E 5

```
LLL                  OOOOOOOOO        GGGGGGGGGGGG      IIIIIIIII     NNN          NNN
LLL                  OOOOOOOOO        GGGGGGGGGGGG      IIIIIIIII     NNN          NNN
LLL                  OOOOOOOOO        GGGGGGGGGGGG      IIIIIIIII     NNN          NNN
LLL               OOO         OOO  GGG                     III        NNN          NNN
LLL               OOO         OOO  GGG                     III        NNN          NNN
LLL               OOO         OOO  GGG                     III        NNN          NNN
LLL               OOO         OOO  GGG                     III        NNNNNN       NNN
LLL               OOO         OOO  GGG                     III        NNNNNN       NNN
LLL               OOO         OOO  GGG                     III        NNNNNN       NNN
LLL               OOO         OOO  GGG                     III        NNN   NNN    NNN
LLL               OOO         OOO  GGG                     III        NNN   NNN    NNN
LLL               OOO         OOO  GGG                     III        NNN   NNN    NNN
LLL               OOO         OOO  GGG   GGGGGGGGG         III        NNN       NNNNNN
LLL               OOO         OOO  GGG   GGGGGGGGG         III        NNN       NNNNNN
LLL               OOO         OOO  GGG   GGGGGGGGG         III        NNN       NNNNNN
LLL               OOO         OOO  GGG         GGG         III        NNN          NNN
LLL               OOO         OOO  GGG         GGG         III        NNN          NNN
LLL               OOO         OOO  GGG         GGG         III        NNN          NNN
LLLLLLLLLLLLLLL      OOOOOOOOO        GGGGGGGGG         IIIIIIIII     NNN          NNN
LLLLLLLLLLLLLLL      OOOOOOOOO        GGGGGGGGG         IIIIIIIII     NNN          NNN
LLLLLLLLLLLLLLL      OOOOOOOOO        GGGGGGGGG         IIIIIIIII     NNN          NNN
```

```
**FILE**ID**INTERACT                                        N 6




  IIIIII    NN      NN  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR        AAAAAA      CCCCCCCC  TTTTTTTTTT
  IIIIII    NN      NN  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR        AAAAAA      CCCCCCCC  TTTTTTTTTT
    II      NN      NN      TT      EE          RR      RR  AA      AA  CC              TT
    II      NN      NN      TT      EE          RR      RR  AA      AA  CC              TT
    II      NNNN    NN      TT      EE          RR      RR  AA      AA  CC              TT
    II      NNNN    NN      TT      EE          RR      RR  AA      AA  CC              TT
    II      NN  NN  NN      TT      EEEEEEEE    RRRRRRRR    AA      AA  CC              TT
    II      NN  NN  NN      TT      EEEEEEEE    RRRRRRRR    AA      AA  CC              TT
    II      NN    NNNN      TT      EE          RR  RR      AAAAAAAAAA  CC              TT
    II      NN    NNNN      TT      EE          RR  RR      AAAAAAAAAA  CC              TT
    II      NN      NN      TT      EE          RR    RR    AA      AA  CC              TT          ....
    II      NN      NN      TT      EE          RR    RR    AA      AA  CC              TT          ....
  IIIIII    NN      NN      TT      EEEEEEEEEE  RR      RR  AA      AA    CCCCCCCC      TT          ....
  IIIIII    NN      NN      TT      EEEEEEEEEE  RR      RR  AA      AA    CCCCCCCC      TT          ....

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

INT
V04-

```
0 MODULE interact (IDENT = 'V04-000',
0         ADDRESSING_MODE(EXTERNAL = GENERAL)) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:  Login
1 !
1 ! ABSTRACT:
1 !
1 !       This module handles all processing of interactive jobs.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system.
1 !
1 ! AUTHOR:  Tim Halvorsen, March 1981
1 !
1 ! Modified by:
1 !
1 !       V03-025 ACG0436         Andrew C. Goldstein,    23-Jul-1984  16:32
1 !               Make DISUSER flag apply to all logins; move call to
1 !               UPDATE_UAF_RECORD on failure to after CIA scan has been
1 !               done; fix auditing of invalid username under breakin.
1 !               Make reconnection default to NONE if read times out.
1 !
1 !       V03-024 BLS0332         Benn Schreiber          16-JUL-1984
1 !               Correct punctuation.
1 !
1 !       V03-023 ACG0434         Andrew C. Goldstein,    9-Jul-1984  19:44
1 !               Use SYSGEN parameter to time out all LOGIN reads;
1 !               change reconnection default to true. Set terminal
1 !               to /NOBROADCAST during reading of system password.
1 !
```

```
1 !     V03-022 JRL0017         John R. Lawson, Jr.     6-Jul-1984 16:00
1 !             Move system password from SYS$GQ_PWD to a special record
1 !             in SYSUAF.DAT which AUTHORIZE will ignore.
1 !
1 !     V03-021 MHB0161         Mark Bramhall           28-Jun-1984
1 !             Fix the listing of disconnected processes.
1 !
1 !     V03-020 MHB0144         Mark Bramhall           2-May-1984
1 !             Changes for new account name conventions.
1 !             Limit failing password to NSA$S_PKT_PASSWORD.
1 !             Fix open/connect logic during auto-logins.
1 !
1 !     V03-019 MHB0122         Mark Bramhall           10-Apr-1984
1 !             Finish up automated reconnection code.
1 !             Security audit breakin attempts.
1 !             Security audit successful reconnections.
1 !             Remove forced prefixing of /CLI=xxx and /TABLE=xxx.
1 !             Set terminal name via SET_TERM_NAME.
1 !             Change password zeroing logic.
1 !             Add routine to return ASCIC day of week.
1 !             Call SET_ACCOUNT to clear account name initially.
1 !
1 !     V03-018 MHB0110         Mark Bramhall           21-Mar-1984
1 !             Use LNM services for logical names.
1 !             Clean up "last login" messages.
1 !             Add first cut at automated re-connection.
1 !
1 !     V03-017 PCG0001         Peter George            31-Jan-1984  14:17
1 !             Add secondary password prompting.
1 !             Display UAF information during interactive login.
1 !
1 !     V03-016 ACG0390         Andrew C. Goldstein,    18-Jan-1984  11:36
1 !             Remove unused username from breakin arg list
1 !
1 !     V03-015 ACG0385         Andrew C. Goldstein,    29-Dec-1983  10:07
1 !             Implement job type in JIB; move ALF definitions to LIB
1 !
1 !     V03-014 ACG0379         Andrew C. Goldstein,    6-Dec-1983  19:44
1 !             Make GET_UAFREC return a value to fix OPA0: logins
1 !
1 !     V03-013 ACG0376         Andrew C. Goldstein,    18-Nov-1983  19:31
1 !             Fix system password and autologin handling;
1 !             fix length checks on parsed command line.
1 !             Use GET_INPUT for all terminal reads.
1 !
1 !     V03-012 GAS0183         Gerry Smith     15-Sep-1983
1 !             Add breakin detection.
1 !
1 !     V03-011 GAS0169         Gerry Smith     23-Aug-1983
1 !             Fix comments to show that both SYS$INPUT and SYS$OUTPUT
1 !             are opened.  Also, fix the login retry logic so that
1 !             the welcome isn't displayed for every login retry.
1 !
1 !     V03-010 GAS0162         Gerry Smith     30-Jul-1983
1 !             Add support for the system password.
1 !
1 !     V03-009 GAS0164         Gerry Smith     30-Jul-1983
```

```
1 !                     Change the disable-logical-translation method
1 !                     in RMS calls to use the new LNM_MODE field.
1 !                     Make the "you have (n) new mail messages(s)"
1 !                     message nicer.
1 !
1 !             V03-008 GAS0146         Gerry Smith     23-Jun-1983
1 !                     Add support for the INTER bit, and checking that
1 !                     the interactive process is coming from a terminal.
1 !
1 !             V03-007 GAS0145         Gerry Smith     14-Jun-1983
1 !                     Add login retry logic for interactive processes.
1 !
1 !             V03-006 GAS0138         Gerry Smith     31-May-1983
1 !                     Add CLITABLES, the name of the CLI command table.
1 !
1 !             V03-005 GAS0088         Gerry Smith      4-Oct-1982
1 !                     If the input device is a remote terminal (denoted
1 !                     by having both MNT and TRM bits turned on) then
1 !                     clear the purge type-ahead in the RAB.
1 !                     If /DISK is specified, then make sure that the
1 !                     disk name ends with a ":".  If one isn't there,
1 !                     put one there.
1 !
1 !             V03-003 GAS0069         Gerry Smith      1-Apr-1982
1 !                     Add a password mask of overstriking characters
1 !                     if SYS$INPUT is a local echo terminal.  Also change
1 !                     code to allow for no uaf record on interactive login.
1 !
1 !             V03-001 GAS0059         Gerry Smith     17-Feb-1982
1 !                     Fix various auto-login problems.  Make sure that
1 !                     a username of <login> is used until it is determined
1 !                     that a valid UAF has been found.  Use only the system
1 !                     logical name table to translate during open/creates.
1 !
1 !             V03-014 GAS0043         Gerry Smith      5-Feb-1982
1 !                     Change $CHARCOUNT to %CHARCOUNT.
1 !
1 !             V03-013 GAS0041         Gerry Smith     03-Feb-1982
1 !                     Force a user-supplied CLI to be in SYS$SYSTEM.
1 !
1 !             V03-012 SPF0051         Steve Forgey    01-Jan-1981
1 !                     Set initial interactive username (CTL$T_USERNAME) tc
1 !                     <login> instead of JOBCTL.
1 !
1 !             V03-011 GAS0028         Gerry Smith     30-Dec-1981
1 !                     Zero the password field in the RMS area.  This is to
1 !                     prevent knowledgeable users from gaining access to it
1 !                     during login.
1 !
1 !             V03-010 HRJ0039         Herb Jacobs     19-Dec-1981
1 !                     Allow for accounts without a password for auto-login while
1 !                     not allowing these acoounts to be used interactively via
1 !                     the authorization flag DISACNT.  Add flags in authorization
1 !                     record to allow suppression on new mail and welcome messages.
1 !
1 !             V03-009 HRJ0037         Herb Jacobs     10-Dec-1981
1 !                     Accept passwords in auto login, and handle new device name
```

```
!                       syntax in SYSALF.DAT.
!
!       V008    PCG0001         Peter George    03-Dec-1981
!               Call CLI$END_PARSE after parsing login commmand.
!
!       V03-007 HRJ0032         Herb Jacobs     13-Nov-1981
!               Accept null passwords by using new validate_pass entry.
!
!       V006    TMH0006         Tim Halvorsen   22-Oct-1981
!               Add missing support for CAPTIVE UAF flag.
!
!       V005    SPF0030         Steve Forgey    15-Sep-1981
!               Set terminal name in PCB before validating username and
!               password.
!
!       V004    TMH0004         Tim Halvorsen   17-Jul-1981
!               Change the wording on the new mail message.
!
!       V003    TMH0003         Tim Halvorsen   16-Jul-1981
!               Display the actual number of new mail messages
!               when informing the user that new mail has arrived.
!
!       V002    TMH0002         Tim Halvorsen   17-Jun-1981
!               Clear purge-typeahead before password prompt.
!
!       V001    TMH0001         Tim Halvorsen   15-May-1981
!               Output blank line if no user announcement (SYS$ANNOUNCE)
!               message.
!--

!
! Include files
!

LIBRARY 'SYS$LIBRARY:LIB';                      ! VAX/VMS system definitions
REQUIRE 'SHRLIB$:UTILDEF';                      ! Common BLISS definitions
REQUIRE 'LIB$:PPDDEF';                          ! Process permanent data region
REQUIRE 'LIB$:LGIDEF';                          ! LOGINOUT private permanent storage
```

```
1 !
1 ! Table of contents
1 !
1
1 FORWARD ROUTINE
1     init_interactive:    NOVALUE,        ! Initialize interactive job
1     auto_login,                          ! Check if automatic login enabled
1     interactive_validation,              ! Perform interactive validation
1     get_password,                        ! Acquire password from terminal
1     write_announcement,                  ! Write user-supplied announcement msg
1     announce:            NOVALUE,        ! Announce successful login
1     zero_password:       NOVALUE,        ! Zero password in RMS buffer
1     get_syspwd:          NOVALUE,        ! Get system password
1     check_connection:    NOVALUE,        ! Check for (re-)connection(s)
1     ascic_day_of_week;                   ! Return ASCIC day of week
1
1 !
1 ! External routines
1 !
1
1 EXTERNAL ROUTINE
1     open_input:          NOVALUE,        ! Open primary input file
1     open_output:         NOVALUE,        ! Open primary output file
1     get_uafrec,                          ! Read UAF record without validation
1     update_uaf_record:   NOVALUE,        ! Update the login failure count
1     write_file:          NOVALUE,        ! Write file to primary output
1     write_output,                        ! Write to primary output stream
1     write_timeout:       NOVALUE,        ! Cancel read and exit
1     write_fao:           NOVALUE,        ! Write formatted message to output
1     get_input:           NOVALUE,        ! Get record from primary input stream
1     set_uic,                             ! Set process UIC
1     set_username:        NOVALUE,        ! Set username in JIB and P1 space
1     set_term_name:       NOVALUE,        ! Set terminal name in PCB
1     set_sysprv:          NOVALUE,        ! Set SYSPRV privilege
1     clear_sysprv:        NOVALUE,        ! Clear SYSPRV privilege
1     exit_process:        NOVALUE,        ! Exit process
1     lib$day_of_week,                     ! Find day of week from 64-bit time
1     mail$get_new_count,                  ! Get user's mail count
1     lgi$hpwd,                            ! Password masher
1     lgi$check_pass,                      ! Validate password against UAF record
1     lgi$searchuser,                      ! Retrieve a user record
1     security_audit:      NOVALUE,        ! Perform a security audit
1     cia_scan,                            ! Check for suspect/intruder
1     cli$dcl_parse,                       ! Parse DCL command
1     cli$present,                         ! Check if entity present
1     cli$get_value,                       ! Get value from command line
1     cli$end_parse;                       ! Clean up after parsing
1
1 !
1 ! External storage
1 !
1
1 EXTERNAL
1     phy_term_name:       VECTOR,         ! descriptor of physical terminal name
1     terminal_device:     BYTE,           ! True if SYS$INPUT is a terminal
1     input_chan,                          ! Channel assigned to SYS$INPUT
1     dev_dep_2:           $BBLOCK,        ! Particular characteristics
```

```
     dev_char_2:          $BBLOCK,            !  of SYS$INPUT
     job_type,                                ! Job type for JIB
     uaf_record:          REF BBLOCK,         ! Address of UAF record
     uaf_rab:             BBLOCK,             ! RAB for UAF
     uaf_fab:             BBLOCK,             ! FAB for UAF
     sys$input:           VECTOR,             ! Translation of SYS$INPUT
     sys$output:          VECTOR,             ! Translation of SYS$OUTPUT
     fail_password:       VECTOR,             ! Descriptor of failing password
     term_name:           VECTOR,             ! Descriptor of terminal name
     clu_term_name:       VECTOR,             ! Descriptor of cluster terminal name
     input_fab:           BBLOCK,             ! Input FAB
     input_rab:           BBLOCK,             ! Input RAB
     output_fab:          BBLOCK,             ! Output FAB
     ctl$ag_clidata,                          ! Process permanent data region
     sys$gb_pwd_tmo:      BYTE,               ! System password timeout limit
     sys$gb_retry_lim:    BYTE,               ! Number of retries allowed
     sys$gb_retry_tmo:    BYTE;               ! Number of seconds to wait for retries

BIND
     ppd = ctl$ag_clidata: BBLOCK;            ! Address of PPD structure

!
! Literals
!
LITERAL
     bell = 7,                                ! Ring bell
     bs = 8,                                  ! Backspace
     cr = 13,                                 ! Carriage return
     lf = 10;                                 ! Line feed

EXTERNAL LITERAL
     cli$_defaulted,                          ! Qualifier was defaulted
     lgi$_connerr,                            ! Connection failed
     lgi$_disreconnect,                       ! /CONNECT not legal
     lgi$_evade,                              ! evasion in progress
     lgi$_syspwdtmo,                          ! invalid user at terminal
     lgi$_captive,                            ! illegal options on captive account
     lgi$_defcli,                             ! /CLI and /TABLES not legal
     lgi$_notvalid,                           ! invalid user authorization
     lgi$_userauth;                           ! invalid user authorization record

!
! OWN storage
!
OWN
     user_buff : VECTOR[uaf$s_username,BYTE],
     username : VECTOR[2] INITIAL(0, user_buff),
     connect_name_buffer : VECTOR[40,BYTE],
     connect_name : VECTOR[2]                 ! Descriptor of connection device
               INITIAL(0, connect_name_buffer),
     connect_check : INITIAL(0);              ! True if /CONNECT

GLOBAL
     cli_name_buffer: VECTOR [80,BYTE],
     table_name_buffer: VECTOR[80,BYTE],
     disk_name_buffer: VECTOR [40,BYTE],
     com_name_buffer: VECTOR [132,BYTE],
      1      cli_name:     VECTOR [2]                     ! Descriptor of user CLI name
                           INITIAL(0,cli_name_buffer),
      1      table_name: VECTOR[2]                        ! Descriptor of CLI command table
                           INITIAL(0,table_name_buffer),
      1      disk_name:    VECTOR [2]                     ! Descriptor of user disk name
                           INITIAL(0,disk_name_buffer),
      1      com_name:     VECTOR [2]                     ! Descriptor of user login proc
                           INITIAL(0,com_name_buffer),
      1      com_negated: BYTE INITIAL(false);            ! True if /NOCOMMAND
```

```
GLOBAL ROUTINE init_interactive: NOVALUE =

!---
!
!       Initialize an interactive job by requesting the usename
!       and password from the terminal associated with the process.
!
! Inputs:
!
!       None
!
! Outputs:
!
!       uaf_record = Address of UAF record for user
!                       (may be zero if no UAF record read, but login ok)
!---

BEGIN

LOCAL
    status,
    arglist : VECTOR[2]
              INITIAL(1, 0),
    retry_count : BYTE INITIAL(0),      ! Number of retries
    buffer;                             ! Buffer for dummy read

!
! Set initial username of <login>
!
$CMKRNL(ROUTIN = set_username,          ! Set initial username of process
        ARGLST = $DESCRIPTOR('<login>'));

!
! Open the output and input files.
!
$CMEXEC(ROUTIN = open_output);          ! Open output file
$CMEXEC(ROUTIN = open_input);           ! Open input file

!
! Now check the input device to see if it is a terminal.  If not,
! then tell the user to buzz off.  This is so that no matter what
! a user does, it is not possible to ask for a password from a file.
! This helps to discourage people from putting their passwords on a
! disk somewhere.
!
IF .$BBLOCK [input_fab[fab$l_dev], dev$v_fod]
THEN SIGNAL_STOP(lgi$_userauth);

!
! Set the job type according to the characteristics of SYS$INPUT.
! Set no initial typeahead purge for remote terminals.
! Set terminal name in PCB.
!
IF .terminal_device
THEN
    BEGIN
    IF .dev_char_2[dev$v_rtt]           ! If remote terminal,
3       THEN
4           BEGIN
4           job_type = jib$c_remote;           ! Set job type to remote
4           input_rab[rab$v_pta] = 0;          ! Set no initial typeahead purge
4           END
3       ELSE
4           BEGIN
4           IF .dev_dep_2[tt2$v_dialup]        ! Else if dialup terminal,
4           THEN job_type = jib$c_dialup       ! Set job type to dialup
4           ELSE job_type = jib$c_local;       ! Else set job type to local
3           END;
3       set_term_name();                       ! Set terminal name in PCB
2       END;
2
2   !
2   ! Process the system password if there is one.
2   !
2   get_syspwd ();
2
2   !
2   ! Write the system announcement if it exists; else write a blank line.
2   !
2   IF NOT write_announcement (%ASCID 'SYS$ANNOUNCE')
2   THEN write_output (UPLIT (0, 0));
2
2   WHILE true DO
3       BEGIN
3
3   !
3   ! Reset all status information regarding login qualifiers each loop...
3   !
3       cli_name[0] = 0;
3       cli_name[1] = cli_name_buffer;
3       table_name[0] = 0;
3       table_name[1] = table_name_buffer;
3       disk_name[0] = 0;
3       disk_name[1] = disk_name_buffer;
3       com_negated = false;
3       com_name[0] = 0;
3       com_name[1] = com_name_buffer;
3       connect_check = 0;
3       connect_name[0] = 0;
3       connect_name[1] = connect_name_buffer;
3
3   !
3   ! If interactive process, and no automatic login is requested for this
3   ! terminal, then prompt for username & password and read UAF record.
3   !
3       status = auto_login ();                         ! See if autologin
3       IF NOT .status
3       AND .status NEQ lgi$_userauth
3       AND .status NEQ lgi$_notvalid                   ! If not autologin,
3       THEN status = interactive_validation ();        ! try regular interactive.
3
3   !
3   ! Check the DISUSER flag here so that we stay in the retry loop if it's
3   ! set. This preserves the consistency of "invalid user" behavior for
```

```
  3 ! the DISUSER flag. Also, this way attempts on disabled accounts are
  3 ! detected by breakin detection.
  3 !
  3
  3     IF .status
  3     AND .uaf_record NEQ 0
  3     THEN
  4         BEGIN
  4         IF .uaf_record[uaf$v_disacnt]
  5         AND (CH$NEQ (6, UPLIT BYTE ('_OPA0:'), .phy_term_name[0], .phy_term_name[1])
  5            OR CH$NEQ (6, UPLIT BYTE ('SYSTEM'), uaf$s_username, uaf_record[uaf$t_username], ' '))
  4         THEN status = lgi$_notvalid;
  3         END;
  3
  3 !
  3 ! Now run the login attempt against breakin detection. We inform
  3 ! the CIA scan whether the login is successful so far; it informs
  3 ! us if the subject is an intruder or not.
  3 !
  3
  3     IF NOT .status
  3     THEN
  4         BEGIN
  4         arglist[1] = 0;                          ! Want a suspect scan
P 4         IF NOT $CMKRNL (ROUTIN = cia_scan,       ! If evasion is in effect
  5                         ARGLST = arglist)
  4         THEN
  5             BEGIN
P 5             $CMKRNL (ROUTIN = set_username,      ! Set up username unconditionally
  5                     ARGLST = username);
  5             security_audit(nsa$k_rectyp_logb);   !  then audit the breakin
P 5             $CMKRNL (ROUTIN = set_username,      ! Reset process name
  5                     ARGLST = $DESCRIPTOR('<login>'));
  4             END;
  4         IF .status EQL lgi$_notvalid             ! If this was a password failure
  4         AND .uaf_record NEQ 0
  4         THEN update_uaf_record();                ! Update the login failure count
  4         END
  3     ELSE
  4         BEGIN
  4         arglist[1] = 1;                          ! Look for an intruder
P 4         IF NOT $CMKRNL (ROUTIN = cia_scan,
  5                         ARGLST = arglist)
  4         THEN
  5             BEGIN
  5             status = lgi$_notvalid;
  5             ppd[ppd$l_lststatus] = lgi$_evade;
  4             END;
  3         END;
  3
  3     IF .status                                   ! If all done,
  3     THEN RETURN;                                 ! then go away.
  3
  3 !
  3 ! If the login attempt did not succeed, check to see if the retry count
  3 ! has been exceeded.  If not, then change the severity to informational,
  3 ! so that we continue after signalling.  Otherwise, when the signal of
```

