KYTSS, KYDATE, SFLAGS
67          . GLOBL  SETNRT, RSTS, CHNPPN, FILPPN, SPCFLG
68          ;
69          ; GLOBAL REFERENCES
70          ;
71          . GLOBL  SORTP1, SORTP2, TERM
72          . GLOBL  SRTINI, P1TOP, P2TOP
73          ;
74          ; PARAMETERS
75          ;
76          000310  MXFILS  =      200.          ; MAX # OF TEMP FILES
77          000620  MXFIL2  =      2#MXFILS
78          000017  MXKEYS  =      15.          ; MAX # OF SORT KEYS PER RECORD
79          000012  LF      =      12          ; ASCII LINE FEED
80          000007  BELL    =      07          ; ASCII BELL
81
82          . IF DF RT11
83          ;
84          ; SET INITIAL STACK POINTER
85          000000  .ASECT
86          000042  =      42
87          000042  003336' .WORD  STACK
88          000000  .CSECT SORTO
89          . ENDC
90
91          ;
92          ; General control flags stored in SFLAGS:
93          ;
94          000001  F$BIN   =      1          ; Records contain binary data
95          000002  F$VLEN  =      2          ; Records are variable length
96          000004  F$CR    =      4          ; Insert CR-LF at end of output records
97          000010  F$PAUS  =      10         ; Pause on completion of reading input file
98          000020  F$INPL  =      20         ; Do an inplace sort
99          ;
100         ; File type flags stored in ITYPE & OTYPE.
101         ;
102         000001  RT$CBL   =      1          ; COBOL file
103         000002  RT$DBL   =      2          ; DIBOL file
104         000004  RT$FTN   =      4          ; FORTRAN file
105         000010  RT$F11   =      10         ; FILES-11 file
106         000020  RT$TXT   =      20         ; TEXT file
107         000040  RT$X     =      40         ; General fixed length records
108         000100  RT$OS    =      100        ; Sequential organization
109         000200  RT$OR    =      200        ; Relative organization
110         000400  RT$OX    =      400        ; Indexed organization
111         001000  RT$RA    =      1000       ; Ascii recording mode
112         002000  RT$RB    =      2000       ; Binary recording mode
113         040000  RT$CR    =      40000     ; Carriage-return is record terminator
114         100000  RT$VLN   =      100000    ; Blank strip the records
  
```

DATA AREAS

```

115 ;
116 ; FILE MANAGEMENT TABLES
117 ;
118 000000 FILST: .BLKW MXFILS ;FILE STARTING BLOCK NUMBER
119 000620 FILNB: .BLKW MXFILS ;# OF BLOCKS IN FILE
120 001440 FILRN: .BLKW MXFILS ;# OF SORTED RUNS IN FILE
121 ;
122 ; SORT KEY DESCRIPTOR TABLES
123 ;
124 002360 KTYPE: .BLKW MXKEYS ;KEY TYPE
125 002316 KAD: .BLKW MXKEYS ;0=>ASCENDING; 1=>DESCENDING
126 002354 KSTRT: .BLKW MXKEYS ;KEY STARTING COLUMN
127 002412 KLEN: .BLKW MXKEYS ;KEY LENGTH (COLUMNS)
128 ;
129 ; DATA AREAS
130 ;
131 002450 000000 SFLAGS: .WORD 0
132 002452 000000 ITYPE: .WORD 0
133 002454 000000 OTYPE: .WORD 0
134 002456 000000 NUMCOL: .WORD 0
135 002460 000000 NUMLEN: .WORD 0
136
137
138
139
140 002462 000000 CURDFL: .WORD 0
141 002464 000000 TOPMEM: .WORD 0
142 002466 000000 COMPAR: .WORD 0
143 002470 MSGNAM: .BLKW 3
144 002476 000000 NUMKY2: .WORD 0
145 002500 000000 RECLN: .WORD 0
146 002502 000000 ISMARS: .WORD 0
147 002504 000000 RWDLEN: .WORD 0
148 002506 000000 RCNT1: .WORD 0
149 002510 000000 RCNT2: .WORD 0
150 002512 000000 LOWMEM: .WORD 0
151 002514 000000 SKPCNT: .WORD 0
152 002516 000000 SKPVAL: .WORD 0
153 002520 000000 MXREC1: .WORD 0
154 002522 000000 MXREC2: .WORD 0
155 002524 000000 T1SIZ: .WORD 0
156 002526 000000 T2SIZ: .WORD 0
157 002530 000000 INFLEN: .WORD 0
158 002532 000000 000000 SRTCNT: .WORD 0,0
159 002536 000000 002000 TMPCNT: .WORD 0,0
160 002542 AREA: .BLKW 10.
161 002566 000000 OWCNT: .WORD 0
162 002570 000000 OBUFSZ: .WORD 0
163 002572 000000 REC1: .WORD 0
164 002574 000000 REC2: .WORD 0
165 002576 000000 OBUF1: .WORD 0
166 002600 000000 OBUF2: .WORD 0
167 002602 000000 NXTBLK: .WORD 0
168 002604 000000 FILFL: .WORD 0
169 002606 000000 OBFEND: .WORD 0
170 002610 000000 ORPOS: .WORD 0
171 002612 000000 FLBLK: .WORD 0

```

```

172 002614 000000      BLKFC:  .WORD  0
173 002616 000000      IFPNT:  .WORD  0
174 002620 000000      EOFBUF:  .WORD  0
175 002622 000000      FLDMAP:  .WORD  0
176 002624 000000      MAPEND:  .WORD  0
177 002626 000000      SELADR:  .WORD  0
178 002630 000000      INRLEN:  .WORD  0
179 002632 000000      EOFCOL:  .WORD  0
180 002634 000000      CHRCNT:  .WORD  0
181 002636 000000      CURSAV:  .WORD  0
182 002640 000000      ENDSAV:  .WORD  0
183 002642          SAVSPC:  .BLKW  30.
184          ;
185          ;  DEFINE STACK SPACE
186 002736          .BLKW  128.
187 003336      STACK:
188          ;
189          ;  BYTE DATA
190
191
192 003336          000      EOFCHR:  .BYTE  0
193 003337          000      UEOF:    .BYTE  0
194 003340          000      ANDFLG:  .BYTE  0
195 003341          000      DRFLG:   .BYTE  0
196 003342          000      SC1:     .BYTE  0
197 003343          000      SC2:     .BYTE  0
198 003344          000      SPCFLG:  .BYTE  0
199 003345          061      062      063      NUMBUF:  .ASCII  /12345678/
          003350          064      065      066
          003353          067      070
200          .NLIST  BIN
201 003355      PAUSMS:  .ASCIZ  /SORT input complete. Mount SORT output disk./<BELL>
202          .LIST  BIN
203          .EVEN
204          ;
205          ;  Table to convert non-separate sign characters for COBOL and DIBOL
206          ;  signed numeric display values into the natural digit and a
207          ;  sign indication.
208          ;
209          .NLIST  HEX
210          003354'      SINCHR  =      .-60          ;First char in table is "0" (octal 60).
211 003434          060      061      062      .BYTE  +<'0>, +<'1>, +<'2>, +<'3>, +<'4>      ; 060 - 064
212 003441          065      066      067      .BYTE  +<'5>, +<'6>, +<'7>, +<'8>, +<'9>      ; 065 - 071
213 003446          000      000      000      .BYTE           0,           0,           0      ; 072 - 076
214 003453          000      000      061      .BYTE           0,           0, +<'1>, +<'2>, +<'3>      ; 077 - 103
215 003460          064      065      066      .BYTE  +<'4>, +<'5>, +<'6>, +<'7>, +<'8>      ; 104 - 110
216 003465          071      317      316      .BYTE  +<'9>, -<'1>, -<'2>, -<'3>, -<'4>      ; 111 - 115
217 003472          313      312      311      .BYTE  -<'5>, -<'6>, -<'7>, -<'8>, -<'9>      ; 116 - 122
218 003477          000      000      000      .BYTE           0,           0,           0      ; 123 - 127
219 003504          000      000      000      .BYTE           0,           0,           0      ; 130 - 134
220 003511          000      000      000      .BYTE           0,           0,           0      ; 135 - 141
221 003516          000      000      000      .BYTE           0,           0,           0      ; 142 - 146
222 003523          000      000      000      .BYTE           0,           0,           0      ; 147 - 153
223 003530          000      000      000      .BYTE           0,           0,           0, -<'0>      ; 154 - 160
224 003535          317      316      315      .BYTE  -<'1>, -<'2>, -<'3>, -<'4>, -<'5>      ; 161 - 165
225 003542          312      311      310      .BYTE  -<'6>, -<'7>, -<'8>, -<'9>,           0      ; 166 - 172
226 003547          060      000      320      .BYTE  +<'0>,           0, -<'0>      ; 173 - 175
  
```

