```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR      SSSSSSSS  RRRRRRRR      CCCCCCCC  HH      HH    IIIIII    NN      NN
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR      SSSSSSSS  RRRRRRRR      CCCCCCCC  HH      HH    IIIIII    NN      NN
SS              TT      RR      RR  SS          RR      RR  CC          HH      HH      II      NN      NN
SS              TT      RR      RR  SS          RR      RR  CC          HH      HH      II      NN      NN
SS              TT      RR      RR  SS          RR      RR  CC          HH      HH      II      NNNN    NN
SS              TT      RR      RR  SS          RR      RR  CC          HH      HH      II      NNNN    NN
  SSSSSS        TT      RRRRRRRR      SSSSSS    RRRRRRRR    CC          HHHHHHHHHH      II      NN  NN  NN
  SSSSSS        TT      RRRRRRRR      SSSSSS    RRRRRRRR    CC          HHHHHHHHHH      II      NN  NN  NN
        SS      TT      RR  RR              SS  RR  RR      CC          HH      HH      II      NN    NNNN
        SS      TT      RR  RR              SS  RR  RR      CC          HH      HH      II      NN    NNNN
        SS      TT      RR    RR            SS  RR    RR    CC          HH      HH      II      NN      NN      ....
        SS      TT      RR    RR            SS  RR    RR    CC          HH      HH      II      NN      NN      ....
SSSSSSSS        TT      RR      RR  SSSSSSSS  RR      RR    CCCCCCCC  HH      HH    IIIIII    NN      NN      ....
SSSSSSSS        TT      RR      RR  SSSSSSSS  RR      RR    CCCCCCCC  HH      HH    IIIIII    NN      NN      ....

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 STR$$SRCH_INTLK (                      ! Search the string interlock queue
;     2          0002  0                       IDENT = '1-001'          ! File: STRSRCHIN.B32
;     3          0003  0                       ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**********************************************************************
;     7          0007  1 !*                                                                    *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                             *
;    11          0011  1 !*                                                                    *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    17          0017  1 !*  TRANSFERRED.                                                       *
;    18          0018  1 !*                                                                    *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    21          0021  1 !*  CORPORATION.                                                       *
;    22          0022  1 !*                                                                    *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    25          0025  1 !*                                                                    *
;    26          0026  1 !*                                                                    *
;    27          0027  1 !**********************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY: String Library
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the code and data base to support
;    36          0036  1 !       the string interlock macros, used in the string facility.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT: User mode, AST reentrant.
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 29-OCT-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  JBS 29-OCT-1979
;    45          0045  1 !--
;    46          0046  1
;    47          0047  1 !<BLF/PAGE>
```

```
;    49          0048  1 !
;    50          0049  1 ! SWITCHES:
;    51          0050  1 !
;    52          0051  1
;    53          0052  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    54          0053  1
;    55          0054  1 !
;    56          0055  1 ! LINKAGES:
;    57          0056  1 !
;    58          0057  1 !       NONE
;    59          0058  1 !
;    60          0059  1 ! TABLE OF CONTENTS:
;    61          0060  1 !
;    62          0061  1
;    63          0062  1 FORWARD ROUTINE
;    64          0063  1     STR$$SRCH_INTLK;                        ! Search the string interlock queue
;    65          0064  1
;    66          0065  1 !
;    67          0066  1 ! INCLUDE FILES:
;    68          0067  1 !
;    69          0068  1
;    70          0069  1 REQUIRE 'RTLIN:RTLPSECT';                   ! DECLARE_PSECTS macro
;    71          0164  1
;    72          0165  1 LIBRARY 'RTLSTARLE';                        ! System symbols
;    73          0166  1
;    74          0167  1 !
;    75          0168  1 ! EQUATED SYMBOLS:
;    76          0169  1 !
;    77          0170  1 !       NONE
;    78          0171  1 !
;    79          0172  1 ! PSECTS
;    80          0173  1 !
;    81          0174  1 DECLARE_PSECTS (STR);                       ! Define psects
;    82          0175  1 !
;    83          0176  1 ! OWN AND GLOBAL STORAGE:
;    84          0177  1 !
;    85          0178  1
;    86          0179  1 GLOBAL
;    87          0180  1     STR$$Q_INTLK : VECTOR [2, LONG];        ! Root of interlock queue
;    88          0181  1
;    89          0182  1 !
;    90          0183  1 ! EXTERNAL REFERENCES:
;    91          0184  1 !
;    92          0185  1 !+
;    93          0186  1 ! The following are the error codes used in this module:
;    94          0187  1 !-
;    95          0188  1
;    96          0189  1 EXTERNAL LITERAL
;    97          0190  1     STR$_STRIS_INT,                         ! String is interlocked
;    98          0191  1     STR$_FATINTERR;                         ! Fatal internal error
;    99          0192  1
```

```
;  101          0193  1 GLOBAL ROUTINE STR$$SRCH_INTLK (                      ! Search the string interlock queue
;  102          0194  1         OUR_STRING                                    ! String we are searching for
;  103          0195  1     ) =
;  104          0196  1
;  105          0197  1 !++
;  106          0198  1 ! FUNCTIONAL DESCRIPTION:
;  107          0199  1 !
;  108          0200  1 !       If the INSQUE used to place a string in the string interlock
;  109          0201  1 !       queue indicates that there is at least one string already in
;  110          0202  1 !       the queue, this routine is called to see if the string is
;  111          0203  1 !       already interlocked.  If it is, this routine returns
;  112          0204  1 !       STR$_STRIS_INT, which can be passed to LIB$STOP.  If not,
;  113          0205  1 !       this routine returns SS$_NORMAL.
;  114          0206  1 !
;  115          0207  1 ! CALLING SEQUENCE:
;  116          0208  1 !
;  117          0209  1 !       status.wlc.v = STR$$SRCH_INTLK (our_string.rt.dx)
;  118          0210  1 !
;  119          0211  1 ! FORMAL PARAMETERS:
;  120          0212  1 !
;  121          0213  1 !       our_string      The string we are searching for on the string
;  122          0214  1 !                       interlock queue.
;  123          0215  1 !
;  124          0216  1 ! IMPLICIT INPUTS:
;  125          0217  1 !
;  126          0218  1 !       STR$$Q_INTLK    The head of the string interlock queue.
;  127          0219  1 !
;  128          0220  1 ! IMPLICIT OUTPUTS:
;  129          0221  1 !
;  130          0222  1 !       NONE
;  131          0223  1 !
;  132          0224  1 ! COMPLETION CODES:
;  133          0225  1 !
;  134          0226  1 !       SS$_NORMAL      - Successful completion, string not interlocked.
;  135          0227  1 !       STR$_STRIS_INT  - String is interlocked.
;  136          0228  1 !       STR$_FATINTERR  - The string queue is messed up
;  137          0229  1 !
;  138          0230  1 ! SIDE EFFECTS:
;  139          0231  1 !
;  140          0232  1 !       NONE
;  141          0233  1 !
;  142          0234  1 !--
;  143          0235  1
;  144          0236  2     BEGIN
;  145          0237  2
;  146          0238  2     LOCAL
;  147          0239  2         HIT_COUNT,
;  148          0240  2         THIS_STRING : REF VECTOR [3, LONG];     ! Interlock block on string queue
;  149          0241  2
;  150          0242  2 !+
;  151          0243  2 ! Search the string interlock queue, looking for our string.
;  152          0244  2 ! Even though we do not disable ASTs we cannot be led astray by one,
;  153          0245  2 ! because the discipline of the string interlock queue requires that
;  154          0246  2 ! a routine remove anything it adds to the queue, so an AST routine
;  155          0247  2 ! will leave the queue the same as when it started.
;  156          0248  2 !-
;  157          0249  2     HIT_COUNT = 0;
```

```
;     0250  2       THIS_STRING = .STR$$Q_INTLK [0];
;     0251  2
;     0252  2       WHILE (.THIS_STRING NEQA STR$$Q_INTLK [0]) DO
;     0253  3           BEGIN
;     0254  3
;     0255  3           IF (.THIS_STRING [2] EQLA .OUR_STRING) THEN HIT_COUNT = .HIT_COUNT + 1;
;     0256  3
;     0257  3           THIS_STRING = .THIS_STRING [0];
;     0258  2           END;
;     0259  2
;     0260  2   !+
;     0261  2   ! The "hit count" will be 1 if the string is in the queue exactly once,
;     0262  2   ! which is what we expect if the string is not interlocked, since it was
