1 3

```
   CCCCCCCCCCCC  LLL                  IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                  IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                  IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL      IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL      IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL      IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

FILEID**INFO

```
 IIIIII    NN      NN  FFFFFFFFFF     000000
 IIIIII    NN      NN  FFFFFFFFFF     000000
   II      NN      NN  FF           00      00
   II      NN      NN  FF           00      00
   II      NNNN    NN  FF           00      00
   II      NNNN    NN  FF           00      00
   II      NN  NN  NN  FFFFFFFF     00      00
   II      NN  NN  NN  FFFFFFFF     00      00
   II      NN    NNNN  FF           00      00
   II      NN    NNNN  FF           00      00
   II      NN      NN  FF           00      00     ....
   II      NN      NN  FF           00      00     ....
 IIIIII    NN      NN  FF             000000       ....
 IIIIII    NN      NN  FF             000000       ....

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
```

```
MODULE SHOW$PROCESS_CONT (IDENT='V04-000') =
BEGIN

!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************
!
!++
! FACILITY:  SHOW PROCESS/CONTINUOUS utility
!
! ABSTRACT:
!
!       This utility allows anyone to examine a given process
!       extremely closely to watch process statistics.
!
! ENVIRONMENT:
!
!       VAX/VMS operating system.  User mode with CMEXEC, CMKRNL, WORLD privs.
!
! AUTHOR:  Tim Halvorsen, Oct 1979
!
! Modified by:
!
!       V03-020 JRL0005         John R. Lawson, Jr.     19-Jun-1984 15:54
!               Eliminate external references to SCR (shareable image) to
!               expidite image initialization; link to SCR onl when package
!               is required.
!
!       V03-019 MCN0150         Maria del C. Nasr       13-Feb-1984
!               Increase field of working set value display.
!
!       V03-018 LMP0140         L. Mark Pilant,         25-Aug-1983  13:08
!               Add support for alphanumeric UICs.
!
!       V03-017 CWH1002         CW Hobbs        24-Feb-1983
!               Modify to use extended pid as the input pid.
!
```

```
!       V03-016 GAS0099         Gerry Smith       7-Jan-1983
!               Modify slightly to fit with the new SHOW interface.
!
!       V03-015 CWH0015         CW Hobbs         21-Oct-1982
!               Fix use of WSL$C_xxxx symbols.  The pre-X1MF systems had different
!               values for BLISS and MACRO.  Post-X1MF symbols had the same
!               value, so any BLISS uses had to be modified.
!
!       V03-014 TMH0014         Tim Halvorsen    14-Feb-1982
!               Remove code to display whether PC,PSL,SP is current
!               or stored - it slows down the display too much and
!               isn't worth it.
!
!       V03-013 MMD001          Meg Dumont        19-Jan-1982
!               Make INFO callable from SHOW PROCESS with the /CONTINUOUS
!               qualifier.
!       V02-012 BLS0098         Benn Schreiber    4-Nov-1981
!               General addressing mode for scr$ routines
!
!       V02-011 MHB0076         Mark Bramhall    15-Jul-1981
!               Fix process name display to substitute "?" for "funny"
!               characters.  Allow process selection based on PID.
!
!       V02-010 MHB0075         Mark Bramhall    24-Jun-1981
!               Exit handler now goes to screen's last line and only
!               erases that.
!
!       V009    TMH0009         Tim Halvorsen    15-Jun-1981
!               Do not declare exit handler to erase screen until all
!               preliminary error checking is done.
!
!       V008    TMH0008         Tim Halvorsen    11-Jun-1981
!               Replace GETMEM module with GETJPI call.  Fix overlapping
!               display of local event flags.  Erase screen when exiting
!               the program.
!
!       V02-007 MHB0074         Mark Bramhall     9-Jun-1981
!               Modified vmap display to handle 132 column mode with
!               128 pages per row.  Changed from unsolicited input ast's
!               to fullduplex operation with simply a single character,
!               echo off read posted.
!
!       V02-006 WMC0001         Wayne Cardoza    12-May-1981
!               Add display of local event flags.
!
!       V02-005 TMH0001         Tim Halvorsen    19-Dec-1980
!               Add ability to see system working set by typing '*'
!               as the process name.  In vmap display, show 'G' for
!               global section, 'L' for locked page.  Fix optimization
!               which inhibits setting the cursor if its already set
!               to the right place to speed things up.  Add current time
!               to main display.  Fix mode changing so that partial output
!               does not appear on the screen.
!
!       V02-004 GRR0001         Greg Robert      18-Dec-1980
!               Relaxed screen length restrictions.  Removed 'nohardcopy'
!               restriction.  Removed lib$put_output sycronization
```

```
1 !                    call.  Added single spacing option.  Increased
1 !                    wait time when not a terminal.  Changed SHOW_VPN to
1 !                    be more efficient. Add BOLD attribute to PC
1 !                    display in VMAP.
1 !
1 !            V02-003 MHB0069          Mark Bramhall    2-Oct-1980
1 !                    Use LIB$SCREEN_INFO to check and size screen before starting.
1 !                    Add a <CR><LF> to the process name prompt.
1 !                    Exit on CTRL/Z as well as "E".
1 !
1 !            V02-002 MHB0062          Mark Bramhall   15-Sep-1980
1 !                    Exit cleanly on user typed Control/Z.
1 !                    Do initial process search without upper-cased name.
1 !                    Unsolicited 'E' (or 'e') causes a clean exit.
1 !                    'maxvirt' is defined as <#rows-to-use> * 64.
1 !                    Add PC, State, Image name to last line of VMAP display.
1 !
1 !            V001    TMH0001          Tim Halvorsen   27-Jul-1980
1 !                    Slow down display if process is quiesient.
1 !                    Force immediate read of image filespec on entry.
1 !                    Fix image filespec read to test if successfully
1 !                    completed.
1 !--
```

```
!
! Include files
!

LIBRARY 'SYS$LIBRARY:LIB.L32';

!
! Table of contents
!
SWITCHES ADDRESSING_MODE (EXTERNAL=GENERAL);    ! Set longword addressing
FORWARD ROUTINE
    exit_handler:       NOVALUE,        ! Image exit handler
    fao_buffer,                         ! Format an FAO string
    translate_value,                    ! Translate coded value to string
    kernel_get_info,                    ! Get information from PCB/PHD
    wake_ast:           NOVALUE,        ! Unsolicited character AST routine
    proc_cont_display:  NOVALUE;        ! Main routine

!
!       Mechanism to link to SCR
!

external routine

   LIB$FIND_IMAGE_SYMBOL: addressing_mode(general);

own

   $LIB$SCREEN_INFO,
   $SCR$ERASE_LINE,
   $SCR$ERASE_PAGE,
   $SCR$PUT_BUFFER,
   $SCR$PUT_SCREEN,
   $SCR$SET_BUFFER,
   $SCR$SET_CURSOR: long;

macro

   LIB$SCREEN_INFO = ( .$LIB$SCREEN_INFO ) %,
   SCR$ERASE_LINE = ( .$SCR$ERASE_LINE ) %,
   SCR$ERASE_PAGE = ( .$SCR$ERASE_PAGE ) %,
   SCR$PUT_BUFFER = ( .$SCR$PUT_BUFFER ) %,
   SCR$PUT_SCREEN = ( .$SCR$PUT_SCREEN ) %,
   SCR$SET_BUFFER = ( .$SCR$SET_BUFFER ) %,
   SCR$SET_CURSOR = ( .$SCR$SET_CURSOR ) %;

!EXTERNAL ROUTINE
!    lib$screen_info : ADDRESSING_MODE(GENERAL), ! Get screen information
!    scr$erase_line : ADDRESSING_MODE(GENERAL),  ! Erase entire line
!    scr$erase_page : ADDRESSING_MODE(GENERAL),  ! Erase entire screen
!    scr$put_buffer : ADDRESSING_MODE(GENERAL),  ! Flush screen buffer
!    scr$put_screen : ADDRESSING_MODE(GENERAL),  ! Write output string
!    scr$set_buffer : ADDRESSING_MODE(GENERAL),  ! Enable/disable screen buffering
!    scr$set_cursor : ADDRESSING_MODE(GENERAL);  ! Set cursor position


MACRO
```

```
    cstring[] =
        UPLIT BYTE(%CHARCOUNT(%STRING(%REMAINING)),
            %STRING(%REMAINING))%,

    table_entry(prefix)[name] =
        %NAME(prefix,name),cstring(name)%,

    nametable(prefix)[] =
        UPLIT(table_entry(prefix,%REMAINING),-1,-1)%,

    perform(command) =
        IF NOT (status = command) THEN BEGIN
            SIGNAL(.status);
            RETURN true;
            END%,

    at(line,column) =
        scr$set_cursor(line,column)%,

    fao(string) =
        fao_buffer(%ASCID string
        %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI)%,

    write(string) =
        scr$put_screen(fao(string
        %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI))%;

LITERAL
    maxvirt     = 23*128,               ! Maximum virtual pages in vmap
    true        = 1,                    ! Boolean value true
    false       = 0;                    ! Boolean value false

MACRO
    all         = 0,0,8,0%,             ! All bits in a byte
    pc_in_page  = 0,7,1,0%;             ! Use unused bit to indicate "PC page"
!
!       Define a macro to print the proper character in the vmap display
!       given the virtual page number (VPN)
!

MACRO
    show_vpn(page_number) =
        BEGIN
        LOCAL bits: BLOCK [BYTE,BYTE];          ! Bits from working set entry
        bits = .vmap [page_number,all];         ! Get working set bits
        SELECTONEU true
        OF
            SET
            [.bits [pc_in_page]]:
                                scr$put_screen(%ASCID '@',0,0,1);

            [.bits [wsl$v_valid] AND (.bits [wsl$v_wslock] OR .bits [wsl$v_pfnlock])]:
                                scr$put_screen(%ASCID 'L');

            !
            ! The WSL$C symbols are defined as shifted one bit left so that
```

```
                ! things are simpler in MACRO.  We have to shift them back.
                !
                [.bits [wsl$v_valid] AND (.bits [wsl$v_pagtyp] EQL (wsl$c_gblwrt^-1)
                    OR .bits [wsl$v_pagtyp] EQL (wsl$c_global^-1))]:
                                    scr$put_screen(%ASCID 'G');

                [.bits [wsl$v_valid]]:
                                    scr$put_screen(%ASCID '*');

                [OTHERWISE]:
                                    scr$put_screen(%ASCID ' ');
                TES;
            END%;

!
!       Define macro to set the cursor to the position corresponding
!       to a given virtual address
!

MACRO
    at_vpn(vpn) = at((vpn/.vpn_per_col)+1,(vpn MOD .vpn_per_col)+.vpn_1st_col)%;

EXTERNAL
    proc_a_desc:        BLOCK [8,BYTE],         ! Process name descriptor
    proc_z_name:        VECTOR[15,BYTE],        ! Buffer for process name
    proc_l_pid;                                 ! Process identification

PSECT   OWN = INFO_OWN;
PSECT   GLOBAL = INFO_GLOB;
PSECT   PLIT = INFO_PLIT;
PSECT   CODE = INFO_CODE;

OWN

!
!       The following OWN storage is locked into real memory to
!       improve performance, since references are made to almost
!       all pieces during every loop.
!

    lock_start,                         ! Start of locked down OWN storage
    spacing:    INITIAL(2),             ! Default spacing equal 2
    keep_going: INITIAL(true),          ! Time to exit flag
    exit_block: VECTOR [5]              ! Exit control block
                INITIAL(0,exit_handler,1,exit_block [4],0),
    mode_change:        INITIAL(true),  ! Mark display mode changed
    new_display_mode,                   ! New display mode (only if mode_change true)
    dev_flags,                          ! Established by SCR$GET_INFO
    max_row,                            ! Maximum legal row for cursor
    tt_chan:    WORD,                   ! Channel to TT
    tt_buffer:  VECTOR[2,BYTE],         ! Buffer for TT "unsol" chars
    prev_prcnam: VECTOR[15,BYTE],       ! Buffer for previous name
    prev_desc: BLOCK[8,BYTE],           ! Descriptor for above buffer
    prev_image: VECTOR[nam$c_maxrss,BYTE],      ! Previous image name
    prev_imgdesc: VECTOR[2],            ! Descriptor for above buffer
    prev_state,                         ! Previous process values...
    prev_pri,
;   310         0309  1      prev_prib,
;   311         0310  1      prev_grp,
;   312         0311  1      prev_mem,
;   313         0312  1      prev_pc,
;   314         0313  1      prev_sp,
;   315         0314  1      prev_psl,
;   316         0315  1      prev_ppgcnt,
;   317         0316  1      prev_gpgcnt,
;   318         0317  1      prev_cputime,
;   319         0318  1      prev_dirio,
;   320         0319  1      prev_bufio,
;   321         0320  1      prev_pageflts,
;   322         0321  1      prev_locevfl0,
;   323         0322  1      prev_locevfl1,
;   324         0323  1      prev_vause,
;   325         0324  1      prev_vmap:  REF BLOCKVECTOR [maxvirt,BYTE,BYTE],
;   326         0325  1
;   327         0326  1 !
;   328         0327  1 !        The following OWN storage must be locked into real memory to
;   329         0328  1 !        avoid page faults by the kernel mode code running at a
;   330         0329  1 !        high IPL.
;   331         0330  1 !
;   332         0331  1
;   333         0332  1      display_mode,                         ! Display mode (0=normal,1=vmap)
;   334         0333  1      grp,                                  ! Group of UIC
;   335         0334  1      mem,                                  ! Member of UIC
;   336         0335  1      state,                                ! Process state
;   337         0336  1      pri,                                  ! Process priority
;   338         0337  1      prib,                                 ! Process base priority
;   339         0338  1      pc,                                   ! PC register
;   340         0339  1      sp,                                   ! Current SP
;   341         0340  1      ppgcnt,                               ! Process pages in WS
;   342         0341  1      gpgcnt,                               ! Global pages in WS
;   343         0342  1      cputime,                              ! Total CPU time
;   344         0343  1      dirio,                                ! Total direct I/O requests
;   345         0344  1      bufio,                                ! Total buffered I/O requests
;   346         0345  1      pageflts,                             ! Total page faults
;   347         0346  1      vause,                                ! Virtual pages in use
;   348         0347  1      psl,                                  ! Current PSL
;   349         0348  1      locevfl0,                             ! Local event flag cluster 0
;   350         0349  1      locevfl1,                             ! Local event flag cluster 1
;   351         0350  1      vmap: BLOCKVECTOR [maxvirt,BYTE,BYTE], ! Map of virtual address space
;   352         0351  1      lock_end;                             ! End of locked down OWN storage
```