```

227          .EVEN
228          .LIST  BEX
229
230          ;
231          ; The following macro call generates hidden system dependent data
232          ;
233          ;
234 003552          .system
235
236          ;
237          ;
238          ; Define file descriptors and file specification blocks
239          ;
240 004264          fd1::  $FDB          ;file descriptor blocks
241 004274          fd2::  $FDB
242 004304          fd3::  $FDB
243 004314          fd4::  $FDB
244
245 004324          spc1::  $SPC          ;file specification blocks (system dependent names)
246 004336          spc2::  $SPC
247 004350          spc3::  $SPC
248 004362          spc4::  $SPC
249 004374          spc5::  $SPC
250 004406          spc6::  $SPC
251 004420          spc7::  $SPC
252 004432          spc8::  $SPC
253 004444          spc9::  $SPC
254
255          ; table of pointers to SPC's
256 004456 004324' 004336' 004350' files:: .word  spc1,  spc2,  spc3,  spc4,  spc5
          004464 004362' 004374'
257 004470 004406' 004420' 004432' .word  spc6,  spc7,  spc8,  spc9,  0
          004476 004444' 000000
258          ;
259          ; MACRO TO ABORT SORT
260          ;
261          .MACRO  XXX      COD
262          XXX'COD =
263          MOV     #'COD',R0
264          CALL   ABORT
265          .ENDM   XXX
266
267          ; -----
268          ; Debugging breakpoint checkins.
269          ;
270          ; Checkin from SORT1.
271          ;
272 004502 000207          CISRT1: RETURN
273          ;
274          ; Checkin from SORT1.
275          ;
276 004504 000207          CISRT1: RETURN
277          ;
278          ; Checkin from SORT2.
279          ;
280 004506 000207          CISRT2: RETURN

```

```

1          .SBTTL  MAIN SORT FLOW
2          ;-----
3          ;
4 004510   START:
5          ;
6          ;   ENTER SORTI TO DO INITIALIZATION
7          ;
8 004510   000167   000000G   JMP      SRTINI           ;ENTER INITIALIZATION ROUTINE
9          ;
10         ;   SORTI JUMPS TO START1 WHEN IT IS READY TO START A SORT
11        ;
12 004514   000167   000000G   START1: JMP     SORTP1           ;BEGIN SORT PHASE 1
13        ;
14        ;   TERMINATION OF PHASE 1
15        ;
16 004520   P1TERM: $LOCK           ;LOCK USR MODULE IN MEMORY FOR SPEED
17 004522   $KEEP   #FD2           ;CLOSE TEMP FILE 1 CHANNEL
18        ;   CLOSE TEMP FILE # 2 AND DELETE IT IF IT WAS NOT USED.
19 004570   005767   175732   TST     T2SIZ           ;DID WE OPEN TEMP FILE 2?
20 004574   001503   BEQ     7#           ;BR IF NOT
21 004576   $KEEP   #FD4           ;CLOSE TEMP FILE 2 CHANNEL
22 004644   016702   175732   MOV     NXTBLK,R2       ;GET LAST CLUSTER #
23 004650   004767   002022   CALL   DSKCVT          ;GET FILE # FOR IT
24 004654   020427   004314'   CMP    R4,#FD4         ;IS TEMP FILE 2 NEEDED?
25 004660   001451   BEQ    7#           ;BR IF YES
26        ;   TEMP FILE 2 IS NOT NEEDED SO DELETE IT
27 004662   005067   175640   CLR    T2SIZ          ;SAY FILE NOT USED
28 004666   $ERASE  #FD4,#SPC3
29        ;
30        ;   SEE IF WE SHOULD PAUSE WHILE OUTPUT DISK IS MOUNTED.
31        ;
32 005004   032767   000010   175436  7#:   BIT     #F$PAUS,SFLAGS ;IS PAUSE WANTED?
33 005012   001412   BEQ    1#           ;BR IF NOT
34 005014   $UNLOCK
35 005016   $PRINT  #PAUSMS      ;PRINT PAUSE MESSAGE
36 005024   5#:   .TTYIN
37 005030   120027   000012   CMPB   RO,#LF
38 005034   001373   BNE    5#           ;LOOP TILL LF ENTERED
39 005036   $LOCK
40        ;
41        ;   SEE IF AN IN-PLACE SORT IS WANTED.
42        ;
43 005040   032767   000020   175402  1#:   BIT     #F$INPL,SFLAGS ;WANT IN-PLACE SORT?
44 005046   001435   BEQ    4#           ;BR IF NOT
45        ;   IN-PLACE SORT IS WANTED. DO LOOKUP ON INPUT FILE FOR OUTPUT.
46 005050   $REOPEN #FD3,#SPC4     ;DO REOPEN TO ASSOCIATE OUTPUT WITH INPUT FILE
47 005136   000167   000510   JMP    OPNTMP
48        ;   THIS IS NOT AN IN-PLACE SORT.
49        ;   IF OUTPUT FILE EXISTS NOW DELETE IT.
50 005142   4#:   $ERASE  #FD3,#SPC1
51        ;   TRY TO OPEN NEW OUTPUT FILE
52        ;
53        ;   Determine how many blocks to request for output file.
54        ;
55 005260   032767   100000   175166  TONOF: BIT     #RT$VLN,OTYPE ;ARE OUTPUT RECORDS BLANK STRIPPED?
56 005266   001033   BNE    4#           ;BR IF YES
57        ;   Fixed length output records.

```

