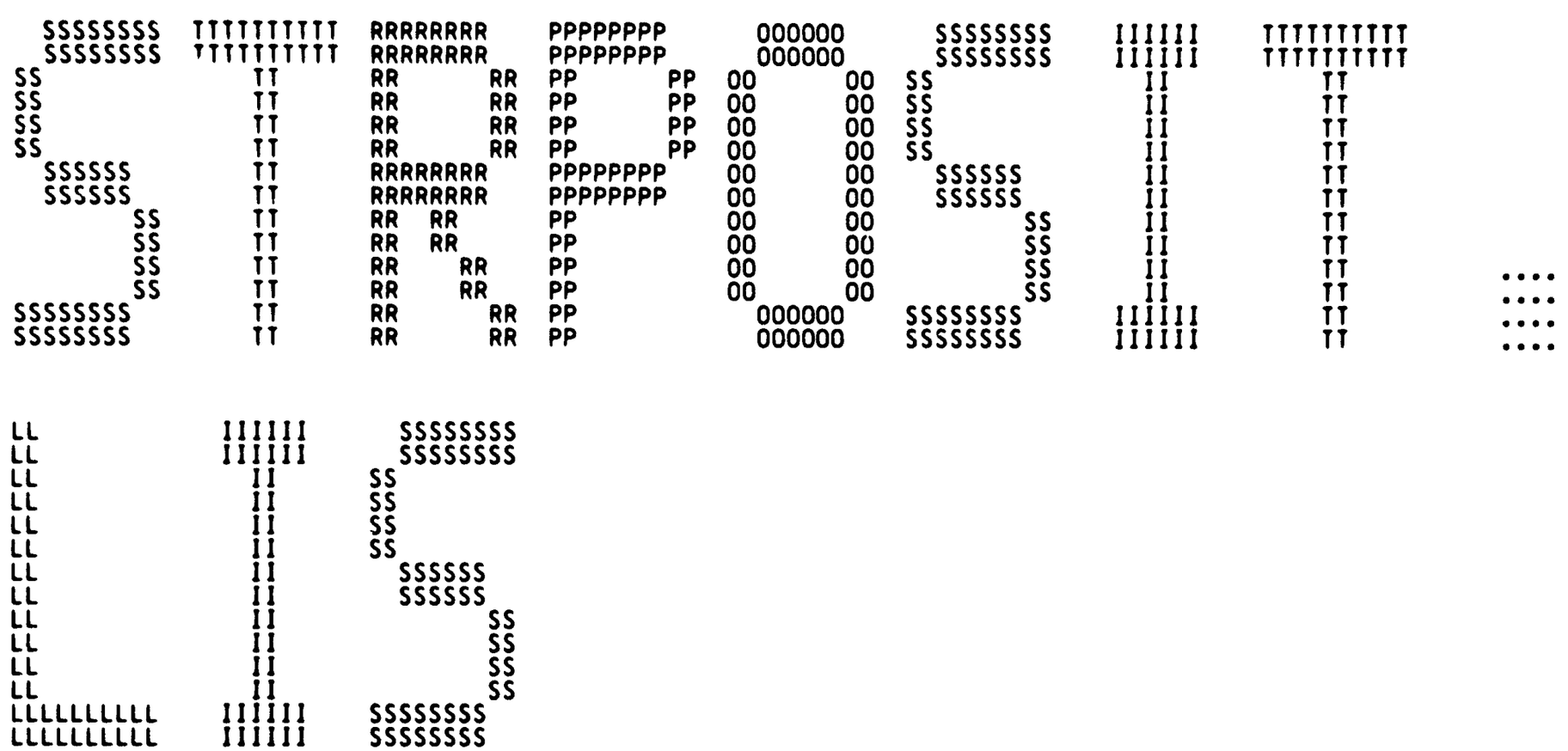
```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        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 PPPPPPPP OOOOOO SSSSSSSS IIIIII TTTTTTTTTT
SSSSSSSS TTTTTTTTTT RRRRRRRR PPPPPPPP OOOOOO SSSSSSSS IIIIII TTTTTTTTTT
SS TT RR RR PP PP OO OO SS II TT
SS TT RR RR PP PP OO OO SS II TT
SS TT RR RR PP PP OO OO SS II TT
SS TT RR RR PP PP OO OO SS II TT
SSSSSS TT RRRRRRRR PPPPPPPP OO OO SSSSSS II TT
SSSSSS TT RRRRRRRR PPPPPPPP OO OO SSSSSS II TT
SS TT RR RR PP OO OO SS II TT
SS TT RR RR PP OO OO SS II TT
SS TT RR RR PP OO OO SS II TT
SS TT RR RR PP OO OO SS II TT
SSSSSSSS TT RR RR PP OOOOOO SSSSSSSS IIIIII TT
SSSSSSSS TT RR RR PP OOOOOO SSSSSSSS IIIIII TT
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$POSITION (           ! Find the position of a substring
;     2          0002  0
;     3          0003  0                 IDENT = '1-014' ! File: STRPOSIT.B32    Edit: RKR1014
;     4          0004  0
;     5          0005  0                     ) =
;     6          0006  1 BEGIN
;     7          0007  1 !
