```
LLL                 000000000          GGGGGGGGGGGG     IIIIIIIII     NNN           NNN
LLL                 000000000          GGGGGGGGGGGG     IIIIIIIII     NNN           NNN
LLL                 000000000          GGGGGGGGGGGG     IIIIIIIII     NNN           NNN
LLL             000           000   GGG                    III      NNN           NNN
LLL             000           000   GGG                    III      NNN           NNN
LLL             000           000   GGG                    III      NNN           NNN
LLL             000           000   GGG                    III      NNNNNN        NNN
LLL             000           000   GGG                    III      NNNNNN        NNN
LLL             000           000   GGG                    III      NNNNNN        NNN
LLL             000           000   GGG                    III      NNN    NNN    NNN
LLL             000           000   GGG                    III      NNN    NNN    NNN
LLL             000           000   GGG                    III      NNN    NNN    NNN
LLL             000           000   GGG    GGGGGGGGG       III      NNN       NNNNNN
LLL             000           000   GGG    GGGGGGGGG       III      NNN       NNNNNN
LLL             000           000   GGG    GGGGGGGGG       III      NNN       NNNNNN
LLL             000           000   GGG          GGG       III      NNN           NNN
LLL             000           000   GGG          GGG       III      NNN           NNN
LLL             000           000   GGG          GGG       III      NNN           NNN
LLLLLLLLLLLLLLL     000000000          GGGGGGGGG        IIIIIIIII     NNN           NNN
LLLLLLLLLLLLLLL     000000000          GGGGGGGGG        IIIIIIIII     NNN           NNN
LLLLLLLLLLLLLLL     000000000          GGGGGGGGG        IIIIIIIII     NNN           NNN
```

```
IIIIII    NN       NN    IIIIII    TTTTTTTTTT  UU      UU    SSSSSSSS  EEEEEEEEEE  RRRRRRRR
IIIIII    NN       NN    IIIIII    TTTTTTTTTT  UU      UU    SSSSSSSS  EEEEEEEEEE  RRRRRRRR
  II      NN       NN      II          TT      UU      UU  SS          EE          RR      RR
  II      NN       NN      II          TT      UU      UU  SS          EE          RR      RR
  II      NNNN     NN      II          TT      UU      UU  SS          EE          RR      RR
  II      NNNN     NN      II          TT      UU      UU  SS          EE          RR      RR
  II      NN NN    NN      II          TT      UU      UU    SSSSSS    EEEEEEEE    RRRRRRRR
  II      NN NN    NN      II          TT      UU      UU    SSSSSS    EEEEEEEE    RRRRRRRR
  II      NN   NNNN       II          TT      UU      UU          SS  EE          RR  RR
  II      NN   NNNN       II          TT      UU      UU          SS  EE          RR  RR
  II      NN     NN       II          TT      UU      UU          SS  EE          RR    RR      ....
  II      NN     NN       II          TT      UU      UU          SS  EE          RR    RR      ....
IIIIII    NN       NN    IIIIII        TT      UUUUUUUUUU  SSSSSSSS  EEEEEEEEEE  RR      RR    ....
IIIIII    NN       NN    IIIIII        TT      UUUUUUUUUU  SSSSSSSS  EEEEEEEEEE  PR      RR    ....

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
```

; R

```
;     1          0001  0 MODULE inituser (IDENT = 'V04-000',
;     2          0002  0         ADDRESSING_MODE(EXTERNAL = GENERAL)) =
;     3          0003  1 BEGIN
;     4          0004  1
;     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 ! FACILITY:  Login
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This module handles all user-specific and CLI initializations.
;    35          0035  1 !
;    36          0036  1 ! ENVIRONMENT:
;    37          0037  1 !
;    38          0038  1 !       VAX/VMS operating system.
;    39          0039  1 !
;    40          0040  1 ! AUTHOR:  Tim Halvorsen, March 1981
;    41          0041  1 !
;    42          0042  1 ! Modified by:
;    43          0043  1 !
;    44          0044  1 !       V03-036 BLS0346         Benn Schreiber          28-AUG-1984
;    45          0045  1 !               Fix cli determination to work for SUBMIT/CLI also.
;    46          0046  1 !
;    47          0047  1 !       V03-035 BLS0343         Benn Schreiber          26-AUG-1984
;    48          0048  1 !               Force CLI to DCL if network process.
;    49          0049  1 !
;    50          0050  1 !       V03-034 LJK0288         Lawrence J. Kenah       9-Aug-1984
;    51          0051  1 !               The AUTHPRI now exists on both the PCB and the PHD.
;    52          0052  1 !
;    53          0053  1 !       V03-033 MHB0162         Mark Bramhall           24-Jul-1984
;    54          0054  1 !               Allow logical name usage activating CLI tables.
;    55          0055  1 !
;    56          0056  1 !       V03-032 ACG0432         Andrew C. Goldstein,    10-Jul-1984  21:25
;    57          0057  1 !               Fix initialization of BYTLM
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-031 MHB0160         Mark Bramhall           26-Jun-1984
;    60          0060  1 !               Fixed restriction vector definition in INIT_KERNEL.
;    61          0061  1 !               Default CLI tables to "clinameTABLES" for all CLIs.
;    62          0062  1 !
;    63          0063  1 !       V03-030 MHB0126         Mark Bramhall           11-Apr-1984
;    64          0064  1 !               Set node name, etc. becomes SET_NODE_NAME.
;    65          0065  1 !               Set terminal name becomes SET_TERM_NAME.
;    66          0066  1 !               Copy activated CLI name into CTL$GT_CLINAME.
;    67          0067  1 !               Use CTL$GT_SPAWNCLI/TABLE as defaults for CLI/CLI table.
;    68          0068  1 !               Change SET_ACCOUNT to NOVALUE and make it GLOBAL.
;    69          0069  1 !               Change SET_USERNAME to NOVALUE.
;    70          0070  1 !
;    71          0071  1 !       V03-029 MHB0117         Mark Bramhall           23-Mar-1984
;    72          0072  1 !               Propagate UAF$V_AUDIT to both PCB$V_SECAUDIT and PCB_STS.
;    73          0073  1 !               Remove any version number from the CTL$GT_TABLENAME string.
;    74          0074  1 !
;    75          0075  1 !       V03-028 MHB0109         Mark Bramhall           21-Mar-1984
;    76          0076  1 !               Use LNM services for logical names.
;    77          0077  1 !               Copy activated CLI table filespec into CTL$GT_TABLENAME.
;    78          0078  1 !               Reference PCB_STS as a BITVECTOR.
;    79          0079  1 !               Change SET_NODENAME to NOVALUE and rework it.
;    80          0080  1 !               Change SET_TERMNAME to NOVALUE and rework it.
;    81          0081  1 !
;    82          0082  1 !       V03-027 PCG0001         Peter George            31-Jan-1984  11:15
;    83          0083  1 !               Rewrite CLI and tables determination logic.
;    84          0084  1 !               Propagate UAF$V_AUDIT to the PCB.
;    85          0085  1 !
;    86          0086  1 !       V03-026 KPL0001         Peter Lieberwirth       27-Jan-1984  13:15
;    87          0087  1 !               Correct problem setting up item-list for creation of group
;    88          0088  1 !               logical name table
;    89          0089  1 !
;    90          0090  1 !       V03-025 ACG0395         Andrew C. Goldstein,    24-Jan-1984  23:04
;    91          0091  1 !               Restore job logical name table code, accidentally dropped
;    92          0092  1 !               in ACG0389
;    93          0093  1 !
;    94          0094  1 !       V03-024 ACG0389         Andrew C. Goldstein,    18-Jan-1984  11:30
;    95          0095  1 !               Condition protecting CLI tables on their being mapped
;    96          0096  1 !
;    97          0097  1 !       V03-023 ACG0385         Andrew C. Goldstein,    28-Dec-1983  17:14
;    98          0098  1 !               Handle longer username and account in UAF; implement
;    99          0099  1 !               job type and per type hourly restrictions. Change UAF
;   100          0100  1 !               working set fields to longwords.
;   101          0101  1 !
;   102          0102  1 !       V03-022 TMK0004         Todd M. Katz            21-Dec-1983
;   103          0103  1 !               Create the group and job logical name tables by calling the
;   104          0104  1 !               exec routine EXE$CRE_JGTABLE instead of issuing the appropriate
;   105          0105  1 !               $CRELNT system services.
;   106          0106  1 !
;   107          0107  1 !       V03-021 ACG0376         Andrew C. Goldstein,    18-Nov-1983  18:32
;   108          0108  1 !               Put virtual and physical terminal names in global buffers.
;   109          0109  1 !               Clarify logic for picking CLI and tables from UAF and params.
;   110          0110  1 !               Clean up mapping of CLI: remove SYS$SYSTEM from default
;   111          0111  1 !               name string, remove UIC changing kluges, restore calls to
;   112          0112  1 !               PROTECT_CLI. Allow for existing use in setting up JIB$W_ENQCNT
;   113          0113  1 !               and JIB$W_SHRFILCNT.
;   114          0114  1 !
```

```
;  115     0115  1 !        V03-020 TMK0003         Todd M. Katz            12-Oct-1983
;  116     0116  1 !                If the process is not a sub-process, create the job-wide
;  117     0117  1 !                logical name table. It is necessary to create this table a
;  118     0118  1 !                second time, because when the table is originally created
;  119     0119  1 !                within PROCSTRT, it is most often created with the wrong UIC
;  120     0120  1 !                and quota. Note that this second creation will delete the
;  121     0121  1 !                existing table.
;  122     0122  1 !
;  123     0123  1 !                Change the name of the routine set_group_lnm_table to
;  124     0124  1 !                set_lnm_tables, and re-create both the group and the job-wide
;  125     0125  1 !                logical name tables within it.
;  126     0126  1 !
;  127     0127  1 !        V03-019 TMK0002         Todd M. Katz            26-Sep-1983
;  128     0128  1 !                Create the group logical name table with a protection of
;  129     0129  1 !                SYSTEM:RWED OWNER: GROUP:R WORLD so that processes with system
;  130     0130  1 !                access rights can access and modify any group table.
;  131     0131  1 !
;  132     0132  1 !        V03-018 GAS0189         Gerry Smith             22-Sep-1983
;  133     0133  1 !                Well, as it turns out, finding the actual physical
;  134     0134  1 !                device associated with virtual terminals wasn't such
;  135     0135  1 !                a good idea after all.  Seems it interferes with the
;  136     0136  1 !                working of terminal broadcasts.  So, just get the immediate
;  137     0137  1 !                device name for the terminal, and find the real physical
;  138     0138  1 !                device elsewhere.
;  139     0139  1 !
;  140     0140  1 !        V03-017 GAS0184         Gerry Smith             16-Sep-1983
;  141     0141  1 !                Add support in SET_TERMNAME for the VT terminals, which
;  142     0142  1 !                actually point to a physical UCB.  This makes sure that,
;  143     0143  1 !                for accounting and security purposes, the actual physical
;  144     0144  1 !                terminal is what is kept track of, rather than floating
;  145     0145  1 !                "virtual devices" whose names mean nothing.
;  146     0146  1 !
;  147     0147  1 !        V03-016 GAS0183         Gerry Smith             15-Sep-1983
;  148     0148  1 !                Change the SET_TERM stuff just a bit, to facilitate
;  149     0149  1 !                breakin evasion.  Also, don't set the terminal name
;  150     0150  1 !                unless SYS$INPUT is really a terminal.
;  151     0151  1 !
;  152     0152  1 !        V03-015 TMK0001         Todd M. Katz            22-Aug-1983
;  153     0153  1 !                Create the Group Logical Name Table with the protection G:R
;  154     0154  1 !                and specify the attributes GROUP and NO_ALIAS on creation.
;  155     0155  1 !
;  156     0156  1 !        V03-014 GAS0166         Gerry Smith             18-Aug-1983
;  157     0157  1 !                When referencing the group logical name table, make
;  158     0158  1 !                sure that the group number is in octal, instead of
;  159     0159  1 !                decimal.  Also obtain the terminal name by calling
;  160     0160  1 !                ioc$cvt_devnam with -1 instead of 0.  NOTE that this
;  161     0161  1 !                call may need to be revisited, if terminals start
;  162     0162  1 !                having tneir node associated with them, or if
;  163     0163  1 !                ioc$cvt_devnam gets rid of the leading underscore.
;  164     0164  1 !
;  165     0165  1 !        V03-013 GAS0161         Gerry Smith             28-Jul-1983
;  166     0166  1 !                Add environmental rights.
;  167     0167  1 !
;  168     0168  1 !        V03-012 GAS0155         Gerry Smith             18-Jul-1983
;  169     0169  1 !                Remove the code that protects the CLI pages.
;  170     0170  1 !
;  171     0171  1 !        V03-011 GAS0152         Gerry Smith              6-Jul-1983
```

```
;  172     0172  1 ;
;  173     0173  1 ;           For calls to CRELNM/CRELNT, pass all parameters by
;  174     0174  1 ;           reference.
;  175     0175  1 ;
;  176     0176  1 ;       V03-010 GAS0138         Gerry Smith             20-Jun-1983
;  177     0177  1 ;           Add CLITABLES, the cli command tables.
;  178     0178  1 ;
;  179     0179  1 ;       V03-009 DMW4047         DMWalp                  10-Jun-1983
;  180     0180  1 ;           Create group logical name tables
;  181     0181  1 ;
;  182     0182  1 ;       V03-008 GAS0126         Gerry Smith             20-Apr-1983
;  183     0183  1 ;           Create the access rights list(s) and attach to PCB.
;  184     0184  1 ;           Also, for network processes, set their base priority
;  185     0185  1 ;           from the NETUAF file.
;  186     0186  1 ;
;  187     0187  1 ;       V03-007 WMC0001         Wayne Cardoza           12-Apr-1983
;  188     0188  1 ;           Add MAXDETACH JIB field.
;  189     0189  1 ;
;  190     0190  1 ;       V03-006 GAS0095         Gerry Smith             22-Nov-1982
;  191     0191  1 ;           Add support for the PPD$V_CAPTIVE bit.  This enables
;  192     0192  1 ;           a CLI to determine whether or not the process is a
;  193     0193  1 ;           captive process.
;  194     0194  1 ;
;  195     0195  1 ;       V03-005 GAS0092         Gerry Smith             21-Oct-1982
;  196     0196  1 ;           Add support for the CLI name being in the compatibility
;  197     0197  1 ;           mode shelf.  This allows the spawning or submission of
;  198     0198  1 ;           processes with a different CLI than what the parent
;  199     0199  1 ;           process is running.
;  200     0200  1 ;
;  201     0201  1 ;       V03-004 TMH0004         Tim Halvorsen           24-Jun-1982
;  202     0202  1 ;           Fix failure to initialize an NFB field.
;  203     0203  1 ;
;  204     0204  1 ;       V03-003 TMH0003         Tim Halvorsen           07-Jun-1982
;  205     0205  1 ;           Modify to use new NETACP QIO interface.
;  206     0206  1 ;
;  207     0207  1 ;       V03-002 GAS0079         Gerry Smith              3-May-1982
;  208     0208  1 ;           When checking for the presence of the DISCTLY bit,
;  209     0209  1 ;           check for the presence of the CAPTIVE bit at well,
;  210     0210  1 ;           since CAPTIVE implies disabled ctrl/y.
;  211     0211  1 ;
;  212     0212  1 ;       V03-01  GAS0076         Gerry Smith             23-Apr-1982
;  213     0213  1 ;           Get NFB definitions from SHRLIB$:NET.L32
;  214     021   1 ;
;  215     021   1 ;       V02-012 GAS0052         Gerry Smith             23-Feb-1982
;  216     02    1 ;           Change the UIC from [10,40] to [1,4], since 10 is
;  217     0217  1 ;           not necessarily a system group-number.
;  218     0218  1 ;
;  219     0219  1 ;       V02-011 SPF0041         Steve Forgey            02-Dec-1981
;  220     0220  1 ;           Add routine to get remote node information.
;  221     0221  1 ;
;  222     0222  1 ;       V02-010 HRJ0032         Herb Jacobs             12-Nov-1981
;  223     0223  1 ;           Fix maximization of WSEXTENT field, set account name
;  224     0224  1 ;           in JIB, and set time of day restrictions in JIB.
;  225     0225  1 ;
;  226     0226  1 ;       V02-009 LJK0068         Lawrence J. Kenah       12-Nov-1981
;  227     0227  1 ;           Add initialization of PHD$B_AUTHPRI with new value
;  228     0228  1 ;           of base priority.
```

```
;  229     0229  1 !       V02-008 TMH0008         Tim Halvorsen           27-Oct-1981
;  230     0230  1 !               Remove code to initialize CLIREG to LOGIN module.
;  231     0231  1 !               Remove code to store ORIGUIC to LOGIN module, since
;  232     0232  1 !               it is now stored in the LGI area.
;  233     0233  1 !               Add extra acmode argument to EXEC_CRELOG routine.
;  234     0234  1 !
;  235     0235  1 !       V02-007 TMH0007         Tim Halvorsen           12-Oct-1981
;  236     0236  1 !               Update size of CLI OWN storage in P1 space using symbol
;  237     0237  1 !               provided by SHELL rather than having the size hard-coded
;  238     0238  1 !               here as well as in SHELL.
;  239     0239  1 !
;  240     0240  1 !       V02-006 LJK0063         Lawrence J. Kenah       16-Sep-1981
;  241     0241  1 !               Change name of external procedure to LIB$P1_MERGE.
;  242     0242  1 !
;  243     0243  1 !       V02-005 SPF0032         Steve Forgey    16-Sep-1981
;  244     0244  1 !               Use $GETDEV to get terminal name and unit number.
;  245     0245  1 !
;  246     0246  1 !       V02-004 SPF0031         Steve Forgey            15-Sep-1981
;  247     0247  1 !               Create a routine to set the terminal name in the PCB.
;  248     0248  1 !
;  249     0249  1 !       V02-003 ROW0021         Ralph O. Weber          19-Aug-1981
;  250     0250  1 !               Make changes for longword Buffered I/O Byte Limit Quota.
;  251     0251  1 !               INIT_KERNEL has been modified to copy UAF$L_BYTLM into
;  252     0252  1 !               JIB$L_BYTLM unless UAF$L_BYTLM is zero.  When UAF$L_BYTLM is
;  253     0253  1 !               zero, INIT_KERNEL copies UAF$W_BYTLM to JIB$L_BYTLM.  Thus
;  254     0254  1 !               INIT_KERNEL whether or not the UAF record contains a valid
;  255     0255  1 !               value in UAF$L_BYTLM (ie: it works properly for both old and
;  256     0256  1 !               new format UAF records).  (NB: the difference between the
;  257     0257  1 !               word, old style, and longword, new style, names).  All
;  258     0258  1 !               programs which operate on the User Authorization File
;  259     0259  1 !               (eg: AUTHORIZE and LOGINOUT) will be modified to first check
;  260     0260  1 !               UAF$L_BYTLM and if it is zero use UAF$W_BYTLM.
;  261     0261  1 !
;  262     0262  1 !       V02-002 HRJ0023         Herb Jacobs     16-Jul-1981
;  263     0263  1 !               Initialize authorized working set extent field PHD$W_WSEXTENT.
;  264     0264  1 !
;  265     0265  1 !       V02-001 TMH0001         Tim Halvorsen   16-Jul-1981
;  266     0266  1 !               Reference SHRLIB$ for shared require files.                    ;
;  267     0267  1 !--
;  268     0268  1
;  269     0269  1 !
;  270     0270  1 ! Include files
;  271     0271  1 !
;  272     0272  1
;  273     0273  1 LIBRARY 'SYS$LIBRARY:LIB';              ! VAX/VMS system definitions
;  274     0274  1
;  275     0275  1 REQUIRE 'SHRLIB$:UTILDEF';              ! Common BLISS definitions
;  276     0460  1
;  277     0461  1 LIBRARY 'SHRLIB$:NET';                  . Network definitions
;  278     0462  1
;  279     0463  1 REQUIRE 'LIB$:PPDDEF';                  ! Process permanent data region
;  280     0610  1
;  281     0611  1 !
;  282     0612  1 ! Declare the linkages to allocate and deallocate nonpaged pool
;  283     0613  1 !
;  284     0614  1 LINKAGE
;  285     0615  1         ALLO    = JSB (REGISTER = 1;            ! R1 = size (on input)
```

```
;  286     0616  1                 REGISTER = 1,                   ! R1 = size of block
;  287     0617  1                 REGISTER = 2):                  ! R2 = address of block
;  288     0618  1                 NOPRESERVE (3,4,5),             ! R3, R4, R5 destroyed
;  289     0619  1          DEALLO = JSB (REGISTER = 0):           ! R0 = address of block
;  290     0620  1                 NOPRESERVE (1,2,3,4,5),         ! R1-R5 destroyed
;  291     0621  1          JGTABLE = JSB(                         ! Create Job and Group Tables
;  292     0622  1                     REGISTER = 7,               ! Job Table Creation Quota
;  293     0623  1                     REGISTER = 10,              ! JIB Address ASCII Equivalent
;  294     0624  1                     REGISTER = 11 ):            ! Group Number ASCII Equivalent
;  295     0625  1                 NOPRESERVE( 1, 2, 3, 4, 5 , 8 );
```

```
;  297          0626  1 !
;  298          0627  1 ! Table of contents
;  299          0628  1 !
;  300          0629  1
;  301          0630  1 FORWARD ROUTINE
;  302          0631  1     init_user:          NOVALUE,    ! Initialize user process quotas, etc.
;  303          0632  1     init_kernel:        NOVALUE,    ! Initialize user in kernel mode
;  304          0633  1     init_cli:           NOVALUE,    ! Initialize CLI image
;  305          0634  1     setup_login_proc:   NOVALUE,    ! Setup login command procedure
;  306          0635  1     map_cli:            NOVALUE,    ! Map the CLI image into P1 space
;  307          0636  1     execute_cli:        NOVALUE,    ! Call the CLI image at its entry point
;  308          0637  1     map_imgact:         NOVALUE,    ! Map image activator code segment
;  309          0638  1     set_p1_base,                    ! Set base address of control region
;  310          0639  1     set_account:        NOVALUE,    ! Set account name in JIB and P1 space
;  311          0640  1     set_username:       NOVALUE,    ! Set username in JIB and P1 space
;  312          0641  1     set_node_name:      NOVALUE,    ! Set remote node info in P1 space
;  313          0642  1     set_term_name:      NOVALUE,    ! Set terminal name in PCB
;  314          0643  1     set_uic,                        ! Set process UIC
;  315          0644  1     create_logical,                 ! Create logical name with LNM services
;  316          0645  1     make_rightslists:   NOVALUE,    ! Create the rights lists
;  317          0646  1     set_localrights,                ! Set up the local rights list
;  318          0647  1     set_more_rights,                ! Set up the extended rights list
;  319          0648  1     set_lnm_tables;                 ! Set up group and job-wide lnm tables
;  320          0649  1
;  321          0650  1 !
;  322          0651  1 ! External routines
;  323          0652  1 !
;  324          0653  1
;  325          0654  1 EXTERNAL ROUTINE
;  326          0655  1     str$append,                     ! Append to a dynamic buffer
;  327          0656  1     set_ppd_prot,                   ! Set page protection on PPD structure
;  328          0657  1     handler,                        ! Condition handler
;  329          0658  1     sys$setddir,                    ! Set default directory
;  330          0659  1     lgi$protect_cli,                ! Read-protect CLI code
;  331          0660  1     execute_image:      NOVALUE,    ! Chain to an image
;  332          0661  1     lib$p1_merge,                   ! Merge image into P1 space
;  333          0662  1     sys$find_held,                  ! RDB routine to find all ID's for a user
;  334          0663  1     exe$deanonpaged: DEALLO,        ! Deallocate non-paged pool
;  335          0664  1     exe$alononpaged: ALLO,          ! Allocate non-paged pool
;  336          0665  1     exe$cre_jgtable: JGTABLE;       ! Create Job and Group Tables
;  337          0666  1
;  338          0667  1 !
;  339          0668  1 ! External storage
;  340          0669  1 !