```

58          ; Multiply record size by number of records to determine needed size.
59 005270 016704 175236          MOV      SRTCNT,R4          ;GET # OF RECORDS PASSED TO WORD (LOW-ORDER)
60 005274 016705 175234          MOV      SRTCNT+2,R5        ;GET HIGH-ORDER COUNT
61 005300 016700 175174          MOV      RECLEN,R0         ;GET OUTPUT RECORD LENGTH
62 005304 032767 040000 175142  BIT      #RT$CR,DTYPE      ;ARE WE TO ADD CR-LF TO OUTPUT RECORDS?
63 005312 001010                BNE      5$                ;BR IF YES
64 005314 032767 000001 175132  BIT      #RT$CBL,DTYPE     ;IS THIS A COBOL FILE?
65 005322 001406                BEQ      6$                ;BR IF NOT
66 005324 032767 002000 175122  BIT      #RT$RB,DTYPE     ;COBOL BINARY FILE?
67 005332 001402                BEQ      6$                ;BR IF NOT
68 005334 062700 000002          5$:    ADD      #2,R0          ;WE NEED 2 MORE CHARS IN RECORD SIZE
69 005340 004767 002112          6$:    CALL     MULTPY        ;MULTIPLY # RECORDS TIMES RECORD SIZE
70 005344 012700 001000          MOV      #512.,R0         ;NOW DIVIDE BY # CHARS PER DISK BLOCK
71 005350 004767 002154          CALL     DIVIDE
72 005354 000436                BR       7$
73          ; Variable length (blank stripped) output records.
74          ; Use size of temp file to detmine space needed for output file.
75 005356 016704 175154          4$:    MOV      TMPcnt,R4          ;GET # DATA BYTES WRITTEN TO TEMP FILE
76 005362 016705 175152          MOV      TMPcnt+2,R5        ;GET HIGH-ORDER VALUE
77 005366 032767 040000 175060  BIT      #RT$CR,DTYPE      ;ARE WE TO ADD CR-LF TO OUTPUT RECORDS?
78 005374 001010                BNE      8$                ;BR IF YES
79 005376 032767 000001 175050  BIT      #RT$CBL,DTYPE     ;IS THIS A COBOL FILE?
80 005404 001416                BEQ      9$                ;BR IF NOT
81 005406 032767 002000 175040  BIT      #RT$RB,DTYPE     ;IS THIS A COBOL BINARY FILE?
82 005414 001412                BEQ      9$                ;BR IF NOT
83 005416 066704 175110          8$:    ADD      SRTCNT,R4          ;ADD # BYTES NEEDED FOR EXTRA CONTROL CHAR
84 005422 005505                ADC      R5                ;PROPOGATE CARRY
85 005424 066705 175104          ADD      SRTCNT+2,R5        ;ADD HIGH-ORDER
86 005430 066704 175076          ADD      SRTCNT,R4         ;ADD FOR 2ND CONTROL CHARACTER
87 005434 005505                ADC      R5
88 005436 066705 175072          ADD      SRTCNT+2,R5
89 005442 012700 001000          9$:    MOV      #512.,R0         ;# BYTES PER DISK BLOCK
90 005446 004767 002056          CALL     DIVIDE           ;CALC # BLOCKS NEEDED
91 005452 062704 000001          7$:    ADD      #1,R4          ;GET 1 EXTRA
92 005456 005505                ADC      R5                ;PROPOGATE CARRY
93 005460 001402                BEQ      10$               ;BR IF AMT NEEDED FITS IN 16 BITS
94 005462 012704 177777          MOV      #-1,R4          ;ASK FOR LARGEST AMT AVAILABLE
95          ; Try to open the output file.
96 005466                10$:   $NEW      #FD3,#SPC1,R4
97 005622 103013                BCC      OPNTMP           ;BR IF NO ERROR
98          ; CANNOT OPEN OUTPUT FILE
99 005624 120027 000001          CMPB    R0,#1            ;NOT ENOUGH SPACE?
100 005630 001004                BNE      2$              ;BR IF SOME OTHER ERROR
101 005632                XXX     10              ;NOT ENOUGH SPACE
102 005642                2$:    XXX     11              ;SOME OTHER OPEN ERROR
103          ; THE OUTPUT FILE IS OPEN --- REOPEN TEMP FILE FOR INPUT.
104 005652                OPNTMP: $OLD    #FD2,#SPC2
105 005776 103004                BCC      3$              ;BR IF NO LOOKUP ERROR
106 006000                9$:    XXX     23              ;CAN'T REOPEN TEMP FILE
107 006010 005767 174512          8$:    TST     T2SIZ        ;IS TEMP FILE 2 NEEDED?
108 006014 001453                BEQ      6$              ;BR IF NOT
109 006016                $OLD    #FD4,#SPC3
110 006142 103716                BCS     9$
111 006144                6$:    $UNLOCK          ;UNLOCK USR MODULE
112          ; BEGIN SECOND PHASE OF SORT.
113          ; (MERGE TEMP FILE SORTED STRINGS)
114 006146 000167 000000G          JMP     SORTP2

```



```
115 ;  
116 ; TERMINATION OF PHASE 2  
117 ;  
118 006152 005000 P2TERM: CLR R0 ; SAY NO ERROR  
119 ;  
120 ; THE XXX MACRO CAUSES A JUMP TO ABORT WITH THE ERROR CODE IN R0.  
121 ;  
122 006154 010046 ABORT: MOV R0, -(SP) ; SAVE ERROR CODE  
123 006156 $UNLOCK ; MAKE SURE USR IS NOT LOCKED  
124 006160 012600 MOV (SP)+, R0  
125 006162 000167 000000G JMP TERM ; REENTER SORTI
```

```

1      . IF DF RT11
2          . SBTTL  RSTS oriented routines
3      ;
4      ; SETNRT
5      ;
6      ; Set the RSTS oriented non-RT11 file parameters for the
7      ; next file operation.  These items are the PPN, protection
8      ; code, mode, and clustersize and are stored in a 4-word
9      ; block whose address immediately follows the JSR R5,SETNRT
10     ; call.  These items were saved off from a previous .CSISPC
11     ; monitor decode and will be loaded in the "one shots" of
12     ; the RT11 runtime's Read/Write stack (which is pointed at
13     ; by the low core RMON word).
14     ;
15     006166 105767 175464 SETNRT: TSTB  RSTS          ;We running under RSTS?
16     006172 001420      BEQ    10$          ;No, just ignore
17     006174 010046      MOV    R0,-(SP)     ;Else get some work registers
18     006176 010146      MOV    R1,-(SP)
19     006200 012500      MOV    (R5)+,R0     ;Point to the saved items
20     006202 013701 000054 MOV    @#RMON,R1      ;Point to the R/W stack
21     006206 012021      MOV    (R0)+,(R1)+  ;Load  PPN
22     006210 001003      BNE    5$
23     006212 017561 000000 177776 MOV    @(R5),-2(R1)
24     006220 012021      5$:  MOV    (R0)+,(R1)+  ; Protection Code
25     006222 012021      MOV    (R0)+,(R1)+  ; Mode
26     006224 012021      MOV    (R0)+,(R1)+  ; Cluster Size
27     006226 012601      MOV    (SP)+,R1      ;Restore the work registers
28     006230 012600      MOV    (SP)+,R0
29     006232 000401      BR     15$
30     006234 005725      10$: TST    (R5)+      ;Skip the save area pointer
31     006236 005725      15$: TST    (R5)+
32     006240 000205      RTS    R5
33     . ENDC
  
```

```

1          .SBTTL  I/O ROUTINES
2          ; FLUSH:  SUBROUTINE TO OUTPUT THE CURRENT
3          ; OUTPUT BUFFER AND SWITCH TO A NEW ONE.
4          ;
5 006242 010146          FLUSH:  MOV     R1,-(SP)      ;SAVE SOME REGISTERS
6 006244 010246          MOV     R2,-(SP)
7 006246 010346          MOV     R3,-(SP)
8 006250 016701 174322   MOV     OBUF1,R1      ;GET ADDRESS OF BUFFER
9 006254 016702 174332   MOV     FLBLK,R2      ;GET ADR OF NXT DISK BLK
10 006260 050211          BIS     R2,(R1)       ;SET FLINK IN BUFFER
11 006262 016701 174174   MOV     CUROFL,R1     ;GET CURRENT FILE #
12 006266 016703 174312   MOV     FILFL,R3      ;GET OLD FLINK
13 006272 010267 174306   MOV     R2,FILFL      ;SAVE NEW FLINK
14 006276 005261 000620' INC     FILNB(R1)     ;COUNT # OF BLOCKS IN FILE
15 006302 004767 000050 1$:  CALL    OUTWRT      ;START WRITE TO FILE
16          ; SWITCH BUFFERS
17 006306 016701 174266   MOV     OBUF2,R1
18 006312 016767 174260 174260   MOV     OBUF1,OBUF2
19 006320 010167 174252   MOV     R1,OBUF1
20 006324 010167 174256   MOV     R1,OBFEND     ;SET ADDR OF END OF BUFFER
21 006330 066767 174234 174250   ADD     OBUFSZ,OBFEND
22          ; INITIALIZE NEW BUFFER
23 006336 005021          CLR     (R1)+         ;CLEAR FLINK
24 006340 005021          CLR     (R1)+         ;CLEAR # OF RECS IN BUFFER
25 006342 010167 174242   MOV     R1,ORPOS      ;POS WHERE NXT REC GOES
26 006346 012603          MOV     (SP)+,R3      ;RESTORE REGISTERS
27 006350 012602          MOV     (SP)+,R2
28 006352 012601          MOV     (SP)+,R1
29 006354 000207          RETURN
30          ;
31          ;
32          ; OUTWRT:  ROUTINE TO WRITE THE CURRENT OUT BUFFER
33          ; TO THE DISK FILE.  WHEN CALLED R3 MUST CONTAIN
34          ; THE DISK GRANULE # WHERE THE BUFFER IS TO BE
35          ; WRITTEN.
36          ; THE BUFFER IS WRITTEN FROM THE POSITION POINTED
37          ; TO BY OBUF1.
38          ;
39 006356 010346          OUTWRT: MOV     R3,-(SP)
40 006360 010446          MOV     R4,-(SP)
41 006362 004767 000310   CALL    DSKCVT        ;CONVERT CLUSTER # TO FILE&BLOCK #
42 006366          $WAIT   R4          ;WAIT FOR PREVIOUS WRITE TO FINISH
43 006500 103004          BCC    1$            ;BR IF NO WRITE ERROR
44 006502          9$:  XXX     15          ;TEMP FILE I/O ERROR
45 006512          1$:  $WRITE R4,OBUF1,OWCNT,R3
46 006666 103705          BCS    9$
47 006670 012604          MOV     (SP)+,R4
48 006672 012603          MOV     (SP)+,R3
49 006674 000207          RETURN

```