```
      3 ! the severe error occurs, the process will be terminated.
      3 !
      3     retry_count = .retry_count + 1;              ! Update retry count
      3     IF .retry_count LSSU .sys$gb_retry_lim        ! If we can retry again,
      4     THEN status = (.status AND NOT sts$m_severity) ! then make the error
      3             OR sts$k_error;                      ! an informational
      3     SIGNAL (.status);                            ! and signal it.
    P 3     $CMKRNL (ROUTIN = set_username,              ! Reset process name
      3             ARGLST = $DESCRIPTOR('<login>'));
      3
      3 !
      3 ! If control returns to here after signalling, then hang a read out on
      3 ! the terminal, with a timeout.
      3 !
      3     input_rab [rab$v_pmt] = 0;                   ! No prompt
      3     input_rab[rab$v_rne] = 1;                    ! and don't echo anything
      3     input_rab[rab$w_usz] = 4;
      3     input_rab[rab$l_ubf] = buffer;
      3     get_input (input_rab, 1);                    ! Wait for next input
      3
      2     END;                                         ! End of WHILE TRUE loop
      2
      1 END;


                                                       .TITLE  INTERACT
                                                       .IDENT  \V04-000\

                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

                3E 6E 69 67 6F 6C 3C  00000 P.AAB:    .ASCII  \<login>\                       ;
                                      00007           .BLKB   1
                           00000007   00008 P.AAA:    .LONG   7                               ;
                           00000000'  0000C           .ADDRESS P.AAB                          ;
   45 43 4E 55 4F 4E 4E 41 24 53 59 53  00010 P.AAD:    .ASCII  \SYS$ANNOUNCE\                  ;
                           010E000C   0001C P.AAC:    .LONG   17694732                        ;
                           00000000'  00020           .ADDRESS P.AAD                          ;
                  00000000 00000000   00024 P.AAE:    .LONG   0, 0                            ;
                3A 30 41 50 4F 5F     0002C P.AAF:    .ASCII  \_OPA0:\                        ;
                4D 45 54 53 59 53     00032 P.AAG:    .ASCII  \SYSTEM\                        ;
                3E 6E 69 67 6F 6C 3C  00038 P.AAI:    .ASCII  \<login>\                       ;
                                      0003F           .BLKB   1
                           00000007   00040 P.AAH:    .LONG   7                               ;
                           00000000'  00044           .ADDRESS P.AAI                          ;
                3E 6E 69 67 6F 6C 3C  00048 P.AAK:    .ASCII  \<login>\                       ;
                                      0004F           .BLKB   1
                           00000007   00050 P.AAJ:    .LONG   7                               ;
                           00000000'  00054           .ADDRESS P.AAK                          ;

                                                       .PSECT  $OWN$,NOEXE,2

                                      00000 USER_BUFF:
                                                       .BLKB   32
                           00000000   00020 USERNAME:
                                                       .LONG   0
                           00000000'  00024           .ADDRESS USER_BUFF                      ;
                                      00028 CONNECT_NAME_BUFFER:
```

```
                                 .BLKB    40
00000000  00050 CONNECT_NAME:
                                 .LONG    0
00000000' 00054                  .ADDRESS CONNECT_NAME_BUFFER                                    ;
00000000  00058 CONNECT_CHECK:
                                 .LONG    0                                                      ;

                                 .PSECT   $GLOBAL$,NOEXE,2

          00000 CLI_NAME_BUFFER::
                                 .BLKB    80
          00050 TABLE_NAME_BUFFER::
                                 .BLKB    80
          000A0 DISK_NAME_BUFFER::
                                 .BLKB    40
          000C8 COM_NAME_BUFFER::
                                 .BLKB    132
00000000  0014C CLI_NAME::
                                 .LONG    0
00000000' 00150                  .ADDRESS CLI_NAME_BUFFER                                        ;
00000000  00154 TABLE_NAME::
                                 .LONG    0                                                      ;
00000000' 00158                  .ADDRESS TABLE_NAME_BUFFER                                      ;
00000000  0015C DISK_NAME::
                                 .LONG    0                                                      ;
00000000' 00160                  .ADDRESS DISK_NAME_BUFFER                                       ;
00000000  00164 COM_NAME::
                                 .LONG    0                                                      ;
00000000' 00168                  .ADDRESS COM_NAME_BUFFER                                        ;
      00  0016C COM_NEGATED::
                                 .BYTE    0                                                      ;

                                 .EXTRN   OPEN_INPUT, OPEN_OUTPUT
                                 .EXTRN   GET_UAFREC, UPDATE_UAF_RECORD
                                 .EXTRN   WRITE_FILE, WRITE_OUTPUT
                                 .EXTRN   WRITE_TIMEOUT, WRITE_FAO
                                 .EXTRN   GET_INPUT, SET_UIC
                                 .EXTRN   SET_USERNAME, SET_TERM_NAME
                                 .EXTRN   SET_SYSPRV, CLEAR_SYSPRV
                                 .EXTRN   EXIT_PROCESS, LIB$DAY_OF_WEEK
                                 .EXTRN   MAIL$GET_NEW_COUNT
                                 .EXTRN   LGI$HPWD, LGI$CHECK_PASS
                                 .EXTRN   LGI$SEARCHUSER, SECURITY_AUDIT
                                 .EXTRN   CIA_SCAN, CLI$DCL_PARSE
                                 .EXTRN   CLI$PRESENT, CLI$GET_VALUE
                                 .EXTRN   CLI$END_PARSE, PHY_TERM_NAME
                                 .EXTRN   TERMINAL_DEVICE
                                 .EXTRN   INPUT_CHAN, DEV_DEP_2
                                 .EXTRN   DEV_CHAR_2, JOB_TYPE
                                 .EXTRN   UAF_RECORD, UAF_RAB
                                 .EXTRN   UAF_FAB, SYS$INPUT
                                 .EXTRN   SYS$OUTPUT, FAIL_PASSWORD
                                 .EXTRN   TERM_NAME, CLU_TERM_NAME
                                 .EXTRN   INPUT_FAB, INPUT_RAB
                                 .EXTRN   OUTPUT_FAB, CTL$AG_CLIDATA
                                 .EXTRN   SYS$GB_PWD_TMO, SYS$GB_RETRY_LIM
                                 .EXTRN   SYS$GB_RETRY_TMO
```

```
                                          .EXTRN  CLI$_DEFAULTED, LGI$_CONNERR
                                          .EXTRN  LGI$_DISRECONNECT
                                          .EXTRN  LGI$_EVADE, LGI$_SYSPWDTMO
                                          .EXTRN  LGI$_CAPTIVE, LGI$_DEFCLI
                                          .EXTRN  LGI$_NOTVALID, LGI$_USERAUTH
                                          .EXTRN  SYS$CMKRNL, SYS$CMEXEC

                                          .PSECT  $CODE$,NOWRT,2

                           0FFC 00000     .ENTRY  INIT_INTERACTIVE, Save R2,R3,R4,R5,R6,R7,-   ; 0733
                                                  R8,R9,R10,R11                                 ;
             5B 00000000G 8F D0 00002     MOVL    #LGI$_NOTVALID, R11                           ;
             5A 00000000G 00 9E 00009     MOVAB   SET_USERNAME, R10                             ;
             59 00000000G 00 9E 00010     MOVAB   INPUT_RAB+4, R9                               ;
             58 00000000G 00 9E 00017     MOVAB   SYS$CMKRNL, R8                                ;
             57     0000' CF 9E 0001E     MOVAB   CLI_NAME, R7                                  ;
             5E           0C C2 00023     SUBL2   #12, SP                                       ;
          04 AE           01 7D 00026     MOVQ    #1, ARGLIST                                   ; 0750
                          56 94 0002A     CLRB    RETRY_COUNT                                   ;
                    0000' CF 9F 0002C     PUSHAB  P.AAA                                         ; 0763
                          5A DD 00030     PUSHL   R10                                           ;
             68           02 FB 00032     CALLS   #2, SYS$CMKRNL                                ;
                          7E D4 00035     CLRL    -(SP)                                         ; 0768
                00000000G 00 9F 00037     PUSHAB  OPEN_OUTPUT                                   ;
   00000000G 00           02 FB 0003D     CALLS   #2, SYS$CMEXEC                                ;
                          7E D4 00044     CLRL    -(SP)                                         ; 0769
                00000000G 00 9F 00046     PUSHAB  OPEN_INPUT                                    ;
   00000000G 00           02 FB 0004C     CALLS   #2, SYS$CMEXEC                                ;
0D 00000000G 00           06 E1 00053     BBC     #6, INPUT_FAB+65, 1$                          ; 0778
                00000000G 8F DD 0005B     PUSHL   #LGI$_USERAUTH                                ; 0779
   00000000G 00           01 FB 00061     CALLS   #1, LIB$STOP                                  ;
             34 00000000G 00 E9 00068 1$: BLBC    TERMINAL_DEVICE, 5$                           ; 0786
0D 00000000G 00           02 E1 0006F     BBC     #2, DEV_CHAR_2, 2$                            ; 0789
   00000000G 00           05 D0 00077     MOVL    #5, JOB_TYPE                                  ; 0792
          03 A9           20 8A 0007E     BICB2   #32, INPUT_RAB+7                              ; 0793
                          18 11 00082     BRB     4$                                            ; 0789
                00000000G 00 95 00084 2$: TSTB    DEV_DEP_2+1                                   ; 0797
                          09 18 0008A     BGEQ    3$                                            ;
   00000000G 00           04 D0 0008C     MOVL    #4, JOB_TYPE                                  ; 0798
                          07 11 00093     BRB     4$                                            ;
   00000000G 00           03 D0 00095 3$: MOVL    #3, JOB_TYPE                                  ; 0799
   00000000G 00           00 FB 0009C 4$: CALLS   #0, SET_TERM_NAME                             ; 0801
       0000V CF           00 FB 000A3 5$: CALLS   #0, GET_SYSPWD                                ; 0807
                    0000' CF 9F 000A8     PUSHAB  P.AAC                                         ; 0812
       0000V CF           01 FB 000AC     CALLS   #1, WRITE_ANNOUNCEMENT                        ;
                       0B 50 E8 000B1     BLBS    R0, 6$                                        ;
                    0000' CF 9F 000B4     PUSHAB  P.AAE                                         ; 0813
   00000000G 00           01 FB 000B8     CALLS   #1, WRITE_OUTPUT                              ;
                          67 D4 000BF 6$: CLRL    CLI_NAME                                      ; 0821
          04 A7      FEB4 C7 9E 000C1     MOVAB   CLI_NAME_BUFFER, CLI_NAME+4                   ; 0822
                       08 A7 D4 000C7     CLRL    TABLE_NAME                                    ; 0823
          0C A7      FF04 C7 9E 000CA     MOVAB   TABLE_NAME_BUFFER, TABLE_NAME+4               ; 0824
                       10 A7 D4 000D0     CLRL    DISK_NAME                                     ; 0825
          14 A7      FF54 C7 9E 000D3     MOVAB   DISK_NAME_BUFFER, DISK_NAME+4                 ; 0826
                       20 A7 94 000D9     CLRB    COM_NEGATED                                   ; 0827
                       18 A7 D4 000DC     CLRL    COM_NAME                                      ; 0828
          1C A7      FF7C C7 9E 000DF     MOVAB   COM_NAME_BUFFER, COM_NAME+4                   ; 0829
```

```
INTERACT                                                 B  8
V04-000                                           16-Sep-1984 01:55:50   VAX-11 Bliss-32 V4.0-742              Page  14
                                                  14-Sep-1984 12:41:07   [LOGIN.SRC]INTERACT.B32;1                  (3)

                                                  0000'  CF D4 000E5        CLRL    CONNECT_CHECK                                  ; 0830
                                                  0000'  CF D4 000E9        CLRL    CONNECT_NAME                                   ; 0831
                                    0000'  CF     0000'  CF 9E 000ED        MOVAB   CONNECT_NAME_BUFFER, CONNECT_NAME+4            ; 0832
                                    0000V  CF            00 FB 000F4        CALLS   #0, AUTO_LOGIN                                 ; 0838
                                           55            50 D0 000F9        MOVL    R0, STATUS                                     ;
                                           19            55 E8 000FC        BLBS    STATUS, 8$                                     ; 0839
                                00000000G  8F            55 D1 000FF        CMPL    STATUS, #LGI$_USERAUTH                         ; 0840
                                                         0D 13 00106        BEQL    7$                                             ;
                                           5B            55 D1 00108        CMPL    STATUS, R11                                    ; 0841
                                                         08 13 0010B        BEQL    7$                                             ;
                                    0000V  CF            00 FB 0010D        CALLS   #0, INTERACTIVE_VALIDATION                     ; 0842
                                           55            50 D0 00112        MOVL    R0, STATUS                                     ;
                                           3B            55 E9 00115 7$:    BLBC    STATUS, 11$                                    ; 0851
                                              00000000G  00 D5 00118 8$:    TSTL    UAF_RECORD                                     ; 0852
                                                         30 13 0011E        BEQL    10$                                            ;
                                           54 00000000G  00 D0 00120        MOVL    UAF_RECORD, R4                                 ; 0855
                          23        01D4   C4            04 E1 00127        BBC     #4, 468(R4), 10$                               ;
                                           50 00000000G  00 D0 0012D        MOVL    PHY_TERM_NAME+4, R0                            ; 0856
00000000G  00             00        0000'  CF            06 2D 00134        CMPC5   #6, P.AAF, #0, PHY_TERM_NAME, (R0)             ;
                                                         60    0013F                                                               ;
                                                         0B 12 00140        BNEQ    9$                                             ;
           20             20        0000'  CF            06 2D 00142        CMPC5   #6, P.AAG, #32, #32, 4(R4)                     ; 0857
                                                     04  A4    00149                                                               ;
                                                         03 13 0014B        BEQL    10$                                            ;
                                           55            5B D0 0014D 9$:    MOVL    R11, STATUS                                    ; 0858
                                           43            55 E8 00150 10$:   BLBS    STATUS, 13$                                    ; 0867
                                                     08  AE D4 00153 11$:   CLRL    ARGLIST+4                                      ; 0870
                                                     04  AE 9F 00156        PUSHAB  ARGLIST                                        ; 0872
                                              00000000G  00 9F 00159        PUSHAB  CIA_SCAN                                       ;
                                           68            02 FB 0015F        CALLS   #2, SYS$CMKRNL                                 ;
                                           1B            50 E8 00162        BLBS    R0, 12$                                        ;
                                                  0000'  CF 9F 00165        PUSHAB  USERNAME                                       ; 0876
                                                         5A DD 00169        PUSHL   R10                                            ;
                                           68            02 FB 0016B        CALLS   #2, SYS$CMKRNL                                 ;
                                                         04 DD 0016E        PUSHL   #4                                             ; 0877
                                00000000G  00            01 FB 00170        CALLS   #1, SECURITY_AUDIT                             ;
                                                  0000'  CF 9F 00177        PUSHAB  P.AAH                                          ; 0879
                                                         5A DD 0017B        PUSHL   R10                                            ;
                                           68            02 FB 0017D        CALLS   #2, SYS$CMKRNL                                 ;
                                           5B            55 D1 00180 12$:   CMPL    STATUS, R11                                    ; 0881
                                                         32 12 00183        BNEQ    14$                                            ;
                                              00000000G  00 D5 00185        TSTL    UAF_RECORD                                     ; 0882
                                                         2A 13 0018B        BEQL    14$                                            ;
                                00000000G  00            00 FB 0018D        CALLS   #0, UPDATE_UAF_RECORD                          ; 0883
                                                         21 11 00194        BRB     14$                                            ; 0867
                                       08  AE            01 D0 00196 13$:   MOVL    #1, ARGLIST+4                                  ; 0887
                                                     04  AE 9F 0019A        PUSHAB  ARGLIST                                        ; 0889
                                              00000000G  00 9F 0019D        PUSHAB  CIA_SCAN                                       ;
                                           68            02 FB 001A3        CALLS   #2, SYS$CMKRNL                                 ;
                                           0E            50 E8 001A6        BLBS    R0, 14$                                        ;
                                           55            5B D0 001A9        MOVL    R11, STATUS                                    ; 0892
                                00000000G  00 00000000G  8F D0 001AC        MOVL    #LGI$_EVADE, PPD+24                            ; 0893
                                           45            55 E8 001B7 14$:   BLBS    STATUS, 16$                                    ; 0897
                                                         56 96 001BA        INCB    RETRY_COUNT                                    ; 0906
                                00000000G  00            56 91 001BC        CMPB    RETRY_COUNT, SYS$GB_RETRY_LIM                  ; 0907
                                                         08 1E 001C3        BGEQU   15$                                            ;
                          50               55            07 CB 001C5        BICL3   #7, STATUS, R0                                 ; 0908
```

```
                 55          50                    02  C9 001C9          BISL3    #2, R0, STATUS                          ; 0909
                                                   55  DD 001CD  15$:    PUSHL    STATUS                                  ; 0910
            00000000G  00                          01  FB 001CF          CALLS    #1, LIB$SIGNAL                          ;
                                       0000'       CF  9F 001D6          PUSHAB   P.AAJ                                   ; 0912
                                                   5A  DD 001DA          PUSHL    R10                                     ;
                             68                    02  FB 001DC          CALLS    #2, SYS$CMKRNL                          ;
                      03     A9           40       8F  8A 001DF          BICB2    #64, INPUT_RAB+7                        ; 0918
                      03     A9                    01  88 001E4          BISB2    #1, INPUT_RAB+7                         ; 0919
                      1C     A9                    04  B0 001E8          MOVW     #4, INPUT_RAB+32                        ; 0920
                      20     A9                    6E  9E 001EC          MOVAB    BUFFER, INPUT_RAB+36                    ; 0921
                                                   01  DD 001F0          PUSHL    #1                                      ; 0922
                                          FC       A9  9F 001F2          PUSHAB   INPUT_RAB                               ;
            00000000G  00                          02  FB 001F5          CALLS    #2, GET_INPUT                           ;
                                              FEC0 31 001FC          BRW      6$                                      ; 0815
                                                   04 001FF  16$:    RET                                              ; 0926

; Routine Size:  512 bytes,    Routine Base:  $CODE$ + 0000
```

```
  1 ROUTINE auto_login =
  1
  1 !---
  1 !
  1 !       Check if any automatic login has been specified for the current
  1 !       terminal in SYSALF.DAT.  If so, obtain the UAF record without
  1 !       prompting for username. Password is still checked if present.
  1 !
  1 ! Inputs:
  1 !
  1 !       sysalf_fab/rab = FAB/RAB for SYSALF file
  1 !       input_rab = RAB for terminal stream
  1 !
  1 ! Outputs:
  1 !
  1 !       routine = True if automatic login enabled, else false
  1 !
  1 !       If automatic login enabled,
  1 !
  1 !       uaf_record = Address of user's UAF record, if automatic login enabled.
  1 !       The typeahead buffer is cleared.
  1 !---
  1
  2 BEGIN
  2
  2 LOCAL
  2     status,
  2     status1,
  2     sysalf_fab: BBLOCK [fab$c_bln],         ! FAB for auto-login file
  2     sysalf_rab: BBLOCK [rab$c_bln],         ! RAB for auto-login file
  2     buffer:     BBLOCK [alf$c_length],      ! SYSALF record buffer
  2     input_buffer: VECTOR [128,BYTE];        ! Input buffer
  2
P 2 $FAB_INIT(FAB = sysalf_fab,
P 2         FNM = 'SYSALF',                     ! Primary filespec
P 2         DNM = 'SYS$SYSTEM:.DAT',            ! Default filespec
P 2         SHR = (GET,PUT,DEL,UPD),            ! Set sharing options
  2         ORG = IDX);                         ! ISAM file
  2 !
  2 ! Disable group and process logical name translation for the open.  This
  2 ! must be done manually, since $FAB_INIT doesn't know about this.
  2 !
  2 sysalf_fab[fab$v_lnm_mode] = psl$c_exec;
  2
P 2 $RAB_INIT(RAB = sysalf_rab,
P 2         FAB = sysalf_fab,                   ! Address of associated FAB
P 2         RAC = KEY,                          ! Keyed record access
P 2         KRF = 0,                            ! Reference by key #0
P 2         USZ = alf$c_length,                 ! Size of entire record
P 2         UBF = buffer,                       ! Address of record buffer
P 2         KSZ = alf$s_devname,                ! Size of key field
  2         KBF = buffer [alf$t_devname]);      ! Address of key field
  2
  2 set_sysprv();                               ! Enable SYSPRV so we can access file
  2
  3 IF NOT $OPEN(FAB = sysalf_fab)              ! Open SYSALF file, if possible
  2 THEN
```

