```
DDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBB      UUU         UUU     GGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG    GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG    GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG    GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG          GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG          GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG          GGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUUU    GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUUU    GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUUU    GGGGGGGGG
```

FILEID**SSIU

H 5

```
 SSSSSSSS    SSSSSSSS    IIIIII    UU      UU
 SSSSSSSS    SSSSSSSS    IIIIII    UU      UU
SS          SS             II      UU      UU
SS          SS             II      UU      UU
SS          SS             II      UU      UU
SS          SS             II      UU      UU
 SSSSSS      SSSSSS        II      UU      UU
 SSSSSS      SSSSSS        II      UU      UU
       SS          SS      II      UU      UU
       SS          SS      II      UU      UU
       SS          SS      II      UU      UU      ....
       SS          SS      II      UU      UU      ....
SSSSSSSS    SSSSSSSS     IIIIII    UUUUUUUUUU      ....
SSSSSSSS    SSSSSSSS     IIIIII    UUUUUUUUUU      ....

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 SSIU (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1
;     4          0004  1 !
;     5          0005  1 !*****************************************************************************
;     6          0006  1 !*                                                                           *
;     7          0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;     8          0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;     9          0009  1 !*  ALL RIGHTS RESERVED.                                                     *
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    12          0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    13          0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    14          0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    15          0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    16          0016  1 !*  TRANSFERRED.                                                             *
;    17          0017  1 !*                                                                           *
;    18          0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    19          0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    20          0020  1 !*  CORPORATION.                                                             *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    23          0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*****************************************************************************
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 ! FACILITY:  VAX/VMS System Service Call Monitor
;    30          0030  1 !
;    31          0031  1 ! ABSTRACT:
;    32          0032  1 !
;    33          0033  1 !       This module is the other portion of SSI.B32.  In this module, user
;    34          0034  1 !       declared routine is setup to be called at interception time.
;    35          0035  1 !       Other information is also stored/retrieved by interfacing with this
;    36          0036  1 !       module.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !
;    40          0040  1 !       VAX/VMS operating system
;    41          0041  1 !
;    42          0042  1 ! Author:
;    43          0043  1 !
;    44          0044  1 !          Ping Sager,  19-Sep-1983
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1
;    48          0048  1 !
;    49          0049  1 ! Include files
;    50          0050  1 !
;    51          0051  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    52          0052  1
```

```
;    54          0053  1 !
;    55          0054  1 ! Table of contents
;    56          0055  1 !
;    57          0056  1 FORWARD ROUTINE
;    58          0057  1     ssiu_start,                     ! Main routine
;    59          0058  1     user_cancel_restore,            ! Cancel/Restore user delcared routine
;    60          0059  1     user_setup;                     ! Setup user declared routine
;    61          0060  1
;    62          0061  1
;    63          0062  1 EXTERNAL ROUTINE
;    64          0063  1     sys$qiow : ADDRESSING_MODE (ABSOLUTE);! Base of transfer vector
;    65          0064  1
;    66          0065  1 EXTERNAL LITERAL
;    67          0066  1     issh_vec_length;                ! Length of monitor code (ISSH)
;    68          0067  1
;    69          0068  1 EXTERNAL
;    70          0069  1     ssi_running_flag,               ! Flag set to indicate this program
;    71          0070  1                                     !   is running
;    72          0071  1     issh_data_beg,                  ! Begin data area (template)
