```
   SSSSSSSSSSS     000000000     RRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333       222222222
   SSSSSSSSSSS     000000000     RRRRRRRRRRRR    TTTTTTTTTTTTTTT     333333333       222222222
   SSSSSSSSSSS     000000000     RRRRRRRRRRRR    TTTTTTTTTTTTTTT     333333333       222222222
SSS              000          000  RRR         RRR        TTT        333          333  222          222
SSS              000          000  RRR         RRR        TTT        333          333  222          222
SSS              000          000  RRR         RRR        TTT        333          333  222          222
SSS              000          000  RRR         RRR        TTT                     333                222
SSS              000          000  RRR         RRR        TTT                     333                222
SSS              000          000  RRR         RRR        TTT                     333                222
   SSSSSSSSS     000          000  RRRRRRRRRRR            TTT                333                   222
   SSSSSSSSS     000          000  RRRRRRRRRRRR           TTT                333                   222
   SSSSSSSSS     000          000  RRRRRRRRRRRR           TTT                333                   222
            SSS  000          000  RRR   RRR              TTT                   333             222
            SSS  000          000  RRR   RRR              TTT                   333             222
            SSS  000          000  RRR   RRR              TTT                   333             222
            SSS  000          000  RRR      RRR           TTT        333        333          222
            SSS  000          000  RRR      RRR           TTT        333        333          222
            SSS  000          000  RRR      RRR           TTT        333        333          222
            SSS  000          000  RRR      RRR           TTT        333        333          222
SSSSSSSSSSS          000000000     RRR         RRR        TTT           333333333       222222222222222
SSSSSSSSSSS          000000000     RRR         RRR        TTT           333333333       222222222222222
SSSSSSSSSSS          000000000     RRR         RRR        TTT           333333333       222222222222222
```

FILEID**SORFILNAM

```
  SSSSSSSS    000000    RRRRRRRR    FFFFFFFFFF    IIIIII    LL          NN      NN    AAAAAA    MM      MM
  SSSSSSSS    000000    RRRRRRRR    FFFFFFFFFF    IIIIII    LL          NN      NN    AAAAAA    MM      MM
SS           00    00   RR     RR   FF            II        LL          NN      NN   AA    AA   MMMM  MMMM
SS           00    00   RR     RR   FF            II        LL          NN      NN   AA    AA   MMMM  MMMM
SS           00    00   RR     RR   FF            II        LL          NNNN    NN   AA    AA   MM  MM  MM
SS           00    00   RR     RR   FF            II        LL          NNNN    NN   AA    AA   MM  MM  MM
  SSSSSS     00    00   RRRRRRRR    FFFFFFF       II        LL          NN  NN  NN   AA    AA   MM      MM
  SSSSSS     00    00   RRRRRRRR    FFFFFFF       II        LL          NN  NN  NN   AA    AA   MM      MM
        SS   00    00   RR   RR     FF            II        LL          NN    NNNN   AAAAAAAAAA   MM      MM
        SS   00    00   RR   RR     FF            II        LL          NN    NNNN   AAAAAAAAAA   MM      MM
        SS   00    00   RR    RR    FF            II        LL          NN      NN   AA    AA   MM      MM      ....
        SS   00    00   RR    RR    FF            II        LL          NN      NN   AA    AA   MM      MM      ....
SSSSSSSS      000000    RR     RR   FF          IIIIII      LLLLLLLLLL  NN      NN   AA    AA   MM      MM      ....
SSSSSSSS      000000    RR     RR   FF          IIIIII      LLLLLLLLLL  NN      NN   AA    AA   MM      MM      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE SOR$FILE_NAME (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1
;     4          0004  1 !*****************************************************************************
;     5          0005  1 !*                                                                           *
;     6          0006  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     7          0007  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;     8          0008  1 !*  ALL RIGHTS RESERVED.                                                      *
;     9          0009  1 !*                                                                           *
;    10          0010  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    11          0011  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE           *
;    12          0012  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER      *
;    13          0013  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
;    14          0014  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY       *
;    15          0015  1 !*  TRANSFERRED.                                                              *
;    16          0016  1 !*                                                                           *
;    17          0017  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE     *
;    18          0018  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT         *
;    19          0019  1 !*  CORPORATION.                                                              *
;    20          0020  1 !*                                                                           *
;    21          0021  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS      *
;    22          0022  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*****************************************************************************
;    26          0026  1 !