```
1 ROUTINE exit_handler (status_address): NOVALUE =
1
1 !---
1 !
1 !       This routine is called during image rundown to cleanup.
1 !       Its sole function is to erase the screen.
1 !
1 ! Inputs:
1 !
1 !       status_address = Address of final status longword
1 !
1 ! Outputs:
1 !
1 !       None
1 !---
1
2 BEGIN
2
2 scr$put_buffer();                         ! Dump contents of current buffer
2 scr$erase_page(.max_row,1);               ! Go to last line and erase it
2
1 END;


                                                    .TITLE  SHOW$PROCESS_CONT
                                                    .IDENT  \V04-000\

                                                    .PSECT  INFO_OWN,NOEXE,2

                                   00000 LOCK_START:
                                                    .BLKB   4
                         00000002  00004 SPACING:.LONG      2                                    ;
                         00000001  00008 KEEP_GOING:
                                                    .LONG   1                                    ;
                         00000000  0000C EXIT_BLOCK:
                                                    .LONG   0                                    ;
                         00000000' 00010            .ADDRESS EXIT_HANDLER                        ;
                         00000001  00014            .LONG   1                                    ;
                         00000000' 00018            .ADDRESS EXIT_BLOCK+16                       ;
                         00000000  0001C            .LONG   0                                    ;
                         00000001  00020 MODE_CHANGE:
                                                    .LONG   1                                    ;
                                   00024 NEW_DISPLAY_MODE:
                                                    .BLKB   4
                                   00028 DEV_FLAGS:
                                                    .BLKB   4
                                   0002C MAX_ROW:.BLKB      4
                                   00030 TT_CHAN:.BLKB      2
                                   00032            .BLKB   2
                                   00034 TT_BUFFER:
                                                    .BLKB   2
                                   00036            .BLKB   2
                                   00038 PREV_PRCNAM:
                                                    .BLKB   15
                                   00047            .BLKB   1
                                   00048 PREV_DESC:
                                                    .BLKB   8
```

```
00050 PREV_IMAGE:
                .BLKB   255
0014F           .BLKB   1
00150 PREV_IMGDESC:
                .BLKB   8
00158 PREV_STATE:
                .BLKB   4
0015C PREV_PRI:
                .BLKB   4
00160 PREV_PRIB:
                .BLKB   4
00164 PREV_GRP:
                .BLKB   4
00168 PREV_MEM:
                .BLKB   4
0016C PREV_PC:.BLKB     4
00170 PREV_SP:.BLKB     4
00174 PREV_PSL:
                .BLKB   4
00178 PREV_PPGCNT:
                .BLKB   4
0017C PREV_GPGCNT:
                .BLKB   4
00180 PREV_CPUTIME:
                .BLKB   4
00184 PREV_DIRIO:
                .BLKB   4
00188 PREV_BUFIO:
                .BLKB   4
0018C PREV_PAGEFLTS:
                .BLKB   4
00190 PREV_LOCEVFL0:
                .BLKB   4
00194 PREV_LOCEVFL1:
                .BLKB   4
00198 PREV_VAUSE:
                .BLKB   4
0019C PREV_VMAP:
                .BLKB   4
001A0 DISPLAY_MODE:
                .BLKB   4
001A4 GRP:      .BLKB   4
001A8 MEM:      .BLKB   4
001AC STATE:    .BLKB   4
001B0 PRI:      .BLKB   4
001B4 PRIB:     .BLKB   4
001B8 PC:       .BLKB   4
001BC SP:       .BLKB   4
001C0 PPGCNT:   .BLKB   4
001C4 GPGCNT:   .BLKB   4
001C8 CPUTIME:.BLKB     4
001CC DIRIO:    .BLKB   4
001D0 BUFIO:    .BLKB   4
001D4 PAGEFLTS:
                .BLKB   4
001D8 VAUSE:    .BLKB   4
001DC PSL:      .BLKB   4
```

```
                                                         001E0 LOCEVFL0:
                                                                      .BLKB   4
                                                         001E4 LOCEVFL1:
                                                                      .BLKB   4
                                                         001E8 VMAP:  .BLKB   2944
                                                         00D68 LOCK_END:
                                                                      .BLKB   4

                                                                      .PSECT  $OWN$,NOEXE,2

                                                         00000 $LIB$SCREEN_INFO:
                                                                      .BLKB   4
                                                         00004 $SCR$ERASE_LINE:
                                                                      .BLKB   4
                                                         00008 $SCR$ERASE_PAGE:
                                                                      .BLKB   4
                                                         0000C $SCR$PUT_BUFFER:
                                                                      .BLKB   4
                                                         00010 $SCR$PUT_SCREEN:
                                                                      .BLKB   4
                                                         00014 $SCR$SET_BUFFER:
                                                                      .BLKB   4
                                                         00018 $SCR$SET_CURSOR:
                                                                      .BLKB   4

                                                                      .EXTRN  LIB$FIND_IMAGE_SYMBOL
                                                                      .EXTRN  PROC_A_DESC, PROC_Z_NAME
                                                                      .EXTRN  PROC_L_PID

                                                                      .PSECT  INFO_CODE,NOWRT,2

                                                    0000 00000 EXIT_HANDLER:
                                                                      .WORD   Save nothing                                  ; 0352
                          0000'  DF           00  FB 00002            CALLS   #0, @$SCR$PUT_BUFFER                          ; 0370
                                              01  DD 00007            PUSHL   #1                                            ; 0371
                                      0000'   CF  DD 00009            PUSHL   MAX_ROW                                       ;
                          0000'  DF           02  FB 0000D            CALLS   #2, @$SCR$ERASE_PAGE                          ;
                                                  04 00012            RET                                                   ; 0373

; Routine Size:  19 bytes,    Routine Base:  INFO_CODE + 0000
```

```
        0374  1 ROUTINE fao_buffer(ctrstr,args) =
        0375  2 BEGIN
        0376  2
        0377  2 !---
        0378  2 !
        0379  2 !       This routine passes an ascii string through the FAO
        0380  2 !       system service with any number of specified parameters.
        0381  2 !
        0382  2 !---
        0383  2
        0384  2 OWN
        0385  2     desc :      VECTOR[2],              ! Result descriptor
        0386  2     buf :       VECTOR[80,BYTE];        ! Output buffer
        0387  2
        0388  2 MAP
        0389  2     ctrstr :    REF VECTOR[2],
        0390  2     args :      VECTOR[4];
        0391  2
        0392  2 desc[0] = 80;                         ! Set up result descriptor
        0393  2 desc[1] = buf;
        0394  2 $faol(ctrstr=.ctrstr,outlen=desc,outbuf=desc,prmlst=args);
        0395  2 RETURN desc;
        0396  1 END;


                                                                .PSECT  INFO_OWN,NOEXE,2

                                              00D6C DESC:       .BLKB   8
                                              00D74 BUF:        .BLKB   80

                                                                .EXTRN  SYS$FAOL

                                                                .PSECT  INFO_CODE,NOWRT,2

                                         0004 00000 FAO_BUFFER:
                                                                .WORD   Save R2                                  ; 0374
                        52      0000'   CF   9E 00002           MOVAB   DESC, R2                                 ;
                        62        50    8F   9A 00007           MOVZBL  #80, DESC                                ; 0392
                   04   A2        08    A2   9E 0000B           MOVAB   BUF, DESC+4                              ; 0393
                                  08    AC   9F 00010           PUSHAB  ARGS                                     ; 0394
                                        52   DD 00013           PUSHL   R2                                       ;
                                        52   DD 00015           PUSHL   R2                                       ;
                                  04    AC   DD 00017           PUSHL   CTRSTR                                   ;
              00000000G 00              04   FB 0001A           CALLS   #4, SYS$FAOL                             ;
                        50              62   9E 00021           MOVAB   DESC, R0                                 ; 0395
                                             04 00024           RET                                              ; 0396

; Routine Size:  37 bytes,    Routine Base:  INFO_CODE + 0013
;  401          0397  1 ROUTINE translate_value(value,table) =
;  402          0398  2 BEGIN
;  403          0399  2
;  404          0400  2 !---
;  405          0401  2 !
;  406          0402  2 !       This routine returns the address of a counted string
;  407          0403  2 !       describing a given value.
;  408          0404  2 !
;  409          0405  2 ! Inputs
;  410          0406  2 !
;  411          0407  2 !       value = The value to be translated
;  412          0408  2 !       table = Address of the table which describes the values:
;  413          0409  2 !
;  414          0410  2 !               cstring-addr,   value
;  415          0411  2 !                   .              .
;  416          0412  2 !                   .              .
;  417          0413  2 !                        -1,       -1
;  418          0414  2 !
;  419          0415  2 ! Outputs
;  420          0416  2 !
;  421          0417  2 !       The value of the routine is the address of the counted
;  422          0418  2 !       ascii stringg.  If the search fails, a pointer to the
;  423          0419  2 !       string "<NONE>" is returned.
;  424          0420  2 !
;  425          0421  2 !---
;  426          0422  2
;  427          0423  2 MAP
;  428          0424  2     table:      REF VECTOR[,LONG];      ! Name table
;  429          0425  2
;  430          0426  2 INCR entry FROM 0 BY 1 DO
;  431          0427  2     SELECT .table[.entry*2] OF SET
;  432          0428  2         [-1]:
;  433          0429  2             RETURN cstring('<NONE>');
;  434          0430  2         [.value]:
;  435          0431  2             RETURN .table[.entry*2+1];
;  436          0432  2     TES
;  437          0433  1 END;


                                                                        .PSECT  INFO_PLIT,NOWRT,NOEXE,2

                                                     06  00000 P.AAA:   .BYTE   6                                        ;
                         3E  45  4E  4F  4E  3C  00001           .ASCII  \<NONE>\                                 ;


                                                                        .PSECT  INFO_CODE,NOWRT,2

                                                   000C  00000 TRANSLATE_VALUE:
                                                                        .WORD   Save R2,R3                               ; 0397
                                                 50  D4  00002          CLRL    ENTRY                                    ; 0430
                  51           50            01  78  00004 1$:          ASHL    #1, ENTRY, R1                            ; 0427
                               53       08 BC41  D0  00008              MOVL    @TABLE[R1], R3                           ;
                     FFFFFFFF  8F            53  D1  0000D              CMPL    R3, #-1                                  ; 0428
                                             09  12  00014              BNEQ    2$                                       ;
                               52     0000'  CF  9E  00016              MOVAB   P.AAA, R2                                ; 0429
                         50          52  D0 0001B        MOVL    R2, R0                              ;
                                         04 0001E        RET                                         ;
                   04    AC          53  D1 0001F 2$:    CMPL    R3, VALUE                           ; 0430
                                     0A  12 00023        BNEQ    3$                                  ;
                         51      08 BC41 DE 00025        MOVAL   @TABLE[R1], R1                      ; 0431
                         50      04   A1 D0 0002A        MOVL    4(R1), R0                           ;
                                         04 0002E        RET                                         ;
              CD         50 7FFFFFFF  8F F3 0002F 3$:    AOBLEQ  #2147483647, ENTRY, 1$              ; 0427
                         50           01 CE 00037        MNEGL   #1, R0                              ; 0433
                                         04 0003A        RET                                         ;