```

1      ;
2      ;-----
3      ; DSKCVT IS CALLED TO CONVERT A TEMP FILE CLUSTER NUMBER
4      ; TO A BLOCK AND FILE NUMBER.
5      ; WHEN CALLED, R3 MUST CONTAIN THE CLUSTER NUMBER.
6      ; ON RETURN, R3 CONTAINS THE FILE BLOCK # AND R4 CONTAINS
7      ; THE CHANNEL NUMBER OF THE TEMP FILE.
8      ; ALL OTHER REGS ARE PRESERVED.
9      ;
10     006676 010346      DSKCVT: MOV     R3, -(SP)      ; PUSH CLUSTER #
11     006700 016704 173710      MOV     BLKFC, R4      ; GET # BLOCKS/CLUSTER
12     006704 000401      BR      1#
13     006706 061603      2#:    ADD     (SP), R3      ; CONVERT CLUSTER # TO BLOCK #
14     006710 005304      1#:    DEC     R4
15     006712 001375      BNE    2#
16     006714 005726      TST    (SP)+          ; CLEAN OFF STACK
17     006716 012704 004274'     MOV     #FD2, R4      ; ASSUME TEMP FILE #1
18     006722 020367 173576      CMP    R3, T1SIZ     ; IS THIS BLOCK IN FILE 1?
19     006726 101411      BLOS   3#            ; BR IF YES
20     006730 166703 173570      SUB    T1SIZ, R3     ; CALC BLOCK # IN FILE 2
21     006734 066703 173654      ADD    BLKFC, R3     ; T1 MAY NOT BE MULTIPLE OF BLOCKING
22     006740 020367 173562      CMP    R3, T2SIZ     ; WILL IT FIT IN FILE 2?
23     006744 101005      BHI    9#            ; BR IF TEMP FILE OVERFLOW
24     006746 012704 004314'     MOV     #FD4, R4      ; SET FOR TEMP FILE 2
25     006752 166703 173636      3#:    SUB    BLKFC, R3 ; 1ST CLUSTER IS CALLED # 1
26     006756 000207      RETURN
27     006760      9#:    XXX    12      ; OVERFLOW OF TEMP FILE
  
```

1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18 006770 060301  
 19 006772 060302  
 20 006774 006203  
 21 006776 024142  
 22 007000 001404  
 23 007002 000207  
 24 007004 024142  
 25 007006 103473  
 26 007010 101067  
 27 007012 005303  
 28 007014 001373  
 29 007016 000207  
 30  
 31  
 32  
 33  
 34 007020 006203  
 35 007022 022122  
 36 007024 001404  
 37 007026 000207  
 38 007030 022122  
 39 007032 103461  
 40 007034 101055  
 41 007036 005303  
 42 007040 001373  
 43 007042 000207

```

.SBTTL KEY COMPARE ROUTINES
-----
;
; The following routines are called to compare two key fields of various types.
; On entry, the following registers are set up:
;   R1 = Address of start of key field 1.
;   R2 = Address of start of key field 2.
;   R3 = Number of bytes in the key field.
;
; On return the condition code must be left set for BGT/BLT/BEQ test.
; R0, R1, R2 and R3 may be destroyed.
; R4 and R5 must be preserved.
-----
;
; Signed computational (binary) values stored with least significant word first.
;
KYCOMP: ADD     R3,R1           ;POINT PAST MOST SIGNIFICANT WORD
        ADD     R3,R2
        ASR    R3              ;GET # WORDS TO COMPARE
        CMP    -(R1),-(R2)    ;COMPARE MOST SIGNIFICANT WORDS (SIGNED)
        BEQ   2$              ;BR IF MOST SIGNIFICANT WORDS ARE EQUAL
        RETURN              ;MOST SIGNIFICANT WORDS ARE UNEQUAL
5$:     CMP    -(R1),-(R2)    ;COMPARE LOW ORDER WORDS
        BLO   SETLT          ;DO THIS AS UNSIGNED COMPARISON
        BHI   SETGT
2$:     DEC    R3              ;MORE TO COMPARE?
        BNE   5$              ;BR IF YES
        RETURN              ;VALUES ARE EQUAL
-----
;
; Compare two binary items stored with most significant word first.
;
KYBIN:  ASR    R3              ;GET # WORDS TO COMPARE
        CMP    (R1)+,(R2)+    ;COMPARE MOST SIGNIFICANT WORDS (SIGNED)
        BEQ   2$              ;BR IF THEY MATCH
        RETURN              ;MOST SIGNIFICANT WORDS ARE UNEQUAL
5$:     CMP    (R1)+,(R2)+    ;COMPARE LOW ORDER WORDS
        BLO   SETLT          ;BR IF 1<2
        BHI   SETGT          ;BR IF 1>2
2$:     DEC    R3              ;MORE TO COMPARE?
        BNE   5$              ;BR IF YES
        RETURN              ;VALUES ARE EQUAL
    
```

```

1      ;-----
2      ; Compare two items with leading separate signs.
3      ;
4 007044 122127 000053 KYLSS:  CMPB  (R1)+,#'+      ;ITEM 1 POSITIVE?
5 007050 001407          BEQ   1#          ;BR IF YES
6 007052 122227 000053          CMPB  (R2)+,#'+      ;ITEM 2 POSITIVE?
7 007056 001447          BEQ   SETLT       ;1ST<0, 2ND>=0
8      ; Both values are negative so reverse order of comparison.
9 007060 010100          MOV   R1,R0
10 007062 010201          MOV   R2,R1
11 007064 010002          MOV   R0,R2
12 007066 000403          BR    2#
13     ; First value is positive.
14 007070 122227 000053 1#:   CMPB  (R2)+,#'+      ;2ND VALUE POSITIVE?
15 007074 001035          BNE   SETGT       ;1ST>0, 2ND<0
16     ; Values have same sign so compare magnitudes.
17 007076 005303 2#:   DEC   R3          ;COUNT DOWN SIGN CHARACTER
18 007100 122122 3#:   CMPB  (R1)+,(R2)+    ;COMPARE CHARACTER STRINGS
19 007102 001002          BNE   9#          ;BR IF UNEQUAL
20 007104 005303          DEC   R3          ;MORE TO COMPARE?
21 007106 003374          BGT   3#          ;BR IF YES
22 007110 000207 9#:   RETURN          ;RETURN WITH CC SET
23
24     ;-----
25     ; Compare two values with trailing separate signs.
26     ;
27 007112 005303 KYTSS:  DEC   R3          ;DON'T COUNT SIGN CHAR IN COMPARISON LOOP
28 007114 060301          ADD   R3,R1        ;POINT TO SIGN CHARACTER
29 007116 060302          ADD   R3,R2
30 007120 121127 000053          CMPB  (R1),#'+      ;1ST VALUE POSITIVE?
31 007124 001407          BEQ   1#          ;BR IF YES
32 007126 121227 000053          CMPB  (R2),#'+      ;2ND VALUE POSITIVE?
33 007132 001421          BEQ   SETLT       ;1ST<0, 2ND>=0
34     ; Both values are negative so reverse order of comparison.
35 007134 010100          MOV   R1,R0
36 007136 010201          MOV   R2,R1
37 007140 010002          MOV   R0,R2
38 007142 000403          BR    2#
39     ; First value is positive.
40 007144 121227 000053 1#:   CMPB  (R2),#'+      ;2ND VALUE POSITIVE?
41 007150 001007          BNE   SETGT       ;1ST VALUE >=0, 2ND<0
42     ; Values have same sign. Compare magnitudes.
43 007152 160301 2#:   SUB   R3,R1        ;POINT TO 1ST DIGIT OF VALUE
44 007154 160302          SUB   R3,R2
45 007156 122122 3#:   CMPB  (R1)+,(R2)+    ;COMPARE DIGIT STRINGS
46 007160 001002          BNE   9#          ;BR IF UNEQUAL
47 007162 005303          DEC   R3          ;MORE TO COMPARE?
48 007164 003374          BGT   3#          ;BR IF YES
49 007166 000207 9#:   RETURN          ;RETURN WITH CC SET
50
51     ; Set condition to cause BGT branch to be taken.
52     ;
53 007170 005727 000001 SETGT: TST   #1          ;SET CC FOR BGT
54 007174 000207          RETURN
55
56     ; Set condition code to cause BLT branch to be taken.
57     ;

```

58 007176 005727 177777  
59 007202 000207

SETLT: TST #-1  
RETURN

;SET CC FOR BLT