```
3       BEGIN
3       clear_sysprv();                         ! Drop SYSPRV on exit
3       RETURN false;
2       END;
2
3   IF NOT (status = $CONNECT(RAB = sysalf_rab)) ! Connect to stream
2   THEN
3       BEGIN
3       IF .status EQL rms$_crmp                ! If global buffers error,
3       THEN
4           BEGIN
4           sysalf_fab [fab$w_gbc] = 0;         ! Turn off global buffers
4           status = $CONNECT(RAB = sysalf_rab); ! Retry connect to stream
3           END;
3       IF NOT .status
3       THEN
4           BEGIN
4           $CLOSE(FAB = sysalf_fab);           ! If error, close file
4           clear_sysprv();                     ! Drop SYSPRV on exit
4           RETURN false;                       ! and return unsuccessful
3           END;
2       END;
2
2                                               ! Copy terminal name to key field
2   CH$COPY(.clu_term_name[0], .clu_term_name[1],
2           ' ', .sysalf_rab [rab$b_ksz], .sysalf_rab [rab$l_kbf]);
2
2   status = $GET(RAB = sysalf_rab);            ! Read record keyed by terminal name
2
2   $CLOSE(FAB = sysalf_fab);                   ! Close file (error or not)
2
2   clear_sysprv();                             ! Drop SYSPRV on exit
2
2   IF NOT .status                              ! If no record found in file,
2   THEN
2       RETURN .status;                         ! then return unsuccessful
2
2   IF .input_rab [rab$v_pta]                   ! If typeahead purge still do be done,
2   THEN
3       BEGIN
3       input_rab [rab$w_usz] = 128;
3       input_rab [rab$l_ubf] = input_buffer;
3       input_rab [rab$b_tmo] = 0;              ! Purge the input typeahead buffer
3       input_rab [rab$v_pmt] = 0;              !  to get rid of unsolicited
3       get_input(input_rab, 0);                !   character that started the job
3       input_rab [rab$v_pta] = 0;              ! Turn off typeahead purge
3       input_rab [rab$b_tmo] = .sys$gb_retry_tmo; !  and reset timeout period
2       END;
2
2   CH$MOVE(uaf$s_username,
2           buffer[alf$t_username],
2           user_buff);
2   username [0] = uaf$s_username;              ! Setup descriptor of username
2
2   status = get_uafrec(username);              ! Get UAF record for user
2
2   IF .uaf_record EQL 0                        ! If no uaf record
```

```
;  644          1041  2 THEN
;  645          1042  3     BEGIN
;  646          1043  3     IF .status                          ! If this is OPA0: and the UAF is gronked
;  647          1044  3     THEN RETURN false                   ! punt the auto-login
;  648          1045  3     ELSE RETURN lgi$_userauth;          ! otherwise return invalid user
;  649          1046  2     END;
;  650          1047  2
;  651        P 1048  2 $CMKRNL(ROUTIN = set_username,        ! Set username of process
;  652          1049  2         ARGLST = username);
;  653          1050  2
;  654          1051  2 status = get_password (0);            ! Acquire and validate primary password
;  655          1052  2 status1 = get_password (1);           ! Acquire and validate secondary password
;  656          1053  2
;  657          1054  2 IF NOT .status                        ! If invalid password
;  658          1055  2 OR NOT .status1                       !
;  659          1056  2 THEN
;  660          1057  3     BEGIN
;  661          1058  3     RETURN lgi$_notvalid;               ! Return an error
;  662          1059  2     END;
;  663          1060  2
;  664          1061  2 IF NOT .uaf_record [uaf$v_disreconnect] ! If reconnections are not inhibited
;  665          1062  2 THEN connect_check = 1;               !  then do connection checking
;  666          1063  2
;  667          1064  2 RETURN true;                          ! Return successful
;  668          1065  2
;  669          1066  1 END;


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                              46  4C  41  53  59  53  00058 P.AAL:  .ASCII  \SYSALF\                                    ;
54  41  44  2E  3A  4D  45  54  53  59  53  24  53  59  53  0005E P.AAM:  .ASCII  \SYS$SYSTEM:.DAT\                           ;

                                                            .EXTRN  SYS$OPEN, SYS$CONNECT
                                                            .EXTRN  SYS$CLOSE, SYS$GET

                                                            .PSECT  $CODE$,NOWRT,2

                                           0FFC 00000 AUTO_LOGIN:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0927
                        5B 00000000G  00  9E 00002          MOVAB   CLEAR_SYSPRV, R11                             ;
                        5A 00000000G  00  9E 00009          MOVAB   SYS$CLOSE, R10                                ;
                        59 00000000G  00  9E 00010          MOVAB   SYS$CONNECT, R9                               ;
                        58     0000'  CF  9E 00017          MOVAB   USERNAME, R8                                  ;
                        57 00000000G  00  9E 0001C          MOVAB   INPUT_RAB+4, R7                               ;
                        5E     FE6C   CE  9E 00023          MOVAB   -404(SP), SP                                  ;
   0050   8F          00        6E            00  2C 00028          MOVC5   #0, (SP), #0, #80, $RMS_PTR                   ; 0964
                                       B0  AD     0002F                                                           ;
                     B0   AD    5003  8F  B0 00031          MOVW    #20483, $RMS_PTR                              ;
                     C6   AD    0F02  8F  B0 00037          MOVW    #3842, $RMS_PTR+22                            ;
                     CD   AD          20  90 0003D          MOVB    #32, $RMS_PTR+29                              ;
                     CF   AD          02  90 00041          MOVB    #2, $RMS_PTR+31                               ;
                     DC   AD    0000' CF  9E 00045          MOVAB   P.AAL, $RMS_PTR+44                            ;
                     E0   AD    0000' CF  9E 0004B          MOVAB   P.AAM, $RMS_PTR+48                            ;
                     E4   AD    0F06  8F  B0 00051          MOVW    #3846, $RMS_PTR+52                            ;
     FA   AD          02        00            01  F0 00057          INSV    #1, #0, #2, SYSALF_FAB+74                     ; 0969
```

```
 0044  8F         00         6E              00  2C 0005D        MOVC5   #0, (SP), #0, #68, $RMS_PTR                    ; 0978
                                     FF6C    CD     00064                                                                ;
              FF6C   CD            4401      8F  B0 00067        MOVW    #17409, $RMS_PTR                               ;
                8A   AD                      01  90 0006E        MOVB    #1, $RMS_PTR+30                                ;
                8C   AD              80      8F  9B 00072        MOVZBW  #128, $RMS_PTR+32                              ;
                90   AD            0080      CE  9E 00077        MOVAB   BUFFER, $RMS_PTR+36                            ;
                9C   AD            0080      CE  9E 0007D        MOVAB   BUFFER, $RMS_PTR+48                            ;
                A0   AD                      3F  90 00083        MOVB    #63, $RMS_PTR+52                               ;
                A8   AD              B0      AD  9E 00087        MOVAB   SYSALF_FAB, $RMS_PTR+60                        ;
         00000000G   00                      00  FB 0008C        CALLS   #0, SET_SYSPRV                                 ; 0980
                                     B0      AD  9F 00093        PUSHAB  SYSALF_FAB                                     ; 0982
         00000000G   00                      01  FB 00096        CALLS   #1, SYS$OPEN                                   ;
                     2C                      50  E9 0009D        BLBC    R0, 2$                                         ;
                                   FF6C      CD  9F 000A0        PUSHAB  SYSALF_RAB                                     ; 0989
                     69                      01  FB 000A4        CALLS   #1, SYS$CONNECT                                ;
                     56                      50  D0 000A7        MOVL    R0, STATUS                                     ;
                     25                      56  E8 000AA        BLBS    STATUS, 3$                                     ;
          0001C14C   8F                      56  D1 000AD        CMPL    STATUS, #115020                                ; 0992
                                             0D  12 000B4        BNEQ    1$                                             ;
                                     F8      AD  B4 000B6        CLRW    SYSALF_FAB+72                                  ; 0995
                                   FF6C      CD  9F 000B9        PUSHAB  SYSALF_RAB                                     ; 0996
                     69                      01  FB 000BD        CALLS   #1, SYS$CONNECT                                ;
                     56                      50  D0 000C0        MOVL    R0, STATUS                                     ;
                     0C                      56  E8 000C3 1$:    BLBS    STATUS, 3$                                     ; 0998
                                     B0      AD  9F 000C6        PUSHAB  SYSALF_FAB                                     ; 1001
                     6A                      01  FB 000C9        CALLS   #1, SYS$CLOSE                                  ;
                     6B                      00  FB 000CC 2$:    CALLS   #0, CLEAR_SYSPRV                               ; 1002
                                           00CE  31 000CF        BRW     10$                                            ; 1003
                     51 00000000G            00  D0 000D2 3$:    MOVL    CLU_TERM_NAME+4, R1                            ; 1008
                     50      A0              AD  9A 000D9        MOVZBL  SYSALF_RAB+52, R0                              ; 1009
    50           20  61 00000000G            00  2C 000DD        MOVC5   CLU_TERM_NAME, (R1), #32, R0, -                ;
                             9C              BD     000E6                @SYSALF_RAB+48                                 ;
                                   FF6C      CD  9F 000E8        PUSHAB  SYSALF_RAB                                     ; 1011
         00000000G   00                      01  FB 000EC        CALLS   #1, SYS$GET                                    ;
                     56                      50  D0 000F3        MOVL    R0, STATUS                                     ;
                                     B0      AD  9F 000F6        PUSHAB  SYSALF_FAB                                     ; 1013
                     6A                      01  FB 000F9        CALLS   #1, SYS$CLOSE                                  ;
                     6B                      00  FB 000FC        CALLS   #0, CLEAR_SYSPRV                               ; 1015
                     04                      56  E8 000FF        BLBS    STATUS, 4$                                     ; 1017
                     50                      56  D0 00102        MOVL    STATUS, R0                                     ; 1019
                                                 04 00105        RET                                                    ;
           29   03   A7                      05  E1 00106 4$:    BBC     #5, INPUT_RAB+7, 5$                            ; 1021
                1C   A7              80      8F  9B 0010B        MOVZBW  #128, INPUT_RAB+32                             ; 1024
                20   A7                      6E  9E 00110        MOVAB   INPUT_BUFFER, INPUT_RAB+36                     ; 1025
                                     1B      A7  94 00114        CLRB    INPUT_RAB+31                                   ; 1026
                03   A7              40      8F  8A 00117        BICB2   #64, INPUT_RAB+7                               ; 1027
                                             7E  D4 0011C        CLRL    -(SP)                                          ; 1028
                                     FC      A7  9F 0011E        PUSHAB  INPUT_RAB                                      ;
         00000000G   00                      02  FB 00121        CALLS   #2, GET_INPUT                                  ;
                03   A7                      20  8A 00128        BICB2   #32, INPUT_RAB+7                               ; 1029
                1B   A7 00000000G            00  90 0012C        MOVB    SYS$GB_RETRY_TMO, INPUT_RAB+31                 ; 1030
      E0   A8   00BF CE                      20  28 00134 5$:    MOVC3   #32, BUFFER+63, USER_BUFF                      ; 1034
                     68                      20  D0 0013B        MOVL    #32, USERNAME                                  ; 1036
                                             58  DD 0013E        PUSHL   R8                                             ; 1038
         00000000G   00                      01  FB 00140        CALLS   #1, GET_UAFREC                                 ;
                     56                      50  D0 00147        MOVL    R0, STATUS                                     ;
                        00000000G            00  D5 0014A        TSTL    UAF_RECORD                                     ; 1040
```

```
                            0B  12 00150          BNEQ    6$                                  ;
                    4B      56  E8 00152          BLBS    STATUS, 10$                         ; 1043
                    50 00000000G 8F  D0 00155     MOVL    #LGI$_USERAUTH, R0                  ; 1045
                            04  0015C             RET                                         ;
                            58  DD 0015D 6$:      PUSHL   R8                                  ; 1049
                 00000000G  00  9F 0015F          PUSHAB  SET_USERNAME                        ;
       00000000G  00        02  FB 00165          CALLS   #2, SYS$CMKRNL                      ;
                            7E  D4 0016C          CLRL    -(SP)                               ; 1051
           0000V  CF        01  FB 0016E          CALLS   #1, GET_PASSWORD                    ;
                  56        50  D0 00173          MOVL    R0, STATUS                          ;
                            01  DD 00176          PUSHL   #1                                  ; 1052
           0000V  CF        01  FB 00178          CALLS   #1, GET_PASSWORD                    ;
                  03        56  E9 0017D          BLBC    STATUS, 7$                          ; 1054
                  08        50  E8 00180          BLBS    STATUS1, 8$                         ; 1055
                  50 00000000G  8F  D0 00183 7$:  MOVL    #LGI$_NOTVALID, R0                  ; 1058
                            04  0018A             RET                                         ;
                  50 00000000G  00  D0 0018B 8$:  MOVL    UAF_RECORD, R0                      ; 1061
     04      01D5  C0       05  E0 00192          BBS     #5, 469(R0), 9$                     ;
               38  A8       01  D0 00198          MOVL    #1, CONNECT_CHECK                   ; 1062
                  50        01  D0 0019C 9$:      MOVL    #1, R0                              ; 1064
                            04  0019F             RET                                         ;
                  50        D4 001A0 10$:         CLRL    R0                                  ; 1066
                            04  001A2             RET                                         ;

; Routine Size:  419 bytes,    Routine Base:  $CODE$ + 0200
```

```
1 ROUTINE interactive_validation =
1
1 !---
1 !
1 !       Perform interactive user validation.  Prompt for the
1 !       username and password, validate them, and read UAF record.
1 !
1 ! Inputs:
1 !
1 !       None
1 !
1 ! Outputs:
1 !
1 !       uaf_record = Address of UAF record
1 !---
1
2 BEGIN
2
2 MACRO
2     string_count = 0,0,16,0 %;             ! String count field of buffer
2
2 EXTERNAL
2     login_command;                         ! Tables describing LOGIN command
2
2 LOCAL
2     status,
2     status1,
2     desc:         VECTOR [2],              ! descriptor
2     string:       BBLOCK [8],              ! Varying string descriptor
2     buffer:       BBLOCK [2+128]           ! Varying string buffer
2                   VOLATILE,
2     input_buffer: VECTOR [128,BYTE];       ! Input buffer
2
2 CH$MOVE(6, UPLIT BYTE('LOGIN '), input_buffer);
2 input_rab [rab$w_usz] = 128 - 6;
2 input_rab [rab$l_ubf] = input_buffer + 6;
2
2 status = false;                            ! Preset parse status
2
2 string [dsc$b_class] = dsc$k_class_vs; ! Setup varying string descriptor
2 string [dsc$a_pointer] = buffer;
2
2 DO
3     BEGIN
3     input_rab[rab$v_pmt] = 1;              ! Set up prompt
3     input_rab[rab$v_rne] = 0;              ! Echo input
3     input_rab[rab$b_psz] = 12;
3     input_rab[rab$l_pbf] = UPLIT BYTE (cr,lf,'Username: ');
3     WHILE true
3     DO
4         BEGIN
4         get_input(input_rab, 0);           ! Prompt for username
4         input_rab[rab$v_pta] = 0;          ! Purge type-ahead first time only
4         IF .input_rab [rab$w_rsz] NEQ 0 ! If non-null input line,
4         THEN
4             EXITLOOP;                      ! then process the line
3         END;
```

```
3
3       desc [0] = .input_rab [rab$w_rsz] + 6;       ! Setup descriptor of line
3       desc [1] = .input_rab [rab$l_rbf] - 6;       ! with LOGIN appended to front
3
3       status = cli$dcl_parse(desc,login_command); ! Parse the LOGIN command line
3
3       IF .status                                  ! If successfully parsed,
3       THEN
4           BEGIN
4           buffer [string_count] = 0;
4           string [dsc$w_maxstrlen] = 39;
4           cli$get_value(%ASCID 'CLI', string);     ! Get value of /CLI
4           CH$MOVE((cli_name [0] = .buffer [string_count]),
4                   buffer + 2, cli_name_buffer);
4           cli$get_value(%ASCID 'TABLES', string); ! Get value of /TABLES
4           CH$MOVE((table_name [0] = .buffer [string_count]),
4                   buffer + 2, table_name_buffer);
4           cli$get_value(%ASCID 'DISK', string);    ! Get value of /DISK
4           IF .buffer [string_count] NEQ 0          ! If a value was specified
4           THEN
5               BEGIN
5               disk_name [0] = .buffer [string_count];
5               CH$MOVE(.buffer[string_count], buffer+2, .disk_name [1]);
5               IF .disk_name_buffer[.buffer[string_count] - 1] NEQ ':'
5               THEN
6                   BEGIN
6                   disk_name_buffer[.buffer[string_count]] = ':';
6                   disk_name[0] = .disk_name[0] + 1;
5                   END;
4               END;
4           connect_check = cli$present(%ASCID 'CONNECT'); ! Check for /CONNECT
4           IF NOT cli$present(%ASCID 'COMMAND')     ! If /NOCOMMAND,
4           THEN
4               com_negated = true                   ! then disable login procedure
4           ELSE
5               BEGIN
5               string [dsc$w_maxstrlen] = 132;      ! Allow up to 132 char filespec
5               cli$get_value(%ASCID 'COMMAND', string);    ! Get value of /COMMAND
5               com_name [0] = .buffer [string_count];
5               CH$MOVE(.com_name [0], buffer+2, .com_name [1]);
4               END;
4           string [dsc$w_maxstrlen] = uaf$s_username;
4           status = cli$get_value(%ASCID 'USERNAME',string); ! Get username string
4           CH$MOVE(.buffer[string_count],
4                   buffer+2,
4                   user_buff);
4           username [0] = .buffer [string_count];  ! Make descriptor of username
3           END;
3       END
2   UNTIL .status;                                  ! Loop until username obtained
2   cli$end_parse ();                               ! Clean up after command parsing
2
2   !
2   ! Check validity of username and password after prompting for both
2   ! to avoid revealing validity of the username by itself.
2   !
2   status = get_uafrec(username);                  ! Lookup the uaf record
```

```
      2 status1 = get_password (0);              ! Acquire and validate primary password
      2 IF NOT .status                           ! If invalid username
      2 THEN RETURN lgi$_notvalid;               ! Return an error
      2 status = get_password (1);               ! Acquire and validate secondary password
      2
P     2 $CMKRNL(ROUTIN = set_username,           ! Set username of process
      2         ARGLST = username);
      2
      2 IF .uaf_record EQL 0                     ! Check for true status but no UAF
      2 THEN RETURN 1;                           ! which is an OPA0: emergency login
      2
      2 IF NOT .status                           ! If invalid password
      2 OR NOT .status1                          !
      2 THEN
      3     BEGIN
      3     RETURN lgi$_notvalid;                ! Return an error
      2     END;
      2
      2 IF .uaf_record [uaf$v_captive]           ! If user not allowed to change things,
      3 AND (.cli_name [0] NEQ 0                 ! and he changed either CLI name,
      3     OR .table_name [0] NEQ 0             !   or CLI table name,
      3     OR .disk_name [0] NEQ 0              !   or DISK name,
      3     OR .com_name [0] NEQ 0               !   or procedure name,
      3     OR .com_negated)                     !   or procedure negated,
      2 THEN RETURN lgi$_captive;                ! return an error
      2
      2 IF .uaf_record [uaf$v_disreconnect]      ! If user not allowed to reconnect,
      2 THEN
      3     BEGIN
      3     IF .connect_check EQL cli$_defaulted ! If connection checking defaulted,
      3     THEN connect_check = 0;              !  turn it off
      3     IF .connect_check                    ! If connection checking,
      3     THEN
      4         BEGIN
      4         IF .uaf_record [uaf$v_captive]
      4         THEN RETURN lgi$_captive         !  return correct
      4         ELSE RETURN lgi$_disreconnect;   !   error message
      3         END;
      2     END;
      2
      2 IF .uaf_record [uaf$v_defcli]            ! If user not allowed to change things,
      3 AND (.cli_name [0] NEQ 0                 ! and he changed either CLI name,
      3     OR .table_name [0] NEQ 0)            !   or CLI table name,
      2 THEN RETURN lgi$_defcli;                 ! return an error
      2
      2 RETURN true;
      1 END;


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                  20  4E  49  47  4F  4C  0006D P.AAN:     .ASCII  \LOGIN \                          ;
                                      0A  0D  00073 P.AAO:     .BYTE   13, 10                            ;
  20  3A  65  6D  61  6E  72  65  73  55  00075            .ASCII  \Username: \                      ;
                                              0007F            .BLKB   1                                 ;
                          00  49  4C  43  00080 P.AAQ:     .ASCII  \CLI\<0>                          ;
```

```
                                  010E0003  00084 P.AAP:   .LONG    17694723                                    ;
                                  00000000' 00088          .ADDRESS P.AAQ                                       ;
            00  00  53  45  4C  42  41  54  0008C P.AAS:   .ASCII   \TABLES\<0><0>                              ;
                                  010E0006  00094 P.AAR:   .LONG    17694726                                    ;
                                  00000000' 00098          .ADDRESS P.AAS                                       ;
                            4B  53  49  44  0009C P.AAU:   .ASCII   \DISK\                                      ;
                                  010E0004  000A0 P.AAT:   .LONG    17694724                                    ;
                                  00000000' 000A4          .ADDRESS P.AAU                                       ;
            00  54  43  45  4E  4E  4F  43  000A8 P.AAW:   .ASCII   \CONNECT\<0>                                ;
                                  010E0007  000B0 P.AAV:   .LONG    17694727                                    ;
                                  00000000' 000B4          .ADDRESS P.AAW                                       ;
            00  44  4E  41  4D  4D  4F  43  000B8 P.AAY:   .ASCII   \COMMAND\<0>                                ;
                                  010E0007  000C0 P.AAX:   .LONG    17694727                                    ;
                                  00000000' 000C4          .ADDRESS P.AAY                                       ;
            00  44  4E  41  4D  4D  4F  43  000C8 P.ABA:   .ASCII   \COMMAND\<0>                                ;
                                  010E0007  000D0 P.AAZ:   .LONG    17694727                                    ;
                                  00000000' 000D4          .ADDRESS P.ABA                                       ;
            45  4D  41  4E  52  45  53  55  000D8 P.ABC:   .ASCII   \USERNAME\                                  ;
                                  010E0008  000E0 P.ABB:   .LONG    17694728                                    ;
                                  00000000' 000E4          .ADDRESS P.ABC                                       ;