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 !
;    30          0030  1 ! FACILITY:     VAX-11 SORT/MERGE
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This module contains routines to copy file names.
;    35          0035  1 !
;    36          0036  1 ! ENVIRONMENT:  VAX/VMS user mode
;    37          0037  1 !
;    38          0038  1 ! AUTHOR: Peter D Gilbert, CREATION DATE: 14-Oct-1982
;    39          0039  1 !
;    40          0040  1 ! MODIFIED BY:
;    41          0041  1 !
;    42          0042  1 !       T03-015         Original
;    43          0043  1 !       T03-016 Changes for hostile environment. PDG 3-Feb-1983
;    44          0044  1 !--
```

```
;    46          0045  1 LIBRARY 'SYS$LIBRARY:STARLET';
;    47          0046  1 REQUIRE 'SRC$:COM';
;    48          0116  1
;    49          0117  1 FORWARD ROUTINE
;    50          0118  1         SOR$$COPY_FILE_NAME:    CAL_CTXREG NOVALUE,
;    51          0119  1         SOR$$BEST_FILE_NAME:    CAL_CTXREG NOVALUE,
;    52          0120  1         SOR$$FREE_FILE_NAME:    CAL_CTXREG NOVALUE;
;    53          0121  1 EXTERNAL ROUTINE
;    54          0122  1         SOR$$ALLOCATE:          CAL_CTXREG,             ! Allocate storage
;    55          0123  1         SOR$$DEALLOCATE:        CAL_CTXREG NOVALUE,     ! Deallocate storage
;    56          0124  1         SOR$$ERROR;
```

```
;    58          0125  1 GLOBAL ROUTINE SOR$$BEST_FILE_NAME
;    59          0126  1         (
;    60          0127  1         FAB:    REF $FAB_DECL,
;    61          0128  1         RES:    REF VECTOR[2]
;    62          0129  1         ): CAL_CTXREG NOVALUE =
;    63          0130  1 !++
;    64          0131  1 !
;    65          0132  1 ! FUNCTIONAL DESCRIPTION:
;    66          0133  1 !
;    67          0134  1 !       This routine determines the best file name available after an OPEN
;    68          0135  1 !       or CREATE, and stores the result string into a length/address vector.
;    69          0136  1 !
;    70          0137  1 ! FORMAL PARAMETERS:
;    71          0138  1 !
;    72          0139  1 !       FAB     Pointer to FAB
;    73          0140  1 !       RES     Length/address of the string
;    74          0141  1 !
;    75          0142  1 ! IMPLICIT INPUTS:
;    76          0143  1 !
;    77          0144  1 !       NONE
;    78          0145  1 !
;    79          0146  1 ! IMPLICIT OUTPUTS:
;    80          0147  1 !
;    81          0148  1 !       NONE
;    82          0149  1 !
;    83          0150  1 ! ROUTINE VALUE:
;    84          0151  1 !
;    85          0152  1 !       NONE
;    86          0153  1 !
;    87          0154  1 !--
;    88          0155  2     BEGIN
;    89          0156  2     EXTERNAL REGISTER
;    90          0157  2         CTX =   COM_REG_CTX:    REF CTX_BLOCK;
;    91          0158  2     BIND
;    92          0159  2         NAM = .FAB[FAB$L_NAM]:  $NAM_DECL;
;    93          0160  2     LOCAL
;    94          0161  2         LEN,
;    95          0162  2         ADR;
;    96          0163  2
;    97          0164  2     ! Get the length/address of the best available file name string
;    98          0165  2     !