;    73          0072  1     issh_data_end,                  ! End data area (template)
;    74          0073  1     issh_prio_mask,                 ! Mask to control the calling of the
;    75          0074  1                                     !  user routines
;    76          0075  1     issh_ctrl_index,                ! User routine index
;    77          0076  1     issh_ctrl_prio_index,           ! User routine index per priority
;    78          0077  1     issh_vec_base;                  ! ISSH base address
;    79          0078  1
```

```
;    81          0079  1 GLOBAL ROUTINE ssiu_start (
;    82          0080  1     USER_SETUP_FLAGS, USER_ADDR, USER_ID, SAVE_MASK) =
;    83          0081  1
;    84          0082  1 !---
;    85          0083  1 !
;    86          0084  1 ! Function:
;    87          0085  1 !
;    88          0086  1 !       This is the main routine of the VAX/VMS to use System Service
;    89          0087  1 !       Monitor.  It calls appropriate actions.
;    90          0088  1 !
;    91          0089  1 ! Inputs:
;    92          0090  1 !
;    93          0091  1 !       USER_SETUP_FLAGS - Set up user declared routine to be called at
;    94          0092  1 !                          interception time, it contains 4 bytes of the
;    95          0093  1 !                          following values:
;    96          0094  1 !
;    97          0095  1 !         SETUP_FLAG     - (0/1) Enable/Disable intercept systerm service for
;    98          0096  1 !                          user declared routine.  When user first time declares
;    99          0097  1 !                          user's routine, SETUP_FLAG must be set to 1.
;   100          0098  1 !
;   101          0099  1 !         USER_PRIO      - (1/2/3/4) Running priority of the declared routine.
;   102          0100  1 !                          Higher priority watches the lower priority.
;   103          0101  1 !
;   104          0102  1 !         USER_MASK      - Enable/Disable user declared routine when intercept
;   105          0103  1 !                          system service is enabled.  Mask has the following
;   106          0104  1 !                          kind of values:
;   107          0105  1 !                            Prio. 1 has 1 bit value,
;   108          0106  1 !                              for example, 1 - enable,
;   109          0107  1 !                                           0 - disable.
;   110          0108  1 !                            Prio. 2 has 2 bits value, for it watches prio. 1
;   111          0109  1 !                              for example, 0 - prio. 1&2 both are inactive,
;   112          0110  1 !                                           1 - prio. 1 active, prio. 2 inactive,
;   113          0111  1 !                                           2 - prio. 1 inactive, prio. 2 active,
;   114          0112  1 !                                           3 - prio. 1&2 both are active.
;   115          0113  1 !                            Prio. 3 has 3 bits value, for it watches prio. 1&2
;   116          0114  1 !                              for example, expand above.
;   117          0115  1 !                            Prio. 4 has 4 bits value, for it watches prio. 1,2&3
;   118          0116  1 !                              for example, expand above.
;   119          0117  1 !
;   120          0118  1 !                          Each priority only sets its own enable/disable bit,
;   121          0119  1 !                          but, higher priority sees more bits values than the
;   122          0120  1 !                          lower priority ones.
;   123          0121  1 !
;   124          0122  1 !         USER_MODE      - (0/1/2/3) The mode of the user declared routine.
;   125          0123  1 !
;   126          0124  1 !       USER_ADDR        - User declared routine address.
;   127          0125  1 !
;   128          0126  1 !       USER_ID          - The address of an identification of the user declared
;   129          0127  1 !                          routine.  (This parameter can be eliminated, explanation
;   130          0128  1 !                          is given in SSI_SETUP, for now, I have used this one as
;   131          0129  1 !                          a handy way to know whether user has declared user's
;   132          0130  1 !                          routine or not).
;   133          0131  1 !
;   134          0132  1 !       SAVE_MASK        - The address of the current state of the user declared
;   135          0133  1 !                          routine, before value is setting by USER_MASK, is
;   136          0134  1 !                          returned to the caller.
;   137          0135  1 !
```

```
;  138          0136  1 ! Outputs:
;  139          0137  1 !
;  140          0138  1 !       Worst status encountered.
;  141          0139  1 !
;  142          0140  1 !---
;  143          0141  1
;  144          0142  2     BEGIN