```

1
2 ; Compare two values that have trailing non-separate signs.
3 ;
4 007204 005303 KYDIS: DEC R3
5 007206 060301 ADD R3,R1 ;POINT TO SIGN CHARACTERS
6 007210 060302 ADD R3,R2
7 007212 111100 MOVB (R1),R0 ;GET SIGN CHAR FROM 1ST VALUE
8 007214 116000 003354' MOVB SINCHR(R0),R0 ;GET UNSIGNED DIGIT
9 007220 002021 BGE 1$ ;BR IF VALUE POSITIVE
10 007222 005400 NEG R0 ;MAKE CHAR POSITIVE
11 007224 110067 174112 MOVB R0,SC1 ;SIGN DIGIT FROM 1ST VALUE
12 ; First value is negative.
13 007230 111200 MOVB (R2),R0 ;GET SIGN CHAR FROM 2ND VALUE
14 007232 116000 003354' MOVB SINCHR(R0),R0 ;GET UNSIGNED DIGIT
15 007236 002357 BGE SETLT ;1ST<0, 2ND>=0
16 007240 005400 NEG R0 ;MAKE CHAR POSITIVE
17 ; Both values are negative so reverse order of comparison.
18 007242 116767 174074 174073 MOVB SC1,SC2 ;REVERSE SIGN DIGITS
19 007250 110067 174066 MOVB R0,SC1
20 007254 010100 MOV R1,R0 ;REVERSE NUMBER STRING POINTERS
21 007256 010201 MOV R2,R1
22 007260 010002 MOV R0,R2
23 007262 000410 BR 2$
24 ; First value is positive.
25 007264 110067 174052 1$: MOVB R0,SC1 ;SAVE SIGN DIGIT FROM 1ST VALUE
26 007270 111200 MOVB (R2),R0 ;GET SIGN CHAR FROM 2ND VALUE
27 007272 116000 003354' MOVB SINCHR(R0),R0 ;GET UNSIGNED DIGIT
28 007276 002734 BLT SETGT ;BR IF 1ST>=0 AND 2ND<0
29 007300 110067 174037 MOVB R0,SC2 ;SAVE MAGNITUDE OF SIGN CHAR 2
30 ; Values have same sign so compare magnitudes.
31 007304 005703 2$: TST R3 ;ANY DIGITS OTHER THAN SIGN DIGITS?
32 007306 001406 BEQ 4$ ;BR IF NOT
33 007310 160301 SUB R3,R1 ;POINT TO FRONT OF DIGIT STRINGS
34 007312 160302 SUB R3,R2
35 007314 122122 3$: CMPB (R1)+,(R2)+ ;COMPARE DIGITS UP TO SIGN CHAR
36 007316 001005 BNE 9$ ;BR IF UNEQUAL
37 007320 005303 DEC R3
38 007322 003374 BGT 3$
39 ; Compare sign digits.
40 007324 126767 174012 174011 4$: CMPB SC1,SC2 ;COMPARE LAST DIGITS OF VALUES
41 007332 000207 9$: RETURN ;RETURN WITH CC SET
    
```



```

1 ; -----
2 ; Compare two dates in the form (mmddy).
3 ;
4 007334 126162 000004 000004 KYDATE: CMPB 4(R1),4(R2) ;COMPARE 1ST DIGIT OF YEAR
5 007342 001012 BNE 2$ ;BR IF UNEQUAL
6 007344 126162 000005 000005 CMPB 5(R1),5(R2) ;COMPARE 2ND DIGIT OF YEAR
7 007352 001006 BNE 2$ ;BR IF UNEQUAL
8 ; Years are qual. Compare month and day.
9 007354 012703 000004 MOV #4,R3 ;COMPARE 4 DIGITS
10 007360 122122 1$: CMPB (R1)+,(R2)+ ;COMPARE MONTH & DAY VALUES
11 007362 001002 BNE 2$ ;BR IF UNEQUAL
12 007364 005303 DEC R3
13 007366 003374 BGT 1$
14 007370 000207 2$: RETURN ;RETURN WITH CC SET
15 ;
16 ; -----
17 ; ROUTINE TO COMPARE TWO ASCII STRINGS WHERE LOWER CASE CHARACTERS
18 ; ARE TO BE CONVERTED TO UPPER CASE.
19 ;
20 ; ON ENTRY,
21 ; R1 = ADDRESS OF STRING 1.
22 ; R2 = ADDRESS OF STRING 2.
23 ; R3 = NUMBER OF CHARACTERS IN EACH STRING.
24 ; R0,R1,R2,R3 AND R5 ARE DESTROYED.
25 ;
26 007372 010367 173236 KYUCA: MOV R3,CHRCNT ;SAVE CHARACTER COUNT
27 007376 112100 3$: MOVB (R1)+,R0 ;GET A CHAR FROM STRING 1
28 007400 112203 MOVB (R2)+,R3 ;GET A CHAR FROM STRING 2
29 007402 020027 000141 CMP R0,#141 ;IS STRING 1 CHAR LOWER CASE?
30 007406 103405 BLO 1$ ;BR IF NOT
31 007410 020027 000172 CMP R0,#172 ;LC(Z)
32 007414 101002 BHI 1$ ;BR IF NOT LC
33 007416 162700 000040 SUB #40,R0 ;CONVERT CHARACTER TO UPPER CASE
34 007422 020327 000141 1$: CMP R3,#141 ;IS STRING 2 CHAR LOWER CASE?
35 007426 103405 BLO 2$ ;BR IF NOT LC
36 007430 020327 000172 CMP R3,#172 ;LC(Z)
37 007434 101002 BHI 2$ ;BR IF NOT LC
38 007436 162703 000040 SUB #40,R3 ;CONVERT CHAR TO UPPER CASE
39 007442 120003 2$: CMPB R0,R3 ;DO THE COMPARISON
40 007444 001003 BNE 9$ ;BR IF FOUND DIFFERENCE
41 007446 005367 173162 DEC CHRCNT ;ANY MORE CHARS TO COMPARE?
42 007452 001351 BNE 3$ ;BR IF YES
43 007454 000207 9$: RETURN

```