; Routine Size:  59 bytes,    Routine Base:  INFO_CODE + 0038
```

```
1 FORWARD
1     info_ipl;                                    ! Longword containing IPL
1
1 ROUTINE kernel_get_info =
1
1 !---
1 !
1 !       This routine runs in kernel mode to obtain
1 !       the necessary information about a process
1 !       from the process control block and process
1 !       header in system space.
1 !
1 ! On input:
1 !       proc_l_pid = Process ID of desired process
1 !
1 !---
1
2 BEGIN
2
2 EXTERNAL
2     sch$gl_nullpcb : ADDRESSING_MODE(GENERAL);  ! Label on null proc pcb (not pointer to null pcb)
2
2 LINKAGE
2     cvt_lnk = JSB (REGISTER=0) : PRESERVE (1,2,3,4,5) NOTUSED (6,7,8,9,10,11);
2
2 EXTERNAL ROUTINE
2     exe$epid_to_pcb : cvt_lnk ADDRESSING_MODE (GENERAL);
2
2 MACRO
2     set_ipl(ipl) = (BUILTIN MTPR; MTPR(%REF(ipl),PR$_IPL))%;
2
2 REGISTER
2     pcb:        REF BLOCK [,BYTE],      ! Address of PCB
2     phd:        REF BLOCK [,BYTE],      ! Address of PHD
2     vpn;                                ! VPN for filling in vmap
2
2 pcb = exe$epid_to_pcb (.proc_l_pid);    ! Convert the EPID to a pcb address
2
2 IF .pcb EQL 0                           ! If we didn't get a pcb address
2 THEN
2     RETURN ss$_nonexpr;                 ! Exit with failure;
2
2 IF  .pcb EQL sch$gl_nullpcb             ! If process went away,
2  OR .pcb [pcb$l_epid] NEQ .proc_l_pid   ! or if the pcb has been reused,
2 THEN
2     RETURN ss$_nonexpr;                 ! then exit with failure
2
2 CH$MOVE(15,pcb[$BYTEOFFSET(pcb$t_lname),8,0,0],proc_z_name);
2 proc_a_desc[dsc$w_length] = .pcb[$BYTEOFFSET(pcb$t_lname),0,8,0];
2 proc_a_desc[dsc$a_pointer] = proc_z_name;
2 grp = .pcb[pcb$w_grp];
2 mem = .pcb[pcb$w_mem];
2 state = .pcb[pcb$w_state];
2 pri = 31-.pcb[pcb$b_pri];
2 prib = 31-.pcb[pcb$b_prib];
2 locevfl0 = .pcb[pcb$l_efcs];
2 locevfl1 = .pcb[pcb$l_efcu];
```

```
2 ppgcnt = .pcb[pcb$w_ppgcnt];
2 gpgcnt = .pcb[pcb$w_gpgcnt];
2
2 IF .display_mode NEQ 0
2 THEN
2     CH$FILL(0,maxvirt,vmap);                ! Initialize vmap to zero if needed
2
2 !
2 !       Set the IPL to SYNCH to avoid outswaps while examining the PHD.
2 !       The IPL is set from a longword at the end of the routine so that
2 !       the remainder of the routine is paged into memory as the IPL is set.
2 !
2
2 SET_IPL(.info_ipl);                         ! Set IPL to avoid outswaps
2
2 phd = .pcb [pcb$l_phd];                     ! Get PHD address
2
2 IF NOT .pcb[pcb$v_res]                      ! If swapped out,
2     OR .phd EQL 0                           ! Or if PHD address zero,
2 THEN
3     BEGIN
3     SET_IPL(0);                             ! Restore IPL
3     RETURN ss$_normal;
2     END;
2
2 cputime = .phd[phd$l_cputim];
2 dirio = .phd[phd$l_diocnt];
2 bufio = .phd[phd$l_biocnt];
2 pageflts = .phd[phd$l_pageflts];
2 vause = .phd[phd$l_frep0va]+(%x'80000000'-.phd [phd$l_frep1va]);
2 pc = .phd[phd$l_pc];
2 psl = .phd [phd$l_psl];
2 sp = .phd[phd$l_usp];
2
2 !
2 !       Copy the WSLE bits into vmap for each page in the working set.
2 !       Only do this if the mode is VMAP to minimize time at SYNCH.
2 !
2
2 IF .display_mode NEQ 0
2 THEN
3     BEGIN
3     REGISTER                                ! Speed the loop up a little
3         maxtest;
3     maxtest = maxvirt;
3
3     INCRA wsle FROM .phd+.phd[phd$w_wslist]*4 TO .phd+.phd[phd$w_wslast]*4 BY 4
3     DO
4         BEGIN
4         MAP wsle: REF BLOCK[,BYTE];                 ! Reference WSLE structure
4         vpn = ..wsle^-9;                    ! Get VPN of page in WS
4         IF .vpn LSSU .maxtest               ! If page within range of vmap,
4         THEN
4             vmap [.vpn,all] = .wsle [all];        !  -then copy attributes to vmap
3         END;
3
3     SET_IPL(0);                             ! Restore IPL back to zero
```

```
;   553          0548  3        vpn = .pc^-9;                       ! Get VPN of current PC
;   554          0549  3
;   555          0550  3        IF .vpn LSSU .maxtest               . If current PC is in vmap
;   556          0551  3        THEN
;   557          0552  3            vmap [.vpn,pc_in_page] = true;  ! Set bit indicating PC in this page
;   558          0553  3
;   559          0554  2        END;        ! End of gathering VMAP statistics
;   560          0555  2
;   561          0556  2 SET IPL(0);                                ! Restore IPL back to zero
;   562          0557  2 RETURN ss$_normal;
;   563          0558  2
;   564          0559  1 END;


                                                                  .EXTRN  SCH$GL_NULLPCB, EXE$EPID_TO_PCB

                                                  07FC 00000 KERNEL_GET_INFO:
                                                                  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10               ; 0437
                              5A 00000000G   00   9E 00002        MOVAB   PROC_L_PID, R10                                ;
                              59 00000000G   00   9E 00009        MOVAB   PROC_Z_NAME, R9                                ;
                              58      0000'  CF   9E 00010        MOVAB   PRI, R8                                        ;
                              50             6A   D0 00015        MOVL    PROC_L_PID, R0                                 ; 0470
                                 00000000G   00   16 00018        JSB     EXE$EPID_TO_PCB                                ;
                              56             50   D0 0001E        MOVL    R0, PCB                                        ;
                                             12   13 00021        BEQL    1$                                             ; 0472
                              50 00000000G   00   9E 00023        MOVAB   SCH$GL_NULLPCB, R0                             ; 0476
                              50             56   D1 0002A        CMPL    PCB, R0                                        ;
                                             06   13 0002D        BEQL    1$                                             ;
                              6A        64   A6   D1 0002F        CMPL    100(PCB), PROC_L_PID                           ; 0477
                                             06   13 00033        BEQL    2$                                             ;
                              50      08E8   8F   3C 00035 1$:    MOVZWL  #2280, R0                                      ; 0479
                                                  04 0003A        RET                                                    ;
                69        71  A6                  0F   28 0003B 2$:    MOVC3   #15, 113(PCB), PROC_Z_NAME                ; 0481
            00000000G    00             70   A6   9B 00040        MOVZBW  112(PCB), PROC_A_DESC                          ; 0482
            00000000G    00                  69   9E 00048        MOVAB   PROC_Z_NAME, PROC_A_DESC+4                     ; 0483
                         F4   A8      00BE   C6   3C 0004F        MOVZWL  190(PCB), GRP                                  ; 0484
                         F8   A8      00BC   C6   3C 00055        MOVZWL  188(PCB), MEM                                  ; 0485
                         FC   A8        2C   A6   3C 0005B        MOVZWL  44(PCB), STATE                                 ; 0486
                              68        0B   A6   9A 00060        MOVZBL  11(PCB), PRI                                   ; 0487
                68            1F             68   C3 00064        SUBL3   PRI, #31, PRI                                  ;
                         04   A8        2F   A6   9A 00068        MOVZBL  47(PCB), PRIB                                  ; 0488
           04   A8            1F        04   A8   C3 0006D        SUBL3   PRIB, #31, PRIB                                ;
                         30   A8        50   A6   7D 00073        MOVQ    80(PCB), LOCEVFLO                              ; 0489
                         10   A8        36   A6   3C 00078        MOVZWL  54(PCB), PPGCNT                                ; 0491
                         14   A8        34   A6   3C 0007D        MOVZWL  52(PCB), GPGCNT                                ; 0492
                                             57   D4 00082        CLRL    R7                                             ; 0494
                                        F0   A8   D5 00084        TSTL    DISPLAY_MODE                                   ;
                                             0B   13 00087        BEQL    3$                                             ;
                                             57   D6 00089        INCL    R7                                             ;
  0B80   8F             00         6E             00   2C 0008B        MOVC5   #0, (SP), #0, #2944, VMAP             ; 0496
                                        38   A8      00092                                                               ;
                              12     0000V   CF   DA 00094 3$:    MTPR    INFO_IPL, #18                                  ; 0504
                              50        6C   A6   D0 00099        MOVL    108(PCB), PHD                                  ; 0506
                              76        24   A6   E9 0009D        BLBC    36(PCB), 7$                                    ; 0508
                                             74   13 000A1        BEQL    7$                                             ; 0509
                         18   A8        38   A0   D0 000A3        MOVL    56(PHD), CPUTIME                               ; 0516
```

```
                   1C   A8       54   A0  7D 000A8         MOVQ    84(PHD), DIRIO                    ; 0517
                   24   A8       4C   A0  D0 000AD         MOVL    76(PHD), PAGEFLTS                 ; 0519
          51       28   A0       30   A0  C3 000B2         SUBL3   48(PHD), 40(PHD), R1              ; 0520
                   28   A8 80000000   E1  9E 000B8         MOVAB   -2147483648(R1), VAUSE            ;
                   08   A8     00C0   C0  D0 000C0         MOVL    192(PHD), PC                      ; 0521
                   2C   A8     00C4   C0  D0 000C6         MOVL    196(PHD), PSL                     ; 0522
                   0C   A8     0084   C0  D0 000CC         MOVL    132(PHD), SP                      ; 0523
                        42            57  E9 000D2         BLBC    R7, 7$                            ; 0530
                        51     0B80   8F  3C 000D5         MOVZWL  #2944, MAXTEST                    ; 0535
                        52       08   A0  3C 000DA         MOVZWL  8(PHD), R2                        ; 0537
                        53          6042  DE 000DE         MOVAL   (PHD)[R2], R3                     ;
                        52       12   A0  3C 000E2         MOVZWL  18(PHD), R2                       ;
                        52          6042  DE 000E6         MOVAL   (PHD)[R2], R2                     ;
                                      12  11 000EA         BRB     6$                                ; 0542
          50            63       F7   8F  78 000EC 4$:     ASHL    #-9, (WSLE), VPN                  ; 0541
                        51            50  D1 000F1         CMPL    VPN, MAXTEST                      ; 0542
                                      05  1E 000F4         BGEQU   5$                                ;
                   38 A840            63  90 000F6         MOVB    (WSLE), VMAP[VPN]                 ; 0544
                        53            04  C0 000FB 5$:     ADDL2   #4, WSLE                          ; 0537
                        52            53  D1 000FE 6$:     CMPL    WSLE, R2                          ;
                                      E9  1B 00101         BLEQU   4$                                ;
                        12            00  DA 00103         MTPR    #0, #18                           ; 0547
          50       08   A8       F7   8F  78 00106         ASHL    #-9, PC, VPN                      ; 0548
                        51            50  D1 0010C         CMPL    VPN, MAXTEST                      ; 0550
                                      06  1E 0010F         BGEQU   7$                                ;
                   38 A840       80   8F  88 00111         BISB2   #128, VMAP[VPN]                   ; 0552
                        12            00  DA 00117 7$:     MTPR    #0, #18                           ; 0556
                        50            01  D0 0011A         MOVL    #1, R0                            ; 0557
                                          04 0011D         RET                                       ; 0559

; Routine Size:  286 bytes,    Routine Base:  INFO_CODE + 0073


;   565         0560  1
;   566         0561  1 !
;   567         0562  1 ! The following PSECT manipulation is used to place the data segment INFO_IPL
;   568         0563  1 ! in the executable PSECT.  This will force the routine to be in-faulted as
;   569         0564  1 ! the IPL is changed.
;   570         0565  1 !
;   571         0566  1 PSECT OWN = INFO_CODE;                ! Make the following OWN follow routine
;   572         0567  1
;   573         0568  1 OWN info_ipl:   INITIAL(ipl$_synch);  ! Used by SET_IPL in previous routine
;   574         0569  1
;   575         0570  1 PSECT OWN = INFO_OWN;                 ! Restore OWN psect attributes
```

```
  1 ROUTINE wake_ast (ast_type) : NOVALUE =
  1
  1 !---
  1 !
  1 !       This routine is invoked when the user types any character or
  1 !       Control/C.
  1 !
  1 !---
  1
  2 BEGIN
  2
  2 OWN
  2     iosb : VECTOR [4, WORD];
  2
  2 IF .ast_type LSS 0 THEN                     ! If Control/C,
  2     tt_buffer [0] = 26;                     ! fake a CTRL/Z
  2
  2 IF .tt_buffer [0] EQL 'V'                   ! If vmap requested,
  2 THEN
  3     BEGIN
  3     new_display_mode = 1;                   ! set new vmap display mode
  3     mode_change = true;                     ! mark mode has changed
  3     END
  2 ELSE IF .tt_buffer [0] EQL ' '             ! If normal requested,
  2 THEN
  3     BEGIN
  3     new_display_mode = 0;                   ! set new normal display mode
  3     mode_change = true;                     ! mark mode has changed
  3     END
  2 ELSE IF .tt_buffer [0] EQL 'E'             ! If exit requested
  2      OR .tt_buffer [0] EQL 26              !  one way or another (CTRL/Z),
  2 THEN
  3     BEGIN
  3     keep_going = false;                     ! say time to exit
  2     END;
  2
  2 IF .keep_going THEN
P 2     keep_going = $QIO(EFN=2,CHAN=.tt_chan,
P 2             FUNC=IO$_READVBLK OR IO$M_NOECHO OR IO$M_CVTLOW,IOSB=iosb,
  2             ASTADR=wake_ast,P1=tt_buffer,P2=1);
  2
  2 !
  2 ! If the above QIO has already completed with an error, then terminate
  2 ! the image.  This fixed a bug which caused an AST level loop when a
  2 ! remote terminal link disappeared.
  2 !
  3 IF (.iosb [0] NEQ 0) AND (.iosb [0] NEQ SS$_NORMAL)
  2 THEN
  2     keep_going = false;
  2
  2 IF NOT .keep_going THEN
  2     $DASSGN(CHAN=.tt_chan);
  2
  1 END;
```