;  145          0143  2
;  146          0144  2     BUILTIN FP;
;  147          0145  2
;  148          0146  2     MAP
;  149          0147  2         user_setup_flags: VECTOR[,BYTE];        ! Important flag values set to
;  150          0148  2                                                 ! make intercept system service
;  151          0149  2                                                 ! to work
;  152          0150  2
;  153          0151  2     LOCAL
;  154          0152  2         intercept,                              ! Address of saved system vector,
;  155          0153  2                                                 !   data, code area in P0
;  156          0154  2         setup_flag,                             ! SETUP_FLAG from
;  157          0155  2         user_prio,                              ! USER_PRIO   USER_SETUP_FLAGS
;  158          0156  2         user_mask,                              ! USER_PRIO   (1 byte each).
;  159          0157  2         user_mode,                              ! USER_MODE
;  160          0158  2         mask_const,                             ! Calculated mask values for
;  161          0159  2                                                 !   each priority
;  162          0160  2         prio_mask: REF VECTOR[,BYTE],           ! Current mask value after set
;  163          0161  2         status;                                 ! Return status
;  164          0162  2
;  165          0163  2
;  166          0164  2     ! Cannot perform futher if there is no interception setup.
;  167          0165  2     !
;  168          0166  2     intercept = .(SYS$QIOW + sgn$c_sysvecpgs * 512 - 4);
;  169          0167  2     IF .intercept EQL 0 THEN RETURN 1;
;  170          0168  2
;  171          0169  2
;  172          0170  2     ! Indicate this program is running.  We never intercept anything from
;  173          0171  2     ! this program.
;  174          0172  2     !
;  175          0173  2     ssi_running_flag = 1;
;  176          0174  2
;  177          0175  2
;  178          0176  2     ! Get setup values from USER_SETUP_FLAGS.
;  179          0177  2     !
;  180          0178  2     setup_flag = .user_setup_flags[0];
;  181          0179  2     user_prio = .user_setup_flags[1];
;  182          0180  2     user_mask = .user_setup_flags[2];
;  183          0181  2     user_mode = .user_setup_flags[3];
;  184          0182  2     status = 1;
;  185          0183  2
;  186          0184  2
;  187          0185  2     ! Set up user declared routine in the table.  Intercept is setup in
;  188          0186  2     ! SSIK.B32 which must be called first, if user first time declares user's
;  189          0187  2     ! routine, enter user declared routine address in the table, if user
;  190          0188  2     ! has already declared user's routine before, simply place the routine
;  191          0189  2     ! in the table given by USER_ID.
;  192          0190  2     !
;  193          0191  2     IF .setup_flag
;  194          0192  2     THEN
```

```
;  195     0193  3          BEGIN
;  196     0194  3
;  197     0195  3
;  198     0196  3          ! Enter User declared routine into the table.
;  199     0197  3          !
;  200     0198  3          IF .user_addr EQL 0
;  201     0199  3          THEN
;  202     0200  4              BEGIN
;  203     0201  4              ssi_running_flag = 0;
;  204     0202  4              RETURN 0;
;  205     0203  4              END
;  206     0204  3          ELSE
;  207     0205  4              BEGIN
;  208     0206  4              IF ..user_id EQL 0
;  209     0207  4              THEN
;  210     0208  4
;  211     0209  4                  ! Locate a spot in the table for the user routine to enter.
;  212     0210  4                  !
;  213     0211  4                  status = user_setup(.user_id, .user_addr, .user_prio, .user_mode)
;  214     0212  4              ELSE
;  215     0213  4
;  216     0214  4                  ! We know the spot, plung it in.
;  217     0215  4                  !
;  218     0216  4                  status = user_cancel_restore(.setup_flag, .user_id, .user_mode,
;  219     0217  4                                               .user_addr);
;  220     0218  4              END;
;  221     0219  3          END
;  222     0220  3
;  223     0221  3
;  224     0222  3          ! Delete user declared routine from the table.  If user declared routine
;  225     0223  3          ! is no longer active.
;  226     0224  3          !
;  227     0225  3          ELSE
;  228     0226  2              BEGIN
;  229     0227  3              IF ..user_id NEQ 0
;  230     0228  3              THEN
;  231     0229  3                  status = user_cancel_restore(.setup_flag, .user_id, .user_mode, 0);
;  232     0230  3              END;
;  233     0231  2
;  234     0232  2
;  235     0233  2          IF NOT .status
;  236     0234  2          THEN
;  237     0235  2              BEGIN
;  238     0236  3              ssi_running_flag = 0;