;     8          0008  1 !*****************************************************************************
;     9          0009  1 !*                                                                           *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                      *
;    13          0013  1 !*                                                                           *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    19          0019  1 !*  TRANSFERRED.                                                              *
;    20          0020  1 !*                                                                           *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    23          0023  1 !*  CORPORATION.                                                              *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*****************************************************************************
;    30          0030  1 !
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 ! FACILITY: String support library
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT: This module takes 2 input strings of any supported class and
;    36          0036  1 !       dtype and returns the position of the substring in the main
;    37          0037  1 !       string starting at an input position
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT: User mode, AST level or not or mixed
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: R. Will, CREATION DATE: 10-Mar-79
;    42          0042  1 !
;    43          0043  1 ! MODIFIED BY:
;    44          0044  1 !
;    45          0045  1 ! R. Will, 10-Mar-79 : VERSION 01
;    46          0046  1 ! 1-001 - Original
;    47          0047  1 ! 1-002 - Fix bug with starting pos > searched string length.
;    48          0048  1 !         RW  14-Jul-79
;    49          0049  1 ! 1-003 - Correct a typo in edit 002.  JBS 25-JUL-1979
;    50          0050  1 ! 1-004 - Make POS agree with DEC BASIC Standard Proposal.
;    51          0051  1 !         RW  23-Sept-79
;    52          0052  1 ! 1-005 - Change name to STR$, string cleanup.  RW  1-NOV-79
;    53          0053  1 ! 1-006 - Correct the psect name.  JBS 02-NOV-1979
;    54          0054  1 ! 1-007 - Use the new interlock macros.  JBS 06-NOV-1979
;    55          0055  1 ! 1-008 - Return only after freeing interlocks.  RW  9-Nov-79
;    56          0056  1 ! 1-009 - Use interlocks in Call entry.  RW  15-Nov-79
;    57          0057  1 ! 1-010 - String speedup.  Remove edit 9.  RW  8-Jan-1980
```

```
;    58          0058  1 ! 1-011 - Remove improper declaration and use of ACTUALCOUNT from
;    59          0059  1 !           STR$POSITION_R6.
;    60          0060  1 !           Add comments to indicate argument registers.  SBL 10-Dec-1980
;    61          0061  1 ! 1-012 - Enhance to recognize additional classes of descriptors by
;    62          0062  1 !           using $STR$GET_LEN_ADDR to extract length and address of
;    63          0063  1 !           first data byte from the given descriptors.  Remove string
;    64          0064  1 !           interlocking code.   RKR 21-APR-81
;    65          0065  1 ! 1-013 - Fix bug relating to START_POS.  SBL 28-Sep-1981
;    66          0066  1 ! 1-014 - Speed up code.  RKR 19-OCT-1981.
;    67          0067  1 !--
;    68          0068  1
;    69          0069  1 !<BLF/PAGE>
```

```
;    71          0070  1 !
;    72          0071  1 ! SWITCHES:
;    73          0072  1 !