;     0263  2   ! put in the queue before we were called.  If the "hit count" is 2 or
;     0264  2   ! greater then the string is interlocked.  The count can be greater than
;     0265  2   ! 2 if the string is in use several times in a read-only context.
;     0266  2   ! If the "hit count" is 0 we have an error.
;     0267  2   !-
;     0268  2
;     0269  3       IF (.HIT_COUNT EQL 1)
;     0270  2       THEN
;     0271  3           RETURN (SS$_NORMAL)
;     0272  2       ELSE
;     0273  2
;     0274  2           IF (.HIT_COUNT GTR 1) THEN RETURN (STR$_STRIS_INT) ELSE RETURN (STR$_FATINTERR);
;     0275  2
;     0276  1       END;                                        ! End of routine STR$$SRCH_INTLK

                                                    .TITLE  STR$$SRCH_INTLK
                                                    .IDENT  \1-001\

                                                    .PSECT  _STR$DATA,NOEXE,  PIC,2

                                 00000 STR$$Q_INTLK::
                                                    .BLKB   8

                                                    .EXTRN  STR$_STRIS_INT, STR$_FATINTERR

                                                    .PSECT  _STR$CODE,NOWRT,  SHR,  PIC,2

                           000C 00000               .ENTRY  STR$$SRCH_INTLK, Save R2,R3                 ; 0193
            53 00000000'  EF  9E 00002               MOVAB   STR$$Q_INTLK, R3                           ;
                          52  D4 00009               CLRL    HIT_COUNT                                  ; 0249
            50            63  D0 0000B               MOVL    STR$$Q_INTLK, THIS_STRING                  ; 0250
            51            63  9E 0000E 1$:           MOVAB   STR$$Q_INTLK, R1                           ; 0252
            51            50  D1 00011               CMPL    THIS_STRING, R1                            ;
                          0E  13 00014               BEQL    3$                                         ;
      04    AC       08   A0  D1 00016               CMPL    8(THIS_STRING), OUR_STRING                 ; 0255
                          02  12 0001B               BNEQ    2$                                         ;
                          52  D6 0001D               INCL    HIT_COUNT                                  ;
            50            60  D0 0001F 2$:           MOVL    (THIS_STRING), THIS_STRING                 ; 0257
                          EA  11 00022               BRB     1$                                         ; 0252
            01            52  D1 00024 3$:           CMPL    HIT_COUNT, #1                              ; 0269
                          04  12 00027               BNEQ    4$                                         ;
            50            01  D0 00029               MOVL    #1, R0                                     ; 0274
                              04 0002C               RET                                                ;
```

```
                                 08  15 0002D 4$:     BLEQ    5$
                      50 00000000G   8F  D0 0002F      MOVL    #STR$_STRIS_INT, R0                    ;
                                     04 00036          RET                                            ;
                      50 00000000G   8F  D0 00037 5$:  MOVL    #STR$_FATINTERR, R0                    ;
                                     04 0003E          RET                                            ; 0276

; Routine Size:  63 bytes,    Routine Base:  _STR$CODE + 0000

;  185          0277  1
;  186          0278  1 END                                     !End of module STR$$SRCH_INTLK
;  187          0279  1
;  188          0280  0 ELUDOM



;                                PSECT SUMMARY
;
;       Name                     Bytes                          Attributes
;
;  _STR$DATA                        8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _STR$CODE                       63  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         1         0       581         00:00.8



;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:STRSRCHIN/OBJ=OBJ$:STRSRCHIN MSRC$:STRSRCHIN/UPDATE=(ENH$:STRSRCHIN
;       )

; Size:         63 code + 8 data bytes
; Run Time:        00:02.7
; Elapsed Time:    00:17.9
; Lines/CPU Min:    6268
; Lexemes/CPU-Min: 18000
; Memory Used:  42 pages
; Compilation Complete
```

LINKER
LIS

STRPOSIT
LIS

STRUNWDEQ
LIS

STRPOSEXT
LIS

STRREPLAC
LIS

STRSRCHN
LIS

STRRIGHT
LIS

LINKER

STRTRIM
LIS

LINK
MAP

PREFIX
REQ

ISDSORT
LIS

STRUPCASE
LIS

STRTRANSL
LIS

DATBAS
MDL

TIRAUX
REQ

ISGENC
REQ

STRPREFIX
LIS

DATBAS
LIS