```
                                          00191            .BLKB   3
                                00000008  00194 INFO_IPL:
                                                           .LONG   8                                         ;

                                                           .PSECT  INFO_OWN,NOEXE,2

                                          00DC4 IOSB:      .BLKB   8

                                                           .EXTRN  SYS$QIO, SYS$DASSGN

                                                           .PSECT  INFO_CODE,NOWRT,2

                                     0004 00000 WAKE_AST:
                                                           .WORD   Save R2                                   ; 0571
                    52      0000'  CF  9E 00002            MOVAB   KEEP_GOING, R2                            ;
                               04  AC  D5 00007            TSTL    AST_TYPE                                  ; 0585
                                   04  18 0000A            BGEQ    1$                                        ;
             2C     A2             1A  90 0000C            MOVB    #26, TT_BUFFER                            ; 0586
                    50         2C  A2  9A 00010 1$:        MOVZBL  TT_BUFFER, R0                             ; 0588
             56     8F             50  91 00014            CMPB    R0, #86                                   ;
                                   06  12 00018            BNEQ    2$                                        ;
             1C     A2             01  D0 0001A            MOVL    #1, NEW_DISPLAY_MODE                      ; 0591
                                   08  11 0001E            BRB     3$                                        ; 0592
                    20             50  91 00020 2$:        CMPB    R0, #32                                   ; 0594
                                   09  12 00023            BNEQ    4$                                        ;
                               1C  A2  D4 00025            CLRL    NEW_DISPLAY_MODE                          ; 0597
             18     A2             01  D0 00028 3$:        MOVL    #1, MODE_CHANGE                           ; 0598
                                   0D  11 0002C            BRB     6$                                        ; 0594
             45     8F             50  91 0002E 4$:        CMPB    R0, #69                                   ; 0600
                                   05  13 00032            BEQL    5$                                        ;
                    1A             50  91 00034            CMPB    R0, #26                                   ; 0601
                                   02  12 00037            BNEQ    6$                                        ;
                                   62  D4 00039 5$:        CLRL    KEEP_GOING                                ; 0604
                    27             62  E9 0003B 6$:        BLBC    KEEP_GOING, 7$                            ; 0607
                                   7E  7C 0003E            CLRQ    -(SP)                                     ; 0610
                                   7E  7C 00040            CLRQ    -(SP)                                     ;
                                   01  DD 00042            PUSHL   #1                                        ;
                               2C  A2  9F 00044            PUSHAB  TT_BUFFER                                 ;
                                   7E  D4 00047            CLRL    -(SP)                                     ;
                               B4  AF  9F 00049            PUSHAB  WAKE_AST                                  ;
                             0DBC  C2  9F 0004C            PUSHAB  IOSB                                      ;
                    7E       0171  8F  3C 00050            MOVZWL  #369, -(SP)                               ;
                    7E         28  A2  3C 00055            MOVZWL  TT_CHAN, -(SP)                            ;
                                   02  DD 00059            PUSHL   #2                                        ;
          00000000G 00             0C  FB 0005B            CALLS   #12, SYS$QIO                              ;
                    62             50  D0 00062            MOVL    R0, KEEP_GOING                            ;
                    50       0DBC  C2  3C 00065 7$:        MOVZWL  IOSB, R0                                  ; 0617
                                   07  13 0006A            BEQL    8$                                        ;
                    01             50  B1 0006C            CMPW    R0, #1                                    ;
                                   02  13 0006F            BEQL    8$                                        ;
                                   62  D4 00071            CLRL    KEEP_GOING                                ; 0619
                    0B             62  E8 00073 8$:        BLBS    KEEP_GOING, 9$                            ; 0621
                    7E         28  A2  3C 00076            MOVZWL  TT_CHAN, -(SP)                            ; 0622
          00000000G 00             01  FB 0007A            CALLS   #1, SYS$DASSGN                            ;
                                       04 00081 9$:        RET                                               ; 0624

; Routine Size:  130 bytes,    Routine Base:  INFO_CODE + 0198
```

```
  1 GLOBAL ROUTINE proc_cont_display : NOVALUE =
  2 BEGIN
  2
  2 !---
  2 !
  2 !       This is the main entry point for the program.
  2 !
  2 !       This routine reads the process name from SYS$INPUT
  2 !       and displays process parameters on the screen.  The
  2 !       display is updated continuously.
  2 !
  2 !---
  2
  2 LITERAL
  2     buflen = 512;                           ! Size of terminal output buffer
  2
  2 LOCAL
  2     prev_vmap_buf: BLOCKVECTOR[maxvirt,BYTE,BYTE],
  2     status,                                 ! Status return from calls
  2     max_col,                                ! Maximum legal column for cursor
  2     vpn_per_col,                            ! # of vpn's per column (64 or 128)
  2     vpn_1st_col,                            ! 1st column used for vpn's
  2     pos,                                    ! Position of last PC on screen
  2     count,                                  ! Interations/image file read
  2     msec,                                   ! Wait time in millisecs
  2     quad_time: VECTOR[2] INITIAL (-100,-1),         ! Quad time for waits
  2     image:      VECTOR[nam$c_maxrss,BYTE],          ! Current image name
  2     image_desc: VECTOR[2],                  ! Descriptor for above buffer
  2     buffer:     VECTOR[buflen,BYTE],        ! Buffer for terminal output
  2     bufdesc:    VECTOR[2],                  ! Descriptor for above buffer
  2     GRP_DESC            : VECTOR [2],       ! Group name descr
  2     GRP_NAME            : VECTOR [KGB$S_NAME,BYTE],      ! Group name storage
  2     MEM_DESC            : VECTOR [2],   ! Member name descr
  2     MEM_NAME            : VECTOR [KGB$S_NAME,BYTE],      ! Member name storage
  2     CONVERTED_UIC       : REF VECTOR;   ! Addr of alpha UIC descr
  2
  2 BIND
P 2     state_table = nametable(sch$c_,
P 2         COLPG,MWAIT,CEF,PFW,LEF,LEFO,HIB,HIBO,SUSP,SUSPO,
  2         FPG,COM,COMO,CUR);
  2
  2 !
  2 !       Post-initialization link to SCR
  2 !
  2
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                       %ascid'LIB$SCREEN_INFO', $LIB$SCREEN_INFO);
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                       %ascid'SCR$ERASE_LINE', $SCR$ERASE_LINE);
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                       %ascid'SCR$ERASE_PAGE', $SCR$ERASE_PAGE);
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                       %ascid'SCR$PUT_BUFFER', $SCR$PUT_BUFFER);
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                       %ascid'SCR$PUT_SCREEN', $SCR$PUT_SCREEN);
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                       %ascid'SCR$SET_BUFFER', $SCR$SET_BUFFER);
  2 LIB$FIND_IMAGE_SYMBOL(%ascid'SCRSHR',
  2                         %ascid'SCR$SET_CURSOR', $SCR$SET_CURSOR);
  2
  2
  2 prev_vmap = prev_vmap_buf;              ! Store the address into the REF
  2
  2 !
  2 !       Make certain portions of OWN storage non-paged to avoid
  2 !       pagefaults by kernel mode code which runs at a high IPL.
  2 !       Make most of the rest of the own storage non-paged for
  2 !       performance.
  2 !
  2
  2 bufdesc [0] = lock_start;               ! Starting address of program
  2 bufdesc [1] = lock_end;                 ! Ending address of program
  2 perform($LKWSET(INADR=bufdesc));        ! Lock down pages to avoid paging
  2
  2 !
  2 !       Make sure we can handle the screen and get its width & height
  2 !
  2
  3 BEGIN
  3     LOCAL
  3         type;
  3     perform(lib$screen_info(dev_flags,type,max_col,max_row));
  4     IF (.max_col LSS 72) OR (.max_row LSS 10)
  3         THEN
  4             BEGIN
  4             SIGNAL(rms$_dev);           ! Call it an inappropriate device...
  4             RETURN;
  3             END;
  3     IF .max_row LSS 20 then spacing = 1;        ! Single space if small screen
  3     IF .max_col LSS 132 THEN                    ! Less than 132 columns gets
  3         vpn_per_col = 64                        !  64 vpn's per column
  3     ELSE                                        ! elsewise
  3         vpn_per_col = 128;                      !  128 vpn's per column
  3     vpn_1st_col = MINU(10,.max_col-.vpn_per_col)+1; ! Find 1st vpn column
  3     max_row = MINU((maxvirt/.vpn_per_col)+1,.max_row); ! Limit max to vmap max
  2 END;
  2
  2 !
  2 ! Get a channel to the terminal, set it to FULLDUPLEX, and post a read on it
  2 !
  2
  3 BEGIN
  3 LOCAL
  3     dib_buf: BLOCK[dib$c_length,BYTE],
  3     dib_buf_desc: VECTOR[2];
  3 perform($ASSIGN(CHAN=tt_chan,DEVNAM=%ASCID 'TT'));
  3 dib_buf_desc[0] = dib$c_length;
  3 dib_buf_desc[1] = dib_buf;
  3 perform($GETCHN(CHAN=.tt_chan,PRIBUF=dib_buf_desc));
  3 dib_buf[dib$l_devdepend] = .dib_buf[dib$l_devdepend] AND NOT tt$m_halfdup;
P 3 perform($QIOW(CHAN=.tt_chan,
  3         FUNC=IO$_SETMODE,P1=dib_buf[dib$b_devclass]));
P 3 perform($QIOW(CHAN=.tt_chan,
  3         FUNC=IO$_SETMODE OR IO$M_CTRLCAST,P1=wake_ast,P2=-1));
```

```
perform($QIO(EFN=2,CHAN=.tt_chan,
        FUNC=IO$_READVBLK OR IO$M_NOECHO OR IO$M_CVTLOW,
        ASTADR=wake_ast,P1=tt_buffer,P2=1));
END;

proc_a_desc [dsc$w_length] = 0;         ! Display name first time

!
!       Declare exit handler
!

perform($DCLEXH(DESBLK = exit_block));  ! Declare exit handler

!
!       Setup screen buffer to optimize screen output
!

bufdesc[0] = buflen;                    ! Buffer length
bufdesc[1] = buffer;                    ! Address of buffer
scr$set_buffer(bufdesc);                ! Enable output buffering

!
!       Display the page heading information
!

DO
    BEGIN
    IF .mode_change                     ! If display mode just changed,
    THEN
        BEGIN
        mode_change = false;            ! then mark it no longer just changed
        display_mode = .new_display_mode; ! and set to new display mode
        scr$erase_page(1,1);            ! Erase the entire screen
        IF .display_mode EQL 0          ! If normal display
        THEN
            BEGIN
            at(.spacing*1,30);  write('Process ');
            at(.spacing*3,5);   write('State');
            at(.spacing*4,5);   write('Cur/base priority');
            at(.spacing*5,5);   write('Current PC');
            at(.spacing*6,5);   write('Current PSL');
            at(.spacing*7,5);   write('Current user SP');
            at(.spacing*8,5);   write('PID');
            at(.spacing*8,25);  write('!XL',.proc_l_pid);
            at(.spacing*9,5);   write('UIC');
            at(.spacing*3,45);  write('Working set');
            at(.spacing*4,45);  write('Virtual pages');
            at(.spacing*5,45);  write('CPU time');
            at(.spacing*6,45);  write('Direct I/O');
            at(.spacing*7,45);  write('Buffered I/O');
            at(.spacing*8,45);  write('Page faults');
            at(.spacing*9,45);  write('Event flags');
            END
        ELSE                            ! Else, if vmap display
            BEGIN
            INCR line FROM 1 TO .max_row-1 ! Label each line with starting VA
            DO
```