                                                           .EXTRN   LOGIN_COMMAND

                                                           .PSECT   $CODE$,NOWRT,2

                                       0FFC 00000 INTERACTIVE_VALIDATION:
                                                           .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 1067
                      5B       0000'   CF  9E 00002        MOVAB    CONNECT_CHECK, R11                          ;
                      5A       0000'   CF  9E 00007        MOVAB    P.AAN, R10                                  ;
                      59  00000000G    00  9E 0000C        MOVAB    CLI$GET_VALUE, R9                           ;
                      58       0000'   CF  9E 00013        MOVAB    CLI_NAME, R8                                ;
                      57  00000000G    00  9E 00018        MOVAB    INPUT_RAB+4, R7                             ;
                      5E       FEEC    CE  9E 0001F        MOVAB    -276(SP), SP                                ;
        6E            6A               06  28 00024        MOVC3    #6, P.AAN, INPUT_BUFFER                     ; 1100
              1C      A7         7A    8F  9B 00028        MOVZBW   #122, INPUT_RAB+32                          ; 1101
              20      A7         06    AE  9E 0002D        MOVAB    INPUT_BUFFER+6, INPUT_RAB+36                ; 1102
                                       56  D4 00032        CLRL     STATUS                                      ; 1104
              F3      AD               0B  90 00034        MOVB     #11, STRING+3                               ; 1106
              F4      AD       0080    CE  9E 00038        MOVAB    BUFFER, STRING+4                            ; 1107
              03      A7         40    8F  88 0003E 1$:    BISB2    #64, INPUT_RAB+7                            ; 1111
              03      A7               01  8A 00043        BICB2    #1, INPUT_RAB+7                             ; 1112
              30      A7               0C  90 00047        MOVB     #12, INPUT_RAB+52                           ; 1113
              2C      A7         06    AA  9E 0004B        MOVAB    P.AAO, INPUT_RAB+48                         ; 1114
                                       7E  D4 00050 2$:    CLRL     -(SP)                                       ; 1118
                                 FC    A7  9F 00052        PUSHAB   INPUT_RAB                                   ;
        00000000G     00               02  FB 00055        CALLS    #2, GET_INPUT                               ;
              03      A7               20  8A 0005C        BICB2    #32, INPUT_RAB+7                            ; 1119
                                 1E    A7  B5 00060        TSTW     INPUT_RAB+34                                ; 1120
                                       EB  13 00063        BEQL     2$                                          ;
              F8      AD         1E    A7  3C 00065        MOVZWL   INPUT_RAB+34, DESC                          ; 1125
              F8      AD               06  C0 0006A        ADDL2    #6, DESC                                    ;
  FC    AD    24      A7               06  C3 0006E        SUBL3    #6, INPUT_RAB+40, DESC+4                    ; 1126
                          00000000G    00  9F 00074        PUSHAB   LOGIN_COMMAND                               ; 1128
                                 F8    AD  9F 0007A        PUSHAB   DESC                                        ;
        00000000G     00               02  FB 0007D        CALLS    #2, CLI$DCL_PARSE                           ;
                      56               50  D0 00084        MOVL     R0, STATUS                                  ;
                      B4               56  E9 00087        BLBC     STATUS, 1$                                  ; 1130
```

```
                          0080    CE   B4 0008A         CLRW     BUFFER                                   ; 1133
          F0    AD                27   B0 0008E         MOVW     #39, STRING                              ; 1134
                            F0    AD   9F 00092         PUSHAB   STRING                                   ; 1135
                            17    AA   9F 00095         PUSHAB   P.AAP                                    ;
                      69          02   FB 00098         CALLS    #2, CLI$GET_VALUE                        ;
                      50  0080    CE   3C 0009B         MOVZWL   BUFFER, R0                               ; 1136
                      68          50   D0 000A0         MOVL     R0, CLI_NAME                             ;
  FEB4    C8        0082  CE      50   28 000A3         MOVC3    R0, BUFFER+2, CLI_NAME_BUFFER            ;
                            F0    AD   9F 000AB         PUSHAB   STRING                                   ; 1138
                            27    AA   9F 000AE         PUSHAB   P.AAR                                    ;
                      69          02   FB 000B1         CALLS    #2, CLI$GET_VALUE                        ;
                      50  0080    CE   3C 000B4         MOVZWL   BUFFER, R0                               ; 1139
                08    A8          50   D0 000B9         MOVL     R0, TABLE_NAME                           ;
  FF04    C8        0082  CE      50   28 000BD         MOVC3    R0, BUFFER+2, TABLE_NAME_BUFFER          ;
                            F0    AD   9F 000C5         PUSHAB   STRING                                   ; 1141
                            33    AA   9F 000C8         PUSHAB   P.AAT                                    ;
                      69          02   FB 000CB         CALLS    #2, CLI$GET_VALUE                        ;
                          0080    CE   B5 000CE         TSTW     BUFFER                                   ; 1142
                                  2A   13 000D2         BEQL     3$                                       ;
                10    A8  0080    CE   3C 000D4         MOVZWL   BUFFER, DISK_NAME                        ; 1145
    14    B8        0082  CE  0080 CE  28 000DA         MOVC3    BUFFER, BUFFER+2, @DISK_NAME+4           ; 1146
                      50  0080    CE   3C 000E3         MOVZWL   BUFFER, R0                               ; 1147
                      3A  FF53  C840   91 000E8         CMPB     DISK_NAME_BUFFER-1[R0], #58              ;
                                  0E   13 000EE         BEQL     3$                                       ;
                      50  0080    CE   3C 000F0         MOVZWL   BUFFER, R0                               ; 1150
          FF54 C840               3A   90 000F5         MOVB     #58, DISK_NAME_BUFFER[R0]                ;
                            10    A8   D6 000FB         INCL     DISK_NAME                                ; 1151
                            43    AA   9F 000FE 3$:     PUSHAB   P.AAV                                    ; 1154
          00000000G   00          01   FB 00101         CALLS    #1, CLI$PRESENT                          ;
                      6B          50   D0 00108         MOVL     R0, CONNECT_CHECK                        ;
                            53    AA   9F 0010B         PUSHAB   P.AAX                                    ; 1155
          00000000G   00          01   FB 0010E         CALLS    #1, CLI$PRESENT                          ;
                      06          50   E8 00115         BLBS     R0, 4$                                   ;
                20    A8          01   90 00118         MOVB     #1, COM_NEGATED                          ; 1157
                                  1C   11 0011C         BRB      5$                                       ;
          F0    AD          84    8F   9B 0011E 4$:     MOVZBW   #132, STRING                             ; 1160
                            F0    AD   9F 00123         PUSHAB   STRING                                   ; 1161
                            63    AA   9F 00126         PUSHAB   P.AAZ                                    ;
                      69          02   FB 00129         CALLS    #2, CLI$GET_VALUE                        ;
                18    A8  0080    CE   3C 0012C         MOVZWL   BUFFER, COM_NAME                         ; 1162
    1C    B8        0082  CE  18  A8   28 00132         MOVC3    COM_NAME, BUFFER+2, @COM_NAME+4          ; 1163
                    F0    AD      20   B0 0013A 5$:     MOVW     #32, STRING                              ; 1165
                            F0    AD   9F 0013E         PUSHAB   STRING                                   ; 1166
                            73    AA   9F 00141         PUSHAB   P.ABB                                    ;
                      69          02   FB 00144         CALLS    #2, CLI$GET_VALUE                        ;
                      56          50   D0 00147         MOVL     R0, STATUS                               ;
    A8    AB        0082  CE  0080 CE  28 0014A         MOVC3    BUFFER, BUFFER+2, USER_BUFF              ; 1167
                    C8    AB  0080 CE  3C 00153         MOVZWL   BUFFER, USERNAME                         ; 1170
                      03          56   E8 00159         BLBS     STATUS, 6$                               ; 1173
                                FEDF   31 0015C         BRW      1$                                       ;
          00000000G   00          00   FB 0015F 6$:     CALLS    #0, CLI$END_PARSE                        ; 1174
                            C8    AB   9F 00166         PUSHAB   USERNAME                                 ; 1180
          00000000G   00          01   FB 00169         CALLS    #1, GET_UAFREC                           ;
                      56          50   D0 00170         MOVL     R0, STATUS                               ;
                                  7E   D4 00173         CLRL     -(SP)                                    ; 1181
              0000V   CF          01   FB 00175         CALLS    #1, GET_PASSWORD                         ;
                      52          50   D0 0017A         MOVL     R0, STATUS1                              ;
```

```
                     29                56  E9 0017D          BLBC    STATUS, 7$                                   ; 1182
                                       01  DD 00180          PUSHL   #1                                           ; 1184
              0000V  CF                01  FB 00182          CALLS   #1, GET_PASSWORD                             ;
                     56                50  D0 00187          MOVL    R0, STATUS                                   ;
                                  C8   AB  9F 0018A          PUSHAB  USERNAME                                     ; 1187
                           00000000G   00  9F 0018D          PUSHAB  SET_USERNAME                                 ;
          00000000G  00                02  FB 00193          CALLS   #2, SYS$CMKRNL                               ;
                     50  00000000G     00  D0 0019A          MOVL    UAF_RECORD, R0                               ; 1189
                                       69  13 001A1          BEQL    15$                                          ;
                     03                56  E9 001A3          BLBC    STATUS, 7$                                   ; 1192
                     08                52  E8 001A6          BLBS    STATUS1, 8$                                  ; 1193
                     50  00000000G     8F  D0 001A9  7$:     MOVL    #LGI$_NOTVALID, R0                           ; 1196
                                       04  001B0             RET                                                  ;
                     51         01D4   C0  9E 001B1  8$:     MOVAB   468(R0), R1                                  ; 1199
          17         61                03  E1 001B6          BBC     #3, (R1), 9$                                 ;
                                       68  D5 001BA          TSTL    CLI_NAME                                     ; 1200
                                       29  12 001BC          BNEQ    11$                                          ;
                                  08   A8  D5 001BE          TSTL    TABLE_NAME                                   ; 1201
                                       24  12 001C1          BNEQ    11$                                          ;
                                  10   A8  D5 001C3          TSTL    DISK_NAME                                    ; 1202
                                       1F  12 001C6          BNEQ    11$                                          ;
                                  18   A8  D5 001C8          TSTL    COM_NAME                                     ; 1203
                                       1A  12 001CB          BNEQ    11$                                          ;
                     16           20   A8  E8 001CD          BLBS    COM_NEGATED, 11$                             ; 1204
          22         61                0D  E1 001D1  9$:     BBC     #13, (R1), 13$                               ; 1207
          00000000G  8F                6B  D1 001D5          CMPL    CONNECT_CHECK, #CLI$_DEFAULTED               ; 1210
                                       02  12 001DC          BNEQ    10$                                          ;
                                       6B  D4 001DE          CLRL    CONNECT_CHECK                                ; 1211
                     14                6B  E9 001E0  10$:    BLBC    CONNECT_CHECK, 13$                           ; 1212
          08         61                03  E1 001E3          BBC     #3, (R1), 12$                                ; 1215
                     50  00000000G     8F  D0 001E7  11$:    MOVL    #LGI$_CAPTIVE, R0                            ; 1217
                                       04  001EE             RET                                                  ;
                     50  00000000G     8F  D0 001EF  12$:    MOVL    #LGI$_DISRECONNECT, R0                       ;
                                       04  001F6             RET                                                  ;
          11         61                01  E1 001F7  13$:    BBC     #1, (R1), 15$                                ; 1221
                                       68  D5 001FB          TSTL    CLI_NAME                                     ; 1222
                                       05  12 001FD          BNEQ    14$                                          ;
                                  08   A8  D5 001FF          TSTL    TABLE_NAME                                   ; 1223
                                       08  13 00202          BEQL    15$                                          ;
                     50  00000000G     8F  D0 00204  14$:    MOVL    #LGI$_DEFCLI, R0                             ; 1224
                                       04  0020B             RET                                                  ;
                     50                01  D0 0020C  15$:    MOVL    #1, R0                                       ; 1226
                                       04  0020F             RET                                                  ; 1227

; Routine Size:  528 bytes,    Routine Base:  $CODE$ + 03A3
```

```
  1 ROUTINE get_password (pwd_number) =
  1
  1 !---
  1 !
  1 !       Acquire a password if one needed and validate it.
  1 !       Return status is true if password check is successful.
  1 !
  1 ! Inputs:
  1 !
  1 !       pwd_number - 0 if validating primary password
  1 !                    1 if validating secondary password
  1 !       uaf_record - Address of UAF record for user, if any
  1 !
  1 ! Outputs:
  1 !
  1 !       routine = True if password validated or none needed, else false.
  1 !---
  1
  2 BEGIN
  2
  2 LOCAL
  2     status,
  2     password_isi,
  2     fab: BBLOCK[fab$c_bln],
  2     rab: BBLOCK[rab$c_bln],
  2     string: VECTOR[24,BYTE],
  2     password:   VECTOR [2];                 ! Password descriptor
  2
  2 $ASSUME ($BYTEOFFSET (uaf$q_pwd2), EQL, $BYTEOFFSET (uaf$q_pwd)+8);
  2 $ASSUME ($BYTEOFFSET (uaf$b_encrypt2), EQL, $BYTEOFFSET (uaf$b_encrypt)+1);
  2
  2 IF .uaf_record NEQ 0                        ! If there is a uaf record and no
  2 THEN                                        ! password is needed
  2 IF  .bblock [uaf_record [uaf$q_pwd],(.pwd_number*8),0,32,0] EQL 0
  2 AND .bblock [uaf_record [uaf$q_pwd],(.pwd_number*8)+4,0,32,0] EQL 0
  2 THEN
  2     RETURN true;                            ! Then return success without prompting
  2
  2 !
  2 ! If SYS$INPUT is a terminal, and is set to be local_echo,
  2 ! then ask for the password with an overstriking mask.
  2 !
  2 IF .terminal_device                                     ! If a terminal
  2 AND .dev_dep_2[tt2$v_localecho]                         ! with local_echo set
  2 THEN
  3     BEGIN
  3
P 3     $FAB_INIT(FAB = fab,
P 3             FNM = 'SYS$OUTPUT',
P 3             FAC = (GET,PUT)
  3             );
  3     fab[fab$v_cr] = 0;
  3
P 3     $RAB_INIT(RAB = rab,                            ! Initialize local RAB
P 3             FAB = fab,
P 3             ROP = (pmt,cvt,tmo,rne),                ! Read with prompt and timeout,
P 3                                                     ! convert to uppercase,
P 3                                             ! read no echo
P 3           PBF = UPLIT BYTE(cr,lf,'Password: ',
P 3                                rep 15 of ('#'), rep 15 of (bs),
P 3                                rep 15 of ('H'), rep 15 of (bs),
P 3                                rep 15 of ('X'), rep 15 of (bs)),
P 3           PSZ = 102,
P 3           UBF = string,
P 3           USZ = %ALLOCATION(string),
  3           TMO = .sys$gb_retry_tmo);
  3
  3     password_isi = (rab[rab$w_isi] = .ppd[ppd$w_inpisi]);
  3     get_input (rab, 0);                     ! Get the password
  3
  3     password [0] = .rab[rab$w_rsz];         ! Store password size
  3     password [1] = .rab[rab$l_rbf];         ! and location
  3
  3     rab[rab$w_isi] = 0;
  3     $OPEN (FAB = fab);
  3     $CONNECT (RAB = rab);
  3
  3     write_output (uplit (26, UPLIT BYTE (cr, 'Password: ', rep 15 of ('@'))),
  3             0, rab);
  3     $CLOSE (FAB = fab);
  3     END
  2 ELSE
  3     BEGIN                                   ! Normal processing
  3
  3     input_rab [rab$l_pbf] = UPLIT BYTE(cr,lf,'Password: ');
  3     input_rab [rab$b_psz] = 12;
  3     input_rab [rab$v_pmt] = true;                   ! Read woth prompt
  3     input_rab [rab$v_rne] = true;                   ! Read no-echo
  3     input_rab [rab$v_pta] = false;                  ! Clear purge-typeahead
  3
  3     password_isi = .input_rab [rab$w_isi];
  3     get_input(input_rab, 0);                        ! Prompt for password
  3
  3     password [0] = .input_rab [rab$w_rsz];          ! Setup descriptor of password
  3     password [1] = .input_rab [rab$l_rbf];
  3
  2     END;
  2
P 2 $CMEXEC(ROUTIN = zero_password,             ! Zero the password in the RMS area
  2         ARGLST = .password_isi);
  2
  3 CH$COPY((fail_password [0] = MINU(.password [0], ! Save password
  2                                   nsa$s_pkt_password)),
  2        .password [1],                               !  for auditing
  2        0,
  2        nsa$s_pkt_password,
  2        .fail_password [1]);
  2
  2 IF .uaf_record EQL 0                         ! If no uaf record
  2 THEN                                        ! return unconditional failure now that
  2     RETURN false;                           ! the prompt has been done
  2
  2 status = lgi$check_pass (password, .uaf_record, .pwd_number); ! Validate user password
  2
```

```
2 RETURN .status;                                            ! Return success/failure
2
1 END;


                                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

    54  55  50  54  55  4F  24  53  59  53   000E8 P.ABD:  .ASCII  \SYS$OUTPUT\
                                        0A  0D   000F2 P.ABE:  .BYTE   13, 10
    20  3A  64  72  6F  77  73  73  61  50   000F4          .ASCII  \Password: \
                                            23   000FE          .ASCII  \#\
                                            23   000FF          .ASCII  \#\
                                            23   00100          .ASCII  \#\
                                            23   00101          .ASCII  \#\
                                            23   00102          .ASCII  \#\
                                            23   00103          .ASCII  \#\
                                            23   00104          .ASCII  \#\
                                            23   00105          .ASCII  \#\
                                            23   00106          .ASCII  \#\
                                            23   00107          .ASCII  \#\
                                            23   00108          .ASCII  \#\
                                            23   00109          .ASCII  \#\
                                            23   0010A          .ASCII  \#\
                                            23   0010B          .ASCII  \#\
                                            23   0010C          .ASCII  \#\
                                           08#   0010D          .BYTE   8[15]
                                            48   0011C          .ASCII  \H\
                                            48   0011D          .ASCII  \H\
                                            48   0011E          .ASCII  \H\
                                            48   0011F          .ASCII  \H\
                                            48   00120          .ASCII  \H\
                                            48   00121          .ASCII  \H\
                                            48   00122          .ASCII  \H\
                                            48   00123          .ASCII  \H\
                                            48   00124          .ASCII  \H\
                                            48   00125          .ASCII  \H\
                                            48   00126          .ASCII  \H\
                                            48   00127          .ASCII  \H\
                                            48   00128          .ASCII  \H\
                                            48   00129          .ASCII  \H\
                                            48   0012A          .ASCII  \H\
                                           08#   0012B          .BYTE   8[15]
                                            58   0013A          .ASCII  \X\
                                            58   0013B          .ASCII  \X\
                                            58   0013C          .ASCII  \X\
                                            58   0013D          .ASCII  \X\
                                            58   0013E          .ASCII  \X\
                                            58   0013F          .ASCII  \X\
                                            58   00140          .ASCII  \X\
                                            58   00141          .ASCII  \X\
                                            58   00142          .ASCII  \X\
                                            58   00143          .ASCII  \X\
                                            58   00144          .ASCII  \X\
                                            58   00145          .ASCII  \X\
                                            58   00146          .ASCII  \X\
                                            58   00147          .ASCII  \X\
INTERACT                                                 E  9
V04-000                                           16-Sep-1984 01:55:50    VAX-11 Bliss-32 V4.0-742                    Page  30
                                                  14-Sep-1984 12:41:07    [LOGIN.SRC]INTERACT.B32;1                          (6)

                                         58  00148          .ASCII  \X\                                         ;
                                        08#  00149          .BYTE   8[15]                                       ;
                                         0D  00158 P.ABG:   .BYTE   13                                          ;
        20  3A  64  72  6F  77  73  73  61  50  00159          .ASCII  \Password: \                                ;
                                         40  00163          .ASCII  \@\                                         ;
                                         40  00164          .ASCII  \@\                                         ;
                                         40  00165          .ASCII  \@\                                         ;
                                         40  00166          .ASCII  \@\                                         ;
                                         40  00167          .ASCII  \@\                                         ;
                                         40  00168          .ASCII  \@\                                         ;
                                         40  00169          .ASCII  \@\                                         ;
                                         40  0016A          .ASCII  \@\                                         ;
                                         40  0016B          .ASCII  \@\                                         ;
                                         40  0016C          .ASCII  \@\                                         ;
                                         40  0016D          .ASCII  \@\                                         ;
                                         40  0016E          .ASCII  \@\                                         ;
                                         40  0016F          .ASCII  \@\                                         ;
                                         40  00170          .ASCII  \@\                                         ;
                                         40  00171          .ASCII  \@\                                         ;
                                             00172          .BLKB   2
                                   0000001A  00174 P.ABF:   .LONG   26                                          ;
                                   00000000' 00178          .ADDRESS P.ABG                                      ;
                                      0A  0D  0017C P.ABH:   .BYTE   13, 10                                      ;
        20  3A  64  72  6F  77  73  73  61  50  0017E          .ASCII  \Password: \                                ;


                                                            .PSECT  $CODE$,NOWRT,2

                                       01FC 00000 GET_PASSWORD:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8                   ; 1228
                      58 00000000G   00   9E 00002          MOVAB   UAF_RECORD, R8                              ;
                      57 00000000G   00   9E 00009          MOVAB   GET_INPUT, R7                               ;
                      56 00000000G   00   9E 00010          MOVAB   INPUT_RAB+4, R6                             ;
                      5E       FF4C   CE   9E 00017          MOVAB   -180(SP), SP                                ;
                      51              68   D0 0001C          MOVL    UAF_RECORD, R1                              ; 1259
                                      1A   13 0001F          BEQL    1$                                          ;
                      50              04   AC D0 00021       MOVL    PWD_NUMBER, R0                              ; 1261
                                 0154 C140   7F 00025          PUSHAQ  340(R1)[R0]                                 ;
                                      9E   D5 0002A          TSTL    @(SP)+                                      ;
                                      0D   12 0002C          BNEQ    1$                                          ;
                                 0158 C140   7F 0002E          PUSHAQ  344(R1)[R0]                                 ; 1262
                                      9E   D5 00033          TSTL    @(SP)+                                      ;
                                      04   12 00035          BNEQ    1$                                          ;
                      50              01   D0 00037          MOVL    #1, R0                                      ; 1264
                                           04 0003A          RET                                                 ;
                      03 00000000G   00   E8 0003B 1$:      BLBS    TERMINAL_DEVICE, 3$                         ; 1270
                                    00B6   31 00042 2$:      BRW     4$                                          ;
                      F6 00000000G   00   E9 00045 3$:      BLBC    DEV_DEP_2, 2$                               ; 1271
    0050    8F              00        6E   00   2C 0004C          MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 1278
                                      64   AE    00053                                                         ;
                 64   AE       5003   8F   B0 00055          MOVW    #20483, $RMS_PTR                            ;
                 7A   AE              03   90 0005B          MOVB    #3, $RMS_PTR+22                             ;
                 CF   AD              02   90 0005F          MOVB    #2, $RMS_PTR+31                             ;
                 DC   AD       0000'  CF   9E 00063          MOVAB   P.ABD, $RMS_PTR+44                          ;
                 E4   AD              0A   90 00069          MOVB    #10, $RMS_PTR+52                            ;
                 CE   AD              02   8A 0006D          BICB2   #2, FAB+30                                  ; 1279
```