;  341          0670  1 EXTERNAL
;  342          0671  1     terminal_device:    BYTE,       ! True if SYS$INPUT is a terminal
;  343          0672  1     term_name:          VECTOR,     ! Terminal name descriptor
;  344          0673  1     dev_char_2:         $BBLOCK,    ! Device characteristics of sys$input
;  345          0674  1     dev_dep_2:          $BBLOCK,    ! Dev-dependent chars of sys$input
;  346          0675  1     pcb_sts:            BITVECTOR,  ! PCB status flags
;  347          0676  1     job_type,                       ! Job type code for JIB
;  348          0677  1     uaf_record:         REF $BBLOCK,! Address of UAF record
;  349          0678  1     sys$input:          VECTOR,     ! Translation of SYS$INPUT
;  350          0679  1     cli_name:           VECTOR,     ! Descriptor of CLI to map
;  351          0680  1     cli_name_buffer:    VECTOR [,BYTE], ! Buffer for CLI name
;  352          0681  1     table_name:         VECTOR,     ! Descriptor of CLI command table
;  353          0682  1     table_name_buffer:  VECTOR [,BYTE]; ! Buffer for CLI command table
```

```
;  354          0683  1      disk_name:          VECTOR,          ! Descriptor of initial default disk
;  355          0684  1      com_name:           VECTOR,          ! Descriptor of procedure to execute
;  356          0685  1      com_negated:        BYTE,            ! True if procedure inhibited
;  357          0686  1      subprocess:         BYTE,            ! True if subprocess
;  358          0687  1      image_activate:     BYTE,            ! True if image to be activated
;  359          0688  1      mmg$imghdrbuf,                       ! Image header buffer
;  360          0689  1      ctl$gl_pcb:         REF $BBLOCK,     ! This process's PCB
;  361          0690  1      ctl$gl_ccbbase,
;  362          0691  1      ctl$gl_uaf_flags:   BITVECTOR,       ! P1 space UAF flags
;  363          0692  1      ctl$gt_cliname:     VECTOR [,BYTE],  ! Activated CLI name (ASCIC)
;  364          0693  1      ctl$gt_tablename:   VECTOR [,BYTE],  ! Activated CLI table name (ASCIC)
;  365          0694  1      ctl$gt_spawncli:    VECTOR [,BYTE],  ! Spawn CLI name (ASCIC)
;  366          0695  1      ctl$gt_spawntable:  VECTOR [,BYTE],  ! Spawn CLI table name (ASCIC)
;  367          0696  1      ctl$ag_cmedata:     VECTOR [,BYTE],  ! CLI passed here from $IMGACT
;  368          0697  1                                           !  if cli image name give to $CREPRC
;  369          0698  1      ctl$ag_climage,                      ! Address of CLI code in control region
;  370          0699  1      ctl$ag_clitable,                     ! Address of CLI command tables
;  371          0700  1      ctl$ag_clidata;                      ! Process permanent data region
;  372          0701  1
;  373          0702  1  BIND
;  374          0703  1      ppd = ctl$ag_clidata: $BBLOCK;       ! Address of PPD structure
;  375          0704  1
;  376          0705  1  !
;  377          0706  1  ! Define message codes
;  378          0707  1  !
;  379          0708  1  EXTERNAL LITERAL
;  380          0709  1      lgi$_clifail,
;  381          0710  1      lgi$_cliprot,
;  382          0711  1      lgi$_clitblfail,
;  383          0712  1      lgi$_clitblprot,
;  384          0713  1      lgi$_clisymtbl;
```

```
;  386     0714  1 GLOBAL ROUTINE init_user: NOVALUE =
;  387     0715  1
;  388     0716  1 !---
;  389     0717  1 !
;  390     0718  1 !       Initialize all user context for the process.  All
;  391     0719  1 !       information from the UAF record is set into the appropriate
;  392     0720  1 !       places in the executive database, such as the UIC, privileges,
;  393     0721  1 !       base priority, limits, quotas, account name, etc.
;  394     0722  1 !
;  395     0723  1 ! Inputs:
;  396     0724  1 !
;  397     0725  1 !       uaf_record = Address of UAF record for user (must be non-zero)
;  398     0726  1 !       disk_name = Descriptor of device name to be used as SYS$DISK
;  399     0727  1 !
;  400     0728  1 ! Outputs:
;  401     0729  1 !
;  402     0730  1 !       None
;  403     0731  1 !---
;  404     0732  1
;  405     0733  2 BEGIN
;  406     0734  2
;  407     0735  2 LOCAL
;  408     0736  2     ptr,
;  409     0737  2     username:   VECTOR [2],          ! Descriptor of username
;  410     0738  2     account:    VECTOR [2],          ! Descriptor of account name
;  411     0739  2     device:     VECTOR [2],          ! Descriptor of default device
;  412     0740  2     directory:  VECTOR [2];          ! Descriptor of default directory
;  413     0741  2
;  414     0742  2 !
;  415     0743  2 ! Set base priority for process
;  416     0744  2 !
;  417     0745  2
;  418     0746  2 IF .pcb_sts[$BITPOSITION(pcb$v_inter)]  ! If interactive
;  419     0747  2 OR .pcb_sts[$BITPOSITION(pcb$v_netwrk)] ! or network process
;  420     0748  2 THEN
;  421     0749  2     $SETPRI(PRI = .uaf_record [uaf$b_pri]); ! Set base priority
;  422     0750  2
;  423     0751  2 !
;  424     0752  2 ! Set default directory
;  425     0753  2 !
;  426     0754  2
;  427     0755  2 directory [0] = CH$RCHAR(uaf_record [uaf$t_defdir]); ! Get descriptor of directory
;  428     0756  2 directory [1] = uaf_record [$BYTEOFFSET(uaf$t_defdir)+1,0,0,0];
;  429     0757  2
;  430     0758  2 SYS$SETDDIR(directory, 0, 0);          ! Set default directory
;  431     0759  2
;  432     0760  2 !
;  433     0761  2 ! Set default disk (logical name SYS$DISK)
;  434     0762  2 !
;  435     0763  2
;  436     0764  2 IF .disk_name [0] EQL 0                ! If no explicit disk specified
;  437     0765  2 THEN
;  438     0766  3     BEGIN
;  439     0767  3     device [0] = CH$RCHAR(uaf_record [uaf$t_defdev]); ! Get UAF disk name
;  440     0768  3     device [1] = uaf_record [$BYTEOFFSET(uaf$t_defdev)+1,0,0,0];
;  441     0769  3     END
;  442     0770  2 ELSE
```

```
;  443          0771  3     BEGIN
;  444          0772  3     device [0] = .disk_name [0];          ! Else, use what user specifies
;  445          0773  3     device [1] = .disk_name [1];
;  446          0774  2     END;
;  447          0775  2
;  448          0776  2 IF .device [0] NEQ 0                      ! If device specified,
;  449          0777  2 THEN
;  450          0778  2     create_logical(%ASCID 'SYS$DISK',
;  451          0779  2                    device,
;  452          0780  2                    psl$c_exec);
;  453          0781  2
;  454          0782  2 !
;  455          0783  2 ! Set the username string
;  456          0784  2 !
;  457          0785  2
;  458          0786  2 ptr = CH$FIND_CH(uaf$s_username, uaf_record [uaf$t_username], ' ');
;  459          0787  2
;  460          0788  2 IF CH$FAIL(.ptr)                          ! If no space found,
;  461          0789  2 THEN
;  462          0790  2     ptr = uaf_record [uaf$t_username] + uaf$s_username; ! Use entire thing
;  463          0791  2
;  464          0792  2 username [0] = CH$DIFF(.ptr, uaf_record [uaf$t_username]);
;  465          0793  2 username [1] = uaf_record [uaf$t_username];
;  466          0794  2
;  467        P 0795  2 $CMKRNL(ROUTIN = set_username,            ! Set username string
;  468          0796  2          ARGLST = username);
;  469          0797  2
;  470          0798  2 !
;  471          0799  2 ! Set the process UIC
;  472          0800  2 !
;  473          0801  2
;  474        P 0802  2 $CMKRNL(ROUTIN = set_uic,                 ! Set the UIC
;  475          0803  2          ARGLST = .uaf_record [uaf$l_uic]);
;  476          0804  2
;  477          0805  2 !
;  478          0806  2 ! Set up the correct group and job-wide logical name tables (ie - redo what
;  479          0807  2 ! PROCSTRT tried to do only this time with the correct UIC and quota
;  480          0808  2 ! information.
;  481          0809  2 !
;  482          0810  2
;  483          0811  3 BEGIN
;  484          0812  3
;  485          0813  3 LOCAL
;  486          0814  3      status;
;  487          0815  3
;  488          0816  4 IF NOT ( status = $CMKRNL( ROUTIN = set_lnm_tables ) )
;  489          0817  3 THEN
;  490          0818  3     SIGNAL_STOP( .status );
;  491          0819  3
;  492          0820  2 END;
;  493          0821  2
;  494          0822  2 !
;  495          0823  2 ! Set the account name for the process
;  496          0824  2 !
;  497          0825  2
;  498          0826  2 account [0] = uaf$s_account;             ! Setup descriptor of string
;  499          0827  2 account [1] = uaf_record [uaf$t_account];
```

```
;  500          0828  2
;  501        P 0829  2 $CMKRNL(ROUTIN = set_account,           ! Set account for process
;  502          0830  2          ARGLST = account);
;  503          0831  2
;  504          0832  2 !
;  505          0833  2 ! Set process name to username.  If this fails, the process name will be
;  506          0834  2 ! left to the orignal name, in the form _TTxx:
;  507          0835  2 !
;  508          0836  2
;  509          0837  2 IF NOT .ppd [ppd$v_mode]               ! If not batch, network or detached,
;  510          0838  2 THEN
;  511          0839  2     $SETPRN(PRCNAM = username);        ! Set process name (ignore errors)
;  512          0840  2
;  513          0841  2 !
;  514          0842  2 ! Create the access rights lists and attach them to the PCB.
;  515          0843  2 !
;  516          0844  2 make_rightslists();
;  517          0845  2
;  518          0846  2 !
;  519          0847  2 ! Set terminal name in PCB if connected to a terminal
;  520          0848  2 !
;  521          0849  2 IF .terminal_device                    ! If connected to a terminal,
;  522          0850  2 THEN set_term_name();                  !  set terminal name in PCB
;  523          0851  2
;  524          0852  2 !
;  525          0853  2 ! Perform kernel mode initialization
;  526          0854  2 !
;  527          0855  2
;  528          0856  2 $CMKRNL(ROUTIN = init_kernel);         ! Copy UAF fields into proper places
;  529          0857  2
;  530          0858  1 END;


                                                .TITLE  INITUSER
                                                .IDENT  \V04-000\

                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                4B  53  49  44  24  53  59  53  00000 P.AAB:   .ASCII  \SYS$DISK\                    ;
                                    010E0008    00008 P.AAA:   .LONG   17694728                     ;
                                    00000000'   0000C          .ADDRESS P.AAB                   ;

                                                .EXTRN  STR$APPEND, SET_PPD_PROT
                                                .EXTRN  HANDLER, SYS$SETDDIR
                                                .EXTRN  LGI$PROTECT_CLI
                                                .EXTRN  EXECUTE_IMAGE, LIB$P1_MERGE
                                                .EXTRN  SYS$FIND_HELD, EXE$DEANONPAGED
                                                .EXTRN  EXE$ALONONPAGED
                                                .EXTRN  EXE$CRE_JGTABLE
                                                .EXTRN  TERMINAL_DEVICE
                                                .EXTRN  TERM_NAME, DEV_CHAR_2
                                                .EXTRN  DEV_DEP_2, PCB_STS
                                                .EXTRN  JOB_TYPE, UAF_RECORD
                                                .EXTRN  SYS$INPUT, CLI_NAME
                                                .EXTRN  CLI_NAME_BUFFER
                                                .EXTRN  TABLE_NAME, TABLE_NAME_BUFFER
                                                .EXTRN  DISK_NAME, COM_NAME
```

```
                                .EXTRN  COM_NEGATED, SUBPROCESS
                                .EXTRN  IMAGE_ACTIVATE, MMG$IMGHDRBUF
                                .EXTRN  CTL$GL_PCB, CTL$GL_CCBBASE
                                .EXTRN  CTL$GL_UAF_FLAGS
                                .EXTRN  CTL$GT_CLINAME, CTL$GT_TABLENAME
                                .EXTRN  CTL$GT_SPAWNCLI
                                .EXTRN  CTL$GT_SPAWNTABLE
                                .EXTRN  CTL$AG_CMEDATA, CTL$AG_CLIMAGE
                                .EXTRN  CTL$AG_CLITABLE
                                .EXTRN  CTL$AG_CLIDATA, LGI$_CLIFAIL
                                .EXTRN  LGI$_CLIPROT, LGI$_CLITBLFAIL
                                .EXTRN  LGI$_CLITBLPROT
                                .EXTRN  LGI$_CLISYMTBL, SYS$SETPRI
                                .EXTRN  SYS$CMKRNL, SYS$SETPRN

                                .PSECT  $CODE$,NOWRT,2

                        001C 00000      .ENTRY  INIT_USER, Save R2,R3,R4                ; 0714
                54 00000000G   00  9E 00002      MOVAB   SYS$CMKRNL, R4                  ;
                53 00000000G   00  9E 00009      MOVAB   UAF_RECORD, R3                  ;
                               5E  20 C2 00010      SUBL2   #32, SP                     ;
    08 00000000G               00  01 E0 00013      BBS     #1, PCB_STS+3, 1$           ; 0746
    13 00000000G               00  05 E1 0001B      BBC     #5, PCB_STS+2, 2$           ; 0747
                                   7E D4 00023 1$:  CLRL    -(SP)                       ; 0749
                               50  63 D0 00025      MOVL    UAF_RECORD, R0              ;
                           7E  0204  C0 9A 00028      MOVZBL  516(R0), -(SP)              ;
                                   7E 7C 0002D      CLRQ    -(SP)                       ;
        00000000G              00  04 FB 0002F      CALLS   #4, SYS$SETPRI              ;
                               50  63 D0 00036 2$:  MOVL    UAF_RECORD, R0              ; 0755
                           6E  0094  C0 9A 00039      MOVZBL  148(R0), DIRECTORY          ;
                     04    AE  0095  C0 9E 0003E      MOVAB   149(R0), DIRECTORY+4        ; 0756
                                   7E 7C 00044      CLRQ    -(SP)                       ; 0758
                                 08  AE 9F 00046      PUSHAB  DIRECTORY                   ;
                00000000G      00  03 FB 00049      CALLS   #3, SYS$SETDDIR             ;
                               50  00000000G  00 D0 00050      MOVL    DISK_NAME, R0       ; 0764
                                   0F 12 00057      BNEQ    3$                          ;
                               50  63 D0 00059      MOVL    UAF_RECORD, R0              ; 0767
                        08     AE  74  A0 9A 0005C      MOVZBL  116(R0), DEVICE         ;
                        0C     AE  75  A0 9E 00061      MOVAB   117(R0), DEVICE+4       ; 0768
                                   0C 11 00066      BRB     4$                          ; 0764
                        08     AE      50 D0 00068 3$:  MOVL    R0, DEVICE              ; 0772
                        0C     AE  00000000G  00 D0 0006C      MOVL    DISK_NAME+4, DEVICE+4 ; 0773
                                 08  AE D5 00074 4$:  TSTL    DEVICE                  ; 0776
                                   0E 13 00077      BEQL    5$                          ;
                                   01 DD 00079      PUSHL   #1                          ; 0778
                                 0C  AE 9F 0007B      PUSHAB  DEVICE                  ;
                               0000' CF 9F 0007E      PUSHAB  P.AAA                       ;
                0000V          CF      03 FB 00082      CALLS   #3, CREATE_LOGICAL      ;
                               52      63 D0 00087 5$:  MOVL    UAF_RECORD, R2          ; 0786
    04    A2                   20      20 3A 0008A      LOCC    #32, #32, 4(R2)         ;
                                   02 12 0008F      BNEQ    6$                          ;
                                   51 D4 00091      CLRL    R1                          ;
                                   51 D5 00093 6$:  TSTL    PTR                     ; 0788
                                   04 12 00095      BNEQ    7$                          ;
                               51      24  A2 9E 00097      MOVAB   36(R2), PTR         ; 0790
                               50      04  A2 9E 0009B 7$:  MOVAB   4(R2), R0           ; 0792
    18    AE                   51      50 C3 0009F      SUBL3   R0, PTR, USERNAME       ;
```

```
                    1C   AE        04   A2  9E 000A4          MOVAB   4(R2), USERNAME+4                ; 0793
                                  18   AE  9F 000A9          PUSHAB  USERNAME                         ; 0796
                               0000V   CF  9F 000AC          PUSHAB  SET_USERNAME
                                    64     02  FB 000B0      CALLS   #2, SYS$CMKRNL
                                    50     63  D0 000B3      MOVL    UAF_RECORD, R0                   ; 0803
                                    24     A0  DD 000B6      PUSHL   36(R0)
                               0000V   CF  9F 000B9          PUSHAB  SET_UIC
                                    64     02  FB 000BD      CALLS   #2, SYS$CMKRNL
                                           7E  D4 000C0      CLRL    -(SP)                            ; 0816
                               0000V   CF  9F 000C2          PUSHAB  SET_LNM_TABLES
                                    64     02  FB 000C6      CALLS   #2, SYS$CMKRNL
                                    09     50  E8 000C9      BLBS    STATUS, 8$                       ; F
                                           50  DD 000CC      PUSHL   STATUS                           ; 0818
                 00000000G   00            01  FB 000CE      CALLS   #1, LIB$STOP
                       10    AE            20  D0 000D5 8$:  MOVL    #32, ACCOUNT                     ; 0826
                14   AE                    34  C1 000D9      ADDL3   #52, UAF_RECORD, ACCOUNT+4       ; 0827
                                  10   AE  9F 000DE          PUSHAB  ACCOUNT                          ; 0830
                               0000V   CF  9F 000E1          PUSHAB  SET_ACCOUNT
                                    64     02  FB 000E5      CALLS   #2, SYS$CMKRNL
                 0A 00000000G   00            01  E0 000E8      BBS     #1, PPD+2, 9$                ; 0837
                                    18   AE  9F 000F0          PUSHAB  USERNAME                         ; 0839
                 00000000G   00            01  FB 000F3      CALLS   #1, SYS$SETPRN
                      0000V   CF            00  FB 000FA 9$:  CALLS   #0, MAKE_RIGHTSLISTS            ; 0844
                              05 00000000G  00  E9 000FF      BLBC    TERMINAL_DEVICE, 10$             ; 0849
                      0000V   CF            00  FB 00106      CALLS   #0, SET_TERM_NAME                ; 0850
                                           7E  D4 0010B 10$:  CLRL    -(SP)                            ; 0856
                               0000V   CF  9F 0010D          PUSHAB  INIT_KERNEL
                                    64     02  FB 00111      CALLS   #2, SYS$CMKRNL
                                               04 00114      RET                                      ; 0858

; Routine Size:  277 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  532     0859  1 ROUTINE init_kernel: NOVALUE =
;  533     0860  1
;  534     0861  1 !---
;  535     0862  1 !
;  536     0863  1 !       Initialize process context in kernel mode.
;  537     0864  1 !
;  538     0865  1 ! Inputs:
;  539     0866  1 !
;  540     0867  1 !       Access mode is kernel.
;  541     0868  1 !
;  542     0869  1 !       uaf_record = Address of UAF record for user (must be non-zero)
;  543     0870  1 !
;  544     0871  1 ! Outputs:
;  545     0872  1 !
;  546     0873  1 !       None
;  547     0874  1 !---
;  548     0875  1
;  549     0876  2 BEGIN
;  550     0877  2
;  551     0878  2 STRUCTURE
;  552     0879  2     threebytevector [i; n, ext=0] =
;  553     0880  2         [n*3]
;  554     0881  2         (threebytevector+i*3)<0, 24, ext>;
;  555     0882  2
;  556     0883  2 EXTERNAL
;  557     0884  2     ctl$gl_phd,                         ! Address of process header
;  558     0885  2     ctl$gl_wspeak,                      ! Peak working set size
;  559     0886  2     ctl$gl_virtpeak,                    ! Peak page file usage
;  560     0887  2     ctl$gq_procpriv,                    ! Process permanent privileges
;  561     0888  2     pfn$gl_phypgcnt,                    ! Total physical pages of memory
;  562     0889  2     sch$gl_freelim,
;  563     0890  2     sgn$gl_maxwscnt;                    ! SYSGEN parameter WSMAX
;  564     0891  2
;  565     0892  2 LOCAL
;  566     0893  2     pcb:        REF BBLOCK,             ! Address of PCB
;  567     0894  2     phd:        REF BBLOCK,             ! Address of PHD
;  568     0895  2     jib:        REF BBLOCK,             ! Address of JIB
;  569     0896  2     arb:        REF BBLOCK,             ! Address of ARB
;  570     0897  2     available_memory;                   ! Amount of available physical memory
;  571     0898  2
;  572     0899  2 !
;  573     0900  2 ! Define a vector structure over the UAF hourly restriction fields
;  574     0901  2 ! for quick reference.
;  575     0902  2 !
;  576     0903  2 $ASSUME (jib$c_network, EQL, 1);
;  577     0904  2 $ASSUME (jib$c_batch, EQL, 2);
;  578     0905  2 $ASSUME (jib$c_local, EQL, 3);
;  579     0906  2 $ASSUME (jib$c_dialup, EQL, 4);
;  580     0907  2 $ASSUME (jib$c_remote, EQL, 5);
;  581     0908  2 $ASSUME ($BYTEOFFSET (uaf$b_network_access_s), EQL, $BYTEOFFSET (uaf$b_network_access_p)+3);
;  582     0909  2 $ASSUME ($BYTEOFFSET (uaf$b_batch_access_p), EQL, $BYTEOFFSET (uaf$b_network_access_s)+3);
;  583     0910  2 $ASSUME ($BYTEOFFSET (uaf$b_batch_access_s), EQL, $BYTEOFFSET (uaf$b_batch_access_p)+3);
;  584     0911  2 $ASSUME ($BYTEOFFSET (uaf$b_local_access_p), EQL, $BYTEOFFSET (uaf$b_batch_access_s)+3);
;  585     0912  2 $ASSUME ($BYTEOFFSET (uaf$b_local_access_s), EQL, $BYTEOFFSET (uaf$b_local_access_p)+3);
;  586     0913  2 $ASSUME ($BYTEOFFSET (uaf$b_dialup_access_p), EQL, $BYTEOFFSET (uaf$b_local_access_s)+3);
;  587     0914  2 $ASSUME ($BYTEOFFSET (uaf$b_dialup_access_s), EQL, $BYTEOFFSET (uaf$b_dialup_access_p)+3);
;  588     0915  2 $ASSUME ($BYTEOFFSET (uaf$b_remote_access_p), EQL, $BYTEOFFSET (uaf$b_dialup_access_s)+3);
```

```
;  589          0916  2 $ASSUME ($BYTEOFFSET (uaf$b_remote_access_s), EQL, $BYTEOFFSET (uaf$b_remote_access_p)+3);
;  590          0917  2
;  591          0918  2 BIND
;  592          0919  2     restrict_vector = uaf_record[uaf$b_network_access_p]
;  593          0920  2                    : threebytevector;
;  594          0921  2
;  595          0922  2 ctl$gl_wspeak = 0;                       ! Initialize peak working set usage
;  596          0923  2 ctl$gl_virtpeak = 0;                     ! Initialize peak page file usage
;  597          0924  2
;  598          0925  2 pcb = .ctl$gl_pcb;                       ! Get address of PCB
;  599          0926  2 jib = .pcb [pcb$l_jib];                  ! Get address of JIB
;  600          0927  2
;  601          0928  2 jib[jib$b_jobtype] = .job_type;
;  602          0929  2
;  603          0930  2 ctl$gl_uaf_flag = .uaf_record [uaf$l_flags];
;  604          0931  2
;  605          0932  2 IF .uaf_record [uaf$v_audit]
;  606          0933  2 THEN
;  607          0934  3     BEGIN
;  608          0935  3     pcb_sts [$BITPOSITION(pcb$v_secaudit)] = 1;
;  609          0936  3     pcb [pcb$v_secaudit] = 1;
;  610          0937  2     END;
;  611          0938  2
;  612          0939  2 pcb [pcb$w_biolm] = .uaf_record [uaf$w_biolm];
;  613          0940  2 pcb [pcb$w_biocnt] = .uaf_record [uaf$w_biolm];
;  614          0941  2 pcb [pcb$w_diolm] = .uaf_record [uaf$w_diolm];
;  615          0942  2 pcb [pcb$w_diocnt] = .uaf_record [uaf$w_diolm];
;  616          0943  2 jib [jib$l_bytlm] = .uaf_record [uaf$l_bytlm]
;  617          0944  2                   + (.jib[jib$l_bytlm] - .jib [jib$l_org_bytlm]);
;  618          0945  2 jib [jib$l_bytcnt] = .uaf_record [uaf$l_bytlm]
;  619          0946  2                   + (.jib[jib$l_bytcnt] - .jib [jib$l_org_bytlm]);
;  620          0947  2 jib [jib$w_prclim] = .uaf_record [uaf$w_prccnt];
;  621          0948  2 jib [jib$w_filcnt] = .uaf_record [uaf$w_fillm]
;  622          0949  2                   + (.jib [jib$w_filcnt] - .jib [jib$w_fillm]);
;  623          0950  2 jib [jib$w_fillm] = .uaf_record [uaf$w_fillm];
;  624          0951  2 IF .job_type NEQ jib$c_detached
;  625          0952  2 THEN
;  626          0953  3     BEGIN
;  627          0954  3     jib [jib$b_daytypes] = .uaf_record [uaf$b_primedays];
;  628          0955  3     jib [jib$l_pdayhours] = .restrict_vector [(.job_type-1)*2];
;  629          0956  3     jib [jib$l_odayhours] = .restrict_vector [(.job_type-1)*2+1];
;  630          0957  2     END;
;  631          0958  2
;  632          0959  2 phd = .ctl$gl_phd;                       ! Get address of PHD
;  633          0960  2
;  634          0961  2 available_memory = MINU(.pfn$gl_phypgcnt - .sch$gl_freelim,
;  635          0962  2                       .sgn$gl_maxwscnt);
;  636          0963  2
;  637          0964  2 phd [phd$w_wsquota] = .phd [phd$w_wslist]-1
;  638          0965  2                   + MINU(.uaf_record [uaf$l_wsquota], .available_memory);
;  639          0966  2
;  640          0967  2 phd [phd$w_wsextent] = .phd [phd$w_wslist]-1
;  641          0968  2                   + MINU(.uaf_record [uaf$l_wsextent], .available_memory);
;  642          0969  2
;  643          0970  2 phd [phd$w_wsextent] = MAXU(.phd [phd$w_wsquota],.phd [phd$w_wsextent]);
;  644          0971  2
;  645          0972  2 phd [phd$w_wsauth] = .phd [phd$w_wsquota];
```

```
;  646          0973  2
;  647          0974  2 phd [phd$w_wsauthext] = .phd [phd$w_wsextent];
;  648          0975  2
;  649          0976  2 phd [phd$w_dfwscnt] = MINU(.phd [phd$w_wsauth],
;  650          0977  2                           .phd [phd$w_wslist]-1 - .uaf_record [uaf$l_dfwscnt]);
;  651          0978  2 jib [jib$l_pgflcnt] = .jib [jib$l_pgflcnt]
;  652          0979  2                     + (.uaf_record [uaf$l_pgflquota] - .jib [jib$l_pgflquota]);
;  653          0980  2 jib [jib$l_pgflquota] = .uaf_record [uaf$l_pgflquota];
;  654          0981  2
;  655          0982  2 phd [phd$w_astlm] = .uaf_record [uaf$w_astlm];
;  656          0983  2 pcb [pcb$w_astcnt] = .uaf_record [uaf$w_astlm];
;  657          0984  2 jib [jib$w_tqlm] = .uaf_record [uaf$w_tqcnt];
;  658          0985  2 jib [jib$w_tqcnt] = .uaf_record [uaf$w_tqcnt];
;  659          0986  2 phd [phd$l_cpulim] = .uaf_record [uaf$l_cputim];
;  660          0987  2 jib [jib$l_cpulim] = .uaf_record [uaf$l_cputim];
;  661          0988  2 jib [jib$w_enqcnt] = .uaf_record [uaf$w_enqlm]
;  662          0989  2                               + (.jib [jib$w_enqcnt] - .jib [jib$w_enqlm]);
;  663          0990  2 jib [jib$w_enqlm] = .uaf_record [uaf$w_enqlm];
;  664          0991  2 jib [jib$w_shrfcnt] = .uaf_record [uaf$w_shrfillm]
;  665          0992  2                                 + (.jib [jib$w_shrfcnt] - .jib [jib$w_shrflim]);
;  666          0993  2 jib [jib$w_shrflim] = .uaf_record [uaf$w_shrfillm];
;  667          0994  2 jib [jib$l_pbytlim] = .uaf_record [uaf$l_pbytlm];
;  668          0995  2 jib [jib$l_pbytcnt] = .uaf_record [uaf$l_pbytlm];
;  669          0996  2 jib [jib$w_maxjobs] = .uaf_record [uaf$w_maxjobs];
;  670          0997  2 jib [jib$w_maxdetach] = .uaf_record [uaf$w_maxdetach];
;  671          0998  2
;  672          0999  2 ! The AUTHPRI cell exists in two places. The $SETPRI system service uses
;  673          1000  2 ! the PCB cell but the PHD cell must exist forever because that is where
;  674          1001  2 ! the JPI item code believes that AUTHPRI is located.