```
  6                 BEGIN
  6                 LOCAL
  6                     vpn_1st_addr;
  6                 vpn_1st_addr = (.line-1)*.vpn_per_col*512;
  6                 at(.line,1);
  6                 IF .vpn_1st_col GEQ 8 THEN
  6                     write('!#XL:', .vpn_1st_col-3, .vpn_1st_addr)
  6                 ELSE
P 6                     write('!#XL:', .vpn_1st_col-3, IF .vpn_per_col EQL 64 THEN
P 6                                                        .vpn_1st_addr/4096
P 6                                                    ELSE
  6                                                      .vpn_1st_addr/65536);
  5                 END;
  5             at(.max_row,3); write('PC:');
  5             at(.max_row,17); write('State:');
  4             END;
  4
  4         pos = 1;                            ! Initial PC display location
  4         image_desc[0] = 0;                  ! Initially null image name
  4         count = 99;                         ! Interations/image read; force 1st one
  4         CH$FILL(0,(.max_row-1)*.vpn_per_col,vmap); ! Initialize vmap
  4         grp = -1;
  4         mem = -1;
  4         state = -1;
  4         pri = -1;
  4         prib = -1;
  4         pc = -1;
  4         sp = -1;
  4         ppgcnt = -1;
  4         gpgcnt = -1;
  4         cputime = -1;
  4         dirio = -1;
  4         bufio = -1;
  4         pageflts = -1;
  4         locevfl0 = -1;
  4         locevfl1 = -1;
  4         vause = -1;
  4         psl = -1;
  4         proc_a_desc [dsc$w_length] = 0;
  3         END;
  3
  3 !
  3 !       Keep refreshing the information that changes
  3 !
  3
  3     status = .prev_pc EQL .pc           ! Set flag true if quiesent
  3                 AND .prev_cputime EQL .cputime;
  3
  3     CH$MOVE(.proc_a_desc[dsc$w_length],.proc_a_desc[dsc$a_pointer],
  3         prev_prcnam);
  3     prev_desc[dsc$w_length] = .proc_a_desc[dsc$w_length];
  3     prev_desc[dsc$a_pointer] = prev_prcnam;
  3     CH$MOVE(.image_desc[0],.image_desc[1],prev_image);
  3     prev_imgdesc[0] = .image_desc[0];
  3     prev_imgdesc[1] = prev_image;
  3     prev_state = .state;
  3     prev_pri = .pri;
  3     prev_prib = .prib;
  3     prev_grp = .grp;
  3     prev_mem = .mem;
  3     prev_pc = .pc;
  3     prev_psl = .psl;
  3     prev_sp = .sp;
  3     prev_ppgcnt = .ppgcnt;
  3     prev_gpgcnt = .gpgcnt;
  3     prev_cputime = .cputime;
  3     prev_dirio = .dirio;
  3     prev_bufio = .bufio;
  3     prev_pageflts = .pageflts;
  3     prev_locevfl0 = .locevfl0;
  3     prev_locevfl1 = .locevfl1;
  3     prev_vause = .vause;
  3     CH$MOVE((.max_row-1)*.vpn_per_col, vmap, .prev_vmap);
  3
  3     !
  3     ! Set the time delay according to type of terminal and activity of
  3     ! process, then update all the stats.
  3     !
  3     msec =                                  ! Compute msecs to wait
  4                 (IF .dev_flags AND 1        !  -if screen oriented device
  5                 THEN (IF .status            !  --and if process quiescent
  5                         THEN 750            !  --then .75 seconds
  5                         ELSE 100)           !  --else .10 seconds
  3                 ELSE 2000);                 !  -if not screen then 2 seconds.
  3     quad_time [0] = (.msec * -10000);       ! Convert to 100 usec units
  3
  3     perform($SETIMR(efn=1,daytim=quad_time));   ! Event flag 1
  3     perform($WAITFR(efn=1));                    ! Wait to complete
  3     perform($CMKRNL(routin=kernel_get_info));   ! Get fresh copies of everything
  3
  3     count = .count+1;                       ! Increment count/image read
  3     IF .count GEQ 5                         ! Every 5 interations,
  3     THEN
  4         BEGIN
  4         LOCAL
  4             iosb:        VECTOR [4,WORD],       ! I/O status block
  4             item_list:  BLOCK [16,BYTE];       ! GETJPI item list
  4
  4         count = 0;                          ! Reset counter
  4         image_desc [1] = image;
  4         item_list [0,0,16,0] = 128;      ! Buffer length
  4         item_list [2,0,16,0] = jpi$_imagname;   ! Request image filespec
  4         item_list [4,0,32,0] = .image_desc [1]; ! Buffer address
  4         item_list [8,0,32,0] = image_desc [0];  ! Return length address
  4         item_list [12,0,32,0] = 0;       ! Terminate list
  4
P 4         IF $GETJPI(ITMLST = item_list,  ! Read image filespec
P 4                 PIDADR = proc_l_pid,
P 4                 EFN = 0,
  5                 IOSB = iosb)
  4         THEN
  5             BEGIN
  5             perform($WAITFR(EFN = 0));  ! Wait for completion
  5
```

```
;  917      0910  5              IF NOT .iosb [0]             ! If deferred error detected
;  918      0911  5              THEN
;  919      0912  5                  image_desc [0] = 0;      ! then force string to null
;  920      0913  5              END
;  921      0914  4          ELSE                             ! If error detected by GETJPI
;  922      0915  4              image_desc [0] = 0;          ! then force string to null
;  923      0916  3          END;
;  924      0917  3
;  925      0918  3      IF .display_mode EQL 0               ! If normal display,
;  926      0919  4      THEN BEGIN
;  927      0920  4      at(.spacing*1,65); write('!8%T',0); ! Write current time
;  928      0921  4      IF CH$NEQ(.prev_desc[dsc$w_length],.prev_desc[dsc$a_pointer],
;  929      0922  4          .proc_a_desc[dsc$w_length],.proc_a_desc[dsc$a_pointer],' ')
;  930      0923  5          THEN BEGIN at(.spacing*1,38);
;  931      0924  5          write('!15<!AF!>', .proc_a_desc[dsc$w_length], .proc_a_desc[dsc$a_pointer]);
;  932      0925  4          END;
;  933      0926  4      IF CH$NEQ(.prev_imgdesc[0],.prev_imgdesc[1],
;  934      0927  4          .image_desc[0],.image_desc[1],' ')
;  935      0928  5          THEN BEGIN
;  936      0929  5          at(.spacing*11,5); write('!AS',image_desc); scr$erase_line();
;  937      0930  5          IF .spacing EQL 1 THEN prev_locevfl1 = .locevfl1+1;
;  938      0931  4          END;
;  939      0932  5      IF .prev_state NEQ .state THEN BEGIN
;  940      0933  5          at(.spacing*3,25); write('!6AC',translate_value(.state,state_table));
;  941      0934  4          END;
;  942      0935  5      IF .prev_pri NEQ .pri THEN BEGIN
;  943      0936  5          at(.spacing*4,24); write('!2UB/',.pri);
;  944      0937  4          END;
;  945      0938  5      IF .prev_prib NEQ .prib THEN BEGIN
;  946      0939  5          at(.spacing*4,27); write('!UB',.prib);
;  947      0940  4          END;
;  948      0941  5      IF .prev_pc NEQ .pc THEN BEGIN
;  949      0942  5          at(.spacing*5,25); write('!XL',.pc);
;  950      0943  4          END;
;  951      0944  5      IF .prev_psl NEQ .psl THEN BEGIN
;  952      0945  5          at(.spacing*6,25); write('!XL',.psl);
;  953      0946  4          END;
;  954      0947  5      IF .prev_sp NEQ .sp THEN BEGIN
;  955      0948  5          at(.spacing*7,25); write('!XL',.sp);
;  956      0949  4          END;
;  957      0950  4      IF .PREV_GRP NEQ .GRP OR .PREV_MEM NEQ .MEM
;  958      0951  4      THEN
;  959      0952  5          BEGIN
;  960      0953  5          LOCAL   UIC;
;  961      0954  5          UIC = .GRP<0,16>^16 OR .MEM<0,16>;
;  962      0955  5          CONVERTED_UIC = FAO('!%I', .UIC);
;  963      0956  5          IF .CONVERTED_UIC[0] LSS 35
;  964      0957  5          THEN
;  965      0958  6              BEGIN
;  966      0959  6              IF .CONVERTED_UIC[0] LEQ 19
;  967      0960  7              THEN (AT (.SPACING*9,25); WRITE('!%I',.UIC))
;  968      0961  6              ELSE (AT (.SPACING*9,9); WRITE('!#< !>!%I',35 - .CONVERTED_UIC[0], .UIC));
;  969      0962  6              END
;  970      0963  5          ELSE
;  971      0964  6              BEGIN
;  972      0965  6              GRP_DESC[0] = KGB$S_NAME;
;  973      0966  6              GRP_DESC[1] = GRP_NAME;
```

```
  6                     MEM_DESC[0] = KGB$S_NAME;
  6                     MEM_DESC[1] = MEM_NAME;
P 6                     $IDTOASC (ID = .GRP^16 OR %X'0000FFFF',
P 6                               NAMLEN = GRP_DESC,
  6                               NAMBUF = GRP_DESC);
P 6                     $IDTOASC (ID = .UIC,
P 6                               NAMLEN = MEM_DESC,
  6                               NAMBUF = MEM_DESC);
  6                     AT (.SPACING*9,10); WRITE('[!AS',GRP_DESC);
  6                     AT (.SPACING*9,11+.GRP_DESC[0]); WRITE (',');
  6                     AT (.SPACING*9+1,11); QRITE ('!AS]',MEM_DESC);
  5                     END;
  4                 END;
  5             IF .prev_ppgcnt+.prev_gpgcnt NEQ .ppgcnt+.gpgcnt THEN BEGIN
  5                 at(.spacing*3,67); write('!6UL',.ppgcnt+.gpgcnt);
  4                 END;
  5             IF .prev_vause NEQ .vause THEN BEGIN
  5                 at(.spacing*4,66); write('!7UL',.vause/512);
  4                 END;
  5             IF .prev_cputime NEQ .cputime THEN BEGIN
P 5                 at(.spacing*5,62); write('!2ZL:!2ZL:!2ZL.!2ZL',
P 5                                 (.cputime/(100*60*60)) MOD 24,
P 5                                 (.cputime/(100*60)) MOD 60,
P 5                                 (.cputime/100) MOD 60,
  5                                 .cputime MOD 100);
  4                 END;
  5             IF .prev_dirio NEQ .dirio THEN BEGIN
  5                 at(.spacing*6,65); write('!8UL',.dirio);
  4                 END;
  5             IF .prev_bufio NEQ .bufio THEN BEGIN
  5                 at(.spacing*7,65); write('!8UL',.bufio);
  4                 END;
  5             IF .prev_pageflts NEQ .pageflts THEN BEGIN
  5                 at(.spacing*8,65); write('!8UL',.pageflts);
  4                 END;
  5             IF .prev_locevfl0 NEQ .locevfl0 THEN BEGIN
  5                 at(.spacing*9,65); write('!XL',.locevfl0);
  4                 END;
  5             IF .prev_locevfl1 NEQ .locevfl1 THEN BEGIN
  5                 at(.spacing*9+1,65-1); write(' !XL',.locevfl1);
  4                 END;
  4             END
  3             ELSE                                ! Else, if display mode is vmap
  4                 BEGIN
  4                 LOCAL vpn,prev_vpn,last_at;
  4                 last_at = -2;
  4                 IF CH$NEQ((.max_row-1)*.vpn_per_col, vmap,
  4                           (.max_row-1)*.vpn_per_col, .prev_vmap, 0)
  4                 THEN
  4                     INCR i FROM 0 TO ((.max_row-1)*.vpn_per_col)-1 ! For each page in virtual space,
  4                     DO
  4                         IF .vmap [.i,all] NEQ .prev_vmap [.i,all] ! If virtual page residency has changed,
  4                         THEN
  5                             BEGIN
  5                             IF .i NEQ .last_at+1        ! If not already at position,
  5                                 OR .i MOD .vpn_per_col EQL 0 ! or skipping to next line
  5                             THEN
```

```
5                       at_vpn(.i);                 ! Set cursor position to VPN
5                   last_at = .i;                   ! Save current cursor position
5                   show_vpn(.i);                   ! Display virtual page
4                   END;
4
4           !
4           ! Put current PC, State, and Image name on last line
4           !
4
5           IF .prev_pc NEQ .pc THEN BEGIN
5               at(.max_row,7); write('!XL',.pc);
4               END;
5           IF .prev_state NEQ .state THEN BEGIN
5               at(.max_row,24); write('!6AC',translate_value(.state,state_table));
4               END;
4           IF CH$NEQ(.prev_imgdesc[0],.prev_imgdesc[1],
4                   .image_desc[0],.image_desc[1],' ')
5               THEN BEGIN
5               at(.max_row,32); write('!AS',image_desc); scr$erase_line();
4               END;
3           END;
3
3       at(.max_row,1);
3       scr$put_buffer();
3       scr$set_buffer(bufdesc);            ! Output contents of buffer
3
2       END WHILE .keep_going;
2
2 RETURN;
1 END;
```

```
                                                  .PSECT  INFO_PLIT,NOWRT,NOEXE,2

                          05  00007 P.AAC:   .BYTE   5                              ;
        47  50  4C  4F  43  00008            .ASCII  \COLPG\                        ;
                          05  0000D P.AAD:   .BYTE   5                              ;
        54  49  41  57  4D  0000E            .ASCII  \MWAIT\                        ;
                          03  00013 P.AAE:   .BYTE   3                              ;
                46  45  43  00014            .ASCII  \CEF\                          ;
                          03  00017 P.AAF:   .BYTE   3                              ;
                57  46  50  00018            .ASCII  \PFW\                          ;
                          03  0001B P.AAG:   .BYTE   3                              ;
                46  45  4C  0001C            .ASCII  \LEF\                          ;
                          04  0001F P.AAH:   .BYTE   4                              ;
            4F  46  45  4C  00020            .ASCII  \LEFO\                         ;
                          03  00024 P.AAI:   .BYTE   3                              ;
                42  49  48  00025            .ASCII  \HIB\                          ;
                          04  00028 P.AAJ:   .BYTE   4                              ;
            4F  42  49  48  00029            .ASCII  \HIBO\                         ;
                          04  0002D P.AAK:   .BYTE   4                              ;
            50  53  55  53  0002E            .ASCII  \SUSP\                         ;
                          05  00032 P.AAL:   .BYTE   5                              ;
        4F  50  53  55  53  00033            .ASCII  \SUSPO\                        ;
                          03  00038 P.AAM:   .BYTE   3                              ;
                47  50  46  00039            .ASCII  \FPG\                          ;
```