;    74          0073  1
;    75          0074  1 SWITCHES ADDRESSING_MODE
;    76          0075  1                 (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    77          0076  1
;    78          0077  1 !
;    79          0078  1 ! LINKAGES:
;    80          0079  1 !
;    81          0080  1
;    82          0081  1 REQUIRE 'RTLIN:STRLNK';         ! Use require file with string linkages
;    83          0266  1
;    84          0267  1 !
;    85          0268  1 ! TABLE OF CONTENTS:
;    86          0269  1 !
;    87          0270  1
;    88          0271  1 FORWARD ROUTINE
;    89          0272  1     STR$POSITION,               ! find substring position
;    90          0273  1     STR$POSITION_R6 : STR$JSB_POSIT;    ! find substring position, JSB
;    91          0274  1
;    92          0275  1 !
;    93          0276  1 ! INCLUDE FILES:
;    94          0277  1 !
;    95          0278  1
;    96          0279  1 REQUIRE 'RTLIN:RTLPSECT';       ! Declare PSECTS code
;    97          0374  1
;    98          0375  1 REQUIRE 'RTLIN:STRMACROS';      ! use string macros to write code
;    99          1291  1
;   100          1292  1 LIBRARY 'RTLSTARLE';            ! STARLET library for macros and symbols
;   101          1293  1
;   102          1294  1 !
;   103          1295  1 ! MACROS:
;   104          1296  1 !
;   105          1297  1 !       NONE
;   106          1298  1 !
;   107          1299  1 ! EQUATED SYMBOLS:
;   108          1300  1 !
;   109          1301  1 !       NONE
;   110          1302  1 !
;   111          1303  1 ! PSECT DECLARATIONS
;   112          1304  1 !
;   113          1305  1 DECLARE_PSECTS (STR);
;   114          1306  1 !
;   115          1307  1 ! OWN STORAGE:
;   116          1308  1 !
;   117          1309  1 !       NONE
;   118          1310  1 !
;   119          1311  1 ! EXTERNAL REFERENCES:
;   120          1312  1 !
;   121          1313  1
;   122          1314  1 EXTERNAL LITERAL
;   123          1315  1     STR$_ILLSTRCLA;                         ! Illegal string class
;   124          1316  1
```

```
;  126     1317  1 GLOBAL ROUTINE STR$POSITION (    ! find position of substring
;  127     1318  1
;  128     1319  1         SRC_DESC,                 ! descriptor of string to search
;  129     1320  1         SUB_DESC,                 ! descriptor of string to find
;  130     1321  1         START_POS                 ! value of position to start search
;  131     1322  1
;  132     1323  1                                 ) =
;  133     1324  1
;  134     1325  1 !++
;  135     1326  1 ! FUNCTIONAL DESCRIPTION:
;  136     1327  1 !
;  137     1328  1 !       This routine takes two source strings of any supported
;  138     1329  1 !       dtype and class and finds the position of the substring in the
;  139     1330  1 !       source string starting at the input starting position.  The
;  140     1331  1 !       routine returns the position of the substring in the source
;  141     1332  1 !       string.
;  142     1333  1 !       The first character of the source string is at position 1.
;  143     1334  1 !       The call entry point is implemented by JSBing to the JSB entry.
;  144     1335  1 !
;  145     1336  1 ! FORMAL PARAMETERS:
;  146     1337  1 !
;  147     1338  1 !       SRC_DESC.rt.dx          pointer to descriptor of string to be
;  148     1339  1 !                               searched
;  149     1340  1 !       SUB_DESC.rt.dx          pointer to descriptor of string to find
;  150     1341  1 !       START_POS.rl.r          position in source to begin
;  151     1342  1 !                               search [optional arg, default = 1]
;  152     1343  1 !
;  153     1344  1 ! IMPLICIT INPUTS:
;  154     1345  1 !
;  155     1346  1 !       NONE
;  156     1347  1 !
;  157     1348  1 ! IMPLICIT OUTPUTS:
;  158     1349  1 !
;  159     1350  1 !       NONE
;  160     1351  1 !
;  161     1352  1 ! ROUTINE VALUE:
;  162     1353  1 !