;  675          1002  2
;  676          1003  2 pcb [pcb$b_authpri] = .pcb [pcb$b_prib];     ! Reset authorized priority
;  677          1004  2 phd [phd$b_authpri] = .pcb [pcb$b_prib];     ! ... in both of its homes
;  678          1005  2
;  679          1006  2 arb = .pcb [pcb$l_arb];                     ! Get address of ARB
;  680          1007  2
;  681          1008  2 move_quad(uaf_record [uaf$q_priv], phd [phd$q_authpriv]);
;  682          1009  2 move_quad(uaf_record [uaf$q_priv], arb [arb$q_priv]);
;  683          1010  2 move_quad(uaf_record [uaf$q_def_priv], ctl$gq_procpriv);
;  684          1011  2
;  685          1012  1 END;


                                                .EXTRN  CTL$GL_PHD, CTL$GL_WSPEAK
                                                .EXTRN  CTL$GL_VIRTPEAK
                                                .EXTRN  CTL$GQ_PROCPRIV
                                                .EXTRN  PFN$GL_PHYPGCNT
                                                .EXTRN  SCH$GL_FREELIM, SGN$GL_MAXWSCNT

                                00FC 00000 INIT_KERNEL:
                                                .WORD   Save R2,R3,R4,R5,R6,R7                  ; 0859
                57 00000000G   00   9E 00002            MOVAB   SGN$GL_MAXWSCNT, R7             ;
                52 00000000G   00   D0 00009            MOVL    UAF_RECORD, R2                  ; 0919
                   00000000G   00   D4 00010            CLRL    CTL$GL_WSPEAK                   ; 0922
                   00000000G   00   D4 00016            CLRL    CTL$GL_VIRTPEAK                 ; 0923
                54 00000000G   00   D0 0001C            MOVL    CTL$GL_PCB, PCB                 ; 0925
                50      0080   C4   D0 00023            MOVL    128(PCB), JIB                   ; 0926
```

```
                                51 00000000G   00    D0 00028          MOVL    JOB_TYPE, R1                                    ; 0928
                             68    A0          51    90 0002F          MOVB    R1, 104(JIB)                                    ;
                 00000000G   00      01D4      C2    D0 00033          MOVL    468(R2), CTL$GL_UAF_FLAGS                       ; 0930
                       0B      01D5  C2        03    E1 0003C          BBC     #3, 469(R2), 1$                                 ; 0932
                 00000000G   00              08    88 00042          BISB2   #8, PCB_STS+3                                   ; 0935
                          27 A4              08    88 00049          BISB2   #8, 39(PCB)                                     ; 0936
                          3C A4      020E    C2    B0 0004D 1$:      MOVW    526(R2), 60(PCB)                                ; 0939
                          3A A4      020E    C2    B0 00053          MOVW    526(R2), 58(PCB)                                ; 0940
                          40 A4      0210    C2    B0 00059          MOVW    528(R2), 64(PCB)                                ; 0941
                          3E A4      0210    C2    B0 0005F          MOVW    528(R2), 62(PCB)                                ; 0942
                       53 24 A0        6C    A0    C3 00065          SUBL3   108(JIB), 36(JIB), R3                           ; 0944
                          24 A0      0230 D243   9E 0006B          MOVAB   @560(R2)[R3], 36(JIB)                           ;
                       53 20 A0        6C    A0    C3 00072          SUBL3   108(JIB), 32(JIB), R3                           ; 0946
                          20 A0      0230 D243   9E 00078          MOVAB   @560(R2)[R3], 32(JIB)                           ;
                          46 A0      020C    C2    B0 0007F          MOVW    524(R2), 70(JIB)                                ; 0947
                             53        30    A0    3C 00085          MOVZWL  48(JIB), R3                                     ; 0949
                             55        32    A0    3C 00089          MOVZWL  50(JIB), R5                                     ;
                             53              55    C2 0008D          SUBL2   R5, R3                                          ;
                    30 A0                    53      0218    C2    A1 00090          ADDW3   536(R2), R3, 48(JIB)                    ;
                          32 A0              53      0218    C2    B0 00097          MOVW    536(R2), 50(JIB)                        ; 0950
                                             51              D5 0009D          TSTL    R1                                      ; 0951
                                             24              13 0009F          BEQL    2$                                      ;
                          0B A0              53      0202    C2    90 000A1          MOVB    514(R2), 11(JIB)                        ; 0954
                       53                    51              01    78 000A7          ASHL    #1, R1, R3                              ; 0955
                       51                    53              03    C5 000AB          MULL3   #3, R3, R1                              ;
    60 A0      01D2 C241                     18              00    EF 000AF          EXTZV   #0, #24, 466(R2)[R1], 96(JIB)           ;
                       51                    53              03    C5 000B8          MULL3   #3, R3, R1                              ; 0956
    64 A0      01D5 C241                     18              00    EF 000BC          EXTZV   #0, #24, 469(R2)[R1], 100(JIB)          ;
                                             51 00000000G    00    D0 000C5 2$:      MOVL    CTL$GL_PHD, PHD                         ; 0959
                    53 00000000G             00 00000000G    00    C3 000CC          SUBL3   SCH$GL_FREELIM, PFN$GL_PHYPGCNT, R3     ; 0961
                                             67              53    D1 000D8          CMPL    R3, SGN$GL_MAXWSCNT                     ; 0962
                                             03              1B 000DB          BLEQU   3$                                      ;
                                             53              67    D0 000DD          MOVL    SGN$GL_MAXWSCNT, R3                     ;
                                             55              53    D0 000E0 3$:      MOVL    R3, AVAILABLE_MEMORY                    ; 0961
                                             53      021C    C2    D0 000E3          MOVL    540(R2), R3                             ; 0965
                                             55              53    D1 000E8          CMPL    R3, AVAILABLE_MEMORY                    ;
                                             03              1B 000EB          BLEQU   4$                                      ;
                                             53              55    D0 000ED          MOVL    AVAILABLE_MEMORY, R3                    ;
                                             56        08    A1    3C 000F0 4$:      MOVZWL  8(PHD), R6                              ;
                                             53              56    C0 000F4          ADDL2   R6, R3                                  ;
                 18 A1                       53              01    A3 000F7          SUBW3   #1, R3, 24(PHD)                         ;
                                             53      0224    C2    D0 000FC          MOVL    548(R2), R3                             ; 0968
                                             55              53    D1 00101          CMPL    R3, AVAILABLE_MEMORY                    ;
                                             03              1B 00104          BLEQU   5$                                      ;
                                             53              55    D0 00106          MOVL    AVAILABLE_MEMORY, R3                    ;
                                             56        08    A1    3C 00109 5$:      MOVZWL  8(PHD), R6                              ;
                                             53              56    C0 0010D          ADDL2   R6, R3                                  ;
                 16 A1                       53              01    A3 00110          SUBW3   #1, R3, 22(PHD)                         ;
                                             53        18    A1    3C 00115          MOVZWL  24(PHD), R3                             ; 0970
                                             53        16    A1    B1 00119          CMPW    22(PHD), R3                             ;
                                             04              1B 0011D          BLEQU   6$                                      ;
                                             53        16    A1    3C 0011F          MOVZWL  22(PHD), R3                             ;
                          16 A1                      53      B0 00123 6$:      MOVW    R3, 22(PHD)                             ;
                          0A A1              18      A1      B0 00127          MOVW    24(PHD), 10(PHD)                        ; 0972
                          14 A1              16      A1      B0 0012C          MOVW    22(PHD), 20(PHD)                        ; 0974
                                             53        08    A1    3C 00131          MOVZWL  8(PHD), R3                              ; 0977
                                             53      0220    C2    C0 00135          ADDL2   544(R2), R3                             ;
```

```
                    55        FF    A3   9E 0013A          MOVAB   -1(R3), R5                                      ;
                    53        0A    A1   3C 0013E          MOVZWL  10(PHD), R3                                     ;
                    55              53   D1 00142          CMPL    R3, R5                                          ;
                                    03   1B 00145          BLEQU   7$                                              ;
                    53              55   D0 00147          MOVL    R5, R3                                          ;
                  1A    A1              53   B0 0014A 7$:      MOVW    R3, 26(PHD)                                     ; 0976
            53  0228    C2        3B    A0   C3 0014E          SUBL3   56(JIB), 552(R2), R3                            ; 0979
                  3C    A0              53   C0 00155          ADDL2   R3, 60(JIB)                                     ;
                  38    A0      0228    C2   D0 00159          MOVL    552(R2), 56(JIB)                                ; 0980
                  40    A1      0214    C2   B0 0015F          MOVW    532(R2), 64(PHD)                                ; 0982
                  38    A4      0214    C2   B0 00165          MOVW    532(R2), 56(PCB)                                ; 0983
                  36    A0      0212    C2   B0 0016B          MOVW    530(R2), 54(JIB)                                ; 0984
                  34    A0      0212    C2   B0 00171          MOVW    530(R2), 52(JIB)                                ; 0985
                  5C    A1      022C    C2   D0 00177          MOVL    556(R2), 92(PHD)                                ; 0986
                  40    A0      022C    C2   D0 0017D          MOVL    556(R2), 64(JIB)                                ; 0987
                    53      4C    A0   3C 00183          MOVZWL  76(JIB), R3                                     ; 0989
                    55      4E    A0   3C 00187          MOVZWL  78(JIB), R5                                     ;
                    53              55   C2 0018B          SUBL2   R5, R3                                          ;
            4C  A0          53      0216    C2   A1 0018E          ADDW3   534(R2), R3, 76(JIB)                            ;
                  4E    A0      0216    C2   B0 00195          MOVW    534(R2), 78(JIB)                                ; 0990
                    53      48    A0   3C 0019B          MOVZWL  72(JIB), R3                                     ; 0992
                    55      4A    A0   3C 0019F          MOVZWL  74(JIB), R5                                     ;
                    53              55   C2 001A3          SUBL2   R5, R3                                          ;
            48  A0          53      021A    C2   A1 001A6          ADDW3   538(R2), R3, 72(JIB)                            ;
                  4A    A0      021A    C2   B0 001AD          MOVW    538(R2), 74(JIB)                                ; 0993
                  2C    A0      0234    C2   D0 001B3          MOVL    564(R2), 44(JIB)                                ; 0994
                  28    A0      0234    C2   D0 001B9          MOVL    564(R2), 40(JIB)                                ; 0995
                  50    A0      0206    C2   B0 001BF          MOVW    518(R2), 80(JIB)                                ; 0996
                  52    A0      020A    C2   B0 001C5          MOVW    522(R2), 82(JIB)                                ; 0997
                  2B    A4        2F    A4   90 001CB          MOVB    47(PCB), 43(PCB)                                ; 1003
                010C    C1        2F    A4   90 001D0          MOVB    47(PCB), 268(PHD)                               ; 1004
                                50      008C    C4   D0 001D6          MOVL    140(PCB), ARB                           ; 1006
                010E    C1      019C    C2   7D 001DB          MOVQ    412(R2), 224(PHD)                               ; 1008
                                60      019C    C2   7D 001E2          MOVQ    412(R2), (ARB)                          ; 1009
                00000000G   00      01A4    C2   7D 001E7          MOVQ    420(R2), CTL$GQ_PROCPRIV                        ; 1010
                                                     04 001F0          RET                                             ; 1012

; Routine Size:  497 bytes,    Routine Base:  $CODE$ + 0115
```

```
;  687          1013  1 GLOBAL ROUTINE init_cli: NOVALUE =
;  688          1014  1
;  689          1015  1 !---
;  690          1016  1 !
;  691          1017  1 !       Initialize the CLI by mapping it into P1 space and
;  692          1018  1 !       setting up the communication region.  The logical
;  693          1019  1 !       names PROC1-N are defined to specify initialization
;  694          1020  1 !       command procedures that the CLI should execute.
;  695          1021  1 !
;  696          1022  1 ! Inputs:
;  697          1023  1 !
;  698          1024  1 !       cli_name = Address of descriptor of CLI name
;  699          1025  1 !       table_name = Address of descriptor of command table
;  700          1026  1 !       uaf_record = Address of UAF record for user, if any
;  701          1027  1 !
;  702          1028  1 ! Outputs:
;  703          1029  1 !
;  704          1030  1 !       None
;  705          1031  1 !---
;  706          1032  1
;  707          1033  2 BEGIN
;  708          1034  2
;  709          1035  2 BIND
;  710          1036  2     dcl_string = UPLIT BYTE('DCL');
;  711          1037  2
;  712          1038  2 LOCAL
;  713          1039  2     restricted_user;            !True if defcli/captive ir p1 or uaf
;  714          1040  2 !
;  715          1041  2 ! Setup fields of the PPD communication region
;  716          1042  2 !
;  717          1043  2
;  718          1044  2 IF .uaf_record NEQ 0                    ! If UAF record valid
;  719          1045  2 THEN
;  720          1046  3     BEGIN                                                          45
;  721          1047  3     IF .uaf_record [uaf$v_disctly]      ! If ctrl/y initially disabled,
;  722          1048  3     OR .uaf_record [uaf$v_captive]      ! (CAPTIVE implies disable ctrl/y
;  723          1049  3     THEN ppd [ppd$v_noctly] = true;     ! then indicate that to CLI
;  724          1050  3     IF .uaf_record [uaf$v_captive]      ! Propagate the captive bit    00
;  725          1051  3     THEN ppd [ppd$v_captive] = true;    !  to PPD communication region
;  726          1052  2     END;
;  727          1053  2
;  728          1054  3 restricted_user = (IF .uaf_record NEQ 0
;  729          1055  4                  THEN (.uaf_record [uaf$v_defcli]
;  730          1056  4                       OR .uaf_record [uaf$v_captive])
;  731          1057  4                  ELSE (.ctl$gl_uaf_flags [$BITPOSITION(uaf$v_defcli)]
;  732          1058  2                       OR .ctl$gl_uaf_flags [$BITPOSITION(uaf$v_captive)]));
;  733          1059  2
;  734          1060  2 !
;  735          1061  2 ! Setup system-wide command procedure, defined by logname SYS$SYLOGIN
;  736          1062  2 !
;  737          1063  2
;  738          1064  2 IF NOT .subprocess
;  739          1065  2 THEN
;  740          1066  3     BEGIN
;  741          1067  3
;  742          1068  3     LOCAL
;  743          1069  3         buffer:                 VECTOR [128,BYTE],
```

```
;  744          1070  3          desc:                       VECTOR [2],
;  745          1071  3          trnlnm_item_list:           BLOCK [1+3+1,LONG];     ! Item list for 1 item
;  746          1072  3
;  747          1073  3      trnlnm_item_list[0, 0,16,0] = (desc[0] = %ALLOCATION(buffer));
;  748          1074  3      trnlnm_item_list[0,16,16,0] = lnm$_string;  ! Fetch name's value string
;  749          1075  3      trnlnm_item_list[1, 0,32,0] = (desc[1] = buffer);
;  750          1076  3      trnlnm_item_list[2, 0,32,0] = desc[0];
;  751          1077  3      trnlnm_item_list[3, 0,32,0] = 0;
;  752          1078  3
;  753        P 1079  3      IF $TRNLNM(TABNAM = %ASCID 'LNM$SYSTEM_TABLE',
;  754        P 1080  3                 LOGNAM = %ASCID 'SYS$SYLOGIN',
;  755        P 1081  3                 ACMODE = UPLIT(psl$c_exec),
;  756          1082  4                 ITMLST = trnlnm_item_list)
;  757          1083  3      EQL ss$_normal
;  758          1084  3      THEN
;  759          1085  3          setup_login_proc(desc);                 ! Tell CLI to execute it
;  760          1086  3
;  761          1087  2      END;
;  762          1088  2
;  763          1089  2 !
;  764          1090  2 ! If not a subprocess, setup the initial command procedure to execute.
;  765          1091  2 !
;  766          1092  2
;  767          1093  2 IF .com_name [0] EQL 0                  ! If no command procedure to execute
;  768          1094  2     AND NOT .com_negated                ! and not explicitly negated,
;  769          1095  2     AND .uaf_record NEQ 0               ! and if UAF record valid,
;  770          1096  2 THEN
;  771          1097  3     BEGIN
;  772          1098  3     com_name [1] = uaf_record [$BYTEOFFSET(uaf$t_lgicmd)+1,0,0,0];
;  773          1099  3     com_name [0] = CH$RCHAR(uaf_record [uaf$t_lgicmd]);
;  774          1100  3
;  775          1101  3     IF .com_name [0] EQL 0              ! If no default in UAF
;  776          1102  3     THEN
;  777          1103  4         BEGIN
;  778          1104  4         com_name [1] = UPLIT BYTE('LOGIN');
;  779          1105  4         com_name [0] = 5;
;  780          1106  3         END;
;  781          1107  2     END;
;  782          1108  2
;  783          1109  2 IF .com_name [0] NEQ 0                  ! If user has login procedure,
;  784          1110  2     AND NOT .subprocess                 ! and not a subprocess
;  785          1111  2 THEN
;  786          1112  2     setup_login_proc(com_name);         ! Tell CLI to execute it
;  787          1113  2
;  788          1114  2 !
;  789          1115  2 ! Get the name of the CLI and tables to map.  If /CLI or /TABLE was
;  790          1116  2 ! specified on the username prompt, then cli_name or table_name will already
;  791          1117  2 ! have initial values.
;  792          1118  2 !
;  793          1119  2
;  794          1120  2 !
;  795          1121  2 ! If no cli specified, and not (captive or defcli), and image activator
;  796          1122  2 ! gave us a cli in cmedata, then use it.
;  797          1123  2 !
;  798          1124  2 IF .cli_name [0] EQL 0                          !If no cli name specified
;  799          1125  2 AND NOT .restricted_user
;  800          1126  3 AND (.ctl$ag_cmedata [0] NEQ 0)                 !And $imgact stored cli name
```

```
;   801          1127  2 THEN
;   802          1128  3     BEGIN
;   803          1129  3     cli_name [0] = .ctl$ag_cmedata [0];
;   804          1130  3     cli_name [1] = ctl$ag_cmedata [1];
;   805          1131  2     END;
;   806          1132  2 !
;   807          1133  2 ! If we don't have a CLI name yet then get CLI and tables from spawn info.
;   808          1134  2 !
;   809          1135  2 IF .cli_name [0] EQL 0                  ! If no CLI has been specified yet
;   810          1136  2 AND NOT .restricted_user                ! and user can specify cli
;   811          1137  2 THEN                                    ! then get it from spawn information
;   812          1138  3     BEGIN
;   813          1139  3     cli_name [0] = .ctl$gt_spawncli [0];
;   814          1140  3     cli_name [1] = ctl$gt_spawncli [1];                                    ; R
;   815          1141  3     IF .table_name [0] EQL 0             ! If no tables have been specified yet
;   816          1142  3     THEN                                 ! then get tables from spawn, too
;   817          1143  4         BEGIN
;   818          1144  4         table_name [0] = .ctl$gt_spawntable [0];
;   819          1145  4         table_name [1] = ctl$gt_spawntable [1];
;   820          1146  3         END;
;   821          1147  2     END;
;   822          1148  2
;   823          1149  2 !
;   824          1150  2 ! If we have a UAF record, but not a CLI or table name yet, or the UAF default
;   825          1151  2 ! CLI flags are set, then get the CLI and tables from the UAF record.
;   826          1152  2 !
;   827          1153  2 IF .uaf_record NEQ 0                   ! If the UAF record is valid
;   828          1154  3 AND (.cli_name [0] EQL 0               !  and no CLI has been specified yet
;   829          1155  3  OR .restricted_user)                  !   or we must use the default CLI
;   830          1156  2 THEN                                   ! Then get the CLI from the UAF
;   831          1157  3     BEGIN
;   832          1158  3     cli_name [0] = .VECTOR [uaf_record [uaf$t_defcli], 0; ,BYTE];
;   833          1159  3     cli_name [1] = uaf_record [uaf$t_defcli] + 1;
;   834          1160  3     IF .table_name [0] EQL 0             ! If no tables have been specified yet
;   835          1161  3     OR .restricted_user                  !  or we must use the default CLI
;   836          1162  3     THEN                                 ! Then the tables from the UAF too
;   837          1163  4         BEGIN
;   838          1164  4         table_name [0] = .VECTOR [uaf_record [uaf$t_clitables], 0; ,BYTE];
;   839          1165  4         table_name [1] = uaf_record [uaf$t_clitables] + 1;
;   840          1166  3         END;
;   841          1167  2     END;
;   842          1168  2
;   843          1169  2 !
;   844          1170  2 ! If we don't have a CLI name yet, or we don't have a UAF record but the UAF
;   845          1171  2 ! flags in P1 space indicate that we must use the default CLI, then get both
;   846          1172  2 ! CLI and tables from P1 space.
;   847          1173  2 !