```
                                                 03  0003C P.AAN:  .BYTE    3                                    ;
                                          4D  4F  43  0003D         .ASCII   \COM\                                ;
                                                 04  00040 P.AAO:  .BYTE    4                                    ;
                                      4F  4D  4F  43  00041         .ASCII   \COMO\                               ;
                                                 03  00045 P.AAP:  .BYTE    3                                    ;
                                          52  55  43  00046         .ASCII   \CUR\                                ;
                                                     00049         .BLKB    3
                                           00000001  0004C P.AAB:  .LONG    1                                    ;
                                           00000000' 00050         .ADDRESS P.AAC                                ;
                                           00000002  00054         .LONG    2                                    ;
                                           00000000' 00058         .ADDRESS P.AAD                                ;
                                           00000003  0005C         .LONG    3                                    ;
                                           00000000' 00060         .ADDRESS P.AAE                                ;
                                           00000004  00064         .LONG    4                                    ;
                                           00000000' 00068         .ADDRESS P.AAF                                ;
                                           00000005  0006C         .LONG    5                                    ;
                                           00000000' 00070         .ADDRESS P.AAG                                ;
                                           00000006  00074         .LONG    6                                    ;
                                           00000000' 00078         .ADDRESS P.AAH                                ;
                                           00000007  0007C         .LONG    7                                    ;
                                           00000000' 00080         .ADDRESS P.AAI                                ;
                                           00000008  00084         .LONG    8                                    ;
                                           00000000' 00088         .ADDRESS P.AAJ                                ;
                                           00000009  0008C         .LONG    9                                    ;
                                           00000000' 00090         .ADDRESS P.AAK                                ;
                                           0000000A  00094         .LONG    10                                   ;
                                           00000000' 00098         .ADDRESS P.AAL                                ;
                                           0000000B  0009C         .LONG    11                                   ;
                                           00000000' 000A0         .ADDRESS P.AAM                                ;
                                           0000000C  000A4         .LONG    12                                   ;
                                           00000000' 000A8         .ADDRESS P.AAN                                ;
                                           0000000D  000AC         .LONG    13                                   ;
                                           00000000' 000B0         .ADDRESS P.AAO                                ;
                                           0000000E  000B4         .LONG    14                                   ;
                                           00000000' 000B8         .ADDRESS P.AAP                                ;
                                  FFFFFFFF FFFFFFFF  000BC         .LONG    -1, -1                               ;
                    00  00  52  48  53  52  43  53  000C4 P.AAR:  .ASCII   \SCRSHR\<0><0>                       ;
                                           010E0006  000CC P.AAQ:  .LONG    17694726                             ;
                                           00000000' 000D0         .ADDRESS P.AAR                                ;
4F  46  4E  49  5F  4E  45  45  52  43  53  24  42  49  4C  000D4 P.AAT:  .ASCII   \LIB$SCREEN_INFO\<0>                 ;
                                                 00  000E3                                                       ;
                                           010E000F  000E4 P.AAS:  .LONG    17694735                             ;
                                           00000000' 000E8         .ADDRESS P.AAT                                ;
                    00  00  52  48  53  52  43  53  000EC P.AAV:  .ASCII   \SCRSHR\<0><0>                       ;
                                           010E0006  000F4 P.AAU:  .LONG    17694726                             ;
                                           00000000' 000F8         .ADDRESS P.AAV                                ;
00  45  4E  49  4C  5F  45  53  41  52  45  24  52  43  53  000FC P.AAX:  .ASCII   \SCR$ERASE_LINE\<0><0>               ;
                                                 00  0010B                                                       ;
                                           010E000E  0010C P.AAW:  .LONG    17694734                             ;
                                           00000000' 00110         .ADDRESS P.AAX                                ;
                    00  00  52  48  53  52  43  53  00114 P.AAZ:  .ASCII   \SCRSHR\<0><0>                       ;
                                           010E0006  0011C P.AAY:  .LONG    17694726                             ;
                                           00000000' 00120         .ADDRESS P.AAZ                                ;
00  45  47  41  50  5F  45  53  41  52  45  24  52  43  53  00124 P.ABB:  .ASCII   \SCR$ERASE_PAGE\<0><0>               ;
                                                 00  00133                                                       ;
                                           010E000E  00134 P.ABA:  .LONG    17694734                             ;
                                           00000000' 00138         .ADDRESS P.ABB                                ;
```

```
                              00  00  52  48  53  52  43  53   0013C P.ABD:   .ASCII   \SCRSHR\<0><0>
                                                  010E0006   00144 P.ABC:   .LONG    17694726
                                                  00000000'  00148          .ADDRESS P.ABD
00  52  45  46  46  55  42  5F  54  55  50  24  52  43  53   0014C P.ABF:   .ASCII   \SCR$PUT_BUFFER\<0><0>
                                                        00   0015B
                                                  010E000E   0015C P.ABE:   .LONG    17694734
                                                  00000000'  00160          .ADDRESS P.ABF
                              00  00  52  48  53  52  43  53   00164 P.ABH:   .ASCII   \SCRSHR\<0><0>
                                                  010E0006   0016C P.ABG:   .LONG    17694726
                                                  00000000'  00170          .ADDRESS P.ABH
00  4E  45  45  52  43  53  5F  54  55  50  24  52  43  53   00174 P.ABJ:   .ASCII   \SCR$PUT_SCREEN\<0><0>
                                                        00   00183
                                                  010E000E   00184 P.ABI:   .LONG    17694734
                                                  00000000'  00188          .ADDRESS P.ABJ
                              00  00  52  48  53  52  43  53   0018C P.ABL:   .ASCII   \SCRSHR\<0><0>
                                                  010E0006   00194 P.ABK:   .LONG    17694726
                                                  00000000'  00198          .ADDRESS P.ABL
00  52  45  46  46  55  42  5F  54  45  53  24  52  43  53   0019C P.ABN:   .ASCII   \SCR$SET_BUFFER\<0><0>
                                                        00   001AB
                                                  010E000E   001AC P.ABM:   .LONG    17694734
                                                  00000000'  001B0          .ADDRESS P.ABN
                              00  00  52  48  53  52  43  53   001B4 P.ABP:   .ASCII   \SCRSHR\<0><0>
                                                  010E0006   001BC P.ABO:   .LONG    17694726
                                                  00000000'  001C0          .ADDRESS P.ABP
00  52  4F  53  52  55  43  5F  54  45  53  24  52  43  53   001C4 P.ABR:   .ASCII   \SCR$SET_CURSOR\<0><0>
                                                        00   001D3
                                                  010E000E   001D4 P.ABQ:   .LONG    17694734
                                                  00000000'  001D8          .ADDRESS P.ABR
                                          00  00  54  54   001DC P.ABT:   .ASCII   \TT\<0><0>
                                                  010E0002   001E0 P.ABS:   .LONG    17694722
                                                  00000000'  001E4          .ADDRESS P.ABT
                              20  73  73  65  63  6F  72  50   001E8 P.ABV:   .ASCII   \Process \
                                                  010E0008   001F0 P.ABU:   .LONG    17694728
                                                  00000000'  001F4          .ADDRESS P.ABV
                              00  00  00  65  74  61  74  53   001F8 P.ABX:   .ASCII   \State\<0><0><0>
                                                  010E0005   00200 P.ABW:   .LONG    17694725
                                                  00000000'  00204          .ADDRESS P.ABX
69  72  6F  69  72  70  20  65  73  61  62  2F  72  75  43   00208 P.ABZ:   .ASCII   \Cur/base priority\<0><0><0>
                                          00  00  00  79  74   00217
                                                  010E0011   0021C P.ABY:   .LONG    17694737
                                                  00000000'  00220          .ADDRESS P.ABZ
                  00  00  43  50  20  74  6E  65  72  72  75  43   00224 P.ACB:   .ASCII   \Current PC\<0><0>
                                                  010E000A   00230 P.ACA:   .LONG    17694730
                                                  00000000'  00234          .ADDRESS P.ACB
                  00  4C  53  50  20  74  6E  65  72  72  75  43   00238 P.ACD:   .ASCII   \Current PSL\<0>
                                                  010E000B   00244 P.ACC:   .LONG    17694731
                                                  00000000'  00248          .ADDRESS P.ACD
50  53  20  72  65  73  75  20  74  6E  65  72  72  75  43   0024C P.ACF:   .ASCII   \Current user SP\<0>
                                                        00   0025B
                                                  010E000F   0025C P.ACE:   .LONG    17694735
                                                  00000000'  00260          .ADDRESS P.ACF
                                          00  44  49  50   00264 P.ACH:   .ASCII   \PID\<0>
                                                  010E0003   00268 P.ACG:   .LONG    17694723
                                                  00000000'  0026C          .ADDRESS P.ACH
                                          00  4C  58  21   00270 P.ACJ:   .ASCII   \!XL\<0>
                                                  010E0003   00274 P.ACI:   .LONG    17694723
                                                  00000000'  00278          .ADDRESS P.ACJ
```

```
                                    00  43  49  55  0027C P.ACL:  .ASCII  \UIC\<0>
                                        010E0003  00280 P.ACK:  .LONG   17694723
                                        00000000' 00284         .ADDRESS P.ACL
            00  74  65  73  20  67  6E  69  6B  72  6F  57  00288 P.ACN:  .ASCII  \Working set\<0>
                                        010E000B  00294 P.ACM:  .LONG   17694731
                                        00000000' 00298         .ADDRESS P.ACN
00  00  73  65  67  61  70  20  6C  61  75  74  72  69  56  0029C P.ACP:  .ASCII  \Virtual pages\<0><0><0>
                                                00  002AB
                                        010E000D  002AC P.ACO:  .LONG   17694733
                                        00000000' 002B0         .ADDRESS P.ACP
                        65  6D  69  74  20  55  50  43  002B4 P.ACR:  .ASCII  \CPU time\
                                        010E0008  002BC P.ACQ:  .LONG   17694728
                                        00000000' 002C0         .ADDRESS P.ACR
            00  00  4F  2F  49  20  74  63  65  72  69  44  002C4 P.ACT:  .ASCII  \Direct I/O\<0><0>
                                        010E000A  002D0 P.ACS:  .LONG   17694730
                                        00000000' 002D4         .ADDRESS P.ACT
            4F  2F  49  20  64  65  72  65  66  66  75  42  002D8 P.ACV:  .ASCII  \Buffered I/O\
                                        010E000C  002E4 P.ACU:  .LONG   17694732
                                        00000000' 002E8         .ADDRESS P.ACV
            00  73  74  6C  75  61  66  20  65  67  61  50  002EC P.ACX:  .ASCII  \Page faults\<0>
                                        010E000B  002F8 P.ACW:  .LONG   17694731
                                        00000000' 002FC         .ADDRESS P.ACX
            00  73  67  61  6C  66  20  74  6E  65  76  45  00300 P.ACZ:  .ASCII  \Event flags\<0>
                                        010E000B  0030C P.ACY:  .LONG   17694731
                                        00000000' 00310         .ADDRESS P.ACZ
                        00  00  00  3A  4C  58  23  21  00314 P.ADB:  .ASCII  \!#XL:\<0><0><0>
                                        010E0005  0031C P.ADA:  .LONG   17694725
                                        00000000' 00320         .ADDRESS P.ADB
                        00  00  00  3A  4C  58  23  21  00324 P.ADD:  .ASCII  \!#XL:\<0><0><0>
                                        010E0005  0032C P.ADC:  .LONG   17694725
                                        00000000' 00330         .ADDRESS P.ADD
                                    00  3A  43  50  00334 P.ADF:  .ASCII  \PC:\<0>
                                        010E0003  00338 P.ADE:  .LONG   17694723
                                        00000000' 0033C         .ADDRESS P.ADF
                        00  00  3A  65  74  61  74  53  00340 P.ADH:  .ASCII  \State:\<0><0>
                                        010E0006  00348 P.ADG:  .LONG   17694726
                                        00000000' 0034C         .ADDRESS P.ADH
                                    54  25  38  21  00350 P.ADJ:  .ASCII  \!8%T\
                                        010E0004  00354 P.ADI:  .LONG   17694724
                                        00000000' 00358         .ADDRESS P.ADJ
            00  00  00  3E  21  46  41  21  3C  35  31  21  0035C P.ADL:  .ASCII  \!15<!AF!>\<0><0><0>
                                        010E0009  00368 P.ADK:  .LONG   17694729
                                        00000000' 0036C         .ADDRESS P.ADL
                                    00  53  41  21  00370 P.ADN:  .ASCII  \!AS\<0>
                                        010E0003  00374 P.ADM:  .LONG   17694723
                                        00000000' 00378         .ADDRESS P.ADN
                                    43  41  36  21  0037C P.ADP:  .ASCII  \!6AC\
                                        010E0004  00380 P.ADO:  .LONG   17694724
                                        00000000' 00384         .ADDRESS P.ADP
                        00  00  00  2F  42  55  32  21  00388 P.ADR:  .ASCII  \!2UB/\<0><0><0>
                                        010E0005  00390 P.ADQ:  .LONG   17694725
                                        00000000' 00394         .ADDRESS P.ADR
                                    00  42  55  21  00398 P.ADT:  .ASCII  \!UB\<0>
                                        010E0003  0039C P.ADS:  .LONG   17694723
                                        00000000' 003A0         .ADDRESS P.ADT
                                    00  4C  58  21  003A4 P.ADV:  .ASCII  \!XL\<0>
                                        010E0003  003A8 P.ADU:  .LONG   17694723
```