```
1  
2  
3  
4  
5  
6  
7  
8 007456 010046  
9 007460 010246  
10 007462 010346  
11 007464 010402  
12 007466 010503  
13 007470 005004  
14 007472 005005  
15 007474 000241  
16 007476 006000  
17 007500 103003  
18 007502 060204  
19 007504 005505  
20 007506 060305  
21 007510 006302  
22 007512 006103  
23 007514 005700  
24 007516 001366  
25 007520 012603  
26 007522 012602  
27 007524 012600  
28 007526 000207
```

.SBTTL MULTIPLY & DIVIDE  
-----  
; Routine to multiply the 32-bit value in R4 (low-order) and R5 (high-order)  
; by the 16-bit value in R0.  
; On return the 32-bit product is in R4-R5.  
; R0 is preserved.  
;  
MULTPY: MOV R0, -(SP)  
MOV R2, -(SP)  
MOV R3, -(SP)  
MOV R4, R2 ; MOVE VALUE FROM R4-R5 TO R2-R3  
MOV R5, R3  
CLR R4  
CLR R5  
1\$: CLC  
ROR R0 ; NEED TO ADD THIS TIME?  
BCC 2\$ ; BR IF NOT  
ADD R2, R4 ; ADD LOW-ORDER VALUE  
ADC R5 ; PROPOGATE CARRY  
ADD R3, R5 ; ADD HIGH-ORDER VALUE  
2\$: ASL R2 ; ROTATE MULTIPLYING VALUE  
ROL R3  
TST R0 ; NEED TO DO MORE?  
BNE 1\$ ; BR IF YES  
MOV (SP)+, R3  
MOV (SP)+, R2  
MOV (SP)+, R0  
RETURN

```

1          ; -----
2          ; Routine to divide the 32-bit value in R4 (low-order) and R5 (high-order)
3          ; by the 16-bit quantity in R0.
4          ; The 32-bit quotient is returned in R4-R5 and the remainder is
5          ; returned in R0.
6          ;
7 007530 010146 DIVIDE: MOV     R1, -(SP)
8 007532 010346         MOV     R3, -(SP)
9 007534 005001         CLR     R1             ; INITIALIZE REMAINDER
10 007536 012703 000037 MOV     #31., R3        ; SET SHIFT COUNT
11 007542 006304 1#:   ASL     R4             ; SHIFT BIT OUT OF LOW-ORDER
12 007544 006105         ROL     R5             ; SHIFT THROUGH HIGH-ORDER
13 007546 006101         ROL     R1             ; AND INTO R1
14 007550 020100         CMP     R1, R0        ; GOT ENOUGH TO SUBTRACT YET?
15 007552 103402         BLO     2#           ; BR IF NOT
16 007554 160001         SUB     R0, R1        ; SUBTRACT DIVISOR
17 007556 005204         INC     R4             ; INCREMENT QUOTIENT
18 007560 005303 2#:   DEC     R3             ; COUNT # BITS SHIFTED
19 007562 100367         BPL     1#           ; BR IF MORE TO DO
20 007564 010100         MOV     R1, R0        ; GET REMAINDER TO R0
21 007566 012603         MOV     (SP)+, R3
22 007570 012601         MOV     (SP)+, R1
23 007572 000207         RETURN
24          ;
25 007574          PATCH0: .BLKW 20.           ; PATCH AREA.
26          007644'      POTOP =
27          004510'      .END   START
    