;   848          1174  2 IF .cli_name [0] EQL 0                  ! If no CLI has been specified yet
;   849          1175  3 OR (                                    !  or,
;   850          1176  3     .uaf_record EQL 0                   ! There is no UAF record
;   851          1177  4     AND (                               !  but, P1 flags specify
;   852          1178  4          .restricted_user               !  user is restricted
;   853          1179  4         )
;   854          1180  3    )
;   855          1181  2 THEN                                   ! Then get CLI from P1 space
;   856          1182  3     BEGIN
;   857          1183  3     cli_name [0] = .ctl$gt_cliname [0];
```

```
;  858     1184  3     cli_name [1] = ctl$gt_cliname [1];
;  859     1185  3     IF .table_name [0] EQL 0                ! If no tables have been specified yet
;  860     1186  4     OR (                                    !  or,
;  861     1187  4         .uaf_record EQL 0                   ! There is no UAF record
;  862     1188  5         AND (                               !  but, P1 flags specify
;  863     1189  5              .restricted_user               !   that user is restricted
;  864     1190  5             )
;  865     1191  4        )
;  866     1192  3     THEN                                    ! Then get tables from P1 space too
;  867     1193  4         BEGIN
;  868     1194  4         table_name [0] = .ctl$gt_tablename [0];
;  869     1195  4         table_name [1] = ctl$gt_tablename [1];
;  870     1196  3         END;
;  871     1197  2     END;
;  872     1198  2
;  873     1199  2 !
;  874     1200  2 ! If we still don't have a CLI yet, the use DCL as the final default.  Or,
;  875     1201  2 ! if this is a network process, force DCL, since network requires it.
;  876     1202  2 !
;  877     1203  3 IF (.cli_name[0] EQL 0)                     ! If no CLI has been specified yet
;  878     1204  2     OR .pcb_sts[$BITPOSITION(pcb$v_netwrk)]
;  879     1205  2 THEN                                        ! Then use DCL as the final default
;  880     1206  3     BEGIN
;  881     1207  3     cli_name [0] = 3;
;  882     1208  3     cli_name [1] = dcl_string;
;  883     1209  3     IF .pcb_sts[$BITPOSITION(pcb$v_netwrk)] ! Use default tables if network
;  884     1210  3         THEN table_name [0] = 0;
;  885     1211  2     END;
;  886     1212  2
;  887     1213  2 !
;  888     1214  2 ! Ensure that the CLI image name is prefixed with a device name. We do
;  889     1215  2 ! not want the CLI image to be logical name translatable.
;  890     1216  2 !
;  891     1217  2 IF CH$FAIL (CH$FIND_CH (.cli_name[0], .cli_name[1], ':'))
;  892     1218  2 THEN
;  893     1219  3     BEGIN
;  894     1220  3     CH$COPY (.cli_name[0], .cli_name[1], ' ',
;  895     1221  3              80 - 11, cli_name_buffer[11]);
;  896     1222  3     CH$MOVE (11, UPLIT BYTE ('SYS$SYSTEM:'), cli_name_buffer);
;  897     1223  3     cli_name[0] = .cli_name[0] + 11;
;  898     1224  3     cli_name[1] = cli_name_buffer;
;  899     1225  2     END;
;  900     1226  2
;  901     1227  2 !
;  902     1228  2 ! If no CLI tables name is yet specified and the CLI name is for the form
;  903     1229  2 ! "SYS$SYSTEM:cliname", then create a CLI tables name of the form
;  904     1230  2 ! "clinameTABLES".
;  905     1231  2 !
;  906     1232  2 IF .table_name [0] EQL 0                    ! If no CLI tables name yet
;  907     1233  2 AND .cli_name [0] GTRU 11                   !  and CLI name
;  908     1234  2 AND CH$EQL(11, .cli_name [1],               !   starts with "SYS$SYSTEM:",
;  909     1235  2              11, UPLIT BYTE('SYS$SYSTEM:'))
;  910     1236  2 THEN
;  911     1237  3     BEGIN                                   ! Form "clinameTABLES"
;  912     1238  3     CH$COPY(.cli_name [0] - 11, .cli_name [1] + 11,
;  913     1239  3             6, UPLIT BYTE('TABLES'),
;  914     1240  3             ' ',
```

```
;  915          1241  3              80, table_name_buffer);
;  916          1242  3      table_name [0] = .cli_name [0] - 11 + 6;
;  917          1243  3      table_name [1] = table_name_buffer;
;  918          1244  2      END;
;  919          1245  2
;  920          1246  2 !
;  921          1247  2 ! Map the CLI image into the control region.
;  922          1248  2 !
;  923          1249  2
;  924          1250  2 $CMEXEC(ROUTIN = map_cli);              ! Map the CLI image
;  925          1251  2
;  926          1252  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        4C  43  44  00010 P.AAC:  .ASCII  \DCL\                                   ;
                                                                    00013         .BLKB   1
4C  42  41  54  5F  4D  45  54  53  59  53  24  4D  4E  4C  00014 P.AAE:  .ASCII  \LNM$SYSTEM_TABLE\                              ;
                                                            45  00023                                                             ;
                                                          010E0010  00024 P.AAD:  .LONG   17694736                                ;
                                                          00000000' 00028         .ADDRESS P.AAE                                  ;
            00  4E  49  47  4F  4C  59  53  24  53  59  53  0002C P.AAG:  .ASCII  \SYS$SYLOGIN\<0>                                ;
                                                          010E000B  00038 P.AAF:  .LONG   17694731                                ;
                                                          00000000' 0003C         .ADDRESS P.AAG                                  ;
                                                          00000001  00040 P.AAH:  .LONG   1                                       ;
                                                    4E  49  47  4F  4C  00044 P.AAI:  .ASCII  \LOGIN\                             ;
                3A  4D  45  54  53  59  53  24  53  59  53  00049 P.AAJ:  .ASCII  \SYS$SYSTEM:\                                  ;
                3A  4D  45  54  53  59  53  24  53  59  53  00054 P.AAK:  .ASCII  \SYS$SYSTEM:\                                  ;
                                    53  45  4C  42  41  54  0005F P.AAL:  .ASCII  \TABLES\                                       ;

                                                                DCL_STRING=     P.AAC
                                                                .EXTRN  SYS$TRNLNM, SYS$CMEXEC

                                                                .PSECT  $CODE$,NOWRT,2

                                                          0FFC 00000     .ENTRY  INIT_CLI, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 1013
                                                                         R11                                                   ;
                                    5B 00000000G    00  9E 00002       MOVAB   COM_NAME, R11                                 ;
                                    5A 00000000G    00  9E 00009       MOVAB   TABLE_NAME, R10                               ;
                                    59 00000000G    00  9E 00010       MOVAB   CLI_NAME, R9                                  ;
                                    5E      FF68    CE  9E 00017       MOVAB   -152(SP), SP                                  ;
                                    50 00000000G    00  D0 0001C       MOVL    UAF_RECORD, R0                                ; 1044
                                                    52  D4 00023       CLRL    R2                                            ;
                                                    50  D5 00025       TSTL    R0                                            ;
                                                    20  13 00027       BEQL    2$                                            ;
                                                    52  D6 00029       INCL    R2                                            ;
                                    51      01D4    C0  9E 0002B       MOVAB   468(R0), R1                                   ; 1047
                                    04              61  E8 00030       BLBS    (R1), 1$                                      ;
                        12          61              03  E1 00033       BBC     #3, (R1), 2$                                  ; 1048
                            000000000G  00          01  88 00037 1$:    BISB2   #1, PPD+2                                     ; 1049
                        07          61              03  E1 0003E       BBC     #3, (R1), 2$                                  ; 1050
                            000000000G  00          08  88 00042       BISB2   #8, PPD+2                                     ; 1051
                                        13          52  E9 00049 2$:    BLBC    R2, 3$                                        ; 1054
52      01D4    C0                  01              01  EF 0004C       EXTZV   #1, #1, 468(R0), RESTRICTED_USER              ; 1056
51      01D4    C0                  01              03  EF 00053       EXTZV   #3, #1, 468(R0), R1                           ;
```

```
                                52              51  C8 0005A          BISL2   R1, RESTRICTED_USER                                     ;
                                                15  11 0005D          BRB     4$                                                      ; 1055
        52 00000000G  00                01      01  EF 0005F 3$:      EXTZV   #1, #1, CTL$GL_UAF_FLAGS, RESTRICTED_USER               ; 1058
        50 00000000G  00                01      03  EF 00068          EXTZV   #3, #1, CTL$GL_UAF_FLAGS, R0                            ;
                                        52      50  C8 00071          BISL2   R0, RESTRICTED_USER                                     ;
                                        44 00000000G    00  E8 00074 4$:      BLBS    SUBPROCESS, 5$                                  ; 1064
                              10        AE      80    8F  9A 0007B          MOVZBL  #128, DESC                                      ; 1073
                                        6E 00020080   8F  D0 00080          MOVL    #131200, TRNLNM_ITEM_LIST                       ;
                                        50      18    AE  9E 00087          MOVAB   BUFFER, R0                                      ; 1075
                              14        AE            50  D0 0008B          MOVL    R0, DESC+4                                      ;
                              04        AE            50  D0 0008F          MOVL    R0, TRNLNM_ITEM_LIST+4                          ;
                              08        AE      10    AE  9E 00093          MOVAB   DESC, TRNLNM_ITEM_LIST+8                        ; 1076
                                        0C      AE  D4 00098          CLRL    TRNLNM_ITEM_LIST+12                             ; 1077
                                                5E  DD 0009B          PUSHL   SP                                              ; 1082
                                      0000'     CF  9F 0009D          PUSHAB  P.AAH                                           ;
                                      0000'     CF  9F 000A1          PUSHAB  P.AAF                                           ;
                                      0000'     CF  9F 000A5          PUSHAB  P.AAD                                           ;
                                                7E  D4 000A9          CLRL    -(SP)                                           ;
                    00000000G   00              05  FB 000AB          CALLS   #5, SYS$TRNLNM                                  ;
                                01              50  D1 000B2          CMPL    R0, #1                                          ; 1083
                                                08  12 000B5          BNEQ    5$                                              ;
                                      10        AE  9F 000B7          PUSHAB  DESC                                            ; 1085
                  0000V CF                      01  FB 000BA          CALLS   #1, SETUP_LOGIN_PROC                            ;
                                                6B  D5 000BF 5$:      TSTL    COM_NAME                                        ; 1093
                                                2C  12 000C1          BNEQ    6$                                              ;
                                        25 00000000G    00  E8 000C3          BLBS    COM_NEGATED, 6$                         ; 1094
                                           00000000G    00  D5 000CA          TSTL    UAF_RECORD                              ; 1095
                                                1D  13 000D0          BEQL    6$                                              ;
                                        50 00000000G    00  D0 000D2          MOVL    UAF_RECORD, R0                          ; 1098
                              04        AB      00D5  C0  9E 000D9          MOVAB   213(R0), COM_NAME+4                             ;
                                        6B      00D4  C0  9A 000DF          MOVZBL  212(R0), COM_NAME                               ; 1099
                                                09  12 000E4          BNEQ    6$                                              ; 1101
                              04        AB      0000'     CF  9E 000E6          MOVAB   P.AAI, COM_NAME+4                               ; 1104
                                        6B              05  D0 000EC          MOVL    #5, COM_NAME                                    ; 1105
                                                6B  D5 000EF 6$:      TSTL    COM_NAME                                        ; 1109
                                                0E  13 000F1          BEQL    7$                                              ;
                                        07 00000000G    00  E8 000F3          BLBS    SUBPROCESS, 7$                          ; 1110
                                                5B  DD 000FA          PUSHL   R11                                             ; 1112
                  0000V CF                      01  FB 000FC          CALLS   #1, SETUP_LOGIN_PROC                            ;
                                                69  D5 00101 7$:      TSTL    CLI_NAME                                        ; 1124
                                                1A  12 00103          BNEQ    8$                                              ;
                                        17              52  E8 00105          BLBS    RESTRICTED_USER, 8$                     ; 1125
                                           00000000G    00  95 00108          TSTB    CTL$AG_CMEDATA                          ; 1126
                                                0F  13 0010E          BEQL    8$                                              ;
                                        69 00000000G    00  9A 00110          MOVZBL  CTL$AG_CMEDATA, CLI_NAME                        ; 1129
                              04        A9 00000000G    00  9E 00117          MOVAB   CTL$AG_CMEDATA+1, CLI_NAME+4                    ; 1130
                                                69  D5 0011F 8$:      TSTL    CLI_NAME                                        ; 1135
                                                25  12 00121          BNEQ    9$                                              ;
                                        22              52  E8 00123          BLBS    RESTRICTED_USER, 9$                     ; 1136
                                        69 00000000G    00  9A 00126          MOVZBL  CTL$GT_SPAWNCLI, CLI_NAME                       ; 1139
                              04        A9 00000000G    00  9E 0012D          MOVAB   CTL$GT_SPAWNCLI+1, CLI_NAME+4                   ; 1140
                                                6A  D5 00135          TSTL    TABLE_NAME                                      ; 1141
                                                0F  12 00137          BNEQ    9$                                              ;
                                        6A 00000000G    00  9A 00139          MOVZBL  CTL$GT_SPAWNTABLE, TABLE_NAME                   ; 1144
                              04        AA 00000000G    00  9E 00140          MOVAB   CTL$GT_SPAWNTABLE+1, TABLE_NAME+4               ; 1145
                                        50 00000000G    00  D0 00148 9$:      MOVL    UAF_RECORD, R0                          ; 1153
                                                24  13 0014F          BEQL    12$                                             ;
```

```
                                    69  D5 00151          TSTL    CLI_NAME                                        ; 1154
                                    C3  13 00153          BEQL    10$                                             ;
                              1D    52  E9 00155          BLBC    RESTRICTED_USER, 12$                            ; 1155
                              69    C0  9A 00158 10$:     MOVZBL  276(R0), CLI_NAME                               ; 1158
                        04    A9    C0  9E 0015D          MOVAB   277(R0), CLI_NAME+4                             ; 1159
                                    6A  D5 00163          TSTL    TABLE_NAME                                      ; 1160
                                    03  13 00165          BEQL    11$                                             ;
                              0B    52  E9 00167          BLBC    RESTRICTED_USER, 12$                            ; 1161
                              6A    C0  9A 0016A 11$:     MOVZBL  308(R0), TABLE_NAME                             ; 1164
                        04    AA    C0  9E 0016F          MOVAB   309(R0), TABLE_NAME+4                           ; 1165
                                    69  D5 00175 12$:     TSTL    CLI_NAME                                        ; 1174
                                    07  13 00177          BEQL    13$                                             ;
                                    50  D5 00179          TSTL    R0                                              ; 1176
                                    2C  12 0017B          BNEQ    15$                                             ;
                              29    52  E9 0017D          BLBC    RESTRICTED_USER, 15$                            ; 1178
                              69 00000000G  00 9A 00180 13$:  MOVZBL  CTL$GT_CLINAME, CLI_NAME                     ; 1183
                        04    A9 00000000G  00 9E 00187     MOVAB   CTL$GT_CLINAME+1, CLI_NAME+4                  ; 1184
                                    6A  D5 0018F          TSTL    TABLE_NAME                                      ; 1185
                                    07  13 00191          BEQL    14$                                             ;
                                    50  D5 00193          TSTL    R0                                              ; 1187
                                    12  12 00195          BNEQ    15$                                             ;
                              0F    52  E9 00197          BLBC    RESTRICTED_USER, 15$                            ; 1189
                              6A 00000000G  00 9A 0019A 14$:  MOVZBL  CTL$GT_TABLENAME, TABLE_NAME                  ; 1194
                        04    AA 00000000G  00 9E 001A1     MOVAB   CTL$GT_TABLENAME+1, TABLE_NAME+4              ; 1195
                                    69  D5 001A9 15$:     TSTL    CLI_NAME                                        ; 1203
                                    08  13 001AB          BEQL    16$                                             ;
                 13 00000000G   00      05  E1 001AD          BBC     #5, PCB_STS+2, 17$                          ; 1204
                                    69  03  D0 001B5 16$:     MOVL    #3, CLI_NAME                                ; 1207
                           04    A9      0000' CF  9E 001B8    MOVAB   DCL_STRING, CLI_NAME+4                      ; 1208
                 02 00000000G   00      05  E1 001BE          BBC     #5, PCB_STS+2, 17$                          ; 1209
                                    6A  D4 001C6          CLRL    TABLE_NAME                                      ; 1210
                                    53  69  D0 001C8 17$:     MOVL    CLI_NAME, R3                                ; 1217
                                    52      04  A9  D0 001CB    MOVL    CLI_NAME+4, R2                              ;
                                    62      53  3A  3A 001CF    LOCC    #58, R3, (R2)                               ;
                                            02  12 001D3       BNEQ    18$                                         ;
                                            51  D4 001D5       CLRL    R1                                          ;
                                            51  D5 001D7 18$:  TSTL    R1                                          ;
                                            21  12 001D9       BNEQ    19$                                         ;
          0045  8F          20          62      53  2C 001DB          MOVC5   R3, (R2), #32, #69, CLI_NAME_BUFFER+11  ; 1221
                                                00000000G  00  001E2
                    00000000G   00    0000' CF  0B  28 001E7          MOVC3   #11, P.AAJ, CLI_NAME_BUFFER             ; 1222
                                            69  0B  C0 001F1          ADDL2   #11, CLI_NAME                           ; 1223
                                      04    A9  00000000G  00  9E 001F4  MOVAB   CLI_NAME_BUFFER, CLI_NAME+4         ; 1224
                                            6A  D5 001FC 19$:  TSTL    TABLE_NAME                                  ; 1232
                                            47  12 001FE       BNEQ    21$                                         ;
                                        0B  69  D1 00200       CMPL    CLI_NAME, #11                               ; 1233
                                            42  1B 00203       BLEQU   21$                                         ;
                                        50      04  A9  D0 00205    MOVL    CLI_NAME+4, R0                          ; 1234
                                  0000' CF      60  0B  29 00209    CMPC3   #11, (R0), P.AAK                        ;
                                            36  12 0020F       BNEQ    21$                                         ;
                                        57  69  0B  C3 00211       SUBL3   #11, CLI_NAME, R7                       ; 1238
                                        50      04  A9  D0 00215    MOVL    CLI_NAME+4, R0                          ;
                                        58      50  8F  9A 00219    MOVZBL  #80, R8                                 ;
                                        56 00000000G  00  9E 0021D    MOVAB   TABLE_NAME_BUFFER, R6                   ;
                          58        20      0B  A0  57  2C 00224    MOVC5   R7, 1T(R0), #32, R8, (R6)               ;
                                                66      0022A                                                       ;
                                            0E  18 0022B       BGEQ    20$                                         ;
```

```
                                  56          57  C0 0022D        ADDL2   R7, R6                              ;
                                  58          57  C2 00230        SUBL2   R7, R8                              ;
        58          20    0000'   CF          06  2C 00233        MOVC5   #6, P.AAL, #32, R8, (R6)            ;
                                              66     0023A                                                    ;
                                  6A       06 A7  9E 0023B 20$:   MOVAB   6(R7), TABLE_NAME                   ; 1242
                             04   AA 00000000G 00  9E 0023F        MOVAB   TABLE_NAME_BUFFER, TABLE_NAME+4     ; 1243
                                              7E  D4 00247 21$:   CLRL    -(SP)                               ; 1250
                                        0000V CF  9F 00249        PUSHAB  MAP_CLI                             ;
                       00000000G  00          02  FB 0024D        CALLS   #2, SYS$CMEXEC                      ;
                                                  04 00254        RET                                         ; 1252

; Routine Size:  597 bytes,    Routine Base:  $CODE$ + 0306
```

```
;  928          1253  1 ROUTINE setup_login_proc (desc): NOVALUE =
;  929          1254  1
;  930          1255  1 !---
;  931          1256  1 !
;  932          1257  1 !       Setup a login command procedure, to be executed initially
;  933          1258  1 !       before starting interactive session.  The CLI will execute
;  934          1259  1 !       the procedures in the order they are give to this routine.
;  935          1260  1 !
;  936          1261  1 ! Inputs:
;  937          1262  1 !
;  938          1263  1 !       desc = Address of descriptor of command procedure
;  939          1264  1 !
;  940          1265  1 ! Outputs:
;  941          1266  1 !
;  942          1267  1 !       None
;  943          1268  1 !---
;  944          1269  1
;  945          1270  2 BEGIN
;  946          1271  2
;  947          1272  2 LOCAL
;  948          1273  2     logbuf:     VECTOR [8,BYTE],        ! Buffer for logical name 'PROC#'
;  949          1274  2     logdesc:    VECTOR [2];             ! Descriptor of above buffer
;  950          1275  2
;  951          1276  2 logdesc [0] = 5;                        ! Setup descriptor of logical name
;  952          1277  2 logdesc [1] = logbuf;
;  953          1278  2
;  954          1279  2 CH$MOVE(4, UPLIT BYTE('PROC'), logbuf); ! Create logical name string
;  955          1280  2
;  956          1281  2 ppd [ppd$b_nprocs] = .ppd [ppd$b_nprocs] + 1; ! Increment # of login procs
;  957          1282  2
;  958          1283  2 logbuf [4] = '0' + .ppd [ppd$b_nprocs]; ! Set procedure index into logname
;  959          1284  2
;  960          1285  2 create_logical(logdesc,                 ! Create PROC# = login file
;  961          1286  2                 .desc,
;  962          1287  2                 psl$c_user);
;  963          1288  2
;  964          1289  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                43  4F  52  50  00065 P.AAM:  .ASCII  \PROC\                                   ;


                                                        .PSECT  $CODE$,NOWRT,2

                                            0004 00000 SETUP_LOGIN_PROC:
                                                        .WORD   Save R2                                         ; 1253
                                52 00000000G    00  9E 00002        MOVAB   PPD+28, R2                          ;
                                5E              0C  C2 00009        SUBL2   #12, SP                             ;
                                                05  DD 0000C        PUSHL   #5                                  ; 1276
                            04  AE      08      AE  9E 0000E        MOVAB   LOGBUF, LOGDESC+4                   ; 1277
                            08  AE    0000'     CF  D0 00013        MOVL    P.AAM, LOGBUF                       ; 1279
                                                62  96 00019        INCB    PPD+28                              ; 1281
                        0C  AE              62      30  81 0001B    ADDB3   #48, PPD+28, LOGBUF+4               ; 1283
```

```
                                 03  DD 00020          PUSHL   #3                                      ; 1285
                              04 AC  DD 00022          PUSHL   DESC                                    ; 1286
                              08 AE  9F 00025          PUSHAB  LOGDESC                                 ; 1285
                    0000V  CF       03  FB 00028      CALLS   #3, CREATE_LOGICAL                      ;
                                        04 0002D       RET                                             ; 1289

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 055B
```

```
;  966     1290  1 ROUTINE map_cli: NOVALUE =
;  967     1291  1
;  968     1292  1 !---
;  969     1293  1 !
;  970     1294  1 !       Map the CLI into the control region.
;  971     1295  1 !
;  972     1296  1 ! Inputs:
;  973     1297  1 !
;  974     1298  1 !       Access mode is executive.
;  975     1299  1 !
;  976     1300  1 !       cli_name = Address of descriptor of CLI name
;  977     1301  1 !
;  978     1302  1 ! Outputs:
;  979     1303  1 !
;  980     1304  1 !       None
;  981     1305  1 !---
;  982     1306  1
;  983     1307  2 BEGIN
;  984     1308  2
;  985     1309  2 OWN
;  986     1310  2     image_name:         VECTOR[2],      ! Image's name descriptor
;  987     1311  2     image_filespec:     VECTOR[2],      ! Image's filespec descriptor
;  988     1312  2     image_header_buf:   VECTOR[128];    ! Image header from $IMGACT
;  989     1313  2 $ASSUME(%ALLOCATION(image_header_buf),EQL,512);
;  990     1314  2
;  991     1315  2 ROUTINE extract_image_name : NOVALUE =  ! Subroutine to extract image's
;  992     1316  3 BEGIN                                    !  filespec and name after $IMGACT
;  993     1317  3 LOCAL
;  994     1318  3     len,
;  995     1319  3     ptr: REF BLOCK[,BYTE];
;  996     1320  3 image_name[0] = 0;
;  997     1321  3 image_filespec[0] = 0;
;  998     1322  3 IF (ptr = .image_header_buf[1]) EQL 0
;  999     1323  3 THEN RETURN;
; 1000     1324  3 image_name[0] = .(ptr[ifd$q_curprog])<0,16>;
; 1001     1325  3 image_name[1] = .(ptr[ifd$q_curprog])<32,32>;
; 1002     1326  3 move_quad(image_name, image_filespec);
; 1003     1327  3 IF (len = .image_name[0]) EQL 0
; 1004     1328  3 THEN RETURN;
; 1005     1329  3 ptr = .image_name[1];
; 1006     1330  3 DO
; 1007     1331  4     BEGIN
; 1008     1332  4     LOCAL
; 1009     1333  4         chr: BYTE;
; 1010     1334  4     chr = CH$RCHAR_A(ptr);
; 1011     1335  4     IF .chr EQL ':'
; 1012     1336  4     OR .chr EQL ']'
; 1013     1337  4     OR .chr EQL '>'
; 1014     1338  4     THEN
; 1015     1339  5         BEGIN
; 1016     1340  5         image_name[0] = .len - 1;
; 1017     1341  5         image_name[1] = .ptr;
; 1018     1342  4         END;
; 1019     1343  4     len = .len - 1;
; 1020     1344  4     END
; 1021     1345  3 WHILE .len GTR 0;
; 1022     1346  3 IF NOT CH$FAIL(ptr = CH$FIND_CH(.image_name[0], .image_name[1], '.'))