;  239     0237  3              RETURN .status;
;  240     0238  3              END;
;  241     0239  2
;  242     0240  2
;  243     0241  2          ! Routine is already declared, set routine enable mask, and return
;  244     0242  2          ! the old one back to the user.  Note: even if the routine is not
;  245     0243  2          ! declared, this piece info. still flows through the program.
;  246     0244  2          !
;  247     0245  2          prio_mask = .intercept + issh_prio_mask - issh_vec_base;
;  248     0246  2
;  249     0247  2
;  250     0248  2          ! 1 - prio. 1, 3 - prio. 2, 7 - prio. 3, 15 - prio. 4).
;  251     0249  2
```

```
;  252     0250  2     !
;  253     0251  2     mask_const = (2 ^ (.user_prio-1)) - 1;
;  254     0252  2
;  255     0253  2
;  256     0254  2     ! Preserved 1 bit value for prio. 1, 2 bits for prio. 2..., and return.
;  257     0255  2     !
;  258     0256  2     .save_mask = ..prio_mask AND .mask_const;
;  259     0257  2
;  260     0258  2
;  261     0259  2     ! Preserve higher bits, through away the bits does not belong, then
;  262     0260  2     ! set its own bits.
;  263     0261  2     !
;  264     0262  2     .prio_mask = (..prio_mask AND (15 - .mask_const)) OR
;  265     0263  2                  (.user_mask AND .mask_const);
;  266     0264  2
;  267     0265  2
;  268     0266  2     ! This program is no longer running.
;  269     0267  2     !
;  270     0268  2     ssi_running_flag = 0;
;  271     0269  2     RETURN ss$_normal;
;  272     0270  2
;  273     0271  1     END;


                                                        .TITLE  SSIU
                                                        .IDENT  \V04-000\

                                                        .EXTRN  SYS$QIOW, ISSH_VEC_LENGTH
                                                        .EXTRN  SSI_RUNNING_FLAG
                                                        .EXTRN  ISSH_DATA_BEG, ISSH_DATA_END
                                                        .EXTRN  ISSH_PRIO_MASK, ISSH_CTRL_INDEX
                                                        .EXTRN  ISSH_CTRL_PRIO_INDEX
                                                        .EXTRN  ISSH_VEC_BASE

                                                        .PSECT  $CODE$,NOWRT,2

                    01FC 00000                          .ENTRY  SSIU_START, Save R2,R3,R4,R5,R6,R7,R8           ; 0079
58      0000G   CF   9E 00002                          MOVAB   SSI_RUNNING_FLAG, R8                            ;
56 00000000G    9F   D0 00007                          MOVL    @#SYS$QIOW+2556, INTERCEPT                      ; 0166
                03   12 0000E                          BNEQ    1$                                              ; 0167
                008C 31 00010                          BRW     8$                                              ;
68              01   D0 00013 1$:                      MOVL    #1, SSI_RUNNING_FLAG                            ; 0173
57          04  AC   9A 00016                          MOVZBL  USER_SETUP_FLAGS, SETUP_FLAG                    ; 0178
52          05  AC   9A 0001A                          MOVZBL  USER_SETUP_FLAGS+1, USER_PRIO                   ; 0179
53          06  AC   9A 0001E                          MOVZBL  USER_SETUP_FLAGS+2, USER_MASK                   ; 0180
55          07  AC   9A 00022                          MOVZBL  USER_SETUP_FLAGS+3, USER_MODE                   ; 0181
50              01   D0 00026                          MOVL    #1, STATUS                                      ; 0182
23              57   E9 00029                          BLBC    SETUP_FLAG, 4$                                  ; 0191
54          08  AC   D0 0002C                          MOVL    USER_ADDR, R4                                   ; 0198
                04   12 00030                          BNEQ    2$                                              ;
                68   D4 00032                          CLRL    SSI_RUNNING_FLAG                                ; 0201
                6D   11 00034                          BRB     9$                                              ; 0202
51          0C  AC   D0 00036 2$:                      MOVL    USER_ID, R1                                     ; 0206
                61   D5 00039                          TSTL    (R1)                                            ;
                0B   12 0003C                          BNEQ    3$                                              ;
                24   BB 0003E                          PUSHR   #^M<R2,R5>                                      ; 0212
                12   BB 00040                          PUSHR   #^M<R1,R4>                                      ;
```

```
                        0000V  CF              04  FB 00042          CALLS   #4, USER_SETUP                                   ;
                                                19  11 00047          BRB     6$                                              ;