```

Symbol table

|         |          |     |          |            |     |         |            |     |          |            |     |         |           |     |
|---------|----------|-----|----------|------------|-----|---------|------------|-----|----------|------------|-----|---------|-----------|-----|
| ABORT   | 006154R0 | 002 | FILNB    | 000620R0   | 002 | LOWMEM  | 002512R0   | 002 | RCNT2    | 002510R0   | 002 | SORTP2= | ***** G   |     |
| ANDFLG  | 003340R0 | 002 | FILPPN   | 003626R0   | 002 | MAPEND  | 002624R0   | 002 | RECLN    | 002500R0   | 002 | SPCFLG  | 003344R0  | 002 |
| AREA    | 002542R  | 002 | FILRN    | 001440R0   | 002 | MODE    | 003653R0   | 002 | REC1     | 002572R0   | 002 | SPCSIZ= | 000005    |     |
| BELL    | = 000007 |     | FILST    | 000000R0   | 002 | MSGNAM  | 002470R0   | 002 | REC2     | 002574R0   | 002 | SPC1    | 004324R0  | 002 |
| BIGEST= | 177777   |     | FLBLK    | 002612R0   | 002 | MULTPY  | 007456R0   | 002 | RMON     | = 000054   |     | SPC2    | 004336R0  | 002 |
| BLKFC   | 002614R0 | 002 | FLDMAP   | 002622R0   | 002 | MXFILS= | 000310 G   |     | RSPCS    | 003662R0   | 002 | SPC3    | 004350R0  | 002 |
| BUFSIZ= | 001000   |     | FLUSH    | 006242R0   | 002 | MXFIL2= | 000620 G   |     | RSTEXT   | 004222R0   | 002 | SPC4    | 004362R0  | 002 |
| CHNBIT= | 000400   |     | FSTIME   | 003652R0   | 002 | MXKEYS= | 000017     |     | RSTS     | 003656R0   | 002 | SPC5    | 004374R0  | 002 |
| CHNEXT  | 004232R0 | 002 | F\$BIN   | = 000001 G |     | MXREC1  | 002520R0   | 002 | RSTSCH   | 004242R0   | 002 | SPC6    | 004406R0  | 002 |
| CHND    | = 000000 |     | F\$CR    | = 000004 G |     | MXREC2  | 002522R0   | 002 | RSTSJB=  | 000404     |     | SPC7    | 004420R0  | 002 |
| CHNPPN  | 003624R0 | 002 | F\$INPL= | 000020 G   |     | NOERR   | = 000001   |     | RT\$CBL= | 000001 G   |     | SPC8    | 004432R0  | 002 |
| CHRCNT  | 002634R  | 002 | F\$PAUS= | 000010 G   |     | NUMBUF  | 003345R0   | 002 | RT\$CR   | = 040000 G |     | SPC9    | 004444R0  | 002 |
| CINNRT  | 003642R0 | 002 | F\$VLEN= | 000002 G   |     | NUMCOL  | 002456R0   | 002 | RT\$DBL= | 000002 G   |     | SRTCNT  | 002532R0  | 002 |
| CINSPC  | 003630R0 | 002 | GTLN     | 003657R0   | 002 | NUMKY2  | 002476R0   | 002 | RT\$FTN= | 000004 G   |     | SRTINI= | ***** G   |     |
| CISRTI  | 004502R0 | 002 | HB\$ARS= | 000006     |     | NUMLEN  | 002460R0   | 002 | RT\$F11= | 000010 G   |     | STACK   | 003336R0  | 002 |
| CISRT1  | 004504R0 | 002 | HB\$DFS= | 000002     |     | NXTBLK  | 002602R0   | 002 | RT\$DR   | = 000200 G |     | START   | 004510R   | 002 |
| CISRT2  | 004506R0 | 002 | HB\$DIP= | 000030     |     | OBFEND  | 002606R0   | 002 | RT\$DS   | = 000100 G |     | START1  | 004514R0  | 002 |
| COMPAR  | 002466R0 | 002 | HB\$DUP= | 000001     |     | OBUFSZ  | 002570R0   | 002 | RT\$DX   | = 000400 G |     | TERM    | = ***** G |     |
| CR      | = 000015 |     | HB\$RIU= | 000032     |     | OBUF1   | 002576R0   | 002 | RT\$RA   | = 001000 G |     | TMPCNT  | 002536R0  | 002 |
| CSPCDF= | 000376   |     | HB\$RSZ= | 000004     |     | OBUF2   | 002600R0   | 002 | RT\$RB   | = 002000 G |     | TONDF   | 005260R   | 002 |
| CTRLZ   | = 000032 |     | HD\$DEV= | 000000     |     | OPNTMP  | 005652R    | 002 | RT\$TXT= | 000020 G   |     | TOPMEM  | 002464R0  | 002 |
| CURDFL  | 002462R0 | 002 | HD\$SZ=  | 000004     |     | ORFLG   | 003341R0   | 002 | RT\$VLN= | 100000 G   |     | TSXFL   | 003654R0  | 002 |
| CURSAV  | 002636R0 | 002 | IFPNT    | 002616R0   | 002 | ORPOS   | 002610R0   | 002 | RT\$X    | = 000040 G |     | T1SIZ   | 002524R0  | 002 |
| DBLKFC= | 000005   |     | INLEN    | 002530R0   | 002 | OTYPE   | 002454R0   | 002 | RT11     | = 000000   |     | T2SIZ   | 002526R0  | 002 |
| DEFEXT  | 004116R0 | 002 | INRLEN   | 002630R0   | 002 | OUTWRT  | 006356R0   | 002 | RWDLEN   | 002504R0   | 002 | UEOF    | 003337R0  | 002 |
| DIVIDE  | 007530R0 | 002 | ISMARS   | 002502R0   | 002 | DVLBIT= | 001000 G   |     | SAVBUF   | 004126R0   | 002 | USERRB= | 000053    |     |
| DSKCVT  | 006676R0 | 002 | ITYPE    | 002452R0   | 002 | DWCNT   | 002566R0   | 002 | SAVSPC   | 002642R0   | 002 | USRLDC= | 000046    |     |
| ENDSAV  | 002640R0 | 002 | JSW      | = 000044 G |     | PATCH0  | 007574R    | 002 | SC1      | 003342R    | 002 | VERSNO= | 000276    |     |
| EOFBUF  | 002620R0 | 002 | KAD      | 002316R0   | 002 | PAUSMS  | 003355R    | 002 | SC2      | 003343R    | 002 | V3FLAG  | 003655R0  | 002 |
| EOFCHR  | 003336R0 | 002 | KLEN     | 002412R0   | 002 | PFSPC   | = 000002   |     | SELADR   | 002626R0   | 002 | WRDS    | = 000004  |     |
| EOFCOL  | 002632R0 | 002 | KSTRT    | 002354R0   | 002 | POBASE= | 000000R0   | 002 | SETGT    | 007170R    | 002 | XBUF    | 004000R0  | 002 |
| ERR     | = 000006 |     | KTYPE    | 002260R0   | 002 | POTOP   | = 007644R0 | 002 | SETLT    | 007176R    | 002 | XEMT    | 004250R0  | 002 |
| ERRLOC= | 000052   |     | KYBIN    | 007020R0   | 002 | P1TERM  | 004520R0   | 002 | SETNRT   | 006166R0   | 002 | XXX10   | = 005632R | 002 |
| FDBSIZ= | 000004   |     | KYCOMP   | 006770R0   | 002 | P1TOP   | = ***** G  |     | SEVERR=  | 000010     |     | XXX11   | = 005642R | 002 |
| FD1     | 004264R0 | 002 | KYDATE   | 007334R0   | 002 | P2TERM  | 006152R0   | 002 | SFLAGS   | 002450R0   | 002 | XXX12   | = 006760R | 002 |
| FD2     | 004274R0 | 002 | KYDIS    | 007204R0   | 002 | P2TOP   | = ***** G  |     | SINCHR=  | 003354R    | 002 | XXX15   | = 006502R | 002 |
| FD3     | 004304R0 | 002 | KYLSS    | 007044R0   | 002 | QBLK    | 003552R0   | 002 | SKPCNT   | 002514R0   | 002 | XXX23   | = 006000R | 002 |
| FD4     | 004314R0 | 002 | KYTSS    | 007112R0   | 002 | QLEN    | = 000003 G |     | SKPVAL   | 002516R0   | 002 | ... V1  | = 000003  |     |
| FILES   | 004456R0 | 002 | KYUCA    | 007372R0   | 002 | RCNT1   | 002506R0   | 002 | SORTP1=  | ***** G    |     | ... V2  | = 000061  |     |
| FILFL   | 002604R0 | 002 | LF       | = 000012   |     |         |            |     |          |            |     |         |           |     |

. ABS. 000044 000 (RW, I, GBL, ABS, DVR)  
 000000 001 (RW, I, LCL, REL, CON)  
 SORTO 007644 002 (RW, I, GBL, REL, DVR)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 15944 Words ( 63 Pages)  
 Size of core pool: 18432 Words ( 72 Pages)  
 Operating system: RT-11

Elapsed time: 00:00:23.48  
 ,DB4: CBSRTO, DK: CBSRTH, CRF=DK: CBSRTH, CBSRTO/C



|        |        |        |        |       |       |       |      |      |      |
|--------|--------|--------|--------|-------|-------|-------|------|------|------|
| FLDMAP | 2-64   | 2-175# |        |       |       |       |      |      |      |
| FLUSH  | 2-57   | 5-5#   |        |       |       |       |      |      |      |
| FSTIME | 2-234# |        |        |       |       |       |      |      |      |
| GTLN   | 2-234# |        |        |       |       |       |      |      |      |
| HB#ARS | 1-74#  |        |        |       |       |       |      |      |      |
| HB#DFS | 1-72#  |        |        |       |       |       |      |      |      |
| HB#DIP | 1-75#  |        |        |       |       |       |      |      |      |
| HB#DUP | 1-71#  |        |        |       |       |       |      |      |      |
| HB#RIU | 1-76#  |        |        |       |       |       |      |      |      |
| HB#RSZ | 1-73#  |        |        |       |       |       |      |      |      |
| HD##SZ | 1-81#  |        |        |       |       |       |      |      |      |
| HD#DEV | 1-80#  |        |        |       |       |       |      |      |      |
| IFPNT  | 2-56   | 2-173# |        |       |       |       |      |      |      |
| INFLN  | 2-44   | 2-157# |        |       |       |       |      |      |      |
| INRLN  | 2-64   | 2-178# |        |       |       |       |      |      |      |
| ISMARS | 2-41   | 2-146# |        |       |       |       |      |      |      |
| ITYPE  | 2-49   | 2-132# |        |       |       |       |      |      |      |
| JSW    | 1-51#  | 2-62   |        |       |       |       |      |      |      |
| KAD    | 2-47   | 2-125# |        |       |       |       |      |      |      |
| KLEN   | 2-47   | 2-127# |        |       |       |       |      |      |      |
| KSTRT  | 2-47   | 2-126# |        |       |       |       |      |      |      |