;  163     1354  1 !       FIND_POS.wlu.v          value of start position of substring in
;  164     1355  1 !                               source string.  0 ==> not found.
;  165     1356  1 !
;  166     1357  1 ! SIDE EFFECTS:
;  167     1358  1 !
;  168     1359  1 !       May signal STR$_ILLSTRCLA
;  169     1360  1 !--
;  170     1361  1
;  171     1362  2     BEGIN
;  172     1363  2
;  173     1364  2     BUILTIN
;  174     1365  2         ACTUALCOUNT;
;  175     1366  2
;  176     1367  2     LOCAL
;  177     1368  2         ACTUAL_START;                           ! if no START_POS then 1
;  178     1369  2
;  179     1370  2     MAP
;  180     1371  2         SRC_DESC : REF $STR$DESCRIPTOR,
;  181     1372  2         SUB_DESC : REF $STR$DESCRIPTOR;
;  182     1373  2
```

```
;  183          1374  3     IF (ACTUALCOUNT () EQL 2)
;  184          1375  2         THEN
;  185          1376  2             ACTUAL_START = 1
;  186          1377  2         ELSE
;  187          1378  2             ACTUAL_START = ..START_POS;
;  188          1379  2
;  189          1380  2     RETURN STR$POSITION_R6 ( SRC_DESC [0,0,0,0],
;  190          1381  2                              SUB_DESC [0,0,0,0],
;  191          1382  2                              .ACTUAL_START);
;  192          1383  1     END;                                    !End of STR$POSITION


                                                    .TITLE  STR$POSITION
                                                    .IDENT  \1-014\

                                                    .EXTRN  STR$_ILLSTRCLA

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

                                    007C 00000      .ENTRY  STR$POSITION, Save %2,R3,R4,R5,R6        ; 1317
                            02      6C  91 00002    CMPB    (AP), #2                                 ; 1374
                                    05  12 00005    BNEQ    1$
                            52      01  D0 00007    MOVL    #1, ACTUAL_START                         ; 1376
                                    04  11 0000A    BRB     2$
                            52  0C  BC  D0 0000C 1$: MOVL    @START_POS, ACTUAL_START                 ; 1378
                            50  04  AC  7D 00010 2$: MOVQ    SRC_DESC, R0                             ; 1381
                                0000V   30 00014    BSBW    STR$POSITION_R6
                                    04  00017      RET                                              ; 1383

; Routine Size:  24 bytes,    Routine Base:  _STR$CODE + 0000
```

```
;  194          1384  1 GLOBAL ROUTINE STR$POSITION_R6 (        ! find position of substring
;  195          1385  1
;  196          1386  1         SRC_DESC,                 ! descriptor of string to search
;  197          1387  1         SUB_DESC,                 ! descriptor of string to find
;  198          1388  1         START_POS                 ! value of position to start search
;  199          1389  1
;  200          1390  1                                 ) : STR$JSB_POSIT =
;  201          1391  1
;  202          1392  1 !++
;  203          1393  1 ! FUNCTIONAL DESCRIPTION:
;  204          1394  1 !
;  205          1395  1 !       This routine takes two source strings of any supported
;  206          1396  1 !       dtype and class and finds the position of the substring in the
;  207          1397  1 !       source string starting at the input starting position.
;  208          1398  1 !       The routine returns the position of the substring in the source
;  209          1399  1 !       string.
;  210          1400  1 !       The first character in the source string is at position 1.
;  211          1401  1 !
;  212          1402  1 ! FORMAL PARAMETERS:
;  213          1403  1 !
;  214          1404  1 !       SRC_DESC.rt.dx = R0     pointer to descriptor of string to be
;  215          1405  1 !                               searched
;  216          1406  1 !       SUB_DESC.rt.dx = R1     pointer to descriptor of string to find
;  217          1407  1 !       START_POS.rl.v = R2     value of position in source to begin
;  218          1408  1 !                               search
;  219          1409  1 !
;  220          1410  1 ! IMPLICIT INPUTS:
;  221          1411  1 !
;  222          1412  1 !       NONE
;  223          1413  1 !