                                                54  DD 00049 3$:      PUSHL   R4                                              ; 0218
                                                22  BB 0004B          PUSHR   #^M<R1,R5>                                      ; 0217
                                                0C  11 0004D          BRB     5$                                              ;
                                     0C         BC  D5 0004F 4$:      TSTL    @USER_ID                                        ; 0228
                                                0E  13 00052          BEQL    6$                                              ;
                                                7E  D4 00054          CLRL    -(SP)                                           ; 0230
                                                55  DD 00056          PUSHL   USER_MODE                                       ;
                                     0C         AC  DD 00058          PUSHL   USER_ID                                         ;
                                                57  DD 0005B 5$:      PUSHL   SETUP_FLAG                                      ;
                        0000V  CF               04  FB 0005D          CALLS   #4, USER_CANCEL_RESTORE                         ;
                               03               50  E8 00062 6$:      BLBS    STATUS, 7$                                      ; 0234
                                                68  D4 00065          CLRL    SSI_RUNNING_FLAG                                ; 0237
                                                04  00067            RET                                                     ; 0238
                               51       0000GCF46  9E 00068 7$:      MOVAB   ISSH_PRIO_MASK[INTERCEPT], R1                   ; 0246
                               50       0000G   CF 9E 0006E          MOVAB   ISSH_VEC_BASE, R0                               ;
                     54        51               50  C3 00073          SUBL3   R0, R1, PRIO_MASK                               ;
                                                52  D7 00077          DECL    R2                                              ; 0251
                     51        02               52  78 00079          ASHL    R2, #2, R1                                      ;
                               50         FF    A1  9E 0007D          MOVAB   -1(R1), MASK_CONST                              ;
                               51               50  D2 00081          MCOML   MASK_CONST, R1                                  ; 0256
                  10   BC                      51  CB 00084          BICL3   R1, @PRIO_MASK, @SAVE_MASK                      ;
                     51        0F               50  C3 00089          SUBL3   MASK_CONST, #15, R1                             ; 0262
                               52               64  D2 0008D          MCOML   @PRIO_MASK, R2                                  ;
                               51               52  CA 00090          BICL2   R2, R1                                          ;
                               52               50  D2 00093          MCOML   MASK_CONST, R2                                  ; 0263
                               53               52  CA 00096          BICL2   R2, R3                                          ;
                     64        53               51  C9 00099          BISL3   R1, R3, @PRIO_MASK                              ;
                                                68  D4 0009D          CLRL    SSI_RUNNING_FLAG                                ; 0268
                               50               01  D0 0009F 8$:      MOVL    #1, R0                                          ; 0269
                                                04  000A2            RET                                                     ;
                                                50  D4 000A3 9$:      CLRL    R0                                              ; 0271
                                                04  000A5            RET                                                     ;

; Routine Size:  166 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  275          0272  1 ROUTINE user_cancel_restore (setup_flag, routine_id, access_mode, routine_addr) =
;  276          0273  1
;  277          0274  1 !---
;  278          0275  1 !
;  279          0276  1 ! Function:
;  280          0277  1 !
;  281          0278  1 !       Delete/Restore the user routine in P0 space according to the given mode
;  282          0279  1 !       and id.
;  283          0280  1 !
;  284          0281  1 ! Inputs:
;  285          0282  1 !
;  286          0283  1 !       setup_flag:     0 - cancel, 1 - restore.
;  287          0284  1 !
;  288          0285  1 !       access_mode:    mode of the user routine.
;  289          0286  1 !
;  290          0287  1 !       routine_id:     user routine identification.
;  291          0288  1 !