| KTYPE  | 2-45   | 2-124# |        |       |       |       |      |      |      |
| KYBIN  | 2-66   | 7-34#  |        |       |       |       |      |      |      |
| KYCOMP | 2-64   | 7-18#  |        |       |       |       |      |      |      |
| KYDATE | 2-66   | 10-4#  |        |       |       |       |      |      |      |
| KYDIS  | 2-60   | 9-4#   |        |       |       |       |      |      |      |
| KYLSS  | 2-66   | 8-4#   |        |       |       |       |      |      |      |
| KYTSS  | 2-66   | 8-27#  |        |       |       |       |      |      |      |
| KYUCA  | 2-44   | 10-26# |        |       |       |       |      |      |      |
| LF     | 1-39#  | 2-79#  | 3-37   |       |       |       |      |      |      |
| LOWMEM | 2-56   | 2-150# |        |       |       |       |      |      |      |
| MAPEND | 2-64   | 2-176# |        |       |       |       |      |      |      |
| MODE   | 2-40   | 2-234# |        |       |       |       |      |      |      |
| MSGNAM | 2-59   | 2-143# |        |       |       |       |      |      |      |
| MULTPY | 2-53   | 3-69   | 11-8#  |       |       |       |      |      |      |
| MXFIL2 | 2-58   | 2-77#  |        |       |       |       |      |      |      |
| MXFILS | 2-57   | 2-76#  | 2-77   | 2-118 | 2-119 | 2-120 |      |      |      |
| MXKEYS | 2-78#  | 2-124  | 2-125  | 2-126 | 2-127 |       |      |      |      |
| MXREC1 | 2-42   | 2-153# |        |       |       |       |      |      |      |
| MXREC2 | 2-42   | 2-154# |        |       |       |       |      |      |      |
| NOERR  | 1-56#  |        |        |       |       |       |      |      |      |
| NUMBUF | 2-55   | 2-199# |        |       |       |       |      |      |      |
| NUMCOL | 2-55   | 2-134# |        |       |       |       |      |      |      |
| NUMKY2 | 2-41   | 2-144# |        |       |       |       |      |      |      |
| NUMLEN | 2-55   | 2-135# |        |       |       |       |      |      |      |
| NXTBLK | 2-48   | 2-167# | 3-22   |       |       |       |      |      |      |
| OBFEND | 2-48   | 2-169# | 5-20*  | 5-21* |       |       |      |      |      |
| OBUF1  | 2-47   | 2-165# | 5-8    | 5-18  | 5-19* | 5-45  |      |      |      |
| OBUF2  | 2-47   | 2-166# | 5-17   | 5-18* |       |       |      |      |      |
| OBUFSZ | 2-45   | 2-162# | 5-21   |       |       |       |      |      |      |
| OPNTMP | 3-47   | 3-97   | 3-104# |       |       |       |      |      |      |
| ORFLG  | 2-63   | 2-195# |        |       |       |       |      |      |      |
| ORPOS  | 2-48   | 2-170# | 5-25*  |       |       |       |      |      |      |
| OTYPE  | 2-49   | 2-133# | 3-55   | 3-62  | 3-64  | 3-66  | 3-77 | 3-79 | 3-81 |
| OUTWRT | 2-57   | 5-15   | 5-39#  |       |       |       |      |      |      |
| OVLBIT | 1-52#  | 2-62   |        |       |       |       |      |      |      |

|        |        |        |       |        |       |        |        |  |
|--------|--------|--------|-------|--------|-------|--------|--------|--|
| DWCNT  | 2-45   | 2-161# | 5-45  | 5-45   |       |        |        |  |
| POBASE | 2-19#  | 2-62   |       |        |       |        |        |  |
| POTOP  | 2-62   | 12-26# |       |        |       |        |        |  |
| P1TERM | 2-45   | 3-16#  |       |        |       |        |        |  |
| P1TOP  | 2-72   |        |       |        |       |        |        |  |
| P2TERM | 2-45   | 3-118# |       |        |       |        |        |  |
| P2TOP  | 2-72   |        |       |        |       |        |        |  |
| PATCHO | 12-25# |        |       |        |       |        |        |  |
| PAUSMS | 2-201# | 3-35   |       |        |       |        |        |  |
| PFSPC  | 1-58#  | 3-17*  | 3-21* | 3-46*  | 3-96* | 3-104* | 3-109* |  |
| QBLK   | 2-40   | 2-234# |       |        |       |        |        |  |
| QLEN   | 1-48#  | 2-40   | 2-234 | 2-234# |       |        |        |  |
| RCNT1  | 2-57   | 2-148# |       |        |       |        |        |  |
| RCNT2  | 2-57   | 2-149# |       |        |       |        |        |  |
| REC1   | 2-60   | 2-163# |       |        |       |        |        |  |
| REC2   | 2-60   | 2-164# |       |        |       |        |        |  |
| RECLEN | 2-41   | 2-145# | 3-61  |        |       |        |        |  |
| RMON   | 1-57#  | 4-20   |       |        |       |        |        |  |
| RSPCS  | 2-234# |        |       |        |       |        |        |  |
| RSTEXT | 2-234# |        |       |        |       |        |        |  |
| RSTS   | 2-67   | 2-234# | 4-15  |        |       |        |        |  |
| RSTSCH | 2-234# |        |       |        |       |        |        |  |
| RSTSJB | 1-64#  |        |       |        |       |        |        |  |
| RT#CBL | 2-50   | 2-102# | 3-64  | 3-79   |       |        |        |  |
| RT#CR  | 2-53   | 2-113# | 3-62  | 3-77   |       |        |        |  |
| RT#DBL | 2-50   | 2-103# |       |        |       |        |        |  |
| RT#F11 | 2-51   | 2-105# |       |        |       |        |        |  |
| RT#FTN | 2-50   | 2-104# |       |        |       |        |        |  |
| RT#OR  | 2-51   | 2-109# |       |        |       |        |        |  |
| RT#OS  | 2-51   | 2-108# |       |        |       |        |        |  |
| RT#OX  | 2-51   | 2-110# |       |        |       |        |        |  |
| RT#RA  | 2-52   | 2-111# |       |        |       |        |        |  |
| RT#RB  | 2-52   | 2-112# | 3-66  | 3-81   |       |        |        |  |
| RT#TXT | 2-50   | 2-106# |       |        |       |        |        |  |
| RT#VLN | 2-53   | 2-114# | 3-55  |        |       |        |        |  |
| RT#X   | 2-50   | 2-107# |       |        |       |        |        |  |
| RT11   | 1-47#  | 2-82   | 4-1   |        |       |        |        |  |
| RWDLEN | 2-56   | 2-147# |       |        |       |        |        |  |
| SAVBUF | 2-234# |        |       |        |       |        |        |  |
| SAVSPC | 2-62   | 2-183# |       |        |       |        |        |  |
| SC1    | 2-196# | 9-11*  | 9-18  | 9-19*  | 9-25* | 9-40   |        |  |
| SC2    | 2-197# | 9-18*  | 9-29* | 9-40   |       |        |        |  |
| SELADR | 2-64   | 2-177# |       |        |       |        |        |  |
| SETGT  | 7-26   | 7-40   | 8-15  | 8-41   | 8-53# | 9-28   |        |  |
| SETLT  | 7-25   | 7-39   | 8-7   | 8-33   | 8-58# | 9-15   |        |  |
| SETNRT | 2-67   | 3-28   | 3-50  | 3-96   | 3-104 | 3-109  | 4-15#  |  |
| SEVERR | 1-55#  |        |       |        |       |        |        |  |
| SFLAGS | 2-45   | 2-131# | 3-32  | 3-43   |       |        |        |  |
| SINCHR | 2-210# | 9-8    | 9-14  | 9-27   |       |        |        |  |
| SKPCNT | 2-42   | 2-151# |       |        |       |        |        |  |
| SKPVAL | 2-52   | 2-152# |       |        |       |        |        |  |
| SORTP1 | 2-71   | 3-12   |       |        |       |        |        |  |
| SORTP2 | 2-71   | 3-114  |       |        |       |        |        |  |
| SPC1   | 2-245# | 2-256  | 3-50  | 3-96   |       |        |        |  |
| SPC2   | 2-246# | 2-256  | 3-104 |        |       |        |        |  |
| SPC3   | 2-247# | 2-256  | 3-28  | 3-109  |       |        |        |  |







|        |        |       |       |       |       |       |      |      |      |      |      |       |       |       |
|--------|--------|-------|-------|-------|-------|-------|------|------|------|------|------|-------|-------|-------|
| .WRITE | 1-101# | 5-45  |       |       |       |       |      |      |      |      |      |       |       |       |
| .WRITW | 1-101# |       |       |       |       |       |      |      |      |      |      |       |       |       |
| POP    | 1-118# | 3-17  | 3-21  | 3-28  | 3-28  | 3-46  | 3-46 | 3-50 | 3-50 | 3-96 | 3-96 | 3-104 | 3-104 | 3-109 |
|        | 3-107  | 5-42  | 5-45  | 5-45  |       |       |      |      |      |      |      |       |       |       |
| PUSH   | 1-112# | 3-17  | 3-21  | 3-28  | 3-28  | 3-46  | 3-46 | 3-50 | 3-50 | 3-96 | 3-96 | 3-104 | 3-104 | 3-109 |
|        | 3-107  | 5-42  | 5-45  | 5-45  |       |       |      |      |      |      |      |       |       |       |
| SETNRT | 1-210# | 3-28  | 3-50  | 3-96  | 3-104 | 3-109 |      |      |      |      |      |       |       |       |
| XXX    | 2-261# | 3-101 | 3-102 | 3-106 | 5-44  | 6-27  |      |      |      |      |      |       |       |       |