;  224          1414  1 ! IMPLICIT OUTPUTS:
;  225          1415  1 !
;  226          1416  1 !       NONE
;  227          1417  1 !
;  228          1418  1 ! ROUTINE VALUE:
;  229          1419  1 !
;  230          1420  1 !       FIND_POS.wlu.v          value of start position of substring in
;  231          1421  1 !                               source string. 0 ==> Not found.
;  232          1422  1 !
;  233          1423  1 ! SIDE EFFECTS:
;  234          1424  1 !
;  235          1425  1 !       may allocate or deallocate dynamic string space
;  236          1426  1 !       may signal STR$_ILLSTRCLA if not an acceptable string class
;  237          1427  1 !
;  238          1428  1 !--
;  239          1429  1
;  240          1430  2     BEGIN
;  241          1431  2
;  242          1432  2     LOCAL
;  243          1433  2         MAIN_IN_LEN,            ! length of main string
;  244          1434  2         MAIN_IN_ADDR,           ! addr of 1st data byte of main string
;  245          1435  2         SUB_IN_LEN,             ! length of substring
;  246          1436  2         SUB_IN_ADDR,            ! addr of 1st data byte of substring
;  247          1437  2         START_OFFSET,           ! Offset into string of start position
;  248          1438  2         FIND_POINTER;           ! position of substring in source string
;  249          1439  2
;  250          1440  2     MAP
```

```
;  251          1441  2          SRC_DESC : REF $STR$DESCRIPTOR,
;  252          1442  2          SUB_DESC : REF $STR$DESCRIPTOR;
;  253          1443  2
;  254          1444  2      !+
;  255          1445  2      ! Set START_OFFSET based on START_POS.  Note that offset is 0-origin.
;  256          1446  2      !-
;  257          1447  2
;  258          1448  2          START_OFFSET = .START_POS - 1;
;  259          1449  2          IF .START_OFFSET LSS 0
;  260          1450  2          THEN
;  261          1451  2              START_OFFSET = 0;
;  262          1452  2
;  263          1453  2      !+
;  264          1454  2      ! Extract the length and address of 1st data byte of strings involved.
;  265          1455  2      ! Signal if we find any errors.
;  266          1456  2      !-
;  267          1457  2          $STR$GET_LEN_ADDR (SRC_DESC, MAIN_IN_LEN, MAIN_IN_ADDR );
;  268          1458  2          $STR$GET_LEN_ADDR ( SUB_DESC, SUB_IN_LEN, SUB_IN_ADDR );
;  269          1459  2
;  270          1460  2      !+
;  271          1461  2      ! Adjust length of source string for starting offset.  Make
;  272          1462  2      ! sure that the length doesn't go negative.
;  273          1463  2      !-
;  274          1464  2
;  275          1465  2          IF .START_OFFSET GTR .MAIN_IN_LEN
;  276          1466  2          THEN
;  277          1467  2              START_OFFSET = .MAIN_IN_LEN;
;  278          1468  2          MAIN_IN_LEN = .MAIN_IN_LEN - .START_OFFSET;
;  279          1469  2
;  280          1470  2      !+
;  281          1471  2      ! Find the address of the substring in the string described by
;  282          1472  2      ! starting at START_OFFSET in the source string
;  283          1473  2      !-
;  284          1474  2          FIND_POINTER = CH$FIND_SUB (
;  285          1475  2              .MAIN_IN_LEN,           ! len of string to search
;  286          1476  2              CH$PLUS (.MAIN_IN_ADDR, .START_OFFSET), ! ptr to string to search
;  287          1477  2              .SUB_IN_LEN,            ! len of str being searched for
;  288          1478  2              .SUB_IN_ADDR);          ! ptr to str being searched for
;  289          1479  2
;  290          1480  2      !+
;  291          1481  2      ! Return 0 if the substring was not found, otherwise return the position
;  292          1482  2      ! of the substring, 1-origin.