;    99          0166  2     ADR = .NAM[NAM$L_RSA];
;   100          0167  2     IF (LEN = .NAM[NAM$B_RSL]) NEQ 0
;   101          0168  2     THEN
;   102          0169  2         0
;   103          0170  2     ELIF (LEN = .NAM[NAM$B_ESL]) NEQ 0
;   104          0171  2     THEN
;   105          0172  2         0
;   106          0173  2     ELSE
;   107          0174  3         BEGIN
;   108          0175  3         LEN = .FAB[FAB$B_FNS];
;   109          0176  3         ADR = .FAB[FAB$L_FNA];
;   110          0177  2         END;
;   111          0178  2
;   112          0179  2     IF .LEN NEQ .RES[0]
;   113          0180  2     THEN
;   114          0181  3         BEGIN
```

```
;  115          0182  3
;  116          0183  3          ! Free the old string
;  117          0184  3          !
;  118          0185  3          SOR$$DEALLOCATE(.RES[0], RES[1]);
;  119          0186  3
;  120          0187  3          ! Allocate space for the new string
;  121          0188  3          !
;  122          0189  3          RES[0] = .LEN;
;  123          0190  3          RES[1] = SOR$$ALLOCATE(.LEN);
;  124          0191  2          END;
;  125          0192  2
;  126          0193  2      ! Move the string
;  127          0194  2      !
;  128          0195  2      CH$MOVE(.LEN, .ADR, .RES[1]);
;  129          0196  2
;  130          0197  1      END;


                                                        .TITLE  SOR$FILE_NAME
                                                        .IDENT  \V04-000\

                                                        .EXTRN  SOR$$ALLOCATE, SOR$$DEALLOCATE
                                                        .EXTRN  SOR$$ERROR

                                                        .PSECT  SOR$RO_CODE,NOWRT,  SHR,  PIC,2

                                        003C 00000      .ENTRY  SOR$$BEST_FILE_NAME, Save R2,R3,R4,R5       ; 0125
                    51      04  AC      D0 00002        MOVL    FAB, R1                                     ; 0159
                    50      28  A1      D0 00006        MOVL    40(R1), R0                                  ;
                    54      04  A0      D0 0000A        MOVL    4(R0), ADR                                  ; 0166
                    53      03  A0      9A 0000E        MOVZBL  3(R0), LEN                                  ; 0167
                                0E      12 00012        BNEQ    1$                                          ;
                    53      0B  A0      9A 00014        MOVZBL  11(R0), LEN                                 ; 0170
                                08      12 00018        BNEQ    1$                                          ;
                    53      34  A1      9A 0001A        MOVZBL  52(R1), LEN                                 ; 0175
                    54      2C  A1      D0 0001E        MOVL    44(R1), ADR                                 ; 0176
                    52      08  AC      D0 00022 1$:    MOVL    RES, R2                                     ; 0179
                    62              53  D1 00026        CMPL    LEN, (R2)                                   ;
                                1C      13 00029        BEQL    2$                                          ;
                            04  A2      9F 0002B        PUSHAB  4(R2)                                       ; 0185
                                62      DD 0002E        PUSHL   (R2)                                        ;
                00000000G       00      02  FB 00030   CALLS   #2, SOR$$DEALLOCATE                        ;
                    62              53  D0 00037        MOVL    LEN, (R2)                                   ; 0189
                                53      DD 0003A        PUSHL   LEN                                         ; 0190
                00000000G       00      01  FB 0003C   CALLS   #1, SOR$$ALLOCATE                          ;
                        04      A2      50  D0 00043    MOVL    R0, 4(R2)                                   ;
            04  B2                  64      53  28 00047 2$:    MOVC3   LEN, (ADR), @4(R2)                  ; 0195
                                                04 0004C    RET                                         ; 0197

; Routine Size:  77 bytes,    Routine Base:  SOR$RO_CODE + 0000
```

```
;  132          0198  1 GLOBAL ROUTINE SOR$$COPY_FILE_NAME
;  133          0199  1         (
;  134          0200  1         DESC:   REF BLOCK[,BYTE],
;  135          0201  1         RES:    REF VECTOR[2]
;  136          0202  1         ): CAL_CTXREG NOVALUE =
;  137          0203  1 !++
;  138          0204  1 !