```
  0044    8F          00          6E              00  2C 00071        MOVC5   #0, (SP), #0, #68, $RMS_PTR              ; 1293
                                           20     AE     00078                                                             :
                       20   AE         4401       8F  B0 0007A        MOVW    #17409, $RMS_PTR                          :
                       24   AE     47000000       8F  D0 00080        MOVL    #1191182336, $RMS_PTR+4                   :
                       3F   AE    00000000G       00  90 00088        MOVB    SYS$GB_RETRY_TMO, $RMS_PTR+31             :
                       40   AE           18       B0     00090        MOVW    #24, $RMS_PTR+32                          :
                       44   AE           08       AE  9E 00094        MOVAB   STRING, $RMS_PTR+36                       :
                       50   AE        0000'       CF  9E 00099        MOVAB   P.ABE, $RMS_PTR+48                        :
                       54   AE           66       8F  90 0009F        MOVB    #102, $RMS_PTR+52                         :
                       5C   AE           64       AE  9E 000A4        MOVAB   FAB, $RMS_PTR+60                          :
                            50    00000000G       00  3C 000A9        MOVZWL  PPD+34, R0                                ; 1295
                       22   AE                    50  B0 000B0        MOVW    R0, RAB+2                                 :
                            52                    50  D0 000B4        MOVL    R0, PASSWORD_ISI                          :
                                                  7E  D4 000B7        CLRL    -(SP)                                     ; 1296
                                         24       AE  9F 000B9        PUSHAB  RAB                                       :
                            67                    02  FB 000BC        CALLS   #2, GET_INPUT                             :
                            6E           42       AE  3C 000BF        MOVZWL  RAB+34, PASSWORD                          ; 1298
                       04   AE           48       AE  D0 000C3        MOVL    RAB+40, PASSWORD+4                        ; 1299
                                         22       AE  B4 000C8        CLRW    RAB+2                                     ; 1301
                                         64       AE  9F 000CB        PUSHAB  FAB                                       ; 1302
              00000000G     00                    01  FB 000CE        CALLS   #1, SYS$OPEN                              :
                                         20       AE  9F 000D5        PUSHAB  RAB                                       ; 1303
              00000000G     00                    01  FB 000D8        CALLS   #1, SYS$CONNECT                           :
                                         20       AE  9F 000DF        PUSHAB  RAB                                       ; 1305
                                                  7E  D4 000E2        CLRL    -(SP)                                     :
                                      0000'       CF  9F 000E4        PUSHAB  P.ABF                                     :
              00000000G     00                    03  FB 000E8        CALLS   #3, WRITE_OUTPUT                          :
                                         64       AE  9F 000EF        PUSHAB  FAB                                       ; 1307
              00000000G     00                    01  FB 000F2        CALLS   #1, SYS$CLOSE                             :
                                                  28  11 000F9        BRB     5$                                        ; 1270
                       2C   A6        0000'       CF  9E 000FB 4$:    MOVAB   P.ABH, INPUT_RAB+48                       ; 1312
                       30   A6                    0C  90 00101        MOVB    #12, INPUT_RAB+52                         ; 1313
                       03   A6           41       8F  88 00105        BISB2   #65, INPUT_RAB+7                          ; 1315
                       03   A6                    20  8A 0010A        BICB2   #32, INPUT_RAB+7                          ; 1316
                            52           FE       A6  3C 0010E        MOVZWL  INPUT_RAB+2, PASSWORD_ISI                 ; 1318
                                                  7E  D4 00112        CLRL    -(SP)                                     ; 1319
                                         FC       A6  9F 00114        PUSHAB  INPUT_RAB                                 :
                            67                    02  FB 00117        CALLS   #2, GET_INPUT                             :
                            6E           1E       A6  3C 0011A        MOVZWL  INPUT_RAB+34, PASSWORD                    ; 1321
                       04   AE           24       A6  D0 0011E        MOVL    INPUT_RAB+40, PASSWORD+4                  ; 1322
                                                  52  DD 00123 5$:    PUSHL   PASSWORD_ISI                              ; 1327
                                      0000V       CF  9F 00125        PUSHAB  ZERO_PASSWORD                             :
              00000000G     00                    02  FB 00129        CALLS   #2, SYS$CMEXEC                            :
                            50                    6E  D0 00130        MOVL    PASSWORD, R0                              ; 1329
                            1F                    50  D1 00133        CMPL    R0, #31                                   :
                                                  03  1B 00136        BLEQU   6$                                        :
                            50                    1F  D0 00138        MOVL    #31, R0                                   :
              00000000G     00                    50  D0 0013B 6$:    MOVL    R0, FAIL_PASSWORD                         :
                            51    00000000G       00  D0 00142        MOVL    FAIL_PASSWORD+4, R1                       ; 1334
          1F          00    04           BE       50  2C 00149        MOVC5   R0, @PASSWORD+4, #0, #31, (R1)            :
                                                  61     0014F                                                             :
                            50                    68  D0 00150        MOVL    UAF_RECORD, R0                            ; 1336
                                                  10  13 00153        BEQL    7$                                        :
                                         04       AC  DD 00155        PUSHL   PWD_NUMBER                                ; 1340
                                                  50  DD 00158        PUSHL   R0                                        :
                                         08       AE  9F 0015A        PUSHAB  PASSWORD                                  :
              00000000G     00                    03  FB 0015D        CALLS   #3, LGI$CHECK_PASS                        :
```

```
                         04 00164          RET                                       : 1342
                      50 D4 00165 7$:      CLRL     R0                               : 1344
                         04 00167          RET                                       :

; Routine Size:  360 bytes,    Routine Base:  $CODE$ + 05B3
```

```
   1 ROUTINE write_announcement (logname) =
   1
   1 !---
   1 !
   1 !         Write an announcement message to the primary output stream.
   1 !         If the logical name given has a translation, it may be of the
   1 !         following two forms:
   1 !
   1 !                 '@filespec'                Write contents of file
   1 !                 'string'                   Write string literally
   1 !
   1 ! Inputs:
   1 !
   1 !         logname = Address of descriptor of logical name
   1 !
   1 ! Outputs:
   1 !
   1 !         routine = True if user-supplied message output, else false
   1 !---
   1
   2 BEGIN
   2
   2 LOCAL
   2     trnlnm_item_list: BLOCK[1*3+1,LONG], ! TRNLNM item list for 1 item
   2     desc:           VECTOR [2],
   2     buffer:         VECTOR [128,BYTE];
   2
   2 trnlnm_item_list[0, 0,16,0] = (desc[0] = 128);
   2 trnlnm_item_list[0,16,16,0] = lnm$_string; ! Fetch name's value string
   2 trnlnm_item_list[1, 0,32,0] = (desc[1] = buffer);
   2 trnlnm_item_list[2, 0,32,0] = desc[0];
   2 trnlnm_item_list[3, 0,32,0] = 0;
   2
P  2 IF $TRNLNM(TABNAM = %ASCID 'LNM$FILE_DEV', ! If translation exists
P  2            LOGNAM = .logname,
   3            ITMLST = trnlnm_item_list)
   2 EQL ss$_normal
   2 THEN
   3     BEGIN
   3     IF .buffer [0] EQL '@'              ! If logname points to file,
   3     THEN
   4         BEGIN
   4         desc [0] = .desc [0] - 1;       ! then remove '@'
   4         desc [1] = .desc [1] + 1;
   4         write_file(desc);               ! and write file to output stream
   4         END
   3     ELSE IF .desc [0] NEQ 0             ! Else if non-null string,
   3     THEN
   4         BEGIN
   4         write_output(UPLIT (0,0));      ! output blank line
   4         write_output(desc);             ! output translation of logname
   4         write_output(UPLIT (0,0));      ! output blank line
   3         END;
   3     RETURN true;                        ! return successful
   2     END;
   2
   2 RETURN false;                           ! return failure
```

```
; 1008          1402  2
; 1009          1403  1 END;



                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

        56  45  44  5F  45  4C  49  46  24  4D  4E  4C  00188 P.ABJ:   .ASCII  \LNM$FILE_DEV\                          ;
                                            010E000C  00194 P.ABI:   .LONG   17694732                                  ;
                                            00000000' 00198          .ADDRESS P.ABJ                                     ;
                                   00000000  00000000  0019C P.ABK:   .LONG   0, 0                                      ;
                                   00000000  00000000  001A4 P.ABL:   .LONG   0, 0                                      ;

                                                                    .EXTRN  SYS$TRNLNM

                                                                    .PSECT  $CODE$,NOWRT,2

                                                 0004 00000 WRITE_ANNOUNCEMENT:
                                                                    .WORD   Save R2                                    ; 1345
                             52 00000000G   00  9E 00002            MOVAB   WRITE_OUTPUT, R2                           ;
                             5E       FF68  CE  9E 00009            MOVAB   -152(SP), SP                               ;
                     E8      AD         80  8F  9A 0000E            MOVZBL  #128, DESC                                 ; 1372
                     F0      AD 00020080   8F  D0 00013            MOVL    #131200, TRNLNM_ITEM_LIST                  ;
                             50             6E  9E 0001B            MOVAB   BUFFER, R0                                 ; 1374
                     EC      AD             50  D0 0001E            MOVL    R0, DESC+4                                 ;
                     F4      AD             50  D0 00022            MOVL    R0, TRNLNM_ITEM_LIST+4                     ;
                     F8      AD         E8  AD  9E 00026            MOVAB   DESC, TRNLNM_ITEM_LIST+8                   ; 1375
                                        FC  AD  D4 0002B            CLRL    TRNLNM_ITEM_LIST+12                        ; 1376
                                        F0  AD  9F 0002E            PUSHAB  TRNLNM_ITEM_LIST                           ; 1380
                                            7E  D4 00031            CLRL    -(SP)                                      ;
                                        04  AC  DD 00033            PUSHL   LOGNAME                                    ;
                                     0000'  CF  9F 00036            PUSHAB  P.ABI                                      ;
                                            7E  D4 0003A            CLRL    -(SP)                                      ;
                  00000000G  00             05  FB 0003C            CALLS   #5, SYS$TRNLNM                             ;
                             01             50  D1 00043            CMPL    R0, #1                                     ; 1381
                                            35  12 00046            BNEQ    3$                                         ;
                     40      8F             6E  91 00048            CMPB    BUFFER, #64                                ; 1384
                                            12  12 0004C            BNEQ    1$                                         ;
                                        E8  AD  D7 0004E            DECL    DESC                                       ; 1387
                                        EC  AD  D6 00051            INCL    DESC+4                                     ; 1388
                                        E8  AD  9F 00054            PUSHAB  DESC                                       ; 1389
                  00000000G  00             01  FB 00057            CALLS   #1, WRITE_FILE                             ;
                                            19  11 0005E            BRB     2$                                         ; 1384
                                        E8  AD  D5 00060 1$:        TSTL    DESC                                       ; 1391
                                            14  13 00063            BEQL    2$                                         ;
                                     0000'  CF  9F 00065            PUSHAB  P.ABK                                      ; 1394
                             62             01  FB 00069            CALLS   #1, WRITE_OUTPUT                           ;
                                        E8  AD  9F 0006C            PUSHAB  DESC                                       ; 1395
                             62             01  FB 0006F            CALLS   #1, WRITE_OUTPUT                           ;
                                     0000'  CF  9F 00072            PUSHAB  P.ABL                                      ; 1396
                             62             01  FB 00076            CALLS   #1, WRITE_OUTPUT                           ;
                             50             01  D0 00079 2$:        MOVL    #1, R0                                     ; 1398
                                                04 0007C            RET                                                ;
                                            50  D4 0007D 3$:        CLRL    R0                                         ; 1401
                                                04 0007F            RET                                                ; 1403

; Routine Size:  128 bytes,    Routine Base:  $CODE$ + 071B
```

```
GLOBAL ROUTINE announce: NOVALUE =

!---
!
!       Issue messages to interactive users announcing successful login,
!       dates of last logins, number of login failures, and number of
!       mail messages.
!
! Inputs:
!
!       None
!
! Outputs:
!
!       None
!---

BEGIN

EXTERNAL
    sys$gq_version;                         ! System version string

LOCAL
    trnlnm_item_list: BLOCK[1*3+1,LONG], ! TRNLNM item list for 1 item
    length,                              ! String variables
    msgcount,
    ptr:        REF VECTOR [,BYTE],
    msg_buffer: VECTOR [128,BYTE],       ! Buffer for announcement message
    bufdesc:    VECTOR [2];              ! Buffer descriptor for above message

IF .uaf_record EQL 0
THEN RETURN;

IF NOT .uaf_record [uaf$v_diswelcom]    ! If a welcome message allowed
THEN
    BEGIN
    IF NOT write_announcement(%ASCID 'SYS$WELCOME') ! If no user welcome
    THEN
        BEGIN
        ptr = CH$MOVE(28, UPLIT BYTE('  Welcome to VAX/VMS version '),
                msg_buffer);
        ptr = CH$MOVE(4, sys$gq_version, .ptr);
        length = .ptr - msg_buffer;     ! Set default length of message

        ptr = CH$MOVE(9, UPLIT BYTE(' on node '), .ptr);

        trnlnm_item_list[0, 0,16,0] = (bufdesc[0] = 16);
        trnlnm_item_list[0,16,16,0] = lnm$_string; ! Fetch name's value string
        trnlnm_item_list[1, 0,32,0] = (bufdesc[1] = .ptr);
        trnlnm_item_list[2, 0,32,0] = bufdesc[0];
        trnlnm_item_list[3, 0,32,0] = 0;

        IF $TRNLNM(TABNAM = %ASCID 'LNM$SYSTEM_TABLE', ! Translate SYS$NODE
                   LOGNAM = %ASCID 'SYS$NODE',
                   ACMODE = UPLIT(psl$c_exec),
                   ITMLST = trnlnm_item_list)
        EQL ss$_normal                  ! If successful,
     4              THEN
     5                  BEGIN
     5                                                  ! Append node name minus ::
     5                  length = .ptr + .bufdesc [0] - 2 - msg_buffer;
     4                  END;
     4
     4              bufdesc [0] = .length;              ! Setup descriptor of message
     4              bufdesc [1] = msg_buffer;
     4
     4              write_output(bufdesc);              ! Write message
     3              END;
     2          END;
     2
     2  !
     2  ! Write messages giving times of last logins and login failure counts.
     2  !
     2  IF NOT .uaf_record [uaf$v_disreport]    ! If login reports allowed
     2  THEN
     3      BEGIN
     3
     3      BIND
     3          lastlogin_i = uaf_record [uaf$q_lastlogin_i] : VECTOR,
     3          lastlogin_n = uaf_record [uaf$q_lastlogin_n] : VECTOR;
     3
     4      IF (.lastlogin_i [0] NEQ 0) OR (.lastlogin_i [1] NEQ 0)
     3      THEN
     3          write_fao(UPLIT BYTE(%ASCIC
     3              '       Last interactive login on !AC, !17%D'),
     3              ascic_day_of_week(lastlogin_i), lastlogin_i);
     3
     4      IF (.lastlogin_n [0] NEQ 0) OR (.lastlogin_n [1] NEQ 0)
     3      THEN
     3          write_fao(UPLIT BYTE(%ASCIC
     3              '       Last non-interactive login on !AC, !17%D'),
     3              ascic_day_of_week(lastlogin_n), lastlogin_n);
     3
     3      IF .uaf_record [uaf$w_logfails] GTRU 0
L    3      THEN write_fao(UPLIT BYTE(%ASCIC %STRING (
L    3          %CHAR(bell),%CHAR(bell),%CHAR(bell),
     3          '           !UW failure!%S since last successful login')),
     3          .uaf_record [uaf$w_logfails]);
     3
     2      END;
     2
     2  !
     2  ! If any new mail since last logged on, issue a message.
     2  !
     2  IF NOT .uaf_record [uaf$v_dismail]      ! If new mail message allowed
     2  AND mail$get_new_count(msgcount, .uaf_record)
     2  THEN
     2      IF .msgcount GTR 0
L    2      THEN write_fao(UPLIT BYTE(%ASCIC %STRING(
L    2          %CHAR(cr),%CHAR(lf),%CHAR(bell),
L    2          '                   You have !UW new Mail message!%S.',
     2          %CHAR(cr),%CHAR(lf))), .msgcount)
     2      ELSE IF .msgcount LSS 0
L    2      THEN write_output(%ASCID %STRING(
; 1125          L 1518  2          %CHAR(cr),%CHAR(lf),%CHAR(bell),
; 1126          L 1519  2          '                You have new Mail messages.',
; 1127            1520  2          %CHAR(cr),%CHAR(lf)));
; 1128            1521  2
; 1129            1522  1 END;



                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

            00  45  4D  4F  43  4C  45  57  24  53  59  53  001AC P.ABN:  .ASCII  \SYS$WELCOME\<0>                                ;
                                                  010E000B  001B8 P.ABM:  .LONG   17694731                                        ;
                                                  00000000' 001BC         .ADDRESS P.ABN                                          ;
58  41  56  20  6F  74  20  65  6D  6F  63  6C  65  57  09  001C0 P.ABO:  .ASCII  <9>\Welcome to VAX/VMS version \                ;
        20  6E  6F  69  73  72  65  76  20  53  4D  56  2F  001CF
                        20  65  64  6F  6E  20  6E  6F  20  001DC P.ABP:  .ASCII  \ on node \                                     ;
                                                            001E5         .BLKB   3                                               ;
4C  42  41  54  5F  4D  45  54  53  59  53  24  4D  4E  4C  001E8 P.ABR:  .ASCII  \LNM$SYSTEM_TABLE\                              ;
                                                        45  001F7
                                                  010E0010  001F8 P.ABQ:  .LONG   17694736                                        ;
                                                  00000000' 001FC         .ADDRESS P.ABR                                          ;
                            45  44  4F  4E  24  53  59  53  00200 P.ABT:  .ASCII  \SYS$NODE\                                      ;
                                                  010E0008  00208 P.ABS:  .LONG   17694728                                        ;
                                                  00000000' 0020C         .ADDRESS P.ABT                                          ;
                                                  00000001  00210 P.ABU:  .LONG   1                                               ;
72  65  74  6E  69  20  74  73  61  4C  20  20  20  20  28  00214 P.ABV:  .ASCII  \(     Last interactive login on !AC, !17%\     ;
6E  6F  20  6E  69  67  6F  6C  20  65  76  69  74  63  61  00223
                    25  37  31  21  20  2C  43  41  21  20  00232
                                                        44  0023C         .ASCII  \D\                                             ;
69  2D  6E  6F  6E  20  74  73  61  4C  20  20  20  20  2C  0023D P.ABW:  .ASCII  \,    Last non-interactive login on !AC, \      ;
69  67  6F  6C  20  65  76  69  74  63  61  72  65  74  6E  0024C
                    20  2C  43  41  21  20  6E  6F  20  6E  0025B
                                        44  25  37  31  21  00265         .ASCII  \!17%D\                                         ;
72  75  6C  69  61  66  20  57  55  21  09  07  07  07  2E  0026A P.ABX:  .ASCII  \.\<7><7><7><9>\!UW failure!%S\                 ;
                                            53  25  21  65  00279
63  75  73  20  74  73  61  6C  20  65  63  6E  69  73  20  0027D         .ASCII  \ since last successful login\                  ;
        6E  69  67  6F  6C  20  6C  75  66  73  73  65  63  0028C
61  68  20  75  6F  59  20  20  20  20  09  07  0A  0D  2B  00299 P.ABY:  .ASCII  \+\<13><10><7><9>\    You have !\               ;
                                            21  20  65  76  002A8
73  65  6D  20  6C  69  61  4D  20  77  65  6E  20  57  55  002AC         .ASCII  \UW new Mail message!%S.\<13><10>               ;
                    0A  0D  2E  53  25  21  65  67  61  73  002BB
                                                            002C5         .BLKB   3                                               ;
76  61  68  20  75  6F  59  20  20  20  20  09  07  0A  0D  002C8 P.ACA:  .ASCII  <13><10><7><9>\     You have new\               ;
                                        77  65  6E  20  65  002D7
2E  73  65  67  61  73  73  65  6D  20  6C  69  61  4D  20  002DC         .ASCII  \ Mail messages.\<13><10><0><0><0>              ;
                                        00  00  00  0A  0D  002EB
                                                  010E0025  002F0 P.ABZ:  .LONG   17694757                                        ;
                                                  00000000' 002F4         .ADDRESS P.ACA                                          ;