```
                                      00000000' 003AC            .ADDRESS P.ADV
                                 00  4C  58  21 003B0 P.ADX:    .ASCII  \!XL\<0>
                                      010E0003  003B4 P.ADW:    .LONG   17694723
                                      00000000' 003B8            .ADDRESS P.ADX
                                 00  4C  58  21 003BC P.ADZ:    .ASCII  \!XL\<0>
                                      010E0003  003C0 P.ADY:    .LONG   17694723
                                      00000000' 003C4            .ADDRESS P.ADZ
                                 00  49  25  21 003C8 P.AEB:    .ASCII  \!%I\<0>
                                      010E0003  003CC P.AEA:    .LONG   17694723
                                      00000000' 003D0            .ADDRESS P.AEB
                                 00  49  25  21 003D4 P.AED:    .ASCII  \!%I\<0>
                                      010E0003  003D8 P.AEC:    .LONG   17694723
                                      00000000' 003DC            .ADDRESS P.AED
         00  00  00  49  25  21  3E  21  20  3C  23  21 003E0 P.AEF:    .ASCII  \!#< !>!%I\<0><0><0>
                                      010E0009  003EC P.AEE:    .LONG   17694729
                                      00000000' 003F0            .ADDRESS P.AEF
                                 53  41  21  5B 003F4 P.AEH:    .ASCII  \[!AS\
                                      010E0004  003F8 P.AEG:    .LONG   17694724
                                      00000000' 003FC            .ADDRESS P.AEH
                                 00  00  00  2C 00400 P.AEJ:    .ASCII  \,\<0><0><0>
                                      010E0001  00404 P.AEI:    .LONG   17694721
                                      00000000' 00408            .ADDRESS P.AEJ
                                 5D  53  41  21 0040C P.AEL:    .ASCII  \!AS]\
                                      010E0004  00410 P.AEK:    .LONG   17694724
                                      00000000' 00414            .ADDRESS P.AEL
                                 4C  55  36  21 00418 P.AEN:    .ASCII  \!6UL\
                                      010E0004  0041C P.AEM:    .LONG   17694724
                                      00000000' 00420            .ADDRESS P.AEN
                                 4C  55  37  21 00424 P.AEP:    .ASCII  \!7UL\
                                      010E0004  00428 P.AEO:    .LONG   17694724
                                      00000000' 0042C            .ADDRESS P.AEP
2E  4C  5A  32  21  3A  4C  5A  32  21  3A  4C  5A  32  21 00430 P.AER:    .ASCII  \!2ZL:!2ZL:!2ZL.!2ZL\<0>
                             00  4C  5A  32  21 0043F
                                      010E0013  00444 P.AEQ:    .LONG   17694739
                                      00000000' 00448            .ADDRESS P.AER
                                 4C  55  38  21 0044C P.AET:    .ASCII  \!8UL\
                                      010E0004  00450 P.AES:    .LONG   17694724
                                      00000000' 00454            .ADDRESS P.AET
                                 4C  55  38  21 00458 P.AEV:    .ASCII  \!8UL\
                                      010E0004  0045C P.AEU:    .LONG   17694724
                                      00000000' 00460            .ADDRESS P.AEV
                                 4C  55  38  21 00464 P.AEX:    .ASCII  \!8UL\
                                      010E0004  00468 P.AEW:    .LONG   17694724
                                      00000000' 0046C            .ADDRESS P.AEX
                                 00  4C  58  21 00470 P.AEZ:    .ASCII  \!XL\<0>
                                      010E0003  00474 P.AEY:    .LONG   17694723
                                      00000000' 00478            .ADDRESS P.AEZ
                                 4C  58  21  20 0047C P.AFB:    .ASCII  \ !XL\
                                      010E0004  00480 P.AFA:    .LONG   17694724
                                      00000000' 00484            .ADDRESS P.AFB
                                 00  00  00  40 00488 P.AFD:    .ASCII  \@\<0><0><0>
                                      010E0001  0048C P.AFC:    .LONG   17694721
                                      00000000' 00490            .ADDRESS P.AFD
                                 00  00  00  4C 00494 P.AFF:    .ASCII  \L\<0><0><0>
                                      010E0001  00498 P.AFE:    .LONG   17694721
                                      00000000' 0049C            .ADDRESS P.AFF
                                 00  00  00  47 004A0 P.AFH:    .ASCII  \G\<0><0><0>
```

```
                      010E0001  004A4 P.AFG:  .LONG    17694721                                   ;
                     00000000'  004A8         .ADDRESS P.AFH                                      ;
              00  00  00  2A  004AC P.AFJ:  .ASCII   \*\<0><0><0>                                ;
                      010E0001  004B0 P.AFI:  .LONG    17694721                                   ;
                     00000000'  004B4         .ADDRESS P.AFJ                                      ;
              00  00  00  20  004B8 P.AFL:  .ASCII   \ \<0><0><0>                                ;
                      010E0001  004BC P.AFK:  .LONG    17694721                                   ;
                     00000000'  004C0         .ADDRESS P.AFL                                      ;
              00  4C  58  21  004C4 P.AFN:  .ASCII   \!XL\<0>                                    ;
                      010E0003  004C8 P.AFM:  .LONG    17694723                                   ;
                     00000000'  004CC         .ADDRESS P.AFN                                      ;
              43  41  36  21  004D0 P.AFP:  .ASCII   \!6AC\                                      ;
                      010E0004  004D4 P.AFO:  .LONG    17694724                                   ;
                     00000000'  004D8         .ADDRESS P.AFP                                      ;
              00  53  41  21  004DC P.AFR:  .ASCII   \!AS\<0>                                    ;
                      010E0003  004E0 P.AFQ:  .LONG    17694723                                   ;
                     00000000'  004E4         .ADDRESS P.AFR                                      ;

                                      STATE_TABLE=         P.AAB
                                              .EXTRN   SYS$LKWSET, SYS$ASSIGN
                                              .EXTRN   SYS$GETCHN, SYS$QIOW
                                              .EXTRN   SYS$DCLEXH, SYS$SETIMR
                                              .EXTRN   SYS$WAITFR, SYS$CMKRNL
                                              .EXTRN   SYS$GETJPI, SYS$IDTOASC

                                              .PSECT   INFO_CODE,NOWRT,2

                             0FFC 00000         .ENTRY   PROC_CONT_DISPLAY, Save R2,R3,R4,R5,R6,R7,- ; 0625
                                                         R8,R9,R10,R11                                ;
                5E      F08C   CE   9E 00002         MOVAB    -3956(SP), SP                                ;
   03EC         CE        9C   8F   98 00007         CVTBL    #-100, QUAD_TIME                             ; 0626
   03F0         CE              01   CE 0000D         MNEGL    #1, QUAD_TIME+4                              ;
                       0000'   CF   9F 00012         PUSHAB   $LIB$SCREEN_INFO                             ; 0670
                       0000'   CF   9F 00016         PUSHAB   P.AAS                                        ;
                       0000'   CF   9F 0001A         PUSHAB   P.AAQ                                        ;
00000000G  00                  03   FB 0001E         CALLS    #3, LIB$FIND_IMAGE_SYMBOL                    ;
                       0000'   CF   9F 00025         PUSHAB   $SCR$ERASE_LINE                              ; 0672
                       0000'   CF   9F 00029         PUSHAB   P.AAW                                        ;
                       0000'   CF   9F 0002D         PUSHAB   P.AAU                                        ;
00000000G  00                  03   FB 00031         CALLS    #3, LIB$FIND_IMAGE_SYMBOL                    ;
                       0000'   CF   9F 00038         PUSHAB   $SCR$ERASE_PAGE                              ; 0674
                       0000'   CF   9F 0003C         PUSHAB   P.ABA                                        ;
                       0000'   CF   9F 00040         PUSHAB   P.AAY                                        ;
00000000G  00                  03   FB 00044         CALLS    #3, LIB$FIND_IMAGE_SYMBOL                    ;
                       0000'   CF   9F 0004B         PUSHAB   $SCR$PUT_BUFFER                              ; 0676
                       0000'   CF   9F 0004F         PUSHAB   P.ABE                                        ;
                       0000'   CF   9F 00053         PUSHAB   P.ABC                                        ;
00000000G  00                  03   FB 00057         CALLS    #3, LIB$FIND_IMAGE_SYMBOL                    ;
                       0000'   CF   9F 0005E         PUSHAB   $SCR$PUT_SCREEN                              ; 0678
                       0000'   CF   9F 00062         PUSHAB   P.ABI                                        ;
                       0000'   CF   9F 00066         PUSHAB   P.ABG                                        ;
00000000G  00                  03   FB 0006A         CALLS    #3, LIB$FIND_IMAGE_SYMBOL                    ;
                       0000'   CF   9F 00071         PUSHAB   $SCR$SET_BUFFER                              ; 0680
                       0000'   CF   9F 00075         PUSHAB   P.ABM                                        ;
                       0000'   CF   9F 00079         PUSHAB   P.ABK                                        ;
00000000G  00                  03   FB 0007D         CALLS    #3, LIB$FIND_IMAGE_SYMBOL                    ;
                       0000'   CF   9F 00084         PUSHAB   $SCR$SET_CURSOR                              ; 0682
```

```
                          0000'  CF  9F 00088          PUSHAB  P.ABQ                                  ;
                          0000'  CF  9F 0008C          PUSHAB  P.ABO                                  ;
    00000000G  00                03  FB 00090          CALLS   #3, LIB$FIND_IMAGE_SYMBOL              ;
        0000'  CF          03F4  CE  9E 00097          MOVAB   PREV_VMAP_BUF, PREV_VMAP               ; 0686
         00DC  CE         0000'  CF  9E 0009E          MOVAB   LOCK_START, BUFDESC                    ; 0695
         00E0  CE         0000'  CF  9E 000A5          MOVAB   LOCK_END, BUFDESC+4                    ; 0696
                                 7E  7C 000AC          CLRQ    -(SP)                                  ; 0697
                           00E4  CE  9F 000AE          PUSHAB  BUFDESC                                ;
    00000000G  00                03  FB 000B2          CALLS   #3, SYS$LKWSET                         ;
               57                50  D0 000B9          MOVL    R0, STATUS                             ;
               16                57  E9 000BC          BLBC    STATUS, 1$                             ;
                          0000'  CF  9F 000BF          PUSHAB  MAX_ROW                                ; 0706
                             0C  AE  9F 000C3          PUSHAB  MAX_COL                                ;
                             14  AE  9F 000C6          PUSHAB  TYPE                                   ;
                          0000'  CF  9F 000C9          PUSHAB  DEV_FLAGS                              ;
        0000'  DF                04  FB 000CD          CALLS   #4, @$LIB$SCREEN_INFO                  ;
               57                50  D0 000D2          MOVL    R0, STATUS                             ;
               78                57  E9 000D5  1$:     BLBC    STATUS, 9$                             ;
    00000048   8F            08  AE  D1 000D8          CMPL    MAX_COL, #72                           ; 0707
                                 07  19 000E0          BLSS    2$                                     ;
               0A         0000'  CF  D1 000E2          CMPL    MAX_ROW, #10                           ;
                                 09  18 000E7          BGEQ    3$                                     ;
                       000184C4  8F  DD 000E9  2$:     PUSHL   #99524                                 ; 0710
                             0592    31 000EF          BRW     32$                                    ;
               14         0000'  CF  D1 000F2  3$:     CMPL    MAX_ROW, #20                           ; 0713
                                 05  18 000F7          BGEQ    4$                                     ;
        0000'  CF                01  D0 000F9          MOVL    #1, SPACING                            ;
    00000084   8F            08  AE  D1 000FE  4$:     CMPL    MAX_COL, #132                          ; 0714
                                 06  18 00106          BGEQ    5$                                     ;
               59            40  8F  9A 00108          MOVZBL  #64, VPN_PER_COL                       ; 0715
                                 04  11 0010C          BRB     6$                                     ;
               59            80  8F  9A 0010E  5$:     MOVZBL  #128, VPN_PER_COL                      ; 0717
56        08   AE                59  C3 00112  6$:     SUBL3   VPN_PER_COL, MAX_COL, R6               ; 0718
               0A                56  D1 00117          CMPL    R6, #10                                ;
                                 03  1B 0011A          BLEQU   7$                                     ;
               56                0A  D0 0011C          MOVL    #10, R6                                ;
                                 56  D6 0011F  7$:     INCL    VPN_1ST_COL                            ;
50 00000B80    8F                59  C7 00121          DIVL3   VPN_PER_COL, #2944, R0                 ; 0719
                                 50  D6 00129          INCL    R0                                     ;
        0000'  CF                50  D1 0012B          CMPL    R0, MAX_ROW                            ;
                                 05  1B 00130          BLEQU   8$                                     ;
               50         0000'  CF  D0 00132          MOVL    MAX_ROW, R0                            ;
        0000'  CF                50  D0 00137  8$:     MOVL    R0, MAX_ROW                            ;
                                 7E  7C 0013C          CLRQ    -(SP)                                  ; 0730
                          0000'  CF  9F 0013E          PUSHAB  TT_CHAN                                ;
                          0000'  CF  9F 00142          PUSHAB  P.ABS                                  ;
    00000000G  00                04  FB 00146          CALLS   #4, SYS$ASSIGN                         ;
               57                50  D0 0014D          MOVL    R0, STATUS                             ;
               6E                57  E9 00150  9$:     BLBC    STATUS, 10$                            ;
          10   AE            74  8F  9A 00153          MOVZBL  #116, DIB_BUF_DESC                     ; 0731
          14   AE            18  AE  9E 00158          MOVAB   DIB_BUF, DIB_BUF_DESC+4                ; 0732
                                 7E  7C 0015D          CLRQ    -(SP)                                  ; 0733
                             18  AE  9F 0015F          PUSHAB  DIB_BUF_DESC                           ;
                                 7E  D4 00162          CLRL    -(SP)                                  ;
               7E         0000'  CF  3C 00164          MOVZWL  TT_CHAN, -(SP)                         ;
    00000000G  00                05  FB 00169          CALLS   #5, SYS$GETCHN                         ;
               57                50  D0 00170          MOVL    R0, STATUS                             ;
```

```
                                         K 15
SHOW$PROCESS_CO                          16-Sep-1984 00:05:41    VAX-11 Bliss-32 V4.0-742                  Page  35
V04-000                                  14-Sep-1984 12:08:33    DISK$VMSMASTER:[CLIUTL.SRC]INFO.B32;1      (8)