;  139          0205  1 ! FUNCTIONAL DESCRIPTION:
;  140          0206  1 !
;  141          0207  1 !       This routine copies a file name (as passed by a user)
;  142          0208  1 !       to a length/address vector.  Note that various descriptor classes
;  143          0209  1 !       are supported (via ANALYZE_SDESC).
;  144          0210  1 !
;  145          0211  1 ! FORMAL PARAMETERS:
;  146          0212  1 !
;  147          0213  1 !       DESC    String descriptor
;  148          0214  1 !       RES     Length/address of the string
;  149          0215  1 !
;  150          0216  1 ! IMPLICIT INPUTS:
;  151          0217  1 !
;  152          0218  1 !       NONE
;  153          0219  1 !
;  154          0220  1 ! IMPLICIT OUTPUTS:
;  155          0221  1 !
;  156          0222  1 !       NONE
;  157          0223  1 !
;  158          0224  1 ! ROUTINE VALUE:
;  159          0225  1 !
;  160          0226  1 !       NONE
;  161          0227  1 !
;  162          0228  1 !--
;  163          0229  2     BEGIN
;  164          0230  2     EXTERNAL REGISTER
;  165          0231  2         CTX =   COM_REG_CTX:    REF CTX_BLOCK;
;  166          0232  2     LOCAL
;  167          0233  2         LEN,
;  168          0234  2         ADR,
;  169          0235  2         STATUS;
;  170          0236  2
;  171          0237  2     LEN = 0;
;  172        L 0238  2     %IF HOSTILE
;  173        U 0239  2     %THEN
;  174        U 0240  2         LEN = .DESC[DSC$W_LENGTH];
;  175        U 0241  2         ADR = .DESC[DSC$A_POINTER];
;  176          0242  2     %ELSE
;  177          0243  3         BEGIN
;  178          0244  3         EXTERNAL ROUTINE
;  179          0245  3             LIB$ANALYZE_SDESC:  ADDRESSING_MODE(GENERAL);
;  180          0246  3         STATUS = LIB$ANALYZE_SDESC(DESC[BASE_], LEN, ADR);
;  181          0247  3         IF NOT .STATUS THEN SOR$$ERROR(SOR$_SHR_SYSERROR, 0, .STATUS);
;  182          0248  2         END;
;  183          0249  2     %FI
;  184          0250  2
;  185          0251  2     RES[0] = .LEN;
;  186          0252  2     RES[1] = SOR$$ALLOCATE(.LEN);
;  187          0253  2
;  188          0254  2     CH$MOVE(.LEN, .ADR, .RES[1]);
```

```
;  189          0255  1     END;


                                                                .EXTRN  LIB$ANALYZE_SDESC

                                                    003C 00000  .ENTRY  SOR$$COPY_FILE_NAME, Save R2,R3,R4,R5      ; 0198
                                 5E                 08  C2 00002  SUBL2  #8, SP                                   ;
                                          04        AE  D4 00005  CLRL   LEN                                      ; 0237
                                                    5E  DD 00008  PUSHL  SP                                       ; 0246
                                          08        AE  9F 0000A  PUSHAB LEN                                      ;
                                          04        AC  DD 0000D  PUSHL  DESC                                     ;
                          00000000G       00        03  FB 00010  CALLS  #3, LIB$ANALYZE_SDESC                     ;
                                          11        50  E8 00017  BLBS   STATUS, 1$                               ; 0247
                                                    50  DD 0001A  PUSHL  STATUS                                   ;
                                                    7E  D4 0001C  CLRL   -(SP)                                    ;
                                       001C11B4     8F  DD 0001E  PUSHL  #1839540                                 ;
                          00000000G       00        03  FB 00024  CALLS  #3, SOR$$ERROR                          ;