                                                                          .EXTRN  SYS$GQ_VERSION

                                                                          .PSECT  $CODE$,NOWRT,2

                                                      07FC  00000         .ENTRY  ANNOUNCE, Save R2,R3,R4,R5,R6,R7,R8,R9,R10      ; 1404
                                       5A 00000000G   00  9E  00002         MOVAB   WRITE_OUTPUT, R10                               ;
                                       59 00000000G   00  9E  00009         MOVAB   UAF_RECORD, R9                                  ;
                                       58 00000000G   00  9E  00010         MOVAB   WRITE_FAO, R8                                   ;
                    57       0000'  CF  9E 00017         MOVAB   P.ABM, R7                              :
                    5E       FF64   CE  9E 0001C         MOVAB   -156(SP), SP                           :
                    50              69  D0 00021         MOVL    UAF_RECORD, R0                         : 1434
                                    01  12 00024         BNEQ    1$                                     :
                                    04  00026            RET                                            :
         78    01D4 C0              05  E0 00027 1$:     BBS     #5, 468(R0), 3$                        : 1437
                                    57  DD 0002D         PUSHL   R7                                     : 1440
                   FF4C  CF         01  FB 0002F         CALLS   #1, WRITE_ANNOUNCEMENT                 :
                         6E         50  E8 00034         BLBS    R0, 3$                                 :
   0C    AE         08   A7         1C  28 00037         MOVC3   #28, P.ABO, MSG_BUFFER                 : 1443
                         83 00000000G   00  D0 0003D     MOVL    SYS$GQ_VERSION, (PTR)+                 : 1445
                         50       0C    AE  9E 00044     MOVAB   MSG_BUFFER, R0                         : 1446
         56              53             50  C3 00048     SUBL3   R0, PTR, LENGTH                        :
         63         24   A7             09  28 0004C     MOVC3   #9, P.ABP, (PTR)                       : 1448
                    04   AE             10  D0 00051     MOVL    #16, BUFDESC                           : 1450
                    F0   AD 00020010    8F  D0 00055     MOVL    #131088, TRNLNM_ITEM_LIST              :
                    08   AE             53  D0 0005D     MOVL    PTR, BUFDESC+4                         : 1452
                    F4   AD             53  D0 00061     MOVL    PTR, TRNLNM_ITEM_LIST+4                :
                    F8   AD        04   AE  9E 00065     MOVAB   BUFDESC, TRNLNM_ITEM_LIST+8            : 1453
                                   FC   AD  D4 0006A     CLRL    TRNLNM_ITEM_LIST+12                    : 1454
                                   F0   AD  9F 0006D     PUSHAB  TRNLNM_ITEM_LIST                       : 1459
                                   58   A7  9F 00070     PUSHAB  P.ABU                                  :
                                   50   A7  9F 00073     PUSHAB  P.ABS                                  :
                                   40   A7  9F 00076     PUSHAB  P.ABQ                                  :
                                        7E  D4 00079     CLRL    -(SP)                                  :
               00000000G 00             05  FB 0007B     CALLS   #5, SYS$TRNLNM                         :
                         01             50  D1 00082     CMPL    R0, #1                                 : 1460
                                        0F  12 00085     BNEQ    2$                                     :
                         53        04   AE  C0 00087     ADDL2   BUFDESC, R3                            : 1464
                         50        0C   AE  9E 0008B     MOVAB   MSG_BUFFER, R0                         :
                         53             50  C2 0008F     SUBL2   R0, R3                                 :
                         56        FE   A3  9E 00092     MOVAB   -2(R3), LENGTH                         :
                    04   AE             56  D0 00096 2$: MOVL    LENGTH, BUFDESC                        : 1467
                    08   AE        0C   AE  9E 0009A     MOVAB   MSG_BUFFER, BUFDESC+4                  : 1468
                                   04   AE  9F 0009F     PUSHAB  BUFDESC                                : 1470
                         6A             01  FB 000A2     CALLS   #1, WRITE_OUTPUT                       :
                         50             69  D0 000A5 3$: MOVL    UAF_RECORD, R0                         : 1477
         52    01D5      C0             04  E0 000A8     BBS     #4, 469(R0), 8$                        :
                         51      018C   C0  9E 000AE     MOVAB   396(R0), R1                            : 1482
                         52      0194   C0  9E 000B3     MOVAB   404(R0), R2                            : 1483
                                        61  D5 000B8     TSTL    (R1)                                   : 1485
                                        05  12 000BA     BNEQ    4$                                     :
                                   04   A1  D5 000BC     TSTL    4(R1)                                  :
                                        11  13 000BF     BEQL    5$                                     :
                                        51  DD 000C1 4$: PUSHL   R1                                     : 1487
                                        51  DD 000C3     PUSHL   R1                                     : 1489
                    0000V  CF           01  FB 000C5     CALLS   #1, ASCIC_DAY_OF_WEEK                  :
                                        50  DD 000CA     PUSHL   R0                                     :
                                   5C   A7  9F 000CC     PUSHAB  P.ABV                                  : 1487
                         68             03  FB 000CF     CALLS   #3, WRITE_FAO                          :
                                        62  D5 000D2 5$: TSTL    (R2)                                   : 1491
                                        05  12 000D4     BNEQ    6$                                     :
                                   04   A2  D5 000D6     TSTL    4(R2)                                  :
                                        12  13 000D9     BEQL    7$                                     :
                                        52  DD 000DB 6$: PUSHL   R2                                     : 1493
                                        52  DD 000DD     PUSHL   R2                                     : 1495
                    0000V  CF           01  FB 000DF     CALLS   #1, ASCIC_DAY_OF_WEEK                  :
```

```
INTERACT                                        B 10
V04-000                                16-Sep-1984 01:55:50    VAX-11 Bliss-32 V4.0-742              Page  40
                                       14-Sep-1984 12:41:07    [LOGIN.SRC]INTERACT.B32;1                   (8)

                                   50  DD 000E4         PUSHL   R0                                        ;
                           0085    C7  9F 000E6         PUSHAB  P.ABW                                     ; 1493
                     68            03  FB 000EA         CALLS   #3, WRITE_FAO                             ;
                     50            69  D0 000ED 7$:     MOVL    UAF_RECORD, R0                            ; 1497
                     50    0164    C0  3C 000F0         MOVZWL  356(R0), R0                               ;
                                   09  13 000F5         BEQL    8$                                        ;
                                   50  DD 000F7         PUSHL   R0                                        ; 1501
                           00B2    C7  9F 000F9         PUSHAB  P.ABX                                     ; 1498
                     68            02  FB 000FD         CALLS   #2, WRITE_FAO                             ;
                     50            69  D0 00100 8$:     MOVL    UAF_RECORD, R0                            ; 1508
         27          01D4          C0  06  E0 00103     BBS     #6, 468(R0), 10$                          ;
                                   50  DD 00109         PUSHL   R0                                        ; 1509
                             04    AE  9F 0010B         PUSHAB  MSGCOUNT                                  ;
             00000000G     00      02  FB 0010E         CALLS   #2, MAIL$GET_NEW_COUNT                    ;
                     18            50  E9 00115         BLBC    R0, 10$                                   ;
                     50            6E  D0 00118         MOVL    MSGCOUNT, R0                              ; 1511
                                   0A  15 0011B         BLEQ    9$                                        ;
                                   50  DD 0011D         PUSHL   R0                                        ; 1515
                           00E1    C7  9F 0011F         PUSHAB  P.ABY                                     ; 1512
                     68            02  FB 00123         CALLS   #2, WRITE_FAO                             ;
                                       04 00126         RET                                               ;
                                   07  18 00127 9$:     BGEQ    10$                                       ; 1516
                           0138    C7  9F 00129         PUSHAB  P.ABZ                                     ; 1520
                     6A            01  FB 0012D         CALLS   #1, WRITE_OUTPUT                          ;
                                       04 00130 10$:    RET                                               ; 1522

; Routine Size:  305 bytes,    Routine Base:  $CODE$ + 079B
; 1131          1523  1 ROUTINE zero_password : NOVALUE =
; 1132          1524  1
; 1133          1525  1 !---
; 1134          1526  1 !
; 1135          1527  1 !       Zero out the password in the RMS buffer.  This must be done in
; 1136          1528  1 !       executive mode, since the RMS buffer is not user-writeable.
; 1137          1529  1 !
; 1138          1530  1 ! Inputs:
; 1139          1531  1 !
; 1140          1532  1 !       Access mode is EXEC.
; 1141          1533  1 !
; 1142          1534  1 !       AP = ISI of RAB which read the password
; 1143          1535  1 !
; 1144          1536  1 ! Outputs:
; 1145          1537  1 !
; 1146          1538  1 !       None.  Password is zeroed.
; 1147          1539  1 !
; 1148          1540  1 !---
; 1149          1541  1
; 1150          1542  2 BEGIN
; 1151          1543  2
; 1152          1544  2 BUILTIN
; 1153          1545  2     AP;
; 1154          1546  2
; 1155          1547  2 LOCAL
; 1156          1548  2     isi;
; 1157          1549  2
; 1158          1550  2 isi = .AP;                                 ! Fetch ISI of RAB reading password
; 1159          1551  2
; 1160          1552  2 ! ************************************************
; 1161          1553  2 ! ***** This routine currently does nothing! *****
; 1162          1554  2 ! ***** It should find the RMS internal      *****
; 1163          1555  2 ! ***** structures to zero out the password  *****
; 1164          1556  2 ! ***** which was read using this ISI.       *****
; 1165          1557  2 ! ************************************************
; 1166          1558  2
; 1167          1559  1 END;



                                         0000 00000 ZERO_PASSWORD:
                                                          .WORD   Save nothing                                  ; 1523
                             50             5C  D0 00002  MOVL    AP, ISI                                       ; 1550
                                                04 00005  RET                                                   ; 1559

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 08CC
```

```
  1 ROUTINE get_syspwd : NOVALUE =
  2 BEGIN
  2
  2 !+++
  2 !
  2 ! Get and validate the system password, if necessary.
  2 !
  2 ! Inputs:
  2 !       None.
  2 !
  2 ! Outputs:
  2 !       None.
  2 !
  2 !---
  2
  2 BUILTIN
  2     cmpm;
  2
  2 LABEL
  2     read_password,
  2     got_channel;
  2
  2 LOCAL
  2     status,
  2     term_char : VECTOR[2],
  2     save_char : VECTOR[2],
  2     channel : WORD,
  2     iosb : VECTOR[2],
  2     buffer : VECTOR[80],
  2     timeout : VECTOR[2],
  2     desc : VECTOR[2],
  2     enc_desc : VECTOR[2],
  2     enc_pwd : VECTOR[2],
  2     uaf_record : BLOCK[UAF$K_LENGTH, byte],
  2     uaf_desc : BLOCK[2] INITIAL(UAF$K_LENGTH, uaf_record);
  2
  2 !
  2 ! If the system password timeout period is zero, then forget it.
  2 !
  2 IF .sys$gb_pwd_tmo EQL 0
  2 THEN RETURN;
  2
  2 !
  2 ! If SYS$INPUT is not a syspassword terminal, or is remote, then return.
  2 !
  2 IF NOT .terminal_device                               ! If not a real terminal
  2 OR NOT .dev_dep_2[tt2$v_syspwd]                       ! or not password_required,
  2 THEN RETURN;                                          ! then just forget it.
  2
  2 !
  2 ! Set the terminal /NOBROADCAST to prevent broadcast messages while
  2 ! receiving the system password. This is essential to preserving the
  2 ! illusion of a dead line.
  2 !
  3 got_channel: BEGIN
P 3 IF NOT $ASSIGN (chan = channel,
  4                 devnam = term_name)
  3 THEN RETURN;
  3
  4 read_password: BEGIN
P 4 IF NOT $QIOW (func=io$_sensemode,
P 4               chan=.channel,
P 4               iosb=iosb,
P 4               p1  =term_char
  5               )
  4 THEN LEAVE got_channel;
  4 IF NOT .iosb THEN LEAVE got_channel;
  4 save_char[0] = .term_char[0];
  4 save_char[1] = .term_char[1];
  4 BBLOCK [term_char[1], TT$V_NOBRDCST] = 1;
P 4 IF NOT $QIOW (func=io$_setmode,
P 4               chan=.channel,
P 4               iosb=iosb,
P 4               p1  =term_char
  5               )
  4 THEN LEAVE read_password;
  4 IF NOT .iosb THEN LEAVE read_password;
  4
  4 !
  4 ! Open the SYSUAF.DAT
  4 !
  4 status = lgi$searchuser(%ASCID'<System+Password>', 0,
  4                         uaf_desc, uaf_fab, uaf_rab, 0);
  4 IF (NOT .status) AND (.status NEQ -2) THEN
  4     LEAVE read_password;
  4
  4 !
  4 ! If no system password, then simply return.
  4 !
  4 IF CMPM(2, uaf_record[uaf$q_pwd], UPLIT(0,0)) EQL 0
  4 THEN LEAVE read_password;
  4
  4 !
  4 ! If here, then SYS$INPUT is a syspwd terminal, and there is a non-null
  4 ! system password.  Set up the input rab to do read-no-echo, and set up a
  4 ! timer, so that we know when to stop trying.
  4 !
  4 input_rab[rab$v_pmt] = 0;                   ! Read with no prompt
  4 input_rab[rab$v_rne] = 1;                   ! Read no echo
  4 input_rab[rab$v_pta] = 0;                   ! Don't purge type-ahead
  4 input_rab[rab$l_ubf] = buffer;              ! Put password here
  4 input_rab[rab$w_usz] = %ALLOCATION(buffer);
  4
  4 enc_desc[0] = %ALLOCATION(enc_pwd);
  4 enc_desc[1] = enc_pwd;
  4
  4 timeout[0] = -10*1000*1000 * .sys$gb_pwd_tmo;
  4 timeout[1] = -1;
  4
  4 ppd[ppd$l_lststatus] = lgi$_syspwdtmo;  ! Set final status = bad sys pwd
  4
P 4 $SETIMR(DAYTIM = timeout,
P 4         ASTADR = write_timeout,
  4         REQIDT = 98);
```

```
; 1283          1674  4
; 1284          1675  4  WHILE true DO
; 1285          1676  5       BEGIN
; 1286          1677  5       get_input (input_rab, 1);
; 1287          1678  5       desc[0] = .input_rab[rab$w_rsz];
; 1288          1679  5       desc[1] = .input_rab[rab$l_rbf];
; 1289          1680  5       lgi$hpwd(enc_desc,                          ! Put smashed pwd here,
; 1290          1681  5                desc,                              ! get password from here
; 1291          1682  5                uaf$c_purdy_v,                     ! using the Purdy algorithm
; 1292          1683  5                .uaf_record[uaf$w_salt],
; 1293          1684  5                %ASCID'<System+Password>' );
; 1294          1685  5       IF CMPM(2,
; 1295          1686  5               uaf_record[uaf$q_pwd],
; 1296          1687  5               enc_pwd) EQL 0
; 1297          1688  5       THEN
; 1298          1689  6           BEGIN
; 1299          1690  6           $CANTIM(REQIDT = 98);
; 1300          1691  6           ppd[ppd$l_lststatus] = 1;
; 1301          1692  6           LEAVE read_password;
; 1302          1693  5           END;
; 1303          1694  4       END;
; 1304          1695  3  END;                                            ! End of block read_password
; 1305          1696  3
; 1306        P 1697  3  $QIOW (func=io$_setmode,
; 1307        P 1698  3         chan=.channel,
; 1308        P 1699  3         iosb=iosb,
; 1309        P 1700  3         p1  =save_char
; 1310          1701  3         );
; 1311          1702  2  END;                                            ! End of block got_channel
; 1312          1703  2
; 1313          1704  2  $DASSGN (chan = .channel);
; 1314          1705  2
; 1315          1706  1  END;



                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

72  6F  77  73  73  61  50  2B  6D  65  74  73  79  53  3C  002F8 P.ACC:  .ASCII  \<System+Password>\<0><0><0>        ;
                                            00  00  00  3E  64  00307                                                ;
                                                  010E0011  0030C P.ACB:  .LONG   17694737                            ;
                                                  00000000' 00310         .ADDRESS P.ACC                              ;
                                        00000000  00000000  00314 P.ACD:  .LONG   0, 0                                ;
72  6F  77  73  73  61  50  2B  6D  65  74  73  79  53  3C  0031C P.ACF:  .ASCII  \<System+Password>\<0><0><0>        ;
                                            00  00  00  3E  64  0032B                                                ;
                                                  010E0011  00330 P.ACE:  .LONG   17694737                            ;
                                                  00000000' 00334         .ADDRESS P.ACF                              ;

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

                                                                          .PSECT  $CODE$,NOWRT,2

                                                        003C 00000 GET_SYSPWD:
                                                                          .WORD   Save R2,R3,R4,R5                    ; 1560
                                      55 00000000G  00  9E 00002          MOVAB   SYS$GB_PWD_TMO, R5                  ;
```

```
              54 00000000G   00   9E 00009          MOVAB   PPD+24, R4                       ;
              53 00000000G   00   9E 00010          MOVAB   SYS$QIOW, R3                     ;
              52 00000000G   00   9E 00017          MOVAB   INPUT_RAB+4, R2                  ;
              5E      F8F8   CE   9E 0001E          MOVAB   -1800(SP), SP                    ;
        04    AE      0584   8F   3C 00023          MOVZWL  #1412, UAF_DESC                  ; 1561
        08    AE        0C   AE   9E 00029          MOVAB   UAF_RECORD, UAF_DESC+4           ;
                              65   95 0002E          TSTB    SYS$GB_PWD_TMO                   ; 1599
                              01   12 00030          BNEQ    1$                               ;
                                   04 00032          RET                                      ;
              01 00000000G   00   E8 00033 1$:      BLBS    TERMINAL_DEVICE, 2$              ; 1605
                                   04 0003A          RET                                      ;
  01 00000000G   00           03   E0 0003B 2$:      BBS     #3, DEV_DEP_2+2, 3$              ; 1606
                                   04 00043          RET                                      ;
                              7E   7C 00044 3$:      CLRQ    -(SP)                            ; 1616
                        08   AE   9F 00046          PUSHAB  CHANNEL                          ;
                 00000000G   00   9F 00049          PUSHAB  TERM_NAME                        ;
     00000000G   00           04   FB 0004F          CALLS   #4, SYS$ASSIGN                   ;
                 01           50   E8 00056          BLBS    R0, 4$                           ;
                                   04 00059          RET                                      ;
                              7E   7C 0005A 4$:      CLRQ    -(SP)                            ; 1624
                              7E   7C 0005C          CLRQ    -(SP)                            ;
                              7E   D4 0005E          CLRL    -(SP)                            ;
                        F8   AD   9F 00060          PUSHAB  TERM_CHAR                        ;
                              7E   7C 00063          CLRQ    -(SP)                            ;
                        E8   AD   9F 00065          PUSHAB  IOSB                             ;
                              27   DD 00068          PUSHL   #39                              ;
              7E        28   AE   3C 0006A          MOVZWL  CHANNEL, -(SP)                   ;
                              7E   D4 0006E          CLRL    -(SP)                            ;
              63              0C   FB 00070          CALLS   #12, SYS$QIOW                    ;
              03              50   E8 00073          BLBS    R0, 6$                           ;
                            014C   31 00076 5$:      BRW     18$                              ;
              F9        E8   AD   E9 00079 6$:      BLBC    IOSB, 5$                         ; 1626
        F0    AD        F8   AD   7D 0007D          MOVQ    TERM_CHAR, SAVE_CHAR             ; 1627
        FE    AD              02   88 00082          BISB2   #2, TERM_CHAR+6                  ; 1629
                              7E   7C 00086          CLRQ    -(SP)                            ; 1634
                              7E   7C 00088          CLRQ    -(SP)                            ;
                              7E   D4 0008A          CLRL    -(SP)                            ;
                        F8   AD   9F 0008C          PUSHAB  TERM_CHAR                        ;
                              7E   7C 0008F          CLRQ    -(SP)                            ;
                        E8   AD   9F 00091          PUSHAB  IOSB                             ;
                              23   DD 00094          PUSHL   #35                              ;
              7E        28   AE   3C 00096          MOVZWL  CHANNEL, -(SP)                   ;
                              7E   D4 0009A          CLRL    -(SP)                            ;
              63              0C   FB 0009C          CALLS   #12, SYS$QIOW                    ;
              03              50   E8 0009F          BLBS    R0, 8$                           ;
                            0107   31 000A2 7$:      BRW     17$                              ;
              F9        E8   AD   E9 000A5 8$:      BLBC    IOSB, 7$                         ; 1636
                              7E   D4 000A9          CLRL    -(SP)                            ; 1641
                 00000000G   00   9F 000AB          PUSHAB  UAF_RAB                          ;
                 00000000G   00   9F 000B1          PUSHAB  UAF_FAB                          ;
                        10   AE   9F 000B7          PUSHAB  UAF_DESC                         ;
                              7E   D4 000BA          CLRL    -(SP)                            ;
                     0000'   CF   9F 000BC          PUSHAB  P.ACB                            ;
     00000000G   00           06   FB 000C0          CALLS   #6, LGI$SEARCHUSER               ;
                 09           50   E8 000C7          BLBS    STATUS, 9$                       ; 1643
      FFFFFFFE   8F           50   D1 000CA          CMPL    STATUS, #-2                      ;
                              CF   12 000D1          BNEQ    7$                               ;
                 50              01  CE 000D3 9$:    MNEGL   #1, R0                                   ; 1649
          0000'  CF        0164  CE  D1 000D6        CMPL    UAF_RECORD+340, P.ACD+4                  ;
                                 11  19 000DD        BLSS    12$                                      ;
                                 0B  14 000DF        BGTR    10$                                      ;
          0000'  CF        0160  CE  D1 000E1        CMPL    UAF_RECORD+340, P.ACD                    ;
                                 04  13 000E8        BEQL    11$                                      ;
                                 04  1F 000EA        BLSSU   12$                                      ;
                                 50  D6 000EC 10$:   INCL    R0                                       ;
                                 50  D6 000EE 11$:   INCL    R0                                       ;
                                 50  D5 000F0 12$:   TSTL    R0                                       ;
                                 AE  13 000F2        BEQL    7$                                       ;
                 03  A2          40  8F  8A 000F4        BICB2   #64, INPUT_RAB+7                         ; 1657
                 03  A2              01  88 000F9        BISB2   #1, INPUT_RAB+7                          ; 1658
                 03  A2              20  8A 000FD        BICB2   #32, INPUT_RAB+7                         ; 1659
                 20  A2        FEA8  CD  9E 00101        MOVAB   BUFFER, INPUT_RAB+36                     ; 1660
                 1C  A2        0140  8F  B0 00107        MOVW    #320, INPUT_RAB+32                       ; 1661
               FE90  CD              08  D0 0010D        MOVL    #8, ENC_DESC                             ; 1663
               FE94  CD        FE88  CD  9E 00112        MOVAB   ENC_PWD, ENC_DESC+4                      ; 1664
                 50              65  9A 00119        MOVZBL  SYS$GB_PWD_TMO, R0                       ; 1666
   FEA0   CD     50  FF676980    8F  C5 0011C        MULL3   #-10000000, R0, TIMEOUT                  ;
               FEA4  CD              01  CE 00126        MNEGL   #1, TIMEOUT+4                            ; 1667
                 64  00000000G   8F  D0 0012B        MOVL    #LGI$_SYSPWDTMO, PPD+24                  ; 1669
                 7E              62  8F  9A 00132        MOVZBL  #98, -(SP)                               ; 1673
                     00000000G   00  9F 00136        PUSHAB  WRITE_TIMEOUT                            ;
                          FEA0   CD  9F 0013C        PUSHAB  TIMEOUT                                  ;
                                 7E  D4 00140        CLRL    -(SP)                                    ;
      00000000G  00              04  FB 00142        CALLS   #4, SYS$SETIMR                           ;
                                 01  DD 00149 13$:   PUSHL   #1                                       ; 1677
                            FC   A2  9F 0014B        PUSHAB  INPUT_RAB                                ;
      00000000G  00              02  FB 0014E        CALLS   #2, GET_INPUT                            ;
               FE98  CD          1E  A2  3C 00155        MOVZWL  INPUT_RAB+34, DESC                       ; 1678
               FE9C  CD          24  A2  D0 0015B        MOVL    INPUT_RAB+40, DESC+4                     ; 1679
                          0000'  CF  9F 00161        PUSHAB  P.ACE                                    ; 1683
                 7E        0176  CE  3C 00165        MOVZWL  UAF_RECORD+358, -(SP)                    ;
                                 02  DD 0016A        PUSHL   #2                                       ; 1680
                          FE98   CD  9F 0016C        PUSHAB  DESC                                     ;
                          FE90   CD  9F 00170        PUSHAB  ENC_DESC                                 ;
      00000000G  00              05  FB 00174        CALLS   #5, LGI$HPWD                             ;
                 50              01  CE 0017B        MNEGL   #1, R0                                   ; 1686
          FE8C   CD        0164  CE  D1 0017E        CMPL    UAF_RECORD+340, ENC_PWD                  ;
                                 11  19 00185        BLSS    16$                                      ;
                                 0B  14 00187        BGTR    14$                                      ;
          FE88   CD        0160  CE  D1 00189        CMPL    UAF_RECORD+340, ENC_PWD                  ;
                                 04  13 00190        BEQL    15$                                      ;
                                 04  1F 00192        BLSSU   16$                                      ;
                                 50  D6 00194 14$:   INCL    R0                                       ;
                                 50  D6 00196 15$:   INCL    R0                                       ;
                                 50  D5 00198 16$:   TSTL    R0                                       ; 1687
                                 AD  12 0019A        BNEQ    13$                                      ;
                                 7E  D4 0019C        CLRL    -(SP)                                    ; 1690
                 7E              62  8F  9A 0019E        MOVZBL  #98, -(SP)                               ;
      00000000G  00              02  FB 001A2        CALLS   #2, SYS$CANTIM                           ;
                 64              01  D0 001A9        MOVL    #1, PPD+24                               ; 1691
                                 7E  7C 001AC 17$:   CLRQ    -(SP)                                    ; 1701
                                 7E  7C 001AE        CLRQ    -(SP)                                    ;
                                 7E  D4 001B0        CLRL    -(SP)                                    ;
                            F0   AD  9F 001B2        PUSHAB  SAVE_CHAR                                ;
```