```

```
;  1023     1347  3 THEN
;  1024     1348  4     BEGIN
;  1025     1349  4     image_name[0] = CH$DIFF(.ptr, .image_name[1]);
;  1026     1350  4     image_filespec[0] = CH$DIFF(.ptr, .image_filespec[1]);
;  1027     1351  3     END;
;  1028     1352  2 END;


                                        .PSECT  $OWN$,NOEXE,2

                                00000 IMAGE_NAME:
                                        .BLKB   8
                                00008 IMAGE_FILESPEC:
                                        .BLKB   8
                                00010 IMAGE_HEADER_BUF:
                                        .BLKB   512


                                        .PSECT  $CODE$,NOWRT,2

                                000C 00000 EXTRACT_IMAGE_NAME:
                                        .WORD   Save R2,R3                              ;  1315
                    53      0000'   CF  9E 00002        MOVAB   IMAGE_NAME, R3          :
                                    63  D4 00007        CLRL    IMAGE_NAME              ;  1320
                                08  A3  D4 00009        CLRL    IMAGE_FILESPEC          ;  1321
                    51          14  A3  D0 0000C        MOVL    IMAGE_HEADER_BUF+4, PTR ;  1322
                                    4C  13 00010        BEQL    5$                      :
                    63          14  A1  3C 00012        MOVZWL  20(PTR), IMAGE_NAME     ;  1324
                04  A3          18  A1  D0 00016        MOVL    24(PTR), IMAGE_NAME+4   ;  1325
                08  A3              63  7D 0001B        MOVQ    IMAGE_NAME, IMAGE_FILESPEC ;  1326
                    52              63  D0 0001F        MOVL    IMAGE_NAME, LEN         ;  1327
                                    3A  13 00022        BEQL    5$                      :
                    51          04  A3  D0 00024        MOVL    IMAGE_NAME+4, PTR       ;  1329
                    50              81  90 00028 1$:    MOVB    (PTR)+, CHR             ;  1334
                    3A              50  91 0002B        CMPB    CHR, #58                ;  1335
                                    0B  13 0002E        BEQL    2$                      :
                            5D  8F      50  91 00030        CMPB    CHR, #93            ;  1336
                                    05  13 00034        BEQL    2$                      :
                    3E              50  91 00036        CMPB    CHR, #62                ;  1337
                                    08  12 00039        BNEQ    3$                      :
                    63      FF  A2  9E 0003B 2$:        MOVAB   -1(R2), IMAGE_NAME      ;  1340
                04  A3              51  D0 0003F        MOVL    PTR, IMAGE_NAME+4       ;  1341
                    E2              52  F5 00043 3$:    SOBGTR  LEN, 1$                 ;  1343
            04  B3              63  2E  3A 00046        LOCC    #46, IMAGE_NAME, @IMAGE_NAME+4 ;  1346
                                    02  12 0004B        BNEQ    4$                      :
                                    51  D4 0004D        CLRL    R1                      :
                                    51  D5 0004F 4$:    TSTL    PTR                     :
                                    0B  13 00051        BEQL    5$                      :
                63              51      04  A3  C3 00053        SUBL3   IMAGE_NAME+4, PTR, IMAGE_NAME ;  1349
            08  A3              51      0C  A3  C3 00058        SUBL3   IMAGE_FILESPEC+4, PTR, IMAGE_FILESPEC ;  1350
                                        04  0005E 5$:    RET                             ;  1352

; Routine Size:  95 bytes,    Routine Base:  $CODE$ + 0589
```

```
;  1029          1353  2
;  1030          1354  2 BUILTIN FP;
;  1031          1355  2
;  1032          1356  2 EXTERNAL
;  1033          1357  2     exe$gl_clitabl;                     ! SYSGEN parameter CLISYMTBL
;  1034          1358  2
;  1035          1359  2 BIND
;  1036          1360  2     clisymtbl = ppd [ppd$q_clisymtbl]: VECTOR;  ! Reference as 2 longwords
;  1037          1361  2
;  1038          1362  2 LOCAL
;  1039          1363  2     status,
;  1040          1364  2     arglist:    VECTOR [2],             ! Arg list to LGI$PROTECT_CLI
;  1041          1365  2     range:      VECTOR [2];             ! Range of CLI symbol table
;  1042          1366  2
;  1043          1367  2 .fp = handler;                          ! Enable condition handler
;  1044          1368  2
;  1045          1369  2 !
;  1046          1370  2 ! Change the page protection on the CLI and its tables to supervisor
;  1047          1371  2 ! write (where writable) and user read, supervisor owned, to prevent
;  1048          1372  2 ! the user from modifying the CLI.
;  1049          1373  2 !
;  1050          1374  2 status = lib$pl_merge(cli_name,          ! Map CLI into control region
;  1051          1375  2         %ASCID 'SYS$SYSTEM:.EXE',       ! Default filespec for CLI
;  1052          1376  2         image_header_buf,               ! Return image header buffer
;  1053          1377  2         ctl$ag_climage);                ! Return address range
;  1054          1378  2
;  1055          1379  2 IF NOT .status                          ! If error detected,
;  1056          1380  2 THEN
;  1057          1381  2     SIGNAL_STOP(lgi$_clifail,1,cli_name,.status); ! then signal fatal error
;  1058          1382  2
;  1059          1383  2 arglist[0] = 1;
;  1060          1384  2 arglist[1] = ctl$ag_climage;
;  1061          1385  2 status = $CMKRNL (ROUTIN = lgi$protect_cli, ARGLST = arglist);
;  1062          1386  2 IF NOT .status
;  1063          1387  2 THEN
;  1064          1388  2     SIGNAL_STOP(lgi$_cliprot,0,.status);
;  1065          1389  2
;  1066          1390  2 extract_image_name();                   ! Extract image name
;  1067          1391  2 CH$MOVE(ctl$gt_cliname[0] = .image_name[0]), ! Load
;  1068          1392  2         .image_name[1],                 !  image name
;  1069          1393  2         ctl$gt_cliname[1]);             !   as ASCIC
;  1070          1394  2
;  1071          1395  2 IF .table_name[0] NEQ 0
;  1072          1396  2 THEN
;  1073          1397  3     BEGIN
;  1074          1398  3     status = lib$pl_merge(table_name,   ! Map command table into control region
;  1075          1399  3         %ASCID 'SYS$SHARE:.EXE',        ! Default filespec for tables
;  1076          1400  3         image_header_buf,               ! Return image header buffer
;  1077          1401  3         ctl$ag_clitable);               ! Return address range
;  1078          1402  3
;  1079          1403  3     IF NOT .status                      ! If error detected,
;  1080          1404  3     THEN
;  1081          1405  3         SIGNAL_STOP(lgi$_clitblfail,1,table_name,.status); ! signal fatal error
;  1082          1406  3     arglist[1] = ctl$ag_clitable;
;  1083          1407  3     status = $CMKRNL (ROUTIN = lgi$protect_cli, ARGLST = arglist);
;  1084          1408  3     IF NOT .status
;  1085          1409  3     THEN
```

```
; 1086          1410  3         SIGNAL_STOP(lgi$_clitblprot,0,.status);
; 1087          1411  3
; 1088          1412  3     extract_image_name();                ! Extract image filespec
; 1089          1413  3     CH$MOVE(ctl$gt_tablename[0] = .image_filespec[0]), ! Load
; 1090          1414  3             .image_filespec[1],          !   image filespec
; 1091          1415  3             ctl$gt_tablename[1]);        !   as ASCIC
; 1092          1416  2     END;
; 1093          1417  2
; 1094          1418  2 !
; 1095          1419  2 ! Create the CLI symbol table space.
; 1096          1420  2 !
; 1097        P 1421  2 status = $EXPREG(PAGCNT = .exe$gl_clitabl,
; 1098        P 1422  2                  RETADR = range,
; 1099        P 1423  2                  ACMODE = psl$c_super,
; 1100          1424  2                  REGION = 1);
; 1101          1425  2
; 1102          1426  2 IF NOT .status
; 1103          1427  2 THEN
; 1104          1428  2     SIGNAL_STOP(lgi$_clisymtbl, 0, .status);
; 1105          1429  2
; 1106        P 1430  2 $CMKRNL(ROUTIN = set_p1_base,            ! Set new base of control region
; 1107          1431  2         ARGLST = .range[1]);
; 1108          1432  2
; 1109          1433  2 clisymtbl[0] = .range[0] - .range[1] + 1;    ! Setup descriptor of storage
; 1110          1434  2 clisymtbl[1] = .range[1];
; 1111          1435  2
; 1112          1436  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        00069           .BLKB   3
45 58 45 2E 3A 4D 45 54 53 59 53 24 53 59 53    0006C P.AAO:    .ASCII  \SYS$SYSTEM:.EXE\<0>                    ;
                                                    00 0007B                                                    ;
                                                010E000F 0007C P.AAN:   .LONG   17694735                        ;
                                                00000000' 00080         .ADDRESS P.AAO                          ;
00 45 58 45 2E 3A 45 52 41 48 53 24 53 59 53    00084 P.AAQ:    .ASCII  \SYS$SHARE:.EXE\<0><0>                  ;
                                                    00 00093                                                    ;
                                                010E000E 00094 P.AAP:   .LONG   17694734                        ;
                                                00000000' 00098         .ADDRESS P.AAQ                          ;

                                                                .EXTRN  EXE$GL_CLITABL, SYS$EXPREG

                                                                .PSECT  $CODE$,NOWRT,2

                                                    0FFC 00000 MAP_CLI:.WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 1290
                                5B 00000000G    00  9E 00002        MOVAB   CTL$AG_CLIMAGE, R11                 ;
                                5A 00000000G    00  9E 00009        MOVAB   TABLE_NAME, R10                     ;
                                59 00000000C    00  9E 00010        MOVAB   SYS$CMKRNL, R9                      ;
                                58     0000'    CF  9E 00017        MOVAB   IMAGE_HEADER_BUF, R8                ;
                                57 00000000G    00  9E 0001C        MOVAB   LIB$STOP, R7                        ;
                                5E              10  C2 00023        SUBL2   #16, SP                             ;
                                6D 00000000G    00  9E 00026        MOVAB   HANDLER, (FP)                       ; 1367
                                       0900     8F  BB 0002D        PUSHR   #^M<R8,R11>                         ; 1374
                                     0000'      CF  9F 00031        PUSHAB  P.AAN                               ;
                                00000000G       00  9F 00035        PUSHAB  CLI_NAME                            ;
```

```
                00000000G  00          04  FB 0003B          CALLS   #4, LIB$P1_MERGE                                    ;
                           56          50  D0 00042          MOVL    R0, STATUS                                         ;
                           13          56  E8 00045          BLBS    STATUS, 1$                                         ; 1379
                                       56  DD 00048          PUSHL   STATUS                                             ; 1381
                               00000000G   00  9F 0004A      PUSHAB  CLI_NAME                                           ;
                                           01  DD 00050      PUSHL   #1                                                 ;
                               00000000G   8F  DD 00052      PUSHL   #LGI$_CLIFAIL                                      ;
                                       67  04  FB 00058      CALLS   #4, LIB$STOP                                       ;
                                    08 AE  01  D0 0005B 1$:   MOVL    #1, ARGLIST                                        ; 1383
                                    0C AE  6B  9E 0005F      MOVAB   CTL$AG_CLIMAGE, ARGLIST+4                          ; 1384
                                        08 AE  9F 00063      PUSHAB  ARGLIST                                            ; 1385
                               00000000G   00  9F 00066      PUSHAB  LGI$PROTECT_CLI                                    ;
                                       69  02  FB 0006C      CALLS   #2, SYS$CMKRNL                                     ;
                                       56  50  D0 0006F      MOVL    R0, STATUS                                         ;
                                       0D  56  E8 00072      BLBS    STATUS, 2$                                         ; 1386
                                           56  DD 00075      PUSHL   STATUS                                             ; 1388
                                           7E  D4 00077      CLRL    -(SP)                                              ;
                               00000000G   8F  DD 00079      PUSHL   #LGI$_CLIPROT                                      ;
                                       67  03  FB 0007F      CALLS   #3, LIB$STOP                                       ;
                                FF1A   CF      00  FB 00082 2$:   CALLS   #0, EXTRACT_IMAGE_NAME                         ; 1390
                                       50      F0  A8  D0 00087  MOVL    IMAGE_NAME, R0                                 ; 1391
                                00000000G  00      50  90 0008B  MOVB    R0, CTL$GT_CLINAME                             ;
00000000G  00                       F4  B8          50  28 00092  MOVC3   R0, @IMAGE_NAME+4, CTL$GT_CLINAME+1          ; 1393
                                           6A  D5 0009B      TSTL    TABLE_NAME                                         ; 1395
                                           6A  13 0009D      BEQL    5$                                                 ;
                               00000000G   00  9F 0009F      PUSHAB  CTL$AG_CLITABLE                                    ; 1398
                                           58  DD 000A5      PUSHL   R8                                                 ;
                                    0000'  CF  9F 000A7      PUSHAB  P.AAP                                              ;
                                           5A  DD 000AB      PUSHL   R10                                                ;
                                00000000G  00      04  FB 000AD  CALLS   #4, LIB$P1_MERGE                              ;
                                           56      50  D0 000B4  MOVL    R0, STATUS                                     ;
                                           0F      56  E8 000B7  BLBS    STATUS, 3$                                     ; 1403
                                                   56  DD 000BA  PUSHL   STATUS                                         ; 1405
                                                   5A  DD 000BC  PUSHL   R10                                          ;
                                                   01  DD 000BE  PUSHL   #1                                             ;
                                       00000000G   8F  DD 000C0  PUSHL   #LGI$_CLITBLFAIL                             ;
                                               67  04  FB 000C6  CALLS   #4, LIB$STOP                                  ;
                                            0C AE  00000000G   00  9E 000C9 3$:   MOVAB   CTL$AG_CLITABLE, ARGLIST+4   ; 1406
                                                       08 AE  9F 000D1  PUSHAB  ARGLIST                                ; 1407
                                               00000000G   00  9F 000D4  PUSHAB  LGI$PROTECT_CLI                       ;
                                                       69  02  FB 000DA  CALLS   #2, SYS$CMKRNL                        ;
                                                       56  50  D0 000DD  MOVL    R0, STATUS                            ;
                                                       0D  56  E8 000E0  BLBS    STATUS, 4$                            ; 1408
                                                           56  DD 000E3  PUSHL   STATUS                                ; 1410
                                                           7E  D4 000E5  CLRL    -(SP)                                 ;
                                               00000000G   8F  DD 000E7  PUSHL   #LGI$_CLITBLPROT                      ;
                                                       67  03  FB 000ED  CALLS   #3, LIB$STOP                         ;
                                             FEAC      CF      00  FB 000F0 4$:   CALLS   #0, EXTRACT_IMAGE_NAME       ; 1412
                                                       50      F8  A8  D0 000F5  MOVL    IMAGE_FILESPEC, R0            ; 1413
                                                00000000G  00      50  90 000F9  MOVB    R0, CTL$GT_TABLENAME           ;
00000000G  00                                       FC  B8          50  28 00100  MOVC3   R0, @IMAGE_FILESPEC+4, CTL$GT_TABLENAME+1  ; 1415
                                                                   01  DD 00109 5$:   PUSHL   #1                       ; 1424
                                                                   02  DD 0010B      PUSHL   #2                       ;
                                                            08 AE  9F 0010D      PUSHAB  RANGE                        ;
                                                       00000000G   00  DD 00110      PUSHL   EXE$GL_CLITABL           ;
                                                00000000G  00          04  FB 00116  CALLS   #4, SYS$EXPREG           ;
                                                           56          50  D0 0011D  MOVL    R0, STATUS             ;
```

```
                    0D              56  E8 00120          BLBS    STATUS, 6$                          ; 1426
                                    56  DD 00123          PUSHL   STATUS                              ; 1428
                                    7E  D4 00125          CLRL    -(SP)                               ;
                        00000000G   8F  DD 00127          PUSHL   #LGI$_CLISYMTBL                     ;
                    67              03  FB 0012D          CALLS   #3, LIB$STOP                        ;
                            04      AE  DD 00130 6$:      PUSHL   RANGE+4                             ; 1431
                          0000V     CF  9F 00133          PUSHAB  SET_P1_BASE                         ;
                    69              02  FB 00137          CALLS   #2, SY$$CMKRNL                      ;
                50          6E      04  AE  C3 0013A      SUBL3   RANGE+4, RANGE, R0                  ; 1433
                00000000G   00      01  A0  9E 0013F      MOVAB   1(R0), CLISYMTBL                    ;
                00000000G   00      04  AE  D0 00147      MOVL    RANGE+4, CLISYMTBL+4                ; 1434
                                        04  0014F          RET                                         ; 1436

; Routine Size:  336 bytes,    Routine Base:  $CODE$ + 05E8
```

```
;  1114          1437  1 GLOBAL ROUTINE execute_cli: NOVALUE =
;  1115          1438  1
;  1116          1439  1 !---
;  1117          1440  1 !
;  1118          1441  1 !       This routine is called to transfer control to the CLI
;  1119          1442  1 !
;  1120          1443  1 ! Inputs:
;  1121          1444  1 !
;  1122          1445  1 !       Access mode is executive.
;  1123          1446  1 !
;  1124          1447  1 ! Outputs:
;  1125          1448  1 !
;  1126          1449  1 !       None
;  1127          1450  1 !---
;  1128          1451  1
;  1129          1452  2 BEGIN
;  1130          1453  2
;  1131          1454  2 BUILTIN FP;
;  1132          1455  2
;  1133          1456  2 MAP FP: REF BBLOCK;                    ! Address of call frame
;  1134          1457  2
;  1135          1458  2 EXTERNAL LITERAL
;  1136          1459  2     exe$c_cmstksz;                     ! # bytes after $CMEXEC frame to PC/PSL
;  1137          1460  2
;  1138          1461  2 LOCAL
;  1139          1462  2     prev_fp:    REF BBLOCK,            ! Address of previous frame
;  1140          1463  2     pcpsl:      REF VECTOR,            ! Address of previous PC/PSL pair
;  1141          1464  2     psl:        REF BBLOCK;            ! Address of previous PSL
;  1142          1465  2
;  1143          1466  2 !
;  1144          1467  2 ! Change the page protection on the PPD structure to allow only supervisor
;  1145          1468  2 ! mode write access for the protection of the CLI data storage.
;  1146          1469  2 !
;  1147          1470  2
;  1148        P 1471  2 $CMEXEC(ROUTIN = set_ppd_prot,         ! Set PPD page protection
;  1149          1472  2         ARGLST = prt$c_ursw);
;  1150          1473  2
;  1151          1474  2 !
;  1152          1475  2 ! Locate the PC/PSL in the $CMEXEC call frame and alter it to point
;  1153          1476  2 ! to the CLI entry point and set the PSL to supervisor mode.
;  1154          1477  2 !
;  1155          1478  2
;  1156          1479  2 prev_fp = .fp [sf$l_save_fp];          ! Get address of CMEXEC call frame
;  1157          1480  2 pcpsl = .prev_fp + exe$c_cmstksz;     ! Point to PC/PSL after argument list
;  1158          1481  2 pcpsl [0] = .ctl$ag_climage;          ! Set PC to CLI entry point
;  1159          1482  2 psl = pcpsl [1];                      ! Get address of PSL
;  1160          1483  2 psl [psl$v_curmod] = psl$c_super;     ! Set access mode to supervisor
;  1161          1484  2 psl [psl$v_prvmod] = psl$c_super;     ! and previous mode as well
;  1162          1485  2
;  1163          1486  2 !
;  1164          1487  2 ! Now, on exit from the $CMEXEC system service, control will be transferred
;  1165          1488  2 ! to the CLI in supervisor mode.
;  1166          1489  2 !
;  1167          1490  2
;  1168          1491  1 END;
```

```
                                                        .EXTRN  EXE$C_CMSTKSZ

                                                        .ENTRY  EXECUTE_CLI, Save nothing                       ; 1437
                                                        PUSHL   #12                                             ; 1472
                                                        PUSHAB  SET_PPD_PROT                                    ;
                                                        CALLS   #2, SYS$CMEXEC                                  ;
                                                        MOVL    12(FP), PREV_FP                                 ; 1479
                                                        ADDL2   #EXE$C_CMSTKSZ, PCPSL                           ; 1480
                                                        MOVL    CTL$AG_CLIMAGE, (PCPSL)+                        ; 1481
                                                        INSV    #2, #0, #2, 3(PSL)                              ; 1483
                                                        INSV    #2, #22, #2, (PSL)                              ; 1484
                                                        RET                                                     ; 1491

; Routine Size:  47 bytes,    Routine Base:  $CODE$ + 0738
```

```
;                1492  1 GLOBAL ROUTINE map_imgact: NOVALUE =
;                1493  1
;                1494  1 !---
;                1495  1 !
;                1496  1 !       Map a code segment into P1 space which, when called, will
;                1497  1 !       unmap the login program and activate a given image.
;                1498  1 !
;                1499  1 ! Inputs:
;                1500  1 !
;                1501  1 !       Access mode is executive.
;                1502  1 !
;                1503  1 !       sys$input = Descriptor of image file specification
;                1504  1 !
;                1505  1 ! Outputs:
;                1506  1 !
;                1507  1 !       ctl$ag_climage = Address of P1 code segment to do the work
;                1508  1 !                        (should be called in executive mode)
;                1509  1 !---
;                1510  1
;                1511  2 BEGIN
;                1512  2
;                1513  2 LOCAL
;                1514  2     range:      VECTOR [2];             ! Range of allocated space in P1
;                1515  2
;                1516  2 BIND
;                1517  2     image_desc = mmg$imghdrbuf: VECTOR; ! Pass image filespec in buffer
;                1518  2
;                1519  2 image_activate = true;                 ! Mark image activate to be done
;                1520  2
;                1521  2 image_desc [0] = .sys$input [0];       ! Store filespec descriptor into buffer
;                1522  2 image_desc [1] = image_desc [2];       ! as well as string itself
;                1523  2 CH$MOVE(.sys$input [0], .sys$input [1], .image_desc [1]);
;                1524  2
;              P 1525  2 $EXPREG(PAGCNT = 1,                   ! Allocate one page in P1 space
;              P 1526  2         RETADR = range,
;              P 1527  2         ACMODE = psl$c_super,
;                1528  2         REGION = 1);
;                1529  2
;              P 1530  2 $CMKRNL(ROUTIN = set_p1_base,         ! Set new base of control region
;                1531  2         ARGLST = .range [1]);        ! so that code stays after rundown
;                1532  2
;                1533  2 CH$MOVE(512, execute_image, .range [1]); ! Copy code into page (max. 1 page)
;                1534  2
;                1535  2 ctl$ag_climage = .range [1];          ! Store address of code segment
;                1536  2
;                1537  1 END;


                                  007C 00000     .ENTRY  MAP_IMGACT, Save R2,R3,R4,R5,R6        ; 1492
                    56 00000000G  00   9E 00002  MOVAB   IMAGE_DESC+4, R6                       ;
                    5E            08   C2 00009  SUBL2   #8, SP                                ;
         00000000G  00            01   90 0000C  MOVB    #1, IMAGE_ACTIVATE                    ; 1519
                    52 00000000G  00   D0 00013  MOVL    SYS$INPUT, R2                         ; 1521
                FC  A6            52   D0 0001A  MOVL    R2, IMAGE_DESC                        ;
```

```
                        66        04    A6  9E 0001E         MOVAB   IMAGE_DESC+8, IMAGE_DESC+4                  ; 1522
                        51 00000000G    00  D0 00022         MOVL    SYS$INPUT+4, R1                            ; 1523
                        50              66  D0 00029         MOVL    IMAGE_DESC+4, R0                           ;
                60      61              52  28 0002C         MOVC3   R2, (R1), (R0)                             ;
                                        01  DD 00030         PUSHL   #1                                         ; 1528
                                        02  DD 00032         PUSHL   #2                                         ;
                                    08  AE  9F 00034         PUSHAB  RANGE                                      ;
                                        01  DD 00037         PUSHL   #1                                         ;
                00000000G   00          04  FB 00039         CALLS   #4, SYS$EXPREG                             ;
                                    04  AE  DD 00040         PUSHL   RANGE+4                                    ; 1531
                                00000V  CF  9F 00043         PUSHAB  SET_P1_BASE                                ;
                00000000G   00          02  FB 00047         CALLS   #2, SYS$CMKRNL                             ;
            04  BE  00000000G   00  0200  8F  28 0004E         MOVC3   #512, EXECUTE_IMAGE, @RANGE+4              ; 1533
                00000000G   00    04  AE  D0 00059         MOVL    RANGE+4, CTL$AG_CLIMAGE                    ; 1535
                                        04  00061         RET                                                ; 1537

; Routine Size:  98 bytes,    Routine Base:  $CODE$ + 0767
```

```
;  1217          1538  1 ROUTINE set_p1_base =
;  1218          1539  1
;  1219          1540  1 !---
;  1220          1541  1 !