                                          52        08  AC  D0 0002B 1$:  MOVL   RES, R2                          ; 0251
                                          62        04  AE  D0 0002F  MOVL   LEN, (R2)                           ;
                                                    04  AE  DD 00033  PUSHL  LEN                                 ; 0252
                          00000000G       00        01  FB 00036  CALLS  #1, SOR$$ALLOCATE                       ;
                                04        A2        50  D0 0003D  MOVL   R0, 4(R2)                               ;
                   04   B2        00    BE        04  AE  28 00041  MOVC3  LEN, @ADR, @4(R2)                     ; 0254
                                                       04 00048  RET                                              ; 0255

; Routine Size:  73 bytes,    Routine Base:  SOR$RO_CODE + 004D
```

```
;  191          0256  1 GLOBAL ROUTINE SOR$$FREE_FILE_NAME
;  192          0257  1         (
;  193          0258  1         RES:    REF VECTOR[2]
;  194          0259  1         ): CAL_CTXREG NOVALUE =
;  195          0260  1 !++
;  196          0261  1 !
;  197          0262  1 ! FUNCTIONAL DESCRIPTION:
;  198          0263  1 !
;  199          0264  1 !       This routine releases the storage used to hold a file name string.
;  200          0265  1 !
;  201          0266  1 ! FORMAL PARAMETERS:
;  202          0267  1 !
;  203          0268  1 !       RES     Length/address of the string
;  204          0269  1 !
;  205          0270  1 ! IMPLICIT INPUTS:
;  206          0271  1 !
;  207          0272  1 !       NONE
;  208          0273  1 !
;  209          0274  1 ! IMPLICIT OUTPUTS:
;  210          0275  1 !
;  211          0276  1 !       NONE
;  212          0277  1 !
;  213          0278  1 ! ROUTINE VALUE:
;  214          0279  1 !
;  215          0280  1 !       NONE
;  216          0281  1 !
;  217          0282  1 !--
;  218          0283  2     BEGIN
;  219          0284  2     EXTERNAL REGISTER
;  220          0285  2         CTX =   COM_REG_CTX:    REF CTX_BLOCK;
;  221          0286  2
;  222          0287  2     SOR$$DEALLOCATE(.RES[0], RES[1]);
;  223          0288  1     END;


                                                0000 00000          .ENTRY  SOR$$FREE_FILE_NAME, Save nothing       ; 0256
                        7E      04    AC              04   C1 00002  ADDL3   #4, RES, -(SP)                         ; 0287
                                                  04    BC   DD 00007  PUSHL   @RES
                            00000000G   00              02   FB 0000A  CALLS   #2, SOR$$DEALLOCATE
                                                            04 00011  RET                                           ; 0288

; Routine Size:  18 bytes,    Routine Base:  SOR$RO_CODE + 0096

;  224          0289  1
;  225          0290  1 END
;  226          0291  0 ELUDOM


;                                       PSECT SUMMARY
```

```
;
;       Name                    Bytes                   Attributes
;
;  SOR$RO_CODE                   168  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;
;
;                           Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776        11         0       581         00:01.0
;  _$255$DUA28:[SORT32.SRC]SORLIB.L32;1    409       105        25        34         00:00.3
;
;
;
;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:SORFILNAM/OBJ=OBJ$:SORFILNAM MSRC$:SORFILNAM/UPDATE=(ENH$:SORFILNAM
;       )
;
; Size:         168 code + 0 data bytes
; Run Time:       00:05.7
; Elapsed Time:   00:24.8
; Lines/CPU Min:   3068
; Lexemes/CPU-Min: 12590
; Memory Used:  71 pages
; Compilation Complete
```

SORINTERF
LIS

SORKEYSUB
LIS

SORFILNAM
LIS

SORCOMPAR
LIS

SORENTRY
LIS