```
                                7E  7C 001B5         CLRQ    -(SP)                              ;
                        E8      AD  9F 001B7         PUSHAB  IOSB                               ;
                                23  DD 001BA         PUSHL   #35                                ;
           7E           28      AE  3C 001BC         MOVZWL  CHANNEL, -(SP)                     ;
                                7E  D4 001C0         CLRL    -(SP)                              ;
           63                   0C  FB 001C2         CALLS   #12, SYS$QIOW                      ;
           7E                   6E  3C 001C5  18$:   MOVZWL  CHANNEL, -(SP)                     ; 1704
 00000000G 00                   01  FB 001C8         CALLS   #1, SYS$DASSGN                     ;
                                    04 001CF         RET                                        ; 1706

; Routine Size:  464 bytes,    Routine Base:  $CODE$ + 08D2
```

```
  1 GLOBAL ROUTINE check_connection: NOVALUE =
  1
  1 !---
  1 !
  1 !       Check for disconnected processes under this username and
  1 !       attempt a (re-)connection to one of them if possible.
  1 !
  1 ! Inputs:
  1 !
  1 !       None
  1 !
  1 ! Outputs:
  1 !
  1 !       This process exits if a (re-)connection is made.
  1 !
  1 !---
  1
  2 BEGIN
  2
  2 IF NOT .connect_check                                        ! If no checking
  2 OR NOT .terminal_device                                      !  or not a real terminal
  2 OR .dev_char_2[dev$v_rtt]                                    !  or terminal is remote
  2 THEN RETURN;                                                 !   then don't check...
  2
  2 IF .connect_name[0] EQL 0                                    ! If not specific
  2 AND .uaf_record NEQ 0                                        !  and UAF record exists
  2 THEN
  3 BEGIN                                                        ! Find 'connect to' process...
  3
  3 LITERAL
  3     num_pids = 16;                                           ! Save up to this many PIDs
  3
  3 LOCAL
  3     num_disconnected,                                        ! Number of disconnecteds
  3     index,                                                   ! Saved PID index
  3     pid_list : VECTOR[num_pids],                             ! Saved PID list
  3     pid_context,                                             ! GETJPI PID context
  3     iosb : VECTOR[4,WORD],                                   ! GETJPI I/O status block
  3     found_username : VECTOR[uaf$s_username,BYTE],  ! Found username
  3     found_terminal : VECTOR[1+15,BYTE],                      ! Found terminal
  3     first_terminal : VECTOR[1+15,BYTE],                      ! First found terminal
  3     found_uic,                                               ! Found UIC
  3     found_pid,                                               ! Found PID
  3     found_procname : VECTOR[16,BYTE],                        ! Found process name
  3     found_imagname : VECTOR[64,BYTE],                        ! Found image name
  3     found_username_len,                                      ! Found username length
  3     found_terminal_desc : VECTOR[2],                         ! Found terminal descriptor
  3     first_terminal_length,                                   ! First found terminal length
  3     found_procname_len,                                      ! Found process name length
  3     found_imagname_len,                                      ! Found image name length
  3     found_devchar2 : $BBLOCK[4],                             ! Found terminal's DEVCHAR2
  3     prompt_buffer : VECTOR [64, BYTE],                       ! Buffer for prompt string
  3     getjpi_item_list : $ITMLST_DECL(ITEMS = 4),              ! GETJPI item list
  3     getdvi_item_list : $ITMLST_DECL(ITEMS = 1);              ! GETDVI item list
  3
P 3 $ITMLST_INIT(ITMLST = getjpi_item_list,                      ! Set up GETJPI item list
P 3             (ITMCOD = jpi$_username,
P 3              BUFSIZ = uaf$s_username,
P 3              BUFADR = found_username,
P 3              RETLEN = found_username_len),
P 3             (ITMCOD = jpi$_terminal,
P 3              BUFSIZ = 15,
P 3              BUFADR = found_terminal[1],
P 3              RETLEN = found_terminal_desc[0]),
P 3             (ITMCOD = jpi$_uic,
P 3              BUFSIZ = 4,
P 3              BUFADR = found_uic),
P 3             (ITMCOD = jpi$_pid,
P 3              BUFSIZ = 4,
  3              BUFADR = found_pid));
  3 found_username_len = 0;
  3 found_terminal_desc[0] = 0;
  3 found_terminal_desc[1] = found_terminal;
  3 found_terminal[0] = '_';
  3 found_uic = 0;
  3 found_pid = 0;
  3
P 3 $ITMLST_INIT(ITMLST = getdvi_item_list,          ! Set up GETDVI item list
P 3             (ITMCOD = dvi$_devchar2,
P 3              BUFSIZ = 4,
  3              BUFADR = found_devchar2));
  3
  3 index = 0;                                       ! Start index at zero
  3 pid_context = -1;                                ! Start the wild card PID
  3
  3 WHILE true DO                                    ! For all processes...
  4     BEGIN
  5     (
  5      LOCAL
  5         status;                                  ! Get job information
P 6      IF NOT (status = $GETJPIW(PIDADR = pid_context,
P 6                                ITMLST = getjpi_item_list,
  6                                IOSB   = iosb))
  5      THEN iosb[0] = .status;
  4     );
  4     IF .iosb[0] EQL SS$_NOMOREPROC               ! Quit if no more processes
  4     THEN EXITLOOP;
  4     IF .iosb[0]                                  ! If found a process
  4     AND .found_terminal_desc[0] NEQ 0            !  and it's interactive
  4     THEN
  4         IF CH$EQL(.found_username_len,found_username, ! If username matches
  4                   uaf$s_username,uaf_record[uaf$t_username],
  4                   ' ')
  4         AND .found_uic EQL .uaf_record[uaf$l_uic] !  and UIC matches
  4         THEN
  5             BEGIN                                ! Get terminal's info
  5             found_terminal_desc[0] = .found_terminal_desc[0] + 1;
P 5             IF $GETDVIW(DEVNAM = found_terminal_desc,
  6                         ITMLST = getdvi_item_list)
  5             AND .found_devchar2[dev$v_det]       ! If disconnected
  5             THEN
  6                 BEGIN
  6                 pid_list[.index] = .found_pid;   ! Save the found PID
  6                 index = .index + 1;              !  and count it as saved
  6                 IF .index GEQU num_pids             ! Quit if up to our limit
  6                     THEN EXITLOOP;
  5                     END;
  4             END;
  3     END;
  3
  3 IF .index EQL 0                                         ! Found nothing disconnected...
  3 THEN RETURN;
  3
P 3 $ITMLST_INIT(ITMLST = getjpi_item_list,                 ! Set up GETJPI item list again
P 3             (ITMCOD = jpi$_terminal,
P 3              BUFSIZ = 15,
P 3              BUFADR = found_terminal[1],
P 3              RETLEN = found_terminal_desc[0]),
P 3             (ITMCOD = jpi$_prcnam,
P 3              BUFSIZ = 16,
P 3              BUFADR = found_procname,
P 3              RETLEN = found_procname_len),
P 3             (ITMCOD = jpi$_imagname,
P 3              BUFSIZ = 64,
P 3              BUFADR = found_imagname,
  3              RETLEN = found_imagname_len));
  3 found_procname_len = 0;
  3 found_imagname_len = 0;
  3
  3 num_disconnected = 0;                                   ! Zero disconnected(s) counter
  3 INCR i FROM 0 TO .index-1 DO                            ! List disconnected(s)
P 3     IF $GETJPIW(PIDADR = pid_list[.i],
P 3                 ITMLST = getjpi_item_list,
  4                 IOSB   = iosb)
  3     AND .iosb[0]
  3     THEN
  4         BEGIN
  4         IF .num_disconnected EQL 0
  4         THEN
  5             BEGIN
  5             first_terminal_length = .found_terminal_desc[0];
  5             ch$move (16, found_terminal, first_terminal);
  6             write_output((IF .index EQL 1
  6                 THEN
  6                 %ASCID '    You have the following disconnected process:'
  6                 ELSE
  5                 %ASCID '    You have the following disconnected processes:'));
  5             write_output(%ASCID 'Terminal   Process name    Image name');
  4             END;
  4         write_fao(UPLIT BYTE (%ASCIC '!10AF !15AF !AF'),
  4                 .found_terminal_desc[0],
  4                 found_terminal[1],
  4                 .found_procname_len,
  4                 found_procname,
  5                 (IF .found_imagname_len EQL 0
  5                  THEN 1+4+1
  4                  ELSE .found_imagname_len),
  5                 (IF .found_imagname_len EQL 0
  5                  THEN UPLIT BYTE ('(none)')
  4                  ELSE found_imagname));
  4         num_disconnected = .num_disconnected+1;
```

```
        END;

IF .num_disconnected GTR 0                      ! If we listed anything
THEN
    BEGIN

    input_rab[rab$w_usz] = %ALLOCATION(connect_name_buffer);
    input_rab[rab$l_ubf] = connect_name_buffer;
    input_rab[rab$v_pta] = 0;                   ! Don't purge typeahead
    input_rab[rab$v_rne] = 0;                   ! Echo input
    input_rab[rab$b_tmo] = .sys$gb_retry_tmo;   ! Standard timeout period
    input_rab[rab$v_pmt] = 1;                   ! Set up for prompt

    IF .num_disconnected EQL 1                  ! If only 1 process listed
    THEN
        BEGIN
        input_rab[rab$b_psz] = 41;
        input_rab[rab$l_pbf] =
            UPLIT BYTE (cr,lf,'Connect to above listed process [YES]: ');
        END
    ELSE
        BEGIN
        CH$COPY (32, UPLIT BYTE (cr,lf,'Enter terminal to connect to ['),
                  .first_terminal_length, first_terminal[1],
                 3, UPLIT BYTE ('): '),
                 ' ', %ALLOCATION (prompt_buffer), prompt_buffer);
        input_rab[rab$b_psz] = 32 + 3 + .first_terminal_length;
        input_rab[rab$l_pbf] = prompt_buffer;
        END;


    get_input(input_rab, 2);                    ! Get user's response

    IF NOT .input_rab[rab$l_sts]                ! If any read error (e.g., timeout)
    THEN RETURN;                                !  treat as "NONE"

    connect_name[0] = .input_rab[rab$w_rsz];    ! Set user's response
    connect_name[1] = .input_rab[rab$l_rbf];    !  as connection terminal

    IF .connect_name[0] NEQ 0                   ! If user did respond
    AND CH$EQL (.connect_name[0], .connect_name[1], ! with "NONE"
                 .connect_name[0], uplit byte ('NONE'))
    THEN RETURN;                                !   then no connection

    IF (.connect_name[0] NEQ 0                  ! If user did respond
         AND CH$EQL (.connect_name[0], .connect_name[1], ! with "YES"
                     .connect_name[0], uplit byte ('YES'))
        )
    OR .connect_name[0] EQL 0                   ! If null response
    THEN                                        !  then connect to first term
        BEGIN
        connect_name[0] = .first_terminal_length+1;
        connect_name[1] = first_terminal;
        END;

    END;
```

```
  2 END;                                                    ! ...find 'connect to' process
  2
  2 IF .connect_name[0] EQL 0                               ! If no name
  2 THEN RETURN;                                            !  then just exit...
  2
  3 BEGIN                                                   ! Connect to terminal...
  3
  3 LOCAL
  3     prev_uic,                                           ! Previous UIC
  3     chan : WORD,                                        ! Connection channel
  3     iosb : VECTOR[4,WORD];                              ! Connection I/O status block
  3
  3 write_fao(UPLIT BYTE (%ASCIC 'Connecting to terminal !AS'),connect_name);
  3
  3 prev_uic = 0;                                           ! No UIC to restore initially
  3 IF .uaf_record NEQ 0                                    ! If UAF record exists,
  3 THEN
P 3     prev_uic = $CMKRNL(ROUTIN = set_uic,                !  set UAF's UIC, save old
  3                        ARGLST = .uaf_record[uaf$l_uic]);
  3
P 4 IF (iosb[0] = $ASSIGN(DEVNAM = term_name,               ! Get a terminal channel
  4                       CHAN   = chan))
  3 THEN
  4     BEGIN
  5     (
  5      LOCAL
  5         status;                                         ! Connect to terminal
P 6      IF NOT (status = $QIOW(CHAN = .chan,
P 6                             FUNC = io$_setmode OR io$m_tt_connect,
P 6                             IOSB = iosb,
  6                             P1   = connect_name))
  5      THEN iosb[0] = .status;
  4     );
  4     $DASSGN(CHAN = .chan);                              ! Free up terminal channel
  3     END;
  3
  3 IF .prev_uic NEQ 0                                      ! If UIC to restore,
  3 THEN
P 3     $CMKRNL(ROUTIN = set_uic,                           !  reset old UIC
  3             ARGLST = .prev_uic);
  3
  3 IF NOT .iosb[0]                                         ! Check for failure
  3 THEN
  4     BEGIN
  4     SIGNAL(lgi$_connerr, 1, connect_name, .iosb[0]); ! Announce the error
  4     RETURN;                                             ! But, continue...
  3     END;
  3
  2 END;                                                    ! ...connect to process
  2
  2 security_audit(nsa$k_rectyp_logi);                      ! Security audit reconnection
  2
  2 $CMEXEC(ROUTIN = exit_process);                         ! Terminate ourselves...
  2
  1 END;
```

```
                                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

68 74 20 65 76 61 68 20 75 6F 59 20 20 20 20 00338 P.ACH:   .ASCII  \    You have the following disconnected \   ;
73 69 64 20 67 6E 69 77 6F 6C 6C 6F 66 20 65 00347                                                           ;
               20 64 65 74 63 65 6E 6E 6F 63 00356                                                           ;
                  3A 73 73 65 63 6F 72 70 00360          .ASCII  \process:\                                  ;
                              010E0030 00368 P.ACG:   .LONG   17694768                                      ;
                              00000000' 0036C          .ADDRESS P.ACH                                        ;
68 74 20 65 76 61 68 20 75 6F 59 20 20 20 20 00370 P.ACJ:   .ASCII  \    You have the following disconnected \   ;
73 69 64 20 67 6E 69 77 6F 6C 6C 6F 66 20 65 0037F                                                           ;
               20 64 65 74 63 65 6E 6E 6F 63 0038E                                                           ;
      00 00 3A 73 65 73 73 65 63 6F 72 70 00398          .ASCII  \processes:\<0><0>                          ;
                              010E0032 003A4 P.ACI:   .LONG   17694770                                      ;
                              00000000' 003A8          .ADDRESS P.ACJ                                        ;
63 6F 72 50 20 20 20 20 6C 61 6E 69 6D 72 65 54 003AC P.ACL:   .ASCII  \Terminal   Process name    Image name\<0>   ;
61 6D 49 20 20 20 20 65 6D 61 6E 20 73 73 65 003BB                                                           ;
                  00 65 6D 61 6E 20 65 67 003CA                                                              ;
                                    00 00 003D2          .ASCII  <0><0>                                      ;
                              010E0025 003D4 P.ACK:   .LONG   17694757                                      ;
                              00000000' 003D8          .ADDRESS P.ACL                                        ;
41 21 20 46 41 35 31 21 20 46 41 30 31 21 0F 003DC P.ACM:   .ASCII  <15>\!10AF !15AF !AF\                    ;
                                          46 003EB                                                           ;
                        29 65 6E 6F 6E 28 003EC P.ACN:   .ASCII  \(none)\                                   ;
                                    0A 0D 003F2 P.ACO:   .BYTE   13, 10                                     ;
76 6F 62 61 20 6F 74 20 74 63 65 6E 6E 6F 43 003F4          .ASCII  \Connect to above listed process [YES]: \  ;
73 65 63 6F 72 70 20 64 65 74 73 69 6C 20 65 00403                                                           ;
               20 3A 5D 53 45 59 5B 20 73 00412                                                              ;
                                    0A 0D 0041B P.ACP:   .BYTE   13, 10                                     ;
20 6C 61 6E 69 6D 72 65 74 20 72 65 74 6E 45 0041D          .ASCII  \Enter terminal to connect to [\            ;
5B 20 6F 74 20 74 63 65 6E 6E 6F 63 20 6F 74 0042C                                                           ;
                                 20 3A 5D 0043B P.ACQ:   .ASCII  \]: \                                      ;
                              45 4E 4F 4E 0043E P.ACR:   .ASCII  \NONE\                                     ;
                                 53 45 59 00442 P.ACS:   .ASCII  \YES\                                      ;
20 6F 74 20 67 6E 69 74 63 65 6E 6E 6F 43 1A 00445 P.ACT:   .ASCII  <26>\Connecting to terminal !AS\          ;
   53 41 21 20 6C 61 6E 69 6D 72 65 74 00454                                                                 ;

                                                        .EXTRN  SYS$GETJPIW, SYS$GETDVIW

                                                        .PSECT  $CODE$,NOWRT,2

                                0FFC 00000              .ENTRY  CHECK_CONNECTION, Save R2,R3,R4,R5,R6,R7,-   ; 1707
                                                                R8,R9,R10,R11                                ;
                 5B      0000'  CF 9E 00002              MOVAB   CONNECT_NAME, R11                            ;
                 5A 00000000G   00 9E 00007              MOVAB   INPUT_RAB+4, R10                             ;
                 5E      FE7C   CE 9E 0000E              MOVAB   -388(SP), SP                                 ;
                 01        08   AB E8 00013              BLBS    CONNECT_CHECK, 1$                            ; 1726
                                  04 00017              RET                                                  ;
                 01 00000000G   00 E8 00018 1$:          BLBS    TERMINAL_DEVICE, 2$                          ; 1727
                                  04 0001F              RET                                                  ;
   01 00000000G  00             02 E1 00020 2$:          BBC     #2, DEV_CHAR_2, 3$                           ; 1728
                                  04 00028              RET                                                  ;
                                6B D5 00029 3$:          TSTL    CONNECT_NAME                                 ; 1731
                                03 13 0002B              BEQL    5$                                           ;
                              0275 31 0002D 4$:          BRW     29$                                          ;
                      00000000G 00 D5 00030 5$:          TSTL    UAF_RECORD                                   ; 1732
                                F5 13 00036              BEQL    4$                                           ;
```

```
                                                                  C 11
INTERACT                                                   16-Sep-1984 01:55:50    VAX-11 Bliss-32 V4.0-742                Page  54
V04-000                                                    14-Sep-1984 12:41:07    [LOGIN.SRC]INTERACT.B32;1                     (11)