;  1221          1542  1 !       This routine resets the base address of the fixed portion
;  1222          1543  1 !       of the control region.
;  1223          1544  1 !
;  1224          1545  1 ! Inputs:
;  1225          1546  1 !
;  1226          1547  1 !       Access mode is kernel.
;  1227          1548  1 !
;  1228          1549  1 !       ap = New base address for fixed P1
;  1229          1550  1 !
;  1230          1551  1 ! Outputs:
;  1231          1552  1 !
;  1232          1553  1 !       routine = status (not signaled)
;  1233          1554  1 !---
;  1234          1555  1
;  1235          1556  2 BEGIN
;  1236          1557  2
;  1237          1558  2 BUILTIN
;  1238          1559  2     ap;
;  1239          1560  2
;  1240          1561  2 EXTERNAL
;  1241          1562  2     ctl$gl_ctlbasva;                  ! Base address of permanent P1 space
;  1242          1563  2
;  1243          1564  2 ctl$gl_ctlbasva = .ap;                ! Set new base of fixed P1 region
;  1244          1565  2
;  1245          1566  2 RETURN true;
;  1246          1567  2
;  1247          1568  1 END;


                                                .EXTRN  CTL$GL_CTLBASVA

                                    0000 00000 SET_P1_BASE:
                                                .WORD   Save nothing                    ; 1538
                00000000G   00          5C  D0 00002        MOVL    AP, CTL$GL_CTLBASVA ; 1564
                            50          01  D0 00009        MOVL    #1, R0              ; 1566
                                            04 0000C        RET                         ; 1568

; Routine Size:  13 bytes,    Routine Base:  $CODE$ + 07C9
```

```
;  1249          1569  1 GLOBAL ROUTINE set_account: NOVALUE =
;  1250          1570  1
;  1251          1571  1 !---
;  1252          1572  1 !
;  1253          1573  1 !       Set the account name in the JIB and P1 space.
;  1254          1574  1 !
;  1255          1575  1 ! Inputs:
;  1256          1576  1 !
;  1257          1577  1 !       Access mode = Kernel
;  1258          1578  1 !
;  1259          1579  1 !       AP = Address of account name descriptor
;  1260          1580  1 !
;  1261          1581  1 ! Outputs:
;  1262          1582  1 !
;  1263          1583  1 !       None.  The JIB and P1 space are updated.
;  1264          1584  1 !
;  1265          1585  1 !---
;  1266          1586  1
;  1267          1587  2 BEGIN
;  1268          1588  2
;  1269          1589  2 BUILTIN
;  1270          1590  2     AP;
;  1271          1591  2
;  1272          1592  2 MAP
;  1273          1593  2     AP:         REF VECTOR;             ! Address of account name descriptor
;  1274          1594  2
;  1275          1595  2 EXTERNAL
;  1276          1596  2     ctl$t_account;                      ! Account name string in P1 space
;  1277          1597  2
;  1278          1598  2 LOCAL
;  1279          1599  2     jib:        REF BBLOCK;             ! Address of JIB
;  1280          1600  2
;  1281          1601  2 CH$COPY(.AP [0], .AP [1],               ! Copy account name string
;  1282          1602  2                                       !  blank padded
;  1283          1603  2         jib$s_account, ctl$t_account);!   to control region
;  1284          1604  2
;  1285          1605  2 jib = .ctl$gl_pcb [pcb$l_jib];         ! Get JIB address
;  1286          1606  2
;  1287          1607  2 CH$COPY(.AP [0], .AP [1],               ! Copy it to JIB as well
;  1288          1608  2
;  1289          1609  2         jib$s_account, jib [jib$t_account]);
;  1290          1610  2
;  1291          1611  1 END;


                                                                .EXTRN  CTL$T_ACCOUNT

                                                  003C 00000    .ENTRY  SET_ACCOUNT, Save R2,R3,R4,R5                   ; 1569
08              20      04      BC                6C   2C 00002 MOVC5   (AP), @4(AP), #32, #8, CTL$T_ACCOUNT      ; 1601
                                        00000000G   00     00008
                                50 00000000G    00   D0 00000 MOVL    CTL$GL_PCB, R0                             ; 1605
                                50      0080    C0   D0 00014 MOVL    128(R0), JIB
08              20      04      BC                6C   2C 00019 MOVC5   (AP), @4(AP), #32, #8, 24(JIB)            ; 1609
                                          18    A0     0001F
                                              04 00021 RET                                                         ; 1611
```

; Routine Size: 34 bytes,   Routine Base: $CODE$ + 07D6

```
;  1293          1612  1 GLOBAL ROUTINE set_username: NOVALUE =
;  1294          1613  1
;  1295          1614  1 !---
;  1296          1615  1 !
;  1297          1616  1 !       Set the username in the JIB and P1 space.
;  1298          1617  1 !
;  1299          1618  1 ! Inputs:
;  1300          1619  1 !
;  1301          1620  1 !       Access mode = Kernel
;  1302          1621  1 !
;  1303          1622  1 !       AP = Address of username descriptor
;  1304          1623  1 !
;  1305          1624  1 ! Outputs:
;  1306          1625  1 !
;  1307          1626  1 !       None.  The JIB and P1 space are updated.
;  1308          1627  1 !
;  1309          1628  1 !---
;  1310          1629  1
;  1311          1630  2 BEGIN
;  1312          1631  2
;  1313          1632  2 BUILTIN
;  1314          1633  2     AP;
;  1315          1634  2
;  1316          1635  2 MAP
;  1317          1636  2     AP:         REF VECTOR;             ! Address of username descriptor
;  1318          1637  2
;  1319          1638  2 EXTERNAL
;  1320          1639  2     ctl$t_username;                     ! Username string in P1 space
;  1321          1640  2
;  1322          1641  2 LOCAL
;  1323          1642  2     jib:        REF BBLOCK;             ! Address of JIB
;  1324          1643  2
;  1325          1644  2 CH$COPY(.AP [0], .AP [1],               ! Copy username string
;  1326          1645  2                                        !  blank padded
;  1327          1646  2         jib$s_username, ctl$t_username); !   to control region
;  1328          1647  2
;  1329          1648  2 jib = .ctl$gl_pcb [pcb$l_jib];          ! Get JIB address
;  1330          1649  2
;  1331          1650  2 CH$COPY(.AP [0], .AP [1],               ! Copy it to JIB as well
;  1332          1651  2
;  1333          1652  2         jib$s_username, jib [jib$t_username]);
;  1334          1653  2
;  1335          1654  1 END;


                                                                .EXTRN  CTL$T_USERNAME

                                                    003C 00000  .ENTRY  SET_USERNAME, Save R2,R3,R4,R5          ;  1612
        0C      20      04      BC          6C    2C 00002      MOVC5   (AP), @4(AP), #32, #12, CTL$T_USERNAME  ;  1644
                                    00000000G   00      00008
                                50  00000000G   00    D0 0000D  MOVL    CTL$GL_PCB, R0                        ;  1648
                                50      0080    C0    D0 00014  MOVL    128(R0), JIB
        0C      20      04      BC          6C    2C 00019      MOVC5   (AP), @4(AP), #32, #12, 12(JIB)       ;  1652
                                                0C    A0 0001F
                                                    04 00021  RET                                             ;  1654
```

: Routine Size. 34 bytes,   Routine Base: $CODE$ + 07F8

```
;1337     1655  1 GLOBAL ROUTINE set_node_name (link): NOVALUE =
;1338     1656  1
;1339     1657  1 !---
;1340     1658  1 !
;1341     1659  1 !       Set the node name, node address, and remote ID strings.
;1342     1660  1 !
;1343     1661  1 ! Inputs:
;1344     1662  1 !
;1345     1663  1 !       link = local link number
;1346     1664  1 !
;1347     1665  1 ! Outputs:
;1348     1666  1 !
;1349     1667  1 !       None.  P1 space is updated.
;1350     1668  1 !
;1351     1669  1 !---
;1352     1670  1
;1353     1671  2 BEGIN
;1354     1672  2
;1355     1673  2 OWN
;1356     1674  2     last_link:          INITIAL (0);            ! Last link processed
;1357     1675  2
;1358     1676  2 ROUTINE load_node_info (bufadr, buflen) =      ! $CMKRNL subroutine to load
;1359     1677  3 BEGIN                                          !  node info into P1 space
;1360     1678  3
;1361     1679  3 EXTERNAL
;1362     1680  3     ctl$t_nodeaddr:     VECTOR [,BYTE],         ! Node address (ASCIC, max=6)
;1363     1681  3     ctl$t_nodename:     VECTOR [,BYTE],         ! Node name (ASCIC, max=6)
;1364     1682  3     ctl$t_remoteid:     VECTOR [,BYTE];         ! Remote id (ASCIC, max=16)
;1365     1683  3
;1366     1684  3 LOCAL
;1367     1685  3     bufend,
;1368     1686  3     srclen,
;1369     1687  3     srcptr;
;1370     1688  3
;1371     1689  3 !
;1372     1690  3 ! Address the returned information and its end
;1373     1691  3 !
;1374     1692  3 bufend = (srcptr = .bufadr) + .buflen;
;1375     1693  3
;1376     1694  3 !
;1377     1695  3 ! Copy remote node address
;1378     1696  3 !
;1379     1697  3 IF .srcptr GEQA .bufend
;1380     1698  3 THEN RETURN 0;
;1381     1699  3 srclen = 4;
;1382     1700  3 CH$COPY((ctl$t_nodeaddr [0] = .srclen), .srcptr, 0, 6, ctl$t_nodeaddr [1]);
;1383     1701  3 srcptr = .srcptr + .srclen;
;1384     1702  3
;1385     1703  3 !
;1386     1704  3 ! Copy remote node name
;1387     1705  3 !
;1388     1706  3 IF .srcptr GEQA .bufend
;1389     1707  3 THEN RETURN 0;
;1390     1708  3 srclen = .(.srcptr)<0,16>;
;1391     1709  3 srcptr = .srcptr + 2;
;1392     1710  3 CH$COPY((ctl$t_nodename [0] = .srclen), .srcptr, 0, 6, ctl$t_nodename [1]);
;1393     1711  3 srcptr = .srcptr + .srclen;
```

```
;  1394          1712  3
;  1395          1713  3 !
;  1396          1714  3 ! Copy remote ID
;  1397          1715  3 !
;  1398          1716  3 IF .srcptr GEQA .bufend
;  1399          1717  3 THEN RETURN 0;
;  1400          1718  3 srclen = .(.srcptr)<0,16>;
;  1401          1719  3 srcptr = .srcptr + 2;
;  1402          1720  3 CH$COPY((ctl$t_remoteid [0] = .srclen), .srcptr, 0, 16, ctl$t_remoteid [1]);
;  1403          1721  3
;  1404          1722  3 RETURN 1;
;  1405          1723  3
;  1406          1724  2 END;


                                                        .PSECT  $OWN$,NOEXE,2

                                00000000  00210 LAST_LINK:
                                                        .LONG   0                                       ;

                                                        .EXTRN  CTL$T_NODEADDR, CTL$T_NODENAME
                                                        .EXTRN  CTL$T_REMOTEID

                                                        .PSECT  $CODE$,NOWRT,2

                                    01FC 00000 LOAD_NODE_INFO:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8               ; 1676
                          56        04   AC  D0 00002    MOVL    BUFADR, SRCPTR                          ; 1692
                58        56        08   AC  C1 00006    ADDL3   BUFLEN, SRCPTR, BUFEND                  ;
                          58             56  D1 0000B    CMPL    SRCPTR, BUFEND                          ; 1697
                                         50  1E 0000E    BGEQU   1$                                      ;
                          57             04  D0 00010    MOVL    #4, SRCLEN                              ; 1699
                    00000000G  00        57  90 00013    MOVB    SRCLEN, CTL$T_NODEADDR                  ; 1700
          06              00   66        57  2C 0001A    MOVC5   SRCLEN, (SRCPTR), #0, #6, CTL$T_NODEADDR+1
                               00000000G 00      0001F
                          56             57  C0 00024    ADDL2   SRCLEN, SRCPTR                          ; 1701
                          58             56  D1 00027    CMPL    SRCPTR, BUFEND                          ; 1706
                                         34  1E 0002A    BGEQU   1$                                      ;
                          57             86  3C 0002C    MOVZWL  (SRCPTR)+, SRCLEN                       ; 1708
                    00000000G  00        57  90 0002F    MOVB    SRCLEN, CTL$T_NODENAME                  ; 1710
          06              00   66        57  2C 00036    MOVC5   SRCLEN, (SRCPTR), #0, #6, CTL$T_NODENAME+1
                               00000000G 00      0003B
                          56             57  C0 00040    ADDL2   SRCLEN, SRCPTR                          ; 1711
                          58             56  D1 00043    CMPL    SRCPTR, BUFEND                          ; 1716
                                         18  1E 00046    BGEQU   1$                                      ;
                          57             86  3C 00048    MOVZWL  (SRCPTR)+, SRCLEN                       ; 1718
                    00000000G  00        57  90 0004B    MOVB    SRCLEN, CTL$T_REMOTEID                  ; 1720
          10              00   66        57  2C 00052    MOVC5   SRCLEN, (SRCPTR), #0, #16, CTL$T_REMOTEID+1
                               00000000G 00      00057
                          50             01  D0 0005C    MOVL    #1, R0                                  ; 1722
                                             04 0005F    RET
                                         50  D4 00060 1$: CLRL    R0                                      ; 1724
                                             04 00062    RET

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 081A
```

```
;  1407          1725  2
;  1408          1726  2 LOCAL
;  1409          1727  2     nfb:                BBLOCK [nfb$c_length + (3 * 4)],
;  1410          1728  2     key:                VECTOR [2],
;  1411          1729  2     buffer:             BBLOCK [4 + (2 + 6) + (2 + 16)],
;  1412          1730  2     chan:               WORD,
;  1413          1731  2     iosb:               VECTOR [4,WORD],
;  1414          1732  2     nfb_desc:           VECTOR [2] INITIAL (%ALLOCATION(nfb), nfb),
;  1415          1733  2     key_desc:           VECTOR [2] INITIAL (%ALLOCATION(key), key),
;  1416          1734  2     buffer_desc:        VECTOR [2] INITIAL (%ALLOCATION(buffer), buffer),
;  1417          1735  2     arglist:            VECTOR [3] INITIAL (2, buffer, 0);
;  1418          1736  2
;  1419          1737  2 !
;  1420          1738  2 ! Set up NFB for NETACP QIO
;  1421          1739  2 !
;  1422          1740  2 CH$FILL(0, %ALLOCATION(nfb), nfb);
;  1423          1741  2 nfb [nfb$b_fct] = nfb$c_fc_show;
;  1424          1742  2 nfb [nfb$b_flags] = nfb$m_noctx;
;  1425          1743  2 nfb [nfb$b_database] = nfb$c_db_lli;
;  1426          1744  2 nfb [nfb$b_oper] = nfb$c_op_eql;
;  1427          1745  2 nfb [nfb$l_srch_key] = nfb$c_lli_lln;
;  1428          1746  2 nfb [nfb$l_fldid] = nfb$c_lli_pna;
;  1429          1747  2 (nfb [nfb$[_fldid]) + 4 = nfb$c_lli_pnn;
;  1430          1748  2 (nfb [nfb$l_fldid]) + 8 = nfb$c_lli_rid;
;  1431          1749  2
;  1432          1750  2 !
;  1433          1751  2 ! Store logical link number as key of reference
;  1434          1752  2 ! Exit without calling NETACP if link was already processed
;  1435          1753  2 !
;  1436          1754  2 key [0] = 0;
;  1437          1755  2 IF (key [1] = .link) EQL .last_link
;  1438          1756  2 THEN RETURN;
;  1439          1757  2
;  1440          1758  2 !
;  1441          1759  2 ! Assign channel to network device
;  1442          1760  2 ! Issue QIO to NETACP
;  1443          1761  2 ! Deassign the channel
;  1444          1762  2 !
;  1445        P 1763  2 IF NOT $ASSIGN(DEVNAM = %ASCID '_NET:',
;  1446          1764  3                  CHAN   = chan)
;  1447          1765  2 THEN RETURN;
;  1448        P 1766  2 IF NOT $QIOW(CHAN = .chan,
;  1449        P 1767  2              FUNC = io$_acpcontrol,
;  1450        P 1768  2              IOSB = iosb,
;  1451        P 1769  2              P1   = nfb_desc,
;  1452        P 1770  2              P2   = key_desc,
;  1453        P 1771  2              P3   = arglist[2],
;  1454          1772  3              P4   = buffer_desc)
;  1455          1773  2 THEN iosb [0] = 0;
;  1456          1774  2 $DASSGN(CHAN = .chan);
;  1457          1775  2 IF NOT .iosb [0]
;  1458          1776  2 THEN RETURN;
;  1459          1777  2
;  1460          1778  2 !
;  1461          1779  2 ! Go load P1 space with the node info
;  1462          1780  2 ! Remember the last link we fully processed
```

```
;  1463          1781  2 !
;  1464          1782  3 IF $CMKRNL(ROUTIN = load_node_info, ARGLST = arglist)
;  1465          1783  2 THEN
;  1466          1784  2     last_link = .link;
;  1467          1785  2
;  1468          1786  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000002  0009C P.AAR:  .LONG   2                                        ;
                                        00000000  00000000  000A0        .LONG   0, 0                                    ;
                                  00  00  00  3A  54  45  4E  5F  000A8 P.AAT:  .ASCII  \_NET:\<0><0><0>                 ;
                                                010E0005  000B0 P.AAS:  .LONG   17694725                                ;
                                                00000000' 000B4        .ADDRESS P.AAT                                   ;

                                                                .EXTRN  SYS$ASSIGN, SYS$QIOW
                                                                .EXTRN  SYS$DASSGN

                                                                .PSECT  $CODE$,NOWRT 2

                                                   003C 00000   .ENTRY  SET_NODE_NAME, Save R2,R3,R4,R5                 ; 1655
                                        5E      8C  AE  9E 00002        MOVAB   -116(SP), SP                            ;
                                     20 AE          1C  D0 00006        MOVL    #28, NFB_DESC                           ; 1671
                                     24 AE      58  AE  9E 0000A        MOVAB   NFB, NFB_DESC+4                         ;
                                     18 AE          08  D0 0000F        MOVL    #8, KEY_DESC                            ;
                                     1C AE      50  AE  9E 00013        MOVAB   KEY, KEY_DESC+4                         ;
                                     10 AE          1E  D0 00018        MOVL    #30, BUFFER_DESC                        ;
                                     14 AF      30  AE  9E 0001C        MOVAB   BUFFER, BUFFER_DESC+4                   ;
                                04 AE  0000'   CF      0C  28 00021     MOVC3   #12, P.AAR, ARGLIST                     ; 1735
                                     08 AE      30  AE  9E 00028        MOVAB   BUFFER, ARGLIST+4                       ; 1671
                             1C         00          6E      00  2C 0002D MOVC5   #0, (SP), #0, #28, NFB                  ; 1740
                                                        58  AE      00032
                                     58 AE 00080422     8F  D0 00034     MOVL    #525346, NFB                           ; 1741
                                     5C AE 08010012     8F  D0 0003C     MOVL    #134283282, NFB+4                       ; 1745
                                     68 AE 08010014     8F  D0 00044     MOVL    #134283284, NFB+16                      ; 1746
                                     6C AE 08020043     8F  D0 0004C     MOVL    #134348867, NFB+20                      ; 1747
                                     70 AE 08020044     8F  D0 00054     MOVL    #134348868, NFB+24                      ; 1748
                                              50  AE    D4  0005C        CLRL    KEY                                     ; 1754
                                        50      04  AC  D0 0005F        MOVL    LINK, R0                                ; 1755
                                     54 AE          50  D0 00063        MOVL    R0, KEY+4                               ;
                                0000' CF            50  D1 00067        CMPL    R0, LAST_LINK                           ;
                                                    60  13 0006C        BEQL    2$                                      ;
                                                    7E  7C 0006E        CLRQ    -(SP)                                   ; 1764
                                              08    AE  9F 00070        PUSHAB  CHAN                                    ;
                                           0000'    CF  9F 00073        PUSHAB  P.AAS                                   ;
                                00000000G  00           04  FB 00077     CALLS   #4, SYS$ASSIGN                          ;
                                           4D           50  E9 0007E     BLBC    R0, 2$                                  ;
                                                        7E  7C 00081     CLRQ    -(SP)                                   ; 1772
                                                 18     AE  9F 00083     PUSHAB  BUFFER_DESC                             ;
                                                 18     AE  9F 00086     PUSHAB  ARGLIST+8                               ;
                                                 28     AE  9F 00089     PUSHAB  KEY_DESC                                ;
                                                 34     AE  9F 0008C     PUSHAB  NFB_DESC                                ;
                                                            7E  7C 0008F CLRQ    -(SP)                                   ;
                                                 48     AE  9F 00091     PUSHAB  IOSB                                    ;
                                                            38  DD 00094 PUSHL   #56                                     ;
```

```
                         7E      28    AE  3C 00096          MOVZWL  CHAN, -(SP)                    ;
                                       7E  D4 0009A          CLRL    -(SP)                        ;
                00000000G  00          0C  FB 0009C          CALLS   #12, SYS$QIOW                ;
                           03          50  E8 000A3          BLBS    R0, 1$                       ;
                                 28    AE  B4 000A6          CLRW    IOSB                         ; 1773
                           7E          6E  3C 000A9 1$:      MOVZWL  CHAN, -(SP)                  ; 1774
                00000000G  00          01  FB 000AC          CALLS   #1, SYS$DASSGN               ;
                           17          28  AE  E9 000B3      BLBC    IOSB, 2$                     ; 1775
                                       04  AE  9F 000B7      PUSHAB  ARGLIST                      ; 1782
                                     FEDF  CF  9F 000BA      PUSHAB  LOAD_NODE_INFO               ;
                00000000G  00          02  FB 000BE          CALLS   #2, SYS$CMKRNL               ;
                           06          50  E9 000C5          BLBC    R0, 2$                       ;
                     0000'  CF     04    AC  D0 000C8          MOVL    LINK, LAST_LINK              ; 1784
                                           04  000CE 2$:      RET                                  ; 1786

; Routine Size:  207 bytes,    Routine Base:  $CODE$ + 087D
```

```
;  1470        1787  1 GLOBAL ROUTINE set_term_name: NOVALUE =
;  1471        1788  1
;  1472        1789  1 !---
;  1473        1790  1 !
;  1474        1791  1 !       Set the terminal name in the PCB.
;  1475        1792  1 !
;  1476        1793  1 ! Inputs:
;  1477        1794  1 !
;  1478        1795  1 !       term_name = Descriptor of terminal name
;  1479        1796  1 !
;  1480        1797  1 ! Outputs:
;  1481        1798  1 !
;  1482        1799  1 !       None.  PCB is updated.
;  1483        1800  1 !