;  293          1483  2      !-
;  294          1484  2
;  295          1485  3          RETURN (
;  296          1486  3              IF CH$FAIL (.FIND_POINTER)
;  297          1487  3              THEN
;  298          1488  3                  0
;  299          1489  3              ELSE
;  300          1490  3                  CH$DIFF (.FIND_POINTER, .MAIN_IN_ADDR) + 1
;  301          1491  2                 );
;  302          1492  2
;  303          1493  1          END;                                        !End of STR$POSITION_R6


                                                        .EXTRN  STR$ANALYZE_SDESC_R1
```

```
                                53              51  D0 00000 STR$POSITION_R6::
                                                             MOVL    R1, R3                                          ; 1384
                                02              52  F4 00003           SOBGEQ  START_OFFSET, 1$                    ; 1448
                                                52  D4 00006           CLRL    START_OFFSET                        ; 1451
                                02      03      A0  91 00008 1$:       CMPB    3(SRC_DESC), #2                     ; 1457
                                                09  1A 0000C           BGTRU   2$
                                55              60  3C 0000E           MOVZWL  (SRC_DESC), MAIN_IN_LEN
                                54      04      A0  D0 00011           MOVL    4(SRC_DESC), MAIN_IN_ADDR
                                                0C  11 00015           BRB     3$
                            00000000G           00  16 00017 2$:       JSB     STR$ANALYZE_SDESC_R1
                                55              50  D0 0001D           MOVL    R0, R5
                                54              51  D0 00020           MOVL    R1, R4
                                02      03      A3  91 00023 3$:       CMPB    3(SUB_DESC), #2                     ; 1458
                                                09  1A 00027           BGTRU   4$
                                56              63  3C 00029           MOVZWL  (SUB_DESC), SUB_IN_LEN
                                51      04      A3  D0 0002C           MOVL    4(SUB_DESC), SUB_IN_ADDR
                                                0C  11 00030           BRB     5$
                                50              53  D0 00032 4$:       MOVL    SUB_DESC, R0
                            00000000G           00  16 00035           JSB     STR$ANALYZE_SDESC_R1
                                56              50  D0 0003B           MOVL    R0, R6
                                55              52  D1 0003E 5$:       CMPL    START_OFFSET, MAIN_IN_LEN           ; 1465
                                                03  15 00041           BLEQ    6$
                                52              55  D0 00043           MOVL    MAIN_IN_LEN, START_OFFSET           ; 1467
                                55              52  C2 00046 6$:       SUBL2   START_OFFSET, MAIN_IN_LEN           ; 1468
    6244                55      61              56  39 00049           MATCHC  SUB_IN_LEN, (SUB_IN_ADDR), MAIN_IN_LEN, -   ; 1478
                                                                                   (START_OFFSET)[MAIN_IN_ADDR]
                                                03  13 0004F           BEQL    7$
                                53              56  D0 00051           MOVL    SUB_IN_LEN, R3
                                53              56  C2 00054 7$:       SUBL2   SUB_IN_LEN, R3
                                                04  12 00057           BNEQ    8$                                  ; 1486
                                                53  D4 00059           CLRL    R3
                                                05  11 0005B           BRB     9$
                                        53      54  C2 0005D 8$:       SUBL2   MAIN_IN_ADDR, R3                    ; 1490
                                                53  D6 00060           INCL    R3
                                        50      53  D0 00062 9$:       MOVL    R3, R0                              ; 1485
                                                    05 00065           RSB                                         ; 1493

; Routine Size:  102 bytes,    Routine Base:  _STR$CODE + 0018

;  304          1494  1
;  305          1495  1 END                                                :End of module
;  306          1496  1
;  307          1497  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;  _STR$CODE                        126  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        4          0       581        00:00.8



;                                       COMMAND QUALIFIERS

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

; Size:           126 code + 0 data bytes
; Run Time:          00:06.0
; Elapsed Time:      00:25.9
; Lines/CPU Min:     15095
; Lexemes/CPU-Min:   34255
; Memory Used:  82 pages
; Compilation Complete
```

LINKER
LIS

STRPOSIT
LIS

STRUNWDEQ
LIS

STRPOSEXT
LIS

STRREPLAC
LIS

STRSRCHIN
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