;  292          0289  1 !       routine_addr:   user routine address.
;  293          0290  1 !
;  294          0291  1 ! Outputs:
;  295          0292  1 !
;  296          0293  1 !       None.
;  297          0294  1 !
;  298          0295  1 !---
;  299          0296  1
;  300          0297  2     BEGIN
;  301          0298  2
;  302          0299  2     LOCAL
;  303          0300  2         intercept,                          ! Address of saved system vector,
;  304          0301  2                                             !   data, code area in P0
;  305          0302  2         data_base : REF VECTOR[,LONG],      ! Address of the data area
;  306          0303  2                                             !   (in which keeps a table of
;  307          0304  2                                             !    user routines indexed
;  308          0305  2                                             !    by (id, mode)
;  309          0306  2         ctrl_id;                            ! True position relative to 0
;  310          0307  2
;  311          0308  2
;  312          0309  2     intercept = .(SYS$QIOW + sgn$c_sysvecpgs * 512 - 4);
;  313          0310  2
;  314          0311  3     IF (..routine_id LSS 1) OR (..routine_id GTR 16)
;  315          0312  2     THEN
;  316          0313  2         RETURN 0;
;  317          0314  2
;  318          0315  2     data_base = .intercept + issh_data_beg - issh_vec_base;
;  319          0316  2     ctrl_id = ..routine_id - 1;
;  320          0317  2
;  321          0318  2
;  322          0319  2     ! Delete/Restore the entry/entries indexed by (id, mode) to (id, PSL$C_USER).
;  323          0320  2     ! Note: there is no check being made in here.
;  324          0321  2     !
;  325          0322  2     INCR i FROM .access_mode to PSL$C_USER DO
;  326          0323  3         BEGIN
;  327          0324  3         IF .setup_flag
;  328          0325  3         THEN
;  329          0326  3             data_base [.ctrl_id * 4] + .i * 4 = .routine_addr
;  330          0327  3         ELSE
;  331          0328  3             data_base [.ctrl_id * 4] + .i * 4 = 0;
```

```
;  332          0329  2          END;
;  333          0330  2
;  334          0331  2      RETURN ss$_normal;
;  335          0332  1      END;


                                        0000 00000 USER_CANCEL_RESTORE:
                                                   .WORD   Save nothing                                  ; 0272
                          50 00000000G   9F   DO 00002                MOVL    @#SYS$QIOW+2556, INTERCEPT    ; 0309
                             08   BC   D5 00009                TSTL    @ROUTINE_ID                   ; 0311
                                  3D   15 0000C                BLEQ    4$
                          10     08   BC   D1 0000E                CMPL    @ROUTINE_ID, #16
                                  37   14 00012                BGTR    4$
                          51  0000GCF   40   9E 00014          MOVAB   ISSH_DATA_BEG[INTERCEPT], R1  ; 0315
                          50  0000G     CF   9E 0001A          MOVAB   ISSH_VEC_BASE, R0
                          51          50   C2 0001F          SUBL2   R0, DATA_BASE
                    50    08   BC           01   C3 00022     SUBL3   #1, @ROUTINE_ID, CTRL_ID      ; 0316
                          50           04   C4 00027          MULL2   #4, R0                        ; 0326
                          51        6140   DE 0002A          MOVAL   (DATA_BASE)[R0], R1
                    50    0C   AC           01   C3 0002E     SUBL3   #1, ACCESS_MODE, I
                                       0E   11 00033          BRB     3$
                             07     04   AC   E9 00035 1$:    BLBC    SETUP_FLAG, 2$
                          6140      10   AC   D0 00039          MOVL    ROUTINE_ADDR, (R1)[I]
                                       03   11 0003E          BRB     3$
                                    6140   D4 00040 2$:       CLRL    (R1)[I]                       ; 0328
                    EE              50   03   F3 00043 3$:    AOBLEQ  #3, I, 1$                     ; 0322
                                    50   01   D0 00047          MOVL    #1, R0                        ; 0331
                                            04 0004A          RET
                                    50   D4 0004B 4$:       CLRL    R0                            ; 0332
                                            04 0004D          RET                                   ;

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 00A6

;  336          0333  1
```

```
;  338     0334  1 ROUTINE user_setup
;  339     0335  1     (routine_id, routine_addr, routine_prio, routine_mode) =
;  340     0336  1
;  341     0337  1 !---
;  342     0338  1 !
;  343     0339  1 ! Function:
;  344     0340  1 !
;  345     0341  1 !       Enter the user routine in P0 space according to the given mode
;  346     0342  1 !       and priority.
;  347     0343  1 !
;  348     0344  1 !       Note:
;  349     0345  1 !       The way I have set up the table in here has wasted a lot of
;  350     0346  1 !       space, ie., this table looks like,
;  351     0347  1 !
;  352     0348  1 !       prio\mode : 3  2  1  0 (kernel)
;  353     0349  1 !       ----------+--------------------
;  354     0350  1 !            1    : x1 x1 x1 x1 (kernel, in prio. 1, x1)
;  355     0351  1 !            2    : x2          (user, in prio. 2, x2)
;  356     0352  1 !            3    :
;  357     0353  1 !            4    : x3          (user, in prio. 4, x3) <-- ctrl_index: 4
;  358     0354  1 !
;  359     0355  1 !       where this table can be condensed into 16 entries:
;  360     0356  1 !       prio\mode : 3  2  1  0 (kernel)
;  361     0357  1 !       ----------+--------------------
;  362     0358  1 !            1    : x1       x2 (kernel and all lowe modes, in prio. 1, x2;
;  363     0359  1 !            2    :                 user in prio. 1, x1)
;  364     0360  1 !            3    :
;  365     0361  1 !            4    :
;  366     0362  1 !
;  367     0363  1 !       USER_ID in here is not needed at all, (easily identified locatation
;  368     0364  1 !       by prio. and mode). <-- future improvements
;  369     0365  1 !
;  370     0366  1 ! Inputs:
;  371     0367  1 !
;  372     0368  1 !       routine_id:     Address of the ID.
;  373     0369  1 !
;  374     0370  1 !       routine_addr:   Address of the routine.
;  375     0371  1 !
;  376     0372  1 !       routine_prio:   Priority of the routine.
;  377     0373  1 !
;  378     0374  1 !       routine_mode:   Mode of the routine.
;  379     0375  1 !
;  380     0376  1 ! Outputs:
;  381     0377  1 !
;  382     0378  1 !       None.
;  383     0379  1 !
;  384     0380  1 !---
;  385     0381  1
;  386     0382  2     BEGIN
;  387     0383  2
;  388     0384  2     LOCAL
;  389     0385  2         intercept,                          ! Address of saved system vector,
;  390     0386  2                                             !   data, code area in P0
;  391     0387  2         ctrl_index : REF VECTOR[,LONG],     ! Total prio. entries.
;  392     0388  2         ctrl_prio_index : REF VECTOR[,BYTE],! Total entries in each prio.
;  393     0389  2         data_base : REF VECTOR[,LONG],      ! Pointer to data area in P0
;  394     0390  2         position,                           ! Exact location in the table
```

```
;  395          0391  2          priority;                               ! Prio. relative to 0
;  396          0392  2
;  397          0393  2
;  398          0394  2      intercept = .(SYS$QIOW + sgn$c_sysvecpgs * 512 - 4);
;  399          0395  2
;  400          0396  2
;  401          0397  2      ! Get the Address of the table in P0.
;  402          0398  2      !
;  403          0399  2      data_base = .intercept + issh_data_beg - issh_vec_base;
;  404          0400  2
;  405          0401  2
;  406          0402  2      ! Calculate the position in the table for the routine to enter.
;  407          0403  2      !
;  408          0404  2      ! 4 modes, 4 priority per mode.
;  409          0405  2      !
;  410          0406  2      priority = .routine_prio - 1;
;  411          0407  2      ctrl_prio_index = .intercept + issh_ctrl_prio_index - issh_vec_base;
;  412          0408  2      position = .ctrl_prio_index[.priority] * 4 + .priority;
;  413          0409  2
;  414          0410  2
;  415          0411  2      ! Table is full.  We have 128 longwords available.  We use half for the
;  416          0412  2      ! table (in reality, 16 longwords is enough), the other half
;  417          0413  2      ! we use it as local storage.
;  418          0414  2      ! so we have 64 longwords / 4 modes = 16 entries.
;  419          0415  2      !
;  420          0416  2      IF .position GEQ 16 THEN RETURN 0;
;  421          0417  2
;  422          0418  2
;  423          0419  2      ! Enter the routine address.
;  424          0420  2      !
;  425          0421  2      INCR i FROM .routine_mode TO 3 DO
;  426          0422  2          data_base [.position * 4] + .i * 4 = .routine_addr;
;  427          0423  2
;  428          0424  2
;  429          0425  2      ! Return the identification back to the user.
;  430          0426  2      !
;  431          0427  2      .routine_id = .position + 1;
;  432          0428  2
;  433          0429  2
;  434          0430  2      ! Update the control indexes, one per priority, and one for the table.
;  435          0431  2      !
;  436          0432  2      ctrl_prio_index[.priority] = .ctrl_prio_index[.priority] + 1;
;  437          0433  2      ctrl_index = .intercept + issh_ctrl_index - issh_vec_base;
;  438          0434  2      ctrl_index[0] = MAX(.ctrl_index[0], ..routine_id);
;  439          0435  2      RETURN 1;
;  440          0436  2
;  441          0437  1      END;


                             007C 00000 USER_SETUP:
                                                .WORD   Save R2,R3,R4,R5,R6                             ; 0334