;  1484        1801  1 !---
;  1485        1802  1
;  1486        1803  2 BEGIN
;  1487        1804  2
;  1488        1805  2 OWN
;  1489        1806  2     link;                                  ! Remote terminal's local link
;  1490        1807  2
;  1491        1808  2 ROUTINE load_term_set_link: NOVALUE =      ! $CMKRNL subroutine to load PCB
;  1492        1809  3 BEGIN                                      !  and fetch remote terminal's link
;  1493        1810  3
;  1494        1811  3 CH$COPY(.term_name [0],                    ! Copy terminal name
;  1495        1812  3         .term_name [1],                    !  with the leading "_"
;  1496        1813  3         ' ',                               !   blank filled
;  1497        1814  3         8,                                 !     into
;  1498        1815  3         ctl$gl_pcb [pcb$t_terminal]);      !       the PCB
;  1499        1816  3 ctl$gl_pcb [$BYTEOFFSET(pcb$t_terminal),0,8,0] = ! Change to ASCIC
;  1500        1817  3                  .term_name [0] - 1;      !  without the leading "_"
;  1501        1818  3
;  1502        1819  3 link = 0;                                  ! Assume no link to fetch
;  1503        1820  3 IF .dev_char_2 [dev$v_rtt]                 ! If this is a remote terminal,
;  1504        1821  3 THEN
;  1505        1822  4     BEGIN
;  1506        1823  4     LOCAL
;  1507        1824  4         chn: WORD,                         ! Channel to terminal
;  1508        1825  4         ucb: REF $BBLOCK;                  ! CCB/UCB pointer
;  1509      P 1826  4     IF $ASSIGN(DEVNAM = term_name,         ! Assign channel to terminal
;  1510        1827  5                CHAN   = chn)
;  1511        1828  4     THEN
;  1512        1829  5         BEGIN
;  1513        1830  5         ucb = .ctl$gl_ccbbase - .chn;      ! Get the CCB
;  1514        1831  5         ucb = .ucb [ccb$l_ucb];            ! Get the UCB
;  1515        1832  5         link = .ucb [ucb$w_rtt_link];      ! Get the local link number
;  1516        1833  5         $DASSGN(CHAN = .chn);              ! Deassign the channel
;  1517        1834  4         END;
;  1518        1835  3     END;
;  1519        1836  3
;  1520        1837  2 END;


                                                .PSECT  $OWN$,NOEXE,2

                                    00214 LINK:     .BLKB   4
```

```
                                                        .PSECT  $CODE$,NOWRT,2

                                                01FC 00000 LOAD_TERM_SET_LINK:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                ; 1808
                                    58 00000000G    00  9E 00002          MOVAB   TERM_NAME, R8
                                    5E              04  C2 00009          SUBL2   #4, SP
                                    57              68  D0 0000C          MOVL    TERM_NAME, R7            ; 1811
                                    50        04    A8  D0 0000F          MOVL    TERM_NAME+4, R0          ; 1812
                                    56 00000000G    00  D0 00013          MOVL    CTL$GL_PCB, R6           ; 1815
                08          20                  60      57  2C 0001A      MOVC5   R7, (R0), #32, #8, 68(R6)
                                                    44  A6      0001F
                        44  A6                  57      01  83 00021      SUBB3   #1, R7, 68(R6)           ; 1817
                                                0000'   CF  D4 00026      CLRL    LINK                     ; 1819
                            30 00000000G        00      02  E1 0002A      BBC     #2, DEV_CHAR_2, 1$       ; 1820
                                                        7E  7C 00032      CLRQ    -(SP)                    ; 1827
                                                    08  AE  9F 00034      PUSHAB  CHN
                                                        58  DD 00037      PUSHL   R8
                                00000000G       00      04  FB 00039      CALLS   #4, SYS$ASSIGN
                                                1F      50  E9 00040      BLBC    R0, 1$
                                                50      6E  3C 00043      MOVZWL  CHN, UCB                 ; 1830
                            50 00000000G        00      50  C3 00046      SUBL3   UCB, CTL$GL_CCBBASE, UCB
                                                50      60  D0 0004E      MOVL    (UCB), UCB               ; 1831
                                    0000'       CF  00D6    C0  3C 00051  MOVZWL  214(UCB), LINK           ; 1832
                                                7E      6E  3C 00058      MOVZWL  CHN, -(SP)               ; 1833
                                00000000G       00      01  FB 0005B      CALLS   #1, SYS$DASSGN
                                                            04  00062 1$:  RET                             ; 1837

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 094C

; 1521          1838  2
; 1522          1839  2 $CMKRNL(ROUTIN = load_term_set_link);   ! Load terminal into PCB, set link
; 1523          1840  2
; 1524          1841  2 IF .link NEQ 0                          ! If there is a link,
; 1525          1842  2 THEN
; 1526          1843  2     set_node_name(.link);               ! Store the node stuff into P1 space
; 1527          1844  2
; 1528          1845  1 END;


                                                  0000 00000        .ENTRY  SET_TERM_NAME, Save nothing      ; 1787
                                                7E  D4 00002        CLRL    -(SP)                            ; 1839
                                              96    AF  9F 00004    PUSHAB  LOAD_TERM_SET_LINK
                                00000000G     00    02  FB 00007    CALLS   #2, SYS$CMKRNL
                                              50        CF  D0 0000E  MOVL    LINK, R0                         ; 1841
                                                        07  13 00013  BEQL    1$
                                                        50  DD 00015  PUSHL   R0                               ; 1843
                                    FEB2        CF      01  FB 00017  CALLS   #1, SET_NODE_NAME
                                                            04  0001C 1$:  RET                              ; 1845

; Routine Size:  29 bytes,    Routine Base:  $CODE$ + 09AF
```

```
;  1530          1846  1 GLOBAL ROUTINE set_uic (new_uic) =
;  1531          1847  1
;  1532          1848  1 !---
;  1533          1849  1 !
;  1534          1850  1 !       Set the process UIC
;  1535          1851  1 !
;  1536          1852  1 ! Inputs:
;  1537          1853  1 !
;  1538          1854  1 !       Access mode = Kernel
;  1539          1855  1 !
;  1540          1856  1 !       ap = New UIC
;  1541          1857  1 !
;  1542          1858  1 ! Outputs:
;  1543          1859  1 !
;  1544          1860  1 !       routine = Previous UIC
;  1545          1861  1 !---
;  1546          1862  1
;  1547          1863  2 BEGIN
;  1548          1864  2
;  1549          1865  2 BUILTIN AP;
;  1550          1866  2
;  1551          1867  2
;  1552          1868  2 LOCAL
;  1553          1869  2     prev_uic;
;  1554          1870  2
;  1555          1871  2 prev_uic = .ctl$gl_pcb [pcb$l_uic];      ! Save previous UIC
;  1556          1872  2
;  1557          1873  2 ctl$gl_pcb [pcb$l_uic] = .ap;           ! Set UIC
;  1558          1874  2
;  1559          1875  2 RETURN .prev_uic;                       ! Return with previous UIC
;  1560          1876  2
;  1561          1877  1 END;


                                        0000 00000      .ENTRY  SET_UIC, Save nothing                   ; 1846
                      50 00000000G   00   D0 00002      MOVL    CTL$GL_PCB, R0                          ; 1871
                      51      00BC   C0   D0 00009      MOVL    188(R0), PREV_UIC                       ;
              00BC    C0             5C   D0 0000E      MOVL    AP, 188(R0)                             ; 1873
                      50             51   D0 00013      MOVL    PREV_UIC, R0                            ; 1875
                                          04 00016      RET                                             ; 1877

; Routine Size:  23 bytes,    Routine Base:  $CODE$ + 09CC
```

```
;  1563          1878  1 GLOBAL ROUTINE create_logical(log_name,eqv_name,acc_mode,att_bute,tbl_name) =
;  1564          1879  1
;  1565          1880  1 !---
;  1566          1881  1 !
;  1567          1882  1 !       Create a logical name using the LNM services.
;  1568          1883  1 !
;  1569          1884  1 ! Inputs:
;  1570          1885  1 !
;  1571          1886  1 !       log_name = Address of descriptor of logical name
;  1572          1887  1 !       eqv_name = Address of descriptor of equivalence name
;  1573          1888  1 !       acc_mode = Access mode for logical name
;  1574          1889  1 !       att_bute = Address of longword of logical name attributes
;  1575          1890  1 !                  Optional: Default is no attributes
;  1576          1891  1 !       tbl_name = Address of descriptor of table name
;  1577          1892  1 !                  Optional. Default is LNM$PROCESS_TABLE
;  1578          1893  1 !
;  1579          1894  1 ! Outputs:
;  1580          1895  1 !
;  1581          1896  1 !       $CRELNM's status is returned.
;  1582          1897  1 !
;  1583          1898  1 !---
;  1584          1899  1
;  1585          1900  2 BEGIN
;  1586          1901  2
;  1587          1902  2 MACRO
;  1588          1903  2     lnm_attsize = 0, 0,16,0%,          ! Attributes size
;  1589          1904  2     lnm_atttype = 0,16,16,0%,          ! Attributes type
;  1590          1905  2     lnm_attaddr = 1, 0,32,0%,          ! Attributes address
;  1591          1906  2     lnm_attrlen = 2, 0,32,0%,          ! Attributes result length
;  1592          1907  2     lnm_strsize = 3, 0,16,0%,          ! String size
;  1593          1908  2     lnm_strtype = 3,16,16,0%,          ! String type
;  1594          1909  2     lnm_straddr = 4, 0,32,0%,          ! String address
;  1595          1910  2     lnm_strrlen = 5, 0,32,0%,          ! String result length
;  1596          1911  2     lnm_endlist = 6, 0,32,0%;          ! End-of-list
;  1597          1912  2
;  1598          1913  2 OWN
;  1599          1914  2     item_list : BLOCK[2*3+1,LONG]      ! $CRELNM item list (2 entries)
;  1600          1915  2                 PRESET([lnm_attsize] = 4,
;  1601          1916  2                        [lnm_atttype] = lnm$_attributes,                    ; Rc
;  1602          1917  2                        [lnm_attaddr] = 0,      ! Filled in...
;  1603          1918  2                        [lnm_attrlen] = 0,
;  1604          1919  2                        [lnm_strsize] = 0,      ! Filled in...
;  1605          1920  2                        [lnm_strtype] = lnm$_string,
;  1606          1921  2                        [lnm_straddr] = 0,      ! Filled in...
;  1607          1922  2                        [lnm_strrlen] = 0,
;  1608          1923  2                        [lnm_endlist] = 0);
;  1609          1924  2
;  1610          1925  2 BUILTIN
;  1611          1926  2     NULLPARAMETER;
;  1612          1927  2
;  1613          1928  2 MAP
;  1614          1929  2     eqv_name : REF VECTOR;
;  1615          1930  2
;  1616          1931  2 LOCAL
;  1617          1932  2     table_name;
;  1618          1933  2
;  1619          1934  2 table_name = %ASCID 'LNM$PROCESS_TABLE';        ! Default table to process
```

```
;  1620          1935  2 IF NOT NULLPARAMETER(5)                        ! If table name supplied
;  1621          1936  2 THEN table_name = .tbl_name;                   !  then use supplied one
;  1622          1937  2
;  1623          1938  2 item_list[lnm_attaddr] = UPLIT(0);             ! Default attributes to 0
;  1624          1939  2 IF NOT NULLPARAMETER(4)                        ! If attributes supplied
;  1625          1940  2 THEN item_list[lnm_attaddr] = .att_bute;       !  then use supplied ones
;  1626          1941  2
;  1627          1942  2 item_list[lnm_strsize] = .eqv_name[0];         ! Set string length
;  1628          1943  2 item_list[lnm_straddr] = .eqv_name[1];         !  and string address
;  1629          1944  2
;  1630        P 1945  2 $CRELNM(TABNAM = .table_name,                  ! Create the logical name
;  1631        P 1946  2         LOGNAM = .log_name,
;  1632        P 1947  2         ACMODE = acc_mode,
;  1633          1948  3         ITMLST = item_list)
;  1634          1949  3
;  1635          1950  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

42  41  54  5F  53  53  45  43  4F  52  50  24  4D  4E  4C  000B8 P.AAV:  .ASCII  \LNM$PROCESS_TABLE\<0><0><0>
                                        00  00  00  45  4C  000C7
                                                    010E0011  000CC P.AAU:  .LONG   17694737
                                                    00000000' 000D0         .ADDRESS P.AAV
                                                    00000000  000D4 P.AAW:  .LONG   0

                                                        .PSECT  $OWN$,NOEXE,2

                                                0003  0004  00218 ITEM_LIST:
                                                                    .WORD   4, 3
                                                00000000  00000000  0021C .LONG   0, 0
                                                          0002  0000  00224 .WORD   0, 2
                                                00000000  00000000  00228 .LONG   0, 0, 0

                                                        .EXTRN  SYS$CRELNM

                                                        .PSECT  $CODE$,NOWRT,2

                                                          0004 00000        .ENTRY  CREATE_LOGICAL, Save R2                 ; 1878
                                            52      0000'  CF  9E 00002     MOVAB   ITEM_LIST+4, R2
                                            51      0000'  CF  9E 00007     MOVAB   P.AAO, TABLE_NAME                       ; 1934
                                            05             6C  91 0000C     CMPB    (AP), #5                                ; 1935
                                                           09  1F 0000F     BLSSU   1$
                                                14         AC  D5 00011     TSTL    20(AP)
                                                           04  13 00014     BEQL    1$
                                            51  14         AC  D0 00016     MOVL    TBL_NAME, TABLE_NAME                    ; 1936
                                            62      0000'  CF  9E 0001A 1$:  MOVAB   P.AAW, ITEM_LIST+4                    ; 1938
                                            04             6C  91 0001F     CMPB    (AP), #4                                ; 1939
                                                           09  1F 00022     BLSSU   2$
                                                10         AC  D5 00024     TSTL    16(AP)
                                                           04  13 00027     BEQL    2$
                                            62  10         AC  D0 00029     MOVL    ATT_BUTE, ITEM_LIST+4                   ; 1940
                                            50  08         AC  D0 0002D 2$:  MOVL    EQV_NAME, R0                            ; 1942
                                      08    A2             60  B0 00031     MOVW    (R0), ITEM_LIST+12
                                      0C    A2         04  A0  D0 00035     MOVL    4(R0), ITEM_LIST+16                     ; 1943
                                                     FC  A2  9F 0003A       PUSHAB  ITEM_LIST                               ; 1948
```

```
                                 0C   AC  9F 0003D          PUSHAB  ACC_MODE                                        ;
                                 04   AC  DD 00040          PUSHL   LOG_NAME                                        ;
                                      51  DD 00043          PUSHL   TABLE_NAME                                      ;
                                      7E  D4 00045          CLRL    -(SP)                                           ;
                          00000000G   00  05  FB 00047      CALLS   #5, SYS$CRELNM                                  ;
                                          04 0004E          RET                                                     ; 1950

; Routine Size:  79 bytes,    Routine Base:  $CODE$ + 09E3
```

```
;1637     1951  1 ROUTINE make_rightslists : NOVALUE =
;1638     1952  2 BEGIN
;1639     1953  2
;1640     1954  2 !+++
;1641     1955  2 !
;1642     1956  2 ! Collect all the identifiers associated with this user, and then
;1643     1957  2 ! copy them into appropriate places in the process PCB, and maybe
;1644     1958  2 ! non-paged pool.
;1645     1959  2 !
;1646     1960  2 ! Inputs:
;1647     1961  2 !       None.  The rights database is interrogated for the information.
;1648     1962  2 !
;1649     1963  2 ! Outputs:
;1650     1964  2 !       None.  The PCB is updated.
;1651     1965  2 !
;1652     1966  2 !---
;1653     1967  2
;1654     1968  2 LITERAL
;1655     1969  2     id_number = (arb$s_localrights/8) - 1;
;1656     1970  2 LOCAL
;1657     1971  2     status,
;1658     1972  2     n : INITIAL(0),
;1659     1973  2     proc_type : INITIAL(0),
;1660     1974  2     term_type : INITIAL(0),
;1661     1975  2     arglist : VECTOR[5],
;1662     1976  2     desc : $BBLOCK[dsc$c_s_bln],                ! Descriptor for extended rights
;1663     1977  2     rights : VECTOR[id_number*2]                ! Local copy of rights list
;1664     1978  2                 INITIAL(REP 2*id_number of (0))    ! preset to zero
;1665     1979  2                 VOLATILE,
;1666     1980  2     context : INITIAL (0) VOLATILE,             ! Context block for SYS$FIND_HELD
;1667     1981  2     holder_block : VECTOR[2] VOLATILE;          ! Identification for this user
;1668     1982  2
;1669     1983  2 !
;1670     1984  2 ! The first step is to get as many identifiers as possible in the local
;1671     1985  2 ! rights list.  Note that the literal ID_NUMBER is actually one less than
;1672     1986  2 ! the number of ID's that go into the local rights list.  This is because the
;1673     1987  2 ! first ID, the UIC, is set elsewhere.
;1674     1988  2 !
;1675     1989  2 ! First, the environmental rights.
;1676     1990  2 !
;1677     1991  2 IF .pcb_sts[$BITPOSITION(pcb$v_batch)]               ! Process type
;1678     1992  2 THEN proc_type = %ASCID 'BATCH'
;1679     1993  2 ELSE IF .pcb_sts[$BITPOSITION(pcb$v_netwrk)]
;1680     1994  2 THEN proc_type = %ASCID 'NETWORK'
;1681     1995  2 ELSE IF .pcb_sts[$BITPOSITION(pcb$v_inter)]          ! For interactive
;1682     1996  2 THEN                                               ! processes, find
;1683     1997  3     BEGIN                                          ! the type of terminal
;1684     1998  3     proc_type = %ASCID 'INTERACTIVE';
;1685     1999  3     IF .terminal_device
;1686     2000  3     THEN
;1687     2001  4         BEGIN
;1688     2002  4         IF .dev_char_2[dev$v_rtt]
;1689     2003  4         THEN term_type = %ASCID 'REMOTE'
;1690     2004  4         ELSE IF .dev_dep_2[tt2$v_dialup]
;1691     2005  4         THEN term_type = %ASCID 'DIALUP'
;1692     2006  4         ELSE term_type = %ASCID 'LOCAL';
;1693     2007  3         END;
```

```
;  1694          2008  2      END;
;  1695          2009  2
;  1696          2010  2  IF .proc_type NEQ 0                          ! If some kind of
;  1697          2011  2  THEN
;  1698          2012  3      BEGIN
;  1699        P 2013  3      IF $ASCTOID(NAME    = .proc_type,
;  1700        P 2014  3                  ID      = rights[2*.n],
;  1701          2015  4                  ATTRIB = rights[(2*.n)+1])
;  1702          2016  3      THEN n = .n + 1;
;  1703          2017  3      IF .term_type NEQ 0
;  1704          2018  3      THEN
;  1705          2019  4          BEGIN
;  1706        P 2020  4          IF $ASCTOID(NAME    = .term_type,
;  1707        P 2021  4                      ID      = rights[2*.n],
;  1708          2022  5                      ATTRIB = rights[(2*.n)+1])
;  1709          2023  4          THEN n = .n + 1;
;  1710          2024  3          END;
;  1711          2025  2      END;
;  1712          2026  2
;  1713          2027  2  !
;  1714          2028  2  ! Get the non-environmental rights.
;  1715          2029  2  !
;  1716          2030  2  holder_block[0] = .uaf_record[uaf$l_uic];
;  1717          2031  2  holder_block[1] = 0;
;  1718          2032  2
;  1719          2033  2  arglist[0] = 4;
;  1720          2034  2  arglist[1] = holder_block;
;  1721          2035  2  arglist[4] = context;
;  1722          2036  2
;  1723          2037  2  INCR i FROM .n TO (id_number - 1) DO
;  1724          2038  3      BEGIN
;  1725          2039  3      arglist[2] = rights[2*.i];
;  1726          2040  3      arglist[3] = rights[2*.i+1];
;  1727        P 2041  4      IF NOT (status = $CMEXEC(ROUTIN = SYS$FIND_HELD,
;  1728          2042  4                               ARGLST = arglist))
;  1729          2043  3      THEN EXITLOOP;
;  1730          2044  2      END;
;  1731          2045  2
;  1732          2046  2  !
;  1733          2047  2  ! Call the kernel-mode routine to set these in place.
;  1734          2048  2  !
;  1735          2049  3  BEGIN
;  1736          2050  3  LOCAL status2;
;  1737        P 2051  4  IF NOT (status2 = $CMKRNL(ROUTIN = set_localrights,
;  1738          2052  4                            ARGLST = rights))
;  1739          2053  3  THEN SIGNAL_STOP(.status2);
;  1740          2054  2  END;
;  1741          2055  2
;  1742          2056  2  !
;  1743          2057  2  ! It may be that there are more than 15 ID's to put into place.  If so,
;  1744          2058  2  ! then keep getting them, but put them in an expandable buffer.
;  1745          2059  2  !
;  1746          2060  2  IF .status
;  1747          2061  2  THEN
;  1748          2062  3      BEGIN
;  1749          2063  3      $init_dyndesc(desc);                    ! Create a dynamic descriptor
;  1750          2064  3
```

```
;  1751          2065  3      rights[0] = 8;                          ! Set up a descriptor pointing to
;  1752          2066  3      rights[1] = rights[2];                  ! the new ID to add to the list
;  1753          2067  3
;  1754          2068  3      arglist[0] = 4;
;  1755          2069  3      arglist[1] = holder_block;
;  1756          2070  3      arglist[2] = rights[2];
;  1757          2071  3      arglist[3] = rights[3];
;  1758          2072  3      arglist[4] = context;
;  1759          2073  3
;  1760        P 2074  3      WHILE $CMEXEC(ROUTIN = SYS$FIND_HELD,
;  1761          2075  4                    ARGLST = arglist)
;  1762          2076  3      DO (str$append(desc, rights));
;  1763          2077  3
;  1764          2078  3      IF .desc[dsc$w_length] NEQ 0            ! If there are more,
;  1765          2079  3      THEN                                    ! then continue
;  1766          2080  4          BEGIN
;  1767          2081  4
;  1768          2082  4 !
;  1769          2083  4 ! Call the kernel-mode routine that will allocate a suitable chunk of
;  1770          2084  4 ! non-paged pool in which to put the rights list, and point to it.
;  1771          2085  4 !