                          50        30   AE  9E 00038              MOVAB   GETJPI_ITEM_LIST, $$ITMBLKPTR                     ; 1776
                          80  02020020   8F  D0 0003C              MOVL    #33685536, ($$ITMBLKPTR)+                         ;
                          80        98   AD  9E 00043              MOVAB   FOUND_USERNAME, ($$ITMBLKPTR)+                    ;
                          80             6E  9E 00047              MOVAB   FOUND_USERNAME_LEN, ($$ITMBLKPTR)+                ;
                          80  031D000F   8F  D0 0004A              MOVL    #52232207, ($$ITMBLKPTR)+                         ;
                          80        89   AD  9E 00051              MOVAB   FOUND_TERMINAL+1, ($$ITMBLKPTR)+                  ;
                          80      00A4   CE  9E 00055              MOVAB   FOUND_TERMINAL_DESC, ($$ITMBLKPTR)+               ;
                          80  03040004   8F  D0 0005A              MOVL    #50593796, ($$ITMBLKPTR)+                         ;
                          80        04   AE  9E 00061              MOVAB   FOUND_UIC, ($$ITMBLKPTR)+                         ;
                                         80  D4 00065              CLRL    ($$ITMBLKPTR)+                                    ;
                          80  03190004   8F  D0 00067              MOVL    #51970052, ($$ITMBLKPTR)+                         ;
                          80        08   AE  9E 0006E              MOVAB   FOUND_PID, ($$ITMBLKPTR)+                         ;
                                         80  7C 00072              CLRQ    ($$ITMBLKPTR)+                                    ;
                                  00A4   CE  D4 00074              CLRL    FOUND_TERMINAL_DESC                               ; 1778
                  00A8    CE        88   AD  9E 00078              MOVAB   FOUND_TERMINAL, FOUND_TERMINAL_DESC+4             ; 1779
                    88    AD        5F   8F  90 0007E              MOVB    #95, FOUND_TERMINAL                               ; 1780
                                         6E  7C 00083              CLRQ    FOUND_USERNAME_LEN                                ; 1777
                                    08   AE  D4 00085              CLRL    FOUND_PID                                         ; 1782
                          50        20   AE  9E 00088              MOVAB   GETDVI_ITEM_LIST, $$ITMBLKPTR                     ; 1787
                          80  00E60004   8F  D0 0008C              MOVL    #15073284, ($$ITMBLKPTR)+                         ;
                          80        0C   AE  9E 00093              MOVAB   FOUND_DEVCHAR2, ($$ITMBLKPTR)+                    ;
                                         80  7C 00097              CLRQ    ($$ITMBLKPTR)+                                    ;
                                         57  D4 00099              CLRL    INDEX                                             ; 1789
                    10    AE             01  CE 0009B              MNEGL   #1, PID_CONTEXT                                   ; 1790
                                         7E  7C 0009F 6$:          CLRQ    -(SP)                                             ; 1799
                                    B8   AD  9F 000A1              PUSHAB  IOSB                                              ;
                                    3C   AE  9F 000A4              PUSHAB  GETJPI_ITEM_LIST                                  ;
                                         7E  D4 000A7              CLRL    -(SP)                                             ;
                                    24   AE  9F 000A9              PUSHAB  PID_CONTEXT                                       ;
                                         7E  D4 000AC              CLRL    -(SP)                                             ;
             00000000G    00             07  FB 000AE              CALLS   #7, SYS$GETJPIW                                   ;
                          04             50  E8 000B5              BLBS    STATUS, 7$                                        ;
                    B8    AD             50  B0 000B8              MOVW    STATUS, IOSB                                      ; 1800
                  09A8    8F        B8   AD  B1 000BC 7$:          CMPW    IOSB, #2472                                       ; 1802
                                         4F  13 000C2              BEQL    8$                                                ;
                          D7        B8   AD  E9 000C4              BLBC    IOSB, 6$                                          ; 1804
                                  00A4   CE  D5 000C8              TSTL    FOUND_TERMINAL_DESC                               ; 1805
                                         D1  13 000CC              BEQL    6$                                                ;
                          54  00000000G  00  D0 000CE              MOVL    UAF_RECORD, R4                                    ; 1808
  20              20      98        AD   6E  2D 000D5              CMPC5   FOUND_USERNAME_LEN, FOUND_USERNAME, #32, -        ;
                                    04   A4     000DB                      #32, 4(R4)                                        ;
                                         C0  12 000DD              BNEQ    6$                                                ;
                    24    A4        04   AE  D1 000DF              CMPL    FOUND_UIC, 36(R4)                                 ; 1810
                                         B9  12 000E4              BNEQ    6$                                                ;
                                  00A4   CE  D6 000E6              INCL    FOUND_TERMINAL_DESC                               ; 1813
                                         7E  7C 000EA              CLRQ    -(SP)                                             ; 1815
                                         7E  7C 000EC              CLRQ    -(SP)                                             ;
                                    30   AE  9F 000EE              PUSHAB  GETDVI_ITEM_LIST                                  ;
                                  00B8   CE  9F 000F1              PUSHAB  FOUND_TERMINAL_DESC                               ;
                                         7E  7C 000F5              CLRQ    -(SP)                                             ;
             00000000G    00             08  FB 000F7              CALLS   #8, SYS$GETDVIW                                   ;
                          9E             50  E9 000FE              BLBC    R0, 6$                                            ;
            99            0C        AE   01  E1 00101              BBC     #1, FOUND_DEVCHAR2, 6$                            ; 1816
                    C0  AD47        08   AE  D0 00106              MOVL    FOUND_PID, PID_LIST[INDEX]                        ; 1819
                                         57  D6 0010C              INCL    INDEX                                             ; 1820
                          10             57  D1 0010E              CMPL    INDEX, #16                                        ; 1821
                                         8C  1F 00111              BLSSU   6$                                                ;
```

```
                            57  D5 00113 8$:    TSTL    INDEX                                             : 1827
                            01  12 00115        BNEQ    9$                                                :
                                04 00117        RET                                                       :
                  50    30  AE  9E 00118 9$:    MOVAB   GETJPI_ITEM_LIST, $$ITMBLKPTR                     : 1842
                  80 031D000F  8F  D0 0011C     MOVL    #52232207, ($$ITMBLKPTR)+                         :
                  80    89  AD  9E 00123        MOVAB   FOUND_TERMINAL+1, ($$ITMBLKPTR)+                  :
                  80  00A4  CE  9E 00127        MOVAB   FOUND_TERMINAL_DESC, ($$ITMBLKPTR)+               :
                  80 031C0010  8F  D0 0012C     MOVL    #52166672, ($$ITMBLKPTR)+                         :
                  80  FF68  CD  9E 00133        MOVAB   FOUND_PROCNAME, ($$ITMBLKPTR)+                    :
                  80    14  AE  9E 00138        MOVAB   FOUND_PROCNAME_LEN, ($$ITMBLKPTR)+                :
                  80 02070040  8F  D0 0013C     MOVL    #34013248, ($$ITMBLKPTR)+                         :
                  80  00AC  CE  9E 00143        MOVAB   FOUND_IMAGNAME, ($$ITMBLKPTR)+                    :
                  80    18  AE  9E 00148        MOVAB   FOUND_IMAGNAME_LEN, ($$ITMBLKPTR)+                :
                            80  D4 0014C        CLRL    ($$ITMBLKPTR)+                                    :
                        14  AE  7C 0014E        CLRQ    FOUND_PROCNAME_LEN                                : 1843
                            58  D4 00151        CLRL    NUM_DISCONNECTED                                  : 1846
                  56        01  CE 00153        MNEGL   #1, I                                             : 1847
                          008F  31 00156        BRW     18$                                               :
                            7E  7C 00159 10$:   CLRQ    -(SP)                                             : 1850
                        B8  AD  9F 0015B        PUSHAB  IOSB                                              :
                        3C  AE  9F 0015E        PUSHAB  GETJPI_ITEM_LIST                                  :
                            7E  D4 00161        CLRL    -(SP)                                             :
                      C0 AD46  DF 00163         PUSHAL  PID_LIST[I]                                       :
                            7E  D4 00167        CLRL    -(SP)                                             :
         00000000G  00      07  FB 00169        CALLS   #7, SYS$GETJPIW                                   :
                    75      50  E9 00170        BLBC    R0, 18$                                           :
                    71  B8  AD  E9 00173        BLBC    IOSB, 18$                                         : 1851
                            58  D5 00177        TSTL    NUM_DISCONNECTED                                  : 1854
                            31  12 00179        BNEQ    13$                                               :
                  59  00A4  CE  D0 0017B        MOVL    FOUND_TERMINAL_DESC, FIRST_TERMINAL_LENGTH        : 1857
   FF78    CD       88  AD  10  28 00180        MOVC3   #16, FOUND_TERMINAL, FIRST_TERMINAL               : 1858
                    01      57  D1 00187        CMPL    INDEX, #1                                         : 1859
                            07  12 0018A        BNEQ    11$                                               :
                  50  0000'  CF  9E 0018C       MOVAB   P.ACG, R0                                         : 1860
                            05  11 00191        BRB     12$                                               :
                  50  0000'  CF  9E 00193 11$:  MOVAB   P.ACI, R0                                         : 1862
                            50  DD 00198 12$:   PUSHL   R0                                                :
         00000000G  00      01  FB 0019A        CALLS   #1, WRITE_OUTPUT                                  : 1859
                      0000'  CF  9F 001A1       PUSHAB  P.ACK                                             : 1864
         00000000G  00      01  FB 001A5        CALLS   #1, WRITE_OUTPUT                                  :
                            51  D4 001AC 13$:   CLRL    R1                                                : 1874
                        18  AE  D5 001AE        TSTL    FOUND_IMAGNAME_LEN                                :
                            09  12 001B1        BNEQ    14$                                               :
                            51  D6 001B3        INCL    R1                                                :
                  50  0000'  CF  9E 001B5       MOVAB   P.ACN, R0                                         : 1875
                            05  11 001BA        BRB     15$                                               :
                  50  00AC  CE  9E 001BC 14$:   MOVAB   FOUND_IMAGNAME, R0                                : 1874
                            50  DD 001C1 15$:   PUSHL   R0                                                :
                  04        51  E9 001C3        BLBC    R1, 16$                                           :
                            06  DD 001C6        PUSHL   #6                                                : 1872
                            03  11 001C8        BRB     17$                                               :
                        1C  AE  DD 001CA 16$:   PUSHL   FOUND_IMAGNAME_LEN                                : 1873
                      FF68  CD  9F 001CD 17$:   PUSHAB  FOUND_PROCNAME                                    : 1868
                        20  AE  DD 001D1        PUSHL   FOUND_PROCNAME_LEN                                : 1869
                        89  AD  9F 001D4        PUSHAB  FOUND_TERMINAL+1                                  : 1868
                      00B8  CE  DD 001D7        PUSHL   FOUND_TERMINAL_DESC                               :
                      0000'  CF  9F 001DB       PUSHAB  P.ACM                                             : 1866
```

```
INTERACT                                          E 11
V04-000                                     16-Sep-1984 01:55:50    VAX-11 Bliss-32 V4.0-742          Page  56
                                            14-Sep-1984 12:41:07    [LOGIN.SRC]INTERACT.B32;1               (11)

                  00000000G  00          07  FB 001DF       CALLS    #7, WRITE_FAO                                      ; 1868
                                         58  D6 001E6       INCL     NUM_DISCONNECTED                                   ; 1877
          02                 56          57  F2 001E8 18$:  AOBLSS   INDEX, I, 19$                                      ; 1848
                                         03  11 001EC       BRB      20$                                                ;
                                       FF68  31 001EE 19$:  BRW      10$                                                ;
                                         58  D5 001F1 20$:  TSTL     NUM_DISCONNECTED                                   ; 1880
                                         03  14 001F3       BGTR     21$                                                ;
                                       00AD  31 001F5       BRW      29$                                                ;
                        1C   AA          28  B0 001F8 21$:  MOVW     #40, INPUT_RAB+32                                  ; 1884
                        20   AA     D8   AB  9E 001FC       MOVAB    CONNECT_NAME_BUFFER, INPUT_RAB+36                  ; 1885
                        03   AA          21  8A 00201       BICB2    #33, INPUT_RAB+7                                   ; 1887
                        1B   AA  00000000G   00  90 00205   MOVB     SYS$GB_RETRY_TMO, INPUT_RAB+31                     ; 1888
                        03   AA     40   8F  88 0020D       BISB2    #64, INPUT_RAB+7                                   ; 1889
                             01          58  D1 00212       CMPL     NUM_DISCONNECTED, #1                               ; 1891
                                         0C  12 00215       BNEQ     22$                                                ;
                        30   AA          29  90 00217       MOVB     #41, INPUT_RAB+52                                  ; 1894
                        2C   AA   0000'  CF  9E 0021B       MOVAB    P.ACO, INPUT_RAB+48                                ; 1896
                                         3A  11 00221       BRB      24$                                                ; 1891
                             56     40   8F  9A 00223 22$:  MOVZBL   #64, R6                                            ; 1901
                             58     64   AE  9E 00227       MOVAB    PROMPT_BUFFER, R8                                  ;
     56          20   0000'  CF          20  2C 0022B       MOVC5    #32, P.ACP, #32, R6, (R8)                          ;
                                         68     00232                                                               ;
                                         1E  18 00233       BGEQ     23$                                                ;
                             58          20  C0 00235       ADDL2    #32, R8                                            ;
                             56          20  C2 00238       SUBL2    #32, R6                                            ;
     56          20   FF79   CD          59  2C 0023B       MOVC5    FIRST_TERMINAL_LENGTH, FIRST_TERMINAL+1, -         ;
                                         68     00242                #32, R6, (R8)                                      ;
                                         0E  18 00243       BGEQ     23$                                                ;
                             58          59  C0 00245       ADDL2    FIRST_TERMINAL_LENGTH, R8                          ;
                             56          59  C2 00248       SUBL2    FIRST_TERMINAL_LENGTH, R6                          ;
     56          20   0000'  CF          03  2C 0024B       MOVC5    #3, P.ACQ, #32, R6, (R8)                           ;
                                         68     00252                                                               ;
            30   AA          59          23  81 00253 23$:  ADDB3    #35, FIRST_TERMINAL_LENGTH, INPUT_RAB+52           ; 1904
                        2C   AA     64   AE  9E 00258       MOVAB    PROMPT_BUFFER, INPUT_RAB+48                        ; 1905
                                         02  DD 0025D 24$:  PUSHL    #2                                                 ; 1909
                                    FC   AA  9F 0025F       PUSHAB   INPUT_RAB                                          ;
                  00000000G  00          02  FB 00262       CALLS    #2, GET_INPUT                                      ;
                             01     04   AA  E8 00269       BLBS     INPUT_RAB+8, 25$                                   ; 1911
                                             04 0026D       RET                                                         ;
                             6B     1E   AA  3C 0026E 25$:  MOVZWL   INPUT_RAB+34, CONNECT_NAME                         ; 1914
                        04   AB     24   AA  D0 00272       MOVL     INPUT_RAB+40, CONNECT_NAME+4                       ; 1915
                             54          6B  D0 00277       MOVL     CONNECT_NAME, R4                                   ; 1917
                                         55  D4 0027A       CLRL     R5                                                 ;
                                         54  D5 0027C       TSTL     R4                                                 ;
                                         0B  13 0027E       BEQL     26$                                                ;
                                         55  D6 00280       INCL     R5                                                 ;
          0000'  CF         04   BB      54  29 00282       CMPC3    R4, @CONNECT_NAME+4, P.ACR                         ; 1918
                                         1C  13 00289       BEQL     30$                                                ;
                             09          55  E9 0028B 26$:  BLBC     R5, 27$                                            ; 1922
          0000'  CF         04   BB      54  29 0028E       CMPC3    R4, @CONNECT_NAME+4, P.ACS                         ; 1923
                                         04  13 00295       BEQL     28$                                                ;
                                         54  D5 00297 27$:  TSTL     R4                                                 ; 1926
                                         0A  12 00299       BNEQ     29$                                                ;
                             6B     01   A9  9E 0029B 28$:  MOVAB    1(R9), CONNECT_NAME                                ; 1929
                        04   AB   FF78   CD  9E 0029F       MOVAB    FIRST_TERMINAL, CONNECT_NAME+4                     ; 1930
                                         6B  D5 002A5 29$:  TSTL     CONNECT_NAME                                       ; 1937
                                         01  12 002A7 30$:  BNEQ     31$                                                ;
                                         04 002A9         RET                                          ;
                              5B         DD 002AA  31$:   PUSHL    R11                                 ; 1947
                                0000'    CF 9F 002AC      PUSHAB   P.ACT                               ;
            00000000G  00                02 FB 002B0      CALLS    #2, WRITE_FAO                       ;
                                         52 D4 002B7      CLRL     PREV_UIC                            ; 1949
                    50 00000000G         00 D0 002B9      MOVL     UAF_RECORD, R0                      ; 1950
                                         13 13 002C0      BEQL     32$                                 ;
                                    24   A0 DD 002C2      PUSHL    36(R0)                              ; 1953
                       00000000G         00 9F 002C5      PUSHAB   SET_UIC                             ;
            00000000G  00                02 FB 002CB      CALLS    #2, SYS$CMKRNL                      ;
                       52                50 D0 002D2      MOVL     R0, PREV_UIC                        ;
                                         7E 7C 002D5  32$:  CLRQ     -(SP)                             ; 1956
                                    24   AE 9F 002D7      PUSHAB   CHAN                                ;
                       00000000G         00 9F 002DA      PUSHAB   TERM_NAME                           ;
            00000000G  00                04 FB 002E0      CALLS    #4, SYS$ASSIGN                      ;
                   F8  AD                50 B0 002E7      MOVW     R0, IOSB                            ;
                       31                50 E9 002EB      BLBC     R0, 34$                             ;
                                         7E 7C 002EE      CLRQ     -(SP)                               ; 1965
                                         7E 7C 002F0      CLRQ     -(SP)                               ;
                                         7E D4 002F2      CLRL     -(SP)                               ;
                                         5B DD 002F4      PUSHL    R11                                 ;
                                         7E 7C 002F6      CLRQ     -(SP)                               ;
                                    F8   AD 9F 002F8      PUSHAB   IOSB                                ;
                       7E         0823   8F 3C 002FB      MOVZWL   #2083, -(SP)                        ;
                       7E           44   AE 3C 00300      MOVZWL   CHAN, -(SP)                         ;
                                         7E D4 00304      CLRL     -(SP)                               ;
            00000000G  00                0C FB 00306      CALLS    #12, SYS$QIOW                       ;
                       04                50 E8 0030D      BLBS     STATUS, 33$                         ;
                   F8  AD                50 B0 00310      MOVW     STATUS, IOSB                        ; 1966
                       7E           1C   AE 3C 00314  33$:  MOVZWL   CHAN, -(SP)                       ; 1968
            00000000G  00                01 FB 00318      CALLS    #1, SYS$DASSGN                      ;
                                         52 D5 0031F  34$:  TSTL     PREV_UIC                          ; 1971
                                         0F 13 00321      BEQL     35$                                 ;
                                         52 DD 00323      PUSHL    PREV_UIC                            ; 1974
                       00000000G         00 9F 00325      PUSHAB   SET_UIC                             ;
            00000000G  00                02 FB 0032B      CALLS    #2, SYS$CMKRNL                      ;
                       16           F8   AD E8 00332  35$:  BLBS     IOSB, 36$                         ; 1976
                       7E           F8   AD 3C 00336      MOVZWL   IOSB, -(SP)                         ; 1979
                                         5B DD 0033A      PUSHL    R11                                 ;
                                         01 DD 0033C      PUSHL    #1                                  ;
                       00000000G         8F DD 0033E      PUSHL    #LGI$_CONNERR                       ;
            00000000G  00                04 FB 00344      CALLS    #4, LIB$SIGNAL                      ;
                                         04 0034B         RET                                          ; 1978
                                         05 DD 0034C  36$:  PUSHL    #5                                ; 1985
            00000000G  00                01 FB 0034E      CALLS    #1, SECURITY_AUDIT                  ;
                                         7E D4 00355      CLRL     -(SP)                               ; 1987
                       00000000G         00 9F 00357      PUSHAB   EXIT_PROCESS                        ;
            00000000G  00                02 FB 0035D      CALLS    #2, SYS$CMEXEC                      ;
                                         04 00364         RET                                          ; 1989

; Routine Size:  869 bytes,    Routine Base:  $CODE$ + 0AA2
; 1601         1990  1 GLOBAL ROUTINE ascic_day_of_week (time) =
; 1602         1991  1
; 1603         1992  1 !---
; 1604         1993  1 !
; 1605         1994  1 !        Return ASCIC day of week given absolute time.
; 1606         1995  1 !
; 1607         1996  1 ! Inputs:
; 1608         1997  1 !
; 1609         1998  1 !        time = Address of absolute time quadword.
; 1610         1999  1 !
; 1611         2000  1 ! Outputs:
; 1612         2001  1 !
; 1613         2002  1 !        Address of ASCIC day of week.
; 1614         2003  1 !
; 1615         2004  1 !---
; 1616         2005  1
; 1617         2006  2 BEGIN
; 1618         2007  2
; 1619         2008  2 BIND
; 1620         2009  2     week_days = UPLIT (UPLIT BYTE(%ASCIC 'Monday'),
; 1621         2010  2                        UPLIT BYTE(%ASCIC 'Tuesday'),
; 1622         2011  2                        UPLIT BYTE(%ASCIC 'Wednesday'),
; 1623         2012  2                        UPLIT BYTE(%ASCIC 'Thursday'),
; 1624         2013  2                        UPLIT BYTE(%ASCIC 'Friday'),
; 1625         2014  2                        UPLIT BYTE(%ASCIC 'Saturday'),
; 1626         2015  2                        UPLIT BYTE(%ASCIC 'Sunday'))
; 1627         2016  2         : VECTOR [7];
; 1628         2017  2
; 1629         2018  2 LOCAL
; 1630         2019  2     day;
; 1631         2020  2
; 1632         2021  2 lib$day_of_week(.time, day);           ! Fetch day of week from time
; 1633         2022  2 RETURN .week_days [.day - 1];          ! Return address of ASCIC week day
; 1634         2023  2
; 1635         2024  1 END;


                                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                        79  61  64  6E  6F  4D  06  00460 P.ACV:  .ASCII  <6>\Monday\                              ;
                    79  61  64  73  65  75  54  07  00467 P.ACW:  .ASCII  <7>\Tuesday\                             ;
            79  61  64  73  65  6E  64  65  57  09  0046F P.ACX:  .ASCII  <9>\Wednesday\                           ;
                79  61  64  73  72  75  68  54  08  00479 P.ACY:  .ASCII  <8>\Thursday\                            ;
                        79  61  64  69  72  46  06  00482 P.ACZ:  .ASCII  <6>\Friday\                              ;
                79  61  64  72  75  74  61  53  08  00489 P.ADA:  .ASCII  <8>\Saturday\                            ;
                        79  61  64  6E  75  53  06  00492 P.ADB:  .ASCII  <6>\Sunday\                              ;
                                                    00499         .BLKB   3
00000000' 00000000' 00000000' 00000000' 00000000' 00000000'  0049C P.ACU:  .ADDRESS P.ACV, P.ACW, P.ACX, P.ACY, P.ACZ, -  ;
                                                  00000000'  004B4                  P.ADA, P.ADB                          ;

                                                             WEEK_DAYS=          P.ACU


                                                                  .PSECT  $CODE$,NOWRT,2

                                                 0000 00000       .ENTRY  ASCIC_DAY_OF_WEEK, Save nothing         ; 1990
            5E                04  C2 00002          SUBL2   #4, SP                                  ;
                              5E  DD 00005          PUSHL   SP                                      ; 2021
                         04   AC  DD 00007          PUSHL   TIME                                    ;
      00000000G  00               02  FB 0000A          CALLS   #2, LIB$DAY_OF_WEEK                     ;
            50                6E  D0 00011          MOVL    DAY, R0                                 ; 2022
            50        0000'CF40   D0 00014          MOVL    WEEK_DAYS-4[R0], R0                     ;
                                  04 0001A          RET                                             ; 2024

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 0E07
; 1637          2025  1 END
; 1638          2026  0 ELUDOM



                                                            .EXTRN  LIB$SIGNAL, LIB$STOP
;                               PSECT SUMMARY
;
;       Name                 Bytes                       Attributes
;
;   $OWN$                        92  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $GLOBAL$                    365  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                     1208  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                     3618  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                            Library Statistics
;
;                                           -------- Symbols --------      Pages       Processing
;       File                                Total     Loaded   Percent     Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1            18619        170         0     1000         00:01.3



;                               COMMAND QUALIFIERS

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

; Size:         3618 code + 1665 data bytes
; Run Time:        00:46.4
; Elapsed Time:    03:03.0
; Lines/CPU Min:     2619
; Lexemes/CPU-Min: 39967
; Memory Used:  365 pages
; Compilation Complete
```

0222 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBHOUR
LIS

LIBDAYWK
LIS

INITUSER
LIS

LOGIN
LIS

INTERACT
LIS