                 56     0000G   CF  9E 00002        MOVAB   ISSH_VEC_BASE, R6                               ;
                 53 00000000G   9F  D0 00007        MOVL    @#SYS$QIOW+2556, INTERCEPT                      ; 0394
                 51     0000GCF43   9E 0000E        MOVAB   ISSH_DATA_BEG[INTERCEPT], R1                    ; 0399
```

```
                50              66  9E 00014         MOVAB   ISSM_VEC_BASE, R0                                    ;
              54 51              50  C3 00017         SUBL3   R0, R1, DATA_BASE                                   ;
              55 0C    AC        01  C3 0001B         SUBL3   #1, ROUTINE_PRIO, PRIORITY                          ; 0406
                51      0000GCF43  9E 00020         MOVAB   ISSM_CTRL_PRIO_INDEX[INTERCEPT], R1                 ; 0407
                50              66  9E 00026         MOVAB   ISSM_VEC_BASE, R0                                   ;
              52 51              50  C3 00029         SUBL3   R0, R1, CTRL_PRIO_INDEX                             ;
                50            6542  9A 0002D         MOVZBL  (PRIORITY)[CTRL_PRIO_INDEX], R0                     ; 0408
                50            6540  DE 00031         MOVAL   (PRIORITY)[R0], POSITION                            ;
                10              50  D1 00035         CMPL    POSITION, #16                                       ; 0416
                                41  18 00038         BGEQ    4$                                                  ;
              51 50              02  78 0003A         ASHL    #2, POSITION, R1                                    ; 0422
                54            6441  DE 0003E         MOVAL   (DATA_BASE)[R1], R4                                 ;
              51 10    AC        01  C3 00042         SUBL3   #1, ROUTINE_MODE, I                                 ;
                                05  11 00047         BRB     2$                                                  ;
                    6441    08  AC  D0 00049 1$:     MOVL    ROUTINE_ADDR, (R4)[I]                               ;
              F7 51             03  F3 0004E 2$:     AOBLEQ  #3, I, 1$                                           ;
                04  BC    01  A0  9E 00052         MOVAB   1(R0), @ROUTINE_ID                                  ; 0427
                            6542  96 00057         INCB    (PRIORITY)[CTRL_PRIO_INDEX]                         ; 0432
                51      0000GCF43  9E 000[illegible] MOVAB   ISSM_CTRL_INDEX[INTERCEPT], R1                      ; 0433
                50              66  9E 00[illegible] MOVAB   ISSM_VEC_BASE, R0                                   ;
              50 51              50  C3 000[illegible] SUBL3   R0, R1, CTRL_INDEX                                 ;
                51              60  D0 00067         MOVL    (CTRL_INDEX), R1                                    ; 0434
                04  BC            51  D1 0006A         CMPL    R1, @ROUTINE_ID                                     ;
                                04  18 0006E         BGEQ    3$                                                  ;
                51    04  BC        D0 00070         MOVL    @ROUTINE_ID, R1                                     ;
                60              51  D0 00074 3$:     MOVL    R1, (CTRL_INDEX)                                    ;
                50              01  D0 00077         MOVL    #1, R0                                              ; 0435
                                    04 0007A         RET                                                         ;
                                50  D4 0007B 4$:     CLRL    R0                                                  ; 0437
                                    04 0007D         RET                                                         ;

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 00F4

;  442          0438  1
;  443          0439  1 END
;  444          0440  0 ELUDOM


;                                   PSECT SUMMARY
;
;      Name                     Bytes                          Attributes
;
; $CODE$                          370  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics
;                                          -------- Symbols --------      Pages       Processing
;      File                                Total    Loaded   Percent      Mapped      Time
;
```

```
;  _$255$DUA28:[SYSLIB]LIB.L32;1              18619          3          0       1000          00:01.9



;                                       COMMAND QUALIFIERS

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

; Size:         370 code + 0 data bytes
; Run Time:         00:09.8
; Elapsed Time:     00:34.3
; Lines/CPU Min:     2707
; Lexemes/CPU-Min:   7187
; Memory Used:  91 pages
; Compilation Complete
```

SSIK
LIS

DELETE

DELEMAIN
LIS

DELTA
LIS

SSITAB
LIS

DELETE
MAP

DELTA

DELTA
MAP

SSIU
LIS

SSIUW
LIS

PURGE
LIS

STRUCDEF
LIS

SSIDISP
LIS

SØDELTA
MAP

DELETE
REQ