;  1772        P 2086  5          IF NOT (status = $CMKRNL(ROUTIN = set_more_rights,
;  1773          2087  5                                   ARGLST = desc))
;  1774          2088  4          THEN SIGNAL_STOP(.status);
;  1775          2089  3          END;
;  1776          2090  2      END;
;  1777          2091  2
;  1778          2092  2 RETURN;
;  1779          2093  1 END;


                                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000000# 000D8 P.AAX:  .LONG   0[14]
                                  00 00 00 48 43 54 41 42 00110 P.AAZ:  .ASCII  \BATCH\<0><0><0>
                                                010E0005 00118 P.AAY:  .LONG   17694725
                                                00000000' 0011C        .ADDRESS P.AAZ
                                  00 4B 52 4F 57 54 45 4E 00120 P.ABB:  .ASCII  \NETWORK\<0>
                                                010E0007 00128 P.ABA:  .LONG   17694727
                                                00000000' 0012C        .ADDRESS P.ABB
00 45 56 49 54 43 41 52 45 54 4E 49 00130 P.ABD:  .ASCII  \INTERACTIVE\<0>
                                                010E000B 0013C P.ABC:  .LONG   17694731
                                                00000000' 00140        .ADDRESS P.ABD
                                  00 00 45 54 4F 4D 45 52 00144 P.ABF:  .ASCII  \REMOTE\<0><0>
                                                010E0006 0014C P.ABE:  .LONG   17694726
                                                00000000' 00150        .ADDRESS P.ABF
                                  00 00 50 55 4C 41 49 44 00154 P.ABH:  .ASCII  \DIALUP\<0><0>
                                                010E0006 0015C P.ABG:  .LONG   17694726
                                                00000000' 00160        .ADDRESS P.ABH
                                  00 00 00 4C 41 43 4F 4C 00164 P.ABJ:  .ASCII  \LOCAL\<0><0><0>
                                                010E0005 0016C P.ABI:  .LONG   17694725
                                                00000000' 00170        .ADDRESS P.ABJ

                                                                  .EXTRN  SYS$ASCTOID

                                                                  .PSECT  $CODE$,NOWRT,2
```

```
                                    0FFC 00000 MAKE_RIGHTSLISTS:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1951
                          5B 00000000G   00   9E 00002        MOVAB   SYS$ASCTOID, R11
                          5A 00000000G   00   9E 00009        MOVAB   PCB_STS, R10
                          59     0000'   CF   9E 00010        MOVAB   P.AAX, R9
                          5E        A0   AE   9E 00015        MOVAB   -96(SP), SP
                                         58   D4 00019        CLRL    PROC_TYPE                             ; 1952
                                         56   7C 0001B        CLRQ    N
                0C    AE          69     38   28 0001D        MOVC3   #56, P.AAX, RIGHTS                    ; 1978
                                   08    AE   D4 00022        CLRL    CONTEXT
                      06        01  AA     06   E1 00025        BBC     #6, PCB_STS+1, 1$                     ; 1991
                                58     40   A9   9E 0002A        MOVAB   P.AAY, PROC_TYPE                      ; 1992
                                         3D   11 0002E        BRB     5$
                      06        02  AA     05   E1 00030 1$:     BBC     #5, PCB_STS+2, 2$                     ; 1993
                                58     50   A9   9E 00035        MOVAB   P.ABA, PROC_TYPE                      ; 1994
                                         32   11 00039        BRB     5$
                      2D        03  AA     01   E1 0003B 2$:     BBC     #1, PCB_STS+3, 5$                     ; 1995
                                58     64   A9   9E 00040        MOVAB   P.ABC, PROC_TYPE                      ; 1998
                                22 00000000G   00   E9 00044        BLBC    TERMINAL_DEVICE, 5$           ; 1999
                      06 00000000G   00        02   E1 0004B        BBC     #2, DEV_CHAR_2, 3$            ; 2002
                                57     74   A9   9E 00053        MOVAB   P.ABE, TERM_TYPE                      ; 2003
                                         14   11 00057        BRB     5$
                             00000000G   00   95 00059 3$:     TSTB    DEV_DEP_2+1                           ; 2004
                                         07   18 0005F        BGEQ    4$
                                57   0084   C9   9E 00061        MOVAB   P.ABG, TERM_TYPE                      ; 2005
                                         05   11 00066        BRB     5$
                                57   0094   C9   9E 00068 4$:     MOVAB   P.ABI, TERM_TYPE                      ; 2006
                                         58   D5 0006D 5$:     TSTL    PROC_TYPE                             ; 2010
                                         38   13 0006F        BEQL    7$
                      52                 56     01   78 00071        ASHL    #1, N, R2                     ; 2015
                                      10 AE42   DF   00075        PUSHAL  RIGHTS+4[R2]
                      52                 56     01   78 00079        ASHL    #1, N, R2
                                      10 AE42   DF   0007D        PUSHAL  RIGHTS[R2]
                                         58   DD 00081        PUSHL   PROC_TYPE
                                6B       03   FB 00083        CALLS   #3, SYS$ASCTOID
                                02       50   E9 00086        BLBC    R0, 6$
                                         56   D6 00089        INCL    N                                     ; 2016
                                         57   D5 0008B 6$:     TSTL    TERM_TYPE                             ; 2017
                                         1A   13 0008D        BEQL    7$
                      52                 56     01   78 0008F        ASHL    #1, N, R2                     ; 2022
                                      10 AE42   DF   00093        PUSHAL  RIGHTS+4[R2]
                      52                 56     01   78 00097        ASHL    #1, N, R2
                                      10 AE42   DF   0009B        PUSHAL  RIGHTS[R2]
                                         57   DD 0009F        PUSHL   TERM_TYPE
                                6B       03   FB 000A1        CALLS   #3, SYS$ASCTOID
                                02       50   E9 000A4        BLBC    R0, 7$
                                         56   D6 000A7        INCL    N                                     ; 2023
                                50 00000000G   00   D0 000A9 7$:     MOVL    UAF_RECORD, R0                ; 2030
                                6E       24   A0   D0 000B0        MOVL    36(R0), HOLDER_BLOCK
                                      04    AE   D4 000B4        CLRL    HOLDER_BLOCK+4                        ; 2031
                          4C    AE          04   D0 000B7        MOVL    #4, ARGLIST                           ; 2033
                          50    AE          6E   9E 000BB        MOVAB   HOLDER_BLOCK, ARGLIST+4               ; 2034
                          5C    AE       08    AE   9E 000BF        MOVAB   CONTEXT, ARGLIST+16                   ; 2035
                                52       FF   A6   9E 000C4        MOVAB   -1(R6), I                             ; 2037
                                         26   11 000C8        BRB     9$
                      56                 52     01   78 000CA 8$:     ASHL    #1, I, R6                     ; 2039
```

```
                54   AE        0C AE46   DE 000CE          MOVAL   RIGHTS[R6], ARGLIST+8
                58   AE        10 AE46   DE 000D4          MOVAL   RIGHTS+4[R6], ARGLIST+12          ;  2040
                                  4C   AE 9F 000DA          PUSHAB  ARGLIST                           ;  2042
                                       00000000G   00 9F 000DD  PUSHAB  SYS$FIND_HELD
          00000000G   00                   02 FB 000E3       CALLS   #2, SYS$CMEXEC
                      53                   50 D0 000EA       MOVL    R0, STATUS
                      04                   53 E9 000ED       BLBC    STATUS, 10$
D6                    52                   06 F3 000F0 9$:   AOBLEQ  #6, I, 8$                 ;  2037
                                  0C   AE 9F 000F4 10$:     PUSHAB  RIGHTS                        ;  2052
                                       0000V  CF 9F 000F7   PUSHAB  SET_LOCALRIGHTS
          00000000G   00                   02 FB 000FB       CALLS   #2, SYS$CMKRNL
                      09                   50 E8 00102       BLBS    STATUS2, 11$
                                           50 DD 00105       PUSHL   STATUS2                       ;  2053
          00000000G   00                   01 FB 00107       CALLS   #1, LIB$STOP
                      6F                   53 E9 0010E 11$:  BLBC    STATUS, 14$                 ;  2060
                44    AE  020E0000    8F   DO 00111         MOVL    #34471936, DESC             ;  2063
                                  48   AE D4 00119          CLRL    DESC+4
                0C    AE                   08 D0 0011C       MOVL    #8, RIGHTS                  ;  2065
                10    AE        14    AE 9E 00120          MOVAB   RIGHTS+8, RIGHTS+4          ;  2066
                4C    AE                   04 D0 00125       MOVL    #4, ARGLIST                 ;  2068
                50    AE                   6E 9E 00129       MOVAB   HOLDER_BLOCK, ARGLIST+4     ;  2069
                54    AE        14    AE 9E 0012D          MOVAB   RIGHTS+8, ARGLIST+8         ;  2070
                58    AE        18    AE 9E 00132          MOVAB   RIGHTS+12, ARGLIST+12       ;  2071
                5C    AE        08    AE 9E 00137          MOVAB   CONTEXT, ARGLIST+16         ;  2072
                                  4C   AE 9F 0013C 12$:     PUSHAB  ARGLIST                       ;  2075
                                       00000000G   00 9F 0013F  PUSHAB  SYS$FIND_HELD
          00000000G   00                   02 FB 00145       CALLS   #2, SYS$CMEXEC
                      0F                   50 E9 0014C       BLBC    R0, 13$
                                  0C   AE 9F 0014F          PUSHAB  RIGHTS                        ;  2076
                                  48   AE 9F 00152          PUSHAB  DESC
                         00000000G   00    02 FB 00155      CALLS   #2, STR$APPEND
                                           DE 11 0015C       BRB     12$
                                  44   AE B5 0015E 13$:     TSTW    DESC                          ;  2078
                                           1D 13 00161       BEQL    14$
                                  44   AE 9F 00163          PUSHAB  DESC                          ;  2087
                                       0000V  CF 9F 00166   PUSHAB  SET_MORE_RIGHTS
                         00000000G   00    02 FB 0016A      CALLS   #2, SYS$CMKRNL
                                     53    50 D0 00171      MOVL    R0, STATUS
                                     09    53 E8 00174      BLBS    STATUS, 14$
                                           53 DD 00177      PUSHL   STATUS                        ;  2088
                         00000000G   00    01 FB 00179      CALLS   #1, LIB$STOP
                                           04 00180 14$:    RET                                   ;  2093

; Routine Size:  385 bytes,    Routine Base:  $CODE$ + 0A32
```

```
;  1781          2094  1 ROUTINE set_localrights =
;  1782          2095  2 BEGIN
;  1783          2096  2
;  1784          2097  2 !+++
;  1785          2098  2 !
;  1786          2099  2 ! Copy the local rights list to the PCB.
;  1787          2100  2 !
;  1788          2101  2 ! Inputs:
;  1789          2102  2 !       mode is KERNEL
;  1790          2103  2 !       AP = address of the local rights list
;  1791          2104  2 !
;  1792          2105  2 ! Outputs:
;  1793          2106  2 !       None.  The PCB is altered.
;  1794          2107  2 !
;  1795          2108  2 !---
;  1796          2109  2
;  1797          2110  2 BUILTIN
;  1798          2111  2     ap;
;  1799          2112  2
;  1800          2113  2
;  1801          2114  2 BIND
;  1802          2115  2     pcb = .ctl$gl_pcb : $BBLOCK,
;  1803          2116  2     arb = .pcb[pcb$l_arb] : $BBLOCK;
;  1804          2117  2
;  1805          2118  2 !
;  1806          2119  2 ! Move the local copy into the PCB.  Note that we skip the first
;  1807          2120  2 ! ID, which is the UIC of the user.
;  1808          2121  2 !
;  1809          2122  2 CH$MOVE(arb$s_localrights - 8, .ap, arb[arb$r_localrights] + 8);
;  1810          2123  2
;  1811          2124  2 RETURN true;
;  1812          2125  1 END;


                                        003C 00000 SET_LOCALRIGHTS:
                                                   .WORD   Save R2,R3,R4,R5                ; 2094
                        50 00000000G    00   D0 00002      MOVL    CTL$GL_PCB, R0          ; 2115
                        50      008C    C0   D0 00009      MOVL    140(R0), R0             ; 2116
                40   A0                 6C   38 0000E      MOVC3   #56, (AP), 64(R0)       ; 2122
                        50              01   D0 00013      MOVL    #1, R0                  ; 2124
                                            04 00016      RET                             ; 2125

; Routine Size:  23 bytes,    Routine Base:  $CODE$ + 0BB3
```

```
;  1814          2126  1 ROUTINE set_more_rights =
;  1815          2127  2 BEGIN
;  1816          2128  2
;  1817          2129  2 !+++
;  1818          2130  2 !
;  1819          2131  2 ! Copy the extended rights list to non-paged pool, and
;  1820          2132  2 ! set a pointer in the PCB.
;  1821          2133  2 !
;  1822          2134  2 ! Inputs:
;  1823          2135  2 !       mode is KERNEL
;  1824          2136  2 !       AP = address of descriptor pointing to the rights list
;  1825          2137  2 !
;  1826          2138  2 ! Outputs:
;  1827          2139  2 !       None.  The PCB is modified.
;  1828          2140  2 !
;  1829          2141  2 !---
;  1830          2142  2
;  1831          2143  2 BUILTIN
;  1832          2144  2     ap;
;  1833          2145  2
;  1834          2146  2
;  1835          2147  2 BIND
;  1836          2148  2     pcb = .ctl$gl_pcb : $BBLOCK,
;  1837          2149  2     arb = .pcb[pcb$l_arb] : $BBLOCK,
;  1838          2150  2     desc = .ap : $BBLOCK,
;  1839          2151  2     rightslist = arb[arb$l_rightslist] : VECTOR;
;  1840          2152  2
;  1841          2153  2 LOCAL
;  1842          2154  2     status,
;  1843          2155  2     size,
;  1844          2156  2     chunk : REF VECTOR;
;  1845          2157  2
;  1846          2158  2 !
;  1847          2159  2 ! Check to see if there is already a rights list.  If so, deallocate it.
;  1848          2160  2 !
;  1849          2161  2 IF .rightslist[2] NEQ 0
;  1850          2162  2 THEN
;  1851          2163  3     BEGIN
;  1852          2164  4     IF NOT (status = exe$deanonpaged(.rightslist[2]))
;  1853          2165  3     THEN RETURN .status;
;  1854          2166  2     END;
;  1855          2167  2
;  1856          2168  2 !
;  1857          2169  2 ! Grab a chunk of non-paged pool large enough for the rights list.  This
;  1858          2170  2 ! must be the size in the descriptor, plus 12 bytes: 8 bytes to store the
;  1859          2171  2 ! descriptor, and another 4 for type and size of the chunk.
;  1860          2172  2 !
;  1861          2173  3 IF NOT (status = exe$alononpaged(.desc[dsc$w_length] + 12; size, chunk))
;  1862          2174  2 THEN RETURN ss$_insfmem;
;  1863          2175  2
;  1864          2176  2 chunk[0] = .desc[dsc$w_length];                ! Set up the descriptor
;  1865          2177  2 chunk[1] = chunk[3];
;  1866          2178  2 chunk[2] = dyn$c_rightslist^16 + .size;        ! Set size and type of block
;  1867          2179  2
;  1868          2180  2 CH$MOVE(.desc[dsc$w_length],                  ! Copy the local data
;  1869          2181  2          .desc[dsc$a_pointer],                ! into the rest of
;  1870          2182  2          chunk[3]);                           ! the block
```

```
;  1871          2183  2
;  1872          2184  2 rightslist[2] = .chunk;                    ! Record address of rights descriptor
;  1873          2185  2
;  1874          2186  2 RETURN true;
;  1875          2187  1 END;


                                                  0FFC 00000 SET_MORE_RIGHTS:
                                                             .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2126
                                  50 00000000G   00   D0 00002       MOVL    CTL$GL_PCB, R0                       ; 2148
                          57      008C   C0         20   C1 00009       ADDL3   #32, 140(R0), R7                     ; 2151
                                            08       A7   D5 0000F       TSTL    8(R7)                                ; 2161
                                                     0D   13 00012       BEQL    1$
                                  50        08       A7   D0 00014       MOVL    8(R7), R0                            ; 2164
                                     00000000G      00   16 00018       JSB     EXE$DEANONPAGED
                                  35                50   E9 0001E       BLBC    STATUS, 3$
                                  51                6C   3C 00021 1$:    MOVZWL  (AP), R1                             ; 2173
                                  51                0C   C0 00024       ADDL2   #12, R1
                                     00000000G      00   16 00027       JSB     EXE$ALONONPAGED
                                  56                52   D0 0002D       MOVL    R2, R6
                                  06                50   E8 00030       BLBS    STATUS, 2$
                                  50        0124    8F   3C 00033       MOVZWL  #292, R0                             ; 2174
                                                     04   00038       RET
                                  66                6C   3C 00039 2$:    MOVZWL  (AP), (CHUNK)                        ; 2176
                             04   A6        0C      A6   9E 0003C       MOVAB   12(CHUNK), 4(CHUNK)                  ; 2177
                             08   A6  00420000      E1   9E 00041       MOVAB   4325376(R1), 8(CHUNK)                ; 2178
                       0C    A6        04   BC      6C   28 00049       MOVC3   (AP), @4(AP), 12(CHUNK)              ; 2182
                             08   A7                56   D0 0004F       MOVL    CHUNK, 8(R7)                         ; 2184
                                  50                01   D0 00053       MOVL    #1, R0                               ; 2186
                                                     04   00056 3$:    RET                                          ; 2187

; Routine Size:  87 bytes,    Routine Base:  $CODE$ + 0BCA
```

```
;  1877          2188  1 GLOBAL ROUTINE set_lnm_tables =
;  1878          2189  1
;  1879          2190  1 !---
;  1880          2191  1 !
;  1881          2192  1 !       Set the correct group and job-wide logical name tables.
;  1882          2193  1 !
;  1883          2194  1 ! Inputs:
;  1884          2195  1 !
;  1885          2196  1 !       subprocess = TRUE if the process is a sub-process
;  1886          2197  1 !       uaf_record = Address of UAF record for user ( must be non-zero )
;  1887          2198  1 !
;  1888          2199  * !---
;  1889          2200  1
;  1890          2201  2 BEGIN
;  1891          2202  2
;  1892          2203  2 LOCAL
;  1893          2204  2     group_table_name:   VECTOR [ 16, BYTE ],
;  1894          2205  2     job_table_name:     VECTOR [ 16, BYTE ],
;  1895          2206  2     table_name_desc:    VECTOR [ 2, LONG ]  INITIAL ( 16, group_table_name ),
;  1896          2207  2     item_list:          VECTOR [ (2*3)+1, LONG ];
;  1897          2208  2
;  1898          2209  2 !
;  1899          2210  2 ! Convert the binary group number to ASCII and append it to "LNM$GROUP_"
;  1900          2211  2 !
;  1901          2212  2
;  1902        P 2213  2 $FAO ( %ASCID 'LNM$GROUP_!OW',
;  1903        P 2214  2        table_name_desc,
;  1904        P 2215  2        table_name_desc,
;  1905          2216  2        .uaf_record [ uaf$w_grp ] );
;  1906          2217  2
;  1907          2218  2 !
;  1908          2219  2 ! Construct the item list for LNM$GROUP and re-create the logical name.
;  1909          2220  2 !
;  1910          2221  2
;  1911          2222  2 item_list [ 0 ] = (LNM$_ATTRIBUTES^16 OR 4);     ! Define the translation to be
;  1912          2223  2 item_list [ 1 ] = UPLIT (LNM$M_TERMINAL);        !  terminal
;  1913          2224  2 item_list [ 2 ] = 0;
;  1914          2225  2
;  1915          2226  2 item_list [ 3 ] = (LNM$_STRING^16 OR .table_name_desc [ 0 ] );
;  1916          2227  2 item_list [ 4 ] = .table_name_desc [ 1 ];        ! Define the translation string
;  1917          2228  2 item_list [ 5 ] = 0;
;  1918          2229  2
;  1919          2230  2 item_list [ 6 ] = 0;                            ! End the item list
;  1920          2231  2
;  1921        P 2232  2 $CRELNM ( ACMODE = %REF(PSL$C_KERNEL),
;  1922        P 2233  2           ITMLST = item_list,
;  1923        P 2234  2           LOGNAM = %ASCID 'LNM$GROUP',
;  1924          2235  2           TABNAM = %ASCID 'LNM$PROCESS_DIRECTORY');
;  1925          2236  2
;  1926          2237  2 !
;  1927          2238  2 ! If the process is not a sub-process, re-create the job-wide and group
;  1928          2239  2 ! logical name tables. While the job table will be created in a fashion that
;  1929          2240  2 ! will causing the existing table (created within PROCSTRT) to be deleted, this
;  1930          2241  2 ! will not be the case with the group table that is to be created. If a group
;  1931          2242  2 ! table with the same name is found, it is left undisturbed and no table
;  1932          2243  2 ! creation takes place.
;  1933          2244  2 !
```

```
;  1934          2245  2
;  1935          2246  2 IF NOT .subprocess
;  1936          2247  2 THEN
;  1937          2248  3     BEGIN
;  1938          2249  3
;  1939          2250  3     !
;  1940          2251  3     ! Construct the name of the job-wide logical name table by appending to
;  1941          2252  3     ! "LNM$JOB_" the ASCII representation of the process's JIB address.
;  1942          2253  3     !
;  1943          2254  3
;  1944          2255  3     table_name_desc[ 1 ] = job_table_name;
;  1945        P 2256  3     $FAO ( %ASCID 'LNM$JOB_!XL',
;  1946        P 2257  3            table_name_desc,
;  1947        P 2258  3            table_name_desc,
;  1948          2259  3            .ctl$gl_pcb [ pcb$l_jib ] );
;  1949          2260  3
;  1950          2261  3     !
;  1951          2262  3     ! Create the job and group logical name tables.
;  1952          2263  3     !
;  1953          2264  3
;  1954          2265  3     RETURN EXE$CRE_JGTABLE( .uaf_record[ uaf$l_jtquota ],
;  1955          2266  3                             job_table_name + 8,
;  1956          2267  3                             group_table_name + 10 );
;  1957          2268  3     END
;  1958          2269  2 ELSE
;  1959          2270  2     RETURN 1;
;  1960          2271  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

00 00 57 4F 21 5F 50 55 4F 52 47 24 4D 4E 4C  00174 P.ABL:   .ASCII  \LNM$GROUP_!OW\<0><0><0>
                                                 00  00183
                                        010E000D 00184 P.ABK:   .LONG   17694733
                                        00000000' 00188         .ADDRESS P.ABL
                                        00000200 0018C P.ABM:   .LONG   512
52 49 44 5F 53 53 45 43 4F 52 50 24 4D 4E 4C  00190 P.ABO:   .ASCII  \LNM$PROCESS_DIRECTORY\<0><0><0>
                     00 00 00 59 52 4F 54 43 45  0019F
                                        010E0015 001A8 P.ABN:   .LONG   17694741
                                        00000000' 001AC         .ADDRESS P.ABO
                  00 00 00 50 55 4F 52 47 24 4D 4E 4C  001B0 P.ABQ:   .ASCII  \LNM$GROUP\<0><0><0>
                                        010E0009 001BC P.ABP:   .LONG   17694729
                                        00000000' 001C0         .ADDRESS P.ABQ
                  00 4C 58 21 5F 42 4F 4A 24 4D 4E 4C  001C4 P.ABS:   .ASCII  \LNM$JOB_!XL\<0>
                                        010E000B 001D0 P.ABR:   .LONG   17694731
                                        00000000' 001D4         .ADDRESS P.ABS

                                                                .EXTRN  SYS$FAO

                                                                .PSECT  $CODE$,NOWRT,2

                                              0FFC 00000        .ENTRY  SET_LNM_TABLES, Save R2,R3,R4,R5,R6,R7,R8,- ; 2188
                                                                        R9,R10,R11
                                59 00000000G   00  9E 00002    MOVAB   SYS$FAO, R9
                                56     0000'   CF  9E 00009    MOVAB   P.ABK, R6
                                5E       B8    AE  9E 0000E    MOVAB   -72(SP), SP
```

```
                20   AE           10   D0 00012          MOVL    #16, TABLE_NAME_DESC                                    ; 2201
                24   AE       38  AE   9E 00016          MOVAB   GROUP_TABLE_NAME, TABLE_NAME_DESC+4                     ;
                50 00000000G  00   D0 0001B          MOVL    UAF_RECORD, R0                                          ; 2216
                7E        26  A0   3C 00022          MOVZWL  38(R0), -(SP)                                           ;
                          24  AE   9F 00026          PUSHAB  TABLE_NAME_DESC                                         ;
                          28  AE   9F 00029          PUSHAB  TABLE_NAME_DESC                                         ;
                              56   DD 0002C          PUSHL   R6                                                      ;
                69            04   FB 0002E          CALLS   #4, SYS$FAO                                             ;
                04   AE 00030004  8F   D0 00031          MOVL    #196612, ITEM_LIST                                      ; 2222
                08   AE       08  A6   9E 00039          MOVAB   P.ABM, ITEM_LIST+4                                      ; 2223
                          0C  AE   D4 0003E          CLRL    ITEM_LIST+8                                             ; 2224
10  AE          20   AE 00020000  8F   C9 00041          BISL3   #131072, TABLE_NAME_DESC, ITEM_LIST+12                  ; 2226
                14   AE       24  AE   D0 0004B          MOVL    TABLE_NAME_DESC+4, ITEM_LIST+16                         ; 2227
                          18  AE   7C 00050          CLRQ    ITEM_LIST+20                                            ; 2228
                          04  AE   9F 00053          PUSHAB  ITEM_LIST                                               ; 2235
                          04  AE   D4 00056          CLRL    4(SP)                                                   ;
                          04  AE   9F 00059          PUSHAB  4(SP)                                                   ;
                          38  A6   9F 0005C          PUSHAB  P.ABP                                                   ;
                          24  A6   9F 0005F          PUSHAB  P.ABN                                                   ;
                              7E   D4 00062          CLRL    -(SP)                                                   ;
          00000000G           00   05   FB 00064          CALLS   #5, SYS$CRELNM                                          ;
                37 00000000G  00   E8 0006B          BLBS    SUBPROCESS, 1$                                          ; 2246
                24   AE       28  AE   9E 00072          MOVAB   JOB_TABLE_NAME, TABLE_NAME_DESC+4                       ; 2255
                50 00000000G  00   D0 00077          MOVL    CTL$GL_PCB, R0                                          ; 2259
                        0080  C0   DD 0007E          PUSHL   128(R0)                                                 ;
                          24  AE   9F 00082          PUSHAB  TABLE_NAME_DESC                                         ;
                          28  AE   9F 00085          PUSHAB  TABLE_NAME_DESC                                         ;
                          4C  A6   9F 00088          PUSHAB  P.ABR                                                   ;
                69            04   FB 0008B          CALLS   #4, SYS$FAO                                             ;
                2B        42  AE   9E 0008E          MOVAB   GROUP_TABLE_NAME+10, R11                                ; 2267
                5A        30  AE   9E 00092          MOVAB   JOB_TABLE_NAME+8, R10                                   ; 2266
                50 00000000G  00   D0 00096          MOVL    UAF_RECORD, R0                                          ; 2265
                57      0238  C0   D0 0009D          MOVL    568(R0), R7                                             ;
                   00000000G  00   16 000A2          JSB     EXE$CRE_JGTABLE                                         ;
                              04   04 000A8          RET                                                             ; 2270
                50            01   D0 000A9 1$:      MOVL    #1, R0                                                  ;
                              04   04 000AC          RET                                                             ; 2271

; Routine Size:  173 bytes,    Routine Base:  $CODE$ + 0C21
```

```
; 1962          2272  1 END
; 1963          2273  0 ELUDOM


                                                .EXTRN  LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;   $PLIT$                       472  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                      3278  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                        564  NOVEC,  WRT,  RD ,NOEXE,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       159         0        1000        00:01.4
;   _$255$DUA28:[SHRLIB]NET.L32;1         1279        15         1          63        00:01.0



;                                       COMMAND QUALIFIERS

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

; Size:         3278 code + 1036 data bytes
; Run Time:       00:44.2
; Elapsed Time:   02:30.7
; Lines/CPU Min:   3088
; Lexemes/CPU-Min: 34633
; Memory Used:  285 pages
; Compilation Complete
```

LIBHOUR
LIS

LIBDAYWK
LIS

INITUSER
LIS

LOGIN
LIS

INTERACT
LIS