                76              57 E9 00173       BLBC    STATUS, 11$                         ;
           22   AE              10 8A 00176       BICB2   #16, DIB_BUF+8                      ; 0734
                                7E 7C 0017A       CLRQ    -(SP)                               ; 0736
                                7E 7C 0017C       CLRQ    -(SP)                               ;
                                7E D4 0017E       CLRL    -(SP)                               ;
                           30   AE 9F 00180       PUSHAB  DIB_BUF+4                           ;
                                7E 7C 00183       CLRQ    -(SP)                               ;
                7E              23 7D 00185       MOVQ    #35, -(SP)                          ;
                7E      0000'   CF 3C 00188       MOVZWL  TT_CHAN, -(SP)                      ;
                                7E D4 0018D       CLRL    -(SP)                               ;
    00000000G   00              0C FB 0018F       CALLS   #12, SYS$QIOW                       ;
                57              50 D0 00196       MOVL    R0, STATUS                          ;
                67              57 E9 00199       BLBC    STATUS, 12$                         ;
                                7E 7C 0019C       CLRQ    -(SP)                               ; 0738
                                7E 7C 0019E       CLRQ    -(SP)                               ;
                7E              01 CE 001A0       MNEGL   #1, -(SP)                           ;
                         FDD7   CF 9F 001A3       PUSHAB  WAKE_AST                            ;
                                7E 7C 001A7       CLRQ    -(SP)                               ;
                                7E D4 001A9       CLRL    -(SP)                               ;
                7E       0123   8F 3C 001AB       MOVZWL  #291, -(SP)                         ;
                7E      0000'   CF 3C 001B0       MOVZWL  TT_CHAN, -(SP)                      ;
                                7E D4 001B5       CLRL    -(SP)                               ;
    00000000G   00              0C FB 001B7       CALLS   #12, SYS$QIOW                       ;
                57              50 D0 001BE       MOVL    R0, STATUS                          ;
                3F              57 E9 001C1 10$:  BLBC    STATUS, 12$                         ;
                                7E 7C 001C4       CLRQ    -(SP)                               ; 0741
                                7E 7C 001C6       CLRQ    -(SP)                               ;
                                01 DD 001C8       PUSHL   #1                                  ;
                        0000'   CF 9F 001CA       PUSHAB  TT_BUFFER                           ;
                                7E D4 001CE       CLRL    -(SP)                               ;
                         FDAA   CF 9F 001D0       PUSHAB  WAKE_AST                            ;
                                7E D4 001D4       CLRL    -(SP)                               ;
                7E       0171   8F 3C 001D6       MOVZWL  #369, -(SP)                         ;
                7E      0000'   CF 3C 001DB       MOVZWL  TT_CHAN, -(SP)                      ;
                                02 DD 001E0       PUSHL   #2                                  ;
    00000000G   00              0C FB 001E2       CALLS   #12, SYS$QIO                        ;
                57              50 D0 001E9       MOVL    R0, STATUS                          ;
                14              57 E9 001EC 11$:  BLBC    STATUS, 12$                         ;
                    00000000G   00 B4 001EF       CLRW    PROC_A_DESC                         ; 0744
                        0000'   CF 9F 001F5       PUSHAB  EXIT_BLOCK                          ; 0750
    00000000G   00              01 FB 001F9       CALLS   #1, SYS$DCLEXH                      ;
                57              50 D0 00200       MOVL    R0, STATUS                          ;
                03              57 E8 00203 12$:  BLBS    STATUS, 13$                         ;
                              0479 31 00206       BRW     31$                                 ;
         00DC   CE       0200   8F 3C 00209 13$:  MOVZWL  #512, BUFDESC                       ; 0756
         00E0   CE       00E4   CE 9E 00210       MOVAB   BUFFER, BUFDESC+4                   ; 0757
                         00DC   CE 9F 00217       PUSHAB  BUFDESC                             ; 0758
        0000'   DF              01 FB 0021B       CALLS   #1, @$SCR$SET_BUFFER                ;
                03      0000'   CF E8 00220 14$:  BLBS    MODE_CHANGE, 15$                    ; 0766
                              02FA 31 00225       BRW     25$                                 ;
                        0000'   CF D4 00228 15$:  CLRL    MODE_CHANGE                         ; 0769
        0000'   CF      0000'   CF D0 0022C       MOVL    NEW_DISPLAY_MODE, DISPLAY_MODE      ; 0770
                                01 DD 00233       PUSHL   #1                                  ; 0771
                                01 DD 00235       PUSHL   #1                                  ;
        0000'   DF              02 FB 00237       CALLS   #2, @$SCR$ERASE_PAGE                ;
                        0000'   CF D5 0023C       TSTL    DISPLAY_MODE                        ; 0772
                                03 13 00240       BEQL    16$                                 ;
```

```
                                      01CB  31 00242          BRW     17$                                  ;
                                        1E  DD 00245 16$:     PUSHL   #30                                  ; 0775
                                 0000'  CF  DD 00247          PUSHL   SPACING                              ;
                 0000'   DF             02  FB 0024B          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 00250          PUSHAB  P.ABU                                ;
                 FBA0    CF             01  FB 00254          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 00259          PUSHL   R0                                   ;
                 0000'   DF             01  FB 0025B          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        05  DD 00260          PUSHL   #5                                   ; 0776
        7E       0000'   CF             03  C5 00262          MULL3   #3, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 00268          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 0026D          PUSHAB  P.ABW                                ;
                 FB83    CF             01  FB 00271          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 00276          PUSHL   R0                                   ;
                 0000'   DF             01  FB 00278          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        05  DD 0027D          PUSHL   #5                                   ; 0777
        7E       0000'   CF             02  78 0027F          ASHL    #2, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 00285          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 0028A          PUSHAB  P.ABY                                ;
                 FB66    CF             01  FB 0028E          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 00293          PUSHL   R0                                   ;
                 0000'   DF             01  FB 00295          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        05  DD 0029A          PUSHL   #5                                   ; 0778
        7E       0000'   CF             05  C5 0029C          MULL3   #5, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 002A2          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 002A7          PUSHAB  P.ACA                                ;
                 FB49    CF             01  FB 002AB          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 002B0          PUSHL   R0                                   ;
                 0000'   DF             01  FB 002B2          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        05  DD 002B7          PUSHL   #5                                   ; 0779
        7E       0000'   CF             06  C5 002B9          MULL3   #6, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 002BF          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 002C4          PUSHAB  P.ACC                                ;
                 FB2C    CF             01  FB 002C8          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 002CD          PUSHL   R0                                   ;
                 0000'   DF             01  FB 002CF          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        05  DD 002D4          PUSHL   #5                                   ; 0780
        7E       0000'   CF             07  C5 002D6          MULL3   #7, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 002DC          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 002E1          PUSHAB  P.ACE                                ;
                 FB0F    CF             01  FB 002E5          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 002EA          PUSHL   R0                                   ;
                 0000'   DF             01  FB 002EC          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        05  DD 002F1          PUSHL   #5                                   ; 0781
        7E       0000'   CF             03  78 002F3          ASHL    #3, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 002F9          CALLS   #2, @SCR$SET_CURSOR                  ;
                                 0000'  CF  9F 002FE          PUSHAB  P.ACG                                ;
                 FAF2    CF             01  FB 00302          CALLS   #1, FAO_BUFFER                       ;
                                        50  DD 00307          PUSHL   R0                                   ;
                 0000'   DF             01  FB 00309          CALLS   #1, @SCR$PUT_SCREEN                  ;
                                        19  DD 0030E          PUSHL   #25                                  ; 0782
        7E       0000'   CF             03  78 00310          ASHL    #3, SPACING, -(SP)                   ;
                 0000'   DF             02  FB 00316          CALLS   #2, @SCR$SET_CURSOR                  ;
                             00000000G  00  DD 0031B          PUSHL   PROC_L_PID                           ;
                                 0000'  CF  9F 00321          PUSHAB  P.ACI                                ;
                 FACF    CF             02  FB 00325          CALLS   #2, FAO_BUFFER                       ;
                                        50  DD 0032A          PUSHL   R0                                   ;
```

```
           0000'  DF           01  FB 0032C         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               05  DD 00331         PUSHL   #5                                      ; 0783
     7E    0000'  CF           09  C5 00333         MULL3   #9, SPACING, -(SP)                      ;
           0000'  DF           02  FB 00339         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 0033E         PUSHAB  P.ACK                                   ;
           FAB2   CF           01  FB 00342         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 00347         PUSHL   R0                                      ;
           0000'  DF           01  FB 00349         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 0034E         PUSHL   #45                                     ; 0784
     7E    0000'  CF           03  C5 00350         MULL3   #3, SPACING, -(SP)                      ;
           0000'  DF           02  FB 00356         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 0035B         PUSHAB  P.ACM                                   ;
           FA95   CF           01  FB 0035F         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 00364         PUSHL   R0                                      ;
           0000'  DF           01  FB 00366         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 0036B         PUSHL   #45                                     ; 0785
     7E    0000'  CF           02  78 0036D         ASHL    #2, SPACING, -(SP)                      ;
           0000'  DF           02  FB 00373         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 00378         PUSHAB  P.ACO                                   ;
           FA78   CF           01  FB 0037C         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 00381         PUSHL   R0                                      ;
           0000'  DF           01  FB 00383         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 00388         PUSHL   #45                                     ; 0786
     7E    0000'  CF           05  C5 0038A         MULL3   #5, SPACING, -(SP)                      ;
           0000'  DF           02  FB 00390         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 00395         PUSHAB  P.ACQ                                   ;
           FA5B   CF           01  FB 00399         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 0039E         PUSHL   R0                                      ;
           0000'  DF           01  FB 003A0         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 003A5         PUSHL   #45                                     ; 0787
     7E    0000'  CF           06  C5 003A7         MULL3   #6, SPACING, -(SP)                      ;
           0000'  DF           02  FB 003AD         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 003B2         PUSHAB  P.ACS                                   ;
           FA3E   CF           01  FB 003B6         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 003BB         PUSHL   R0                                      ;
           0000'  DF           01  FB 003BD         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 003C2         PUSHL   #45                                     ; 0788
     7E    0000'  CF           07  C5 003C4         MULL3   #7, SPACING, -(SP)                      ;
           0000'  DF           02  FB 003CA         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 003CF         PUSHAB  P.ACU                                   ;
           FA21   CF           01  FB 003D3         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 003D8         PUSHL   R0                                      ;
           0000'  DF           01  FB 003DA         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 003DF         PUSHL   #45                                     ; 0789
     7E    0000'  CF           03  78 003E1         ASHL    #3, SPACING, -(SP)                      ;
           0000'  DF           02  FB 003E7         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 003EC         PUSHAB  P.ACW                                   ;
           FA04   CF           01  FB 003F0         CALLS   #1, FAO_BUFFER                          ;
                               50  DD 003F5         PUSHL   R0                                      ;
           0000'  DF           01  FB 003F7         CALLS   #1, @$SCR$PUT_SCREEN                    ;
                               2D  DD 003FC         PUSHL   #45                                     ; 0790
     7E    0000'  CF           09  C5 003FE         MULL3   #9, SPACING, -(SP)                      ;
           0000'  DF           02  FB 00404         CALLS   #2, @$SCR$SET_CURSOR                    ;
                        0000'  CF  9F 00409         PUSHAB  P.ACY                                   ;
                            008E   31 0040D         BRW     24$                                     ;
                  55    0000'  CF  D0 00410 17$:    MOVL    MAX_ROW, R5                             ; 0794
                  54       FD  A6  9E 00415         MOVAB   -3(R6), R4                              ; 0802
```

```
                                          52  D4 00419          CLRL    LINE                                  ;
                                          53  11 0041B          BRB     23$                                   ;
                      50            FF    A2  9E 0041D  18$:    MOVAB   -1(R2), R0                            ; 0799
                      50                  59  C4 00421          MULL2   VPN_PER_COL, R0                       ;
             53       50                  09  78 00424          ASHL    #9, R0, VPN_1ST_ADDR                  ;
                                          01  DD 00428          PUSHL   #1                                    ; 0800
                                          52  DD 0042A          PUSHL   LINE                                  ;
                  0000'   DF              02  FB 0042C          CALLS   #2, @$SCR$SET_CURSOR                  ;
                          08              56  D1 00431          CMPL    VPN_1ST_COL, #8                       ; 0801
                                          0A  19 00434          BLSS    19$                                   ;
                                          53  DD 00436          PUSHL   VPN_1ST_ADDR                          ; 0802
                                          54  DD 00438          PUSHL   R4                                    ;
                                 0000'    CF  9F 0043A          PUSHAB  P.ADA                                 ;
                                          24  11 0043E          BRB     22$                                   ;
               00000040   8F              59  D1 00440  19$:    CMPL    VPN_PER_COL, #64                      ; 0807
                                          0B  12 00447          BNEQ    20$                                   ;
                          53 00001000     8F  C6 00449          DIVL2   #4096, R3                             ;
                                          53  DD 00450          PUSHL   R3                                    ;
                                          0A  11 00452          BRB     21$                                   ;
             50           53 00010000     8F  C7 00454  20$:    DIVL3   #65536, VPN_1ST_ADDR, R0              ;
                                          50  DD 0045C          PUSHL   R0                                    ;
                                          54  DD 0045E  21$:    PUSHL   R4                                    ;
                                 0000'    CF  9F 00460          PUSHAB  P.ADC                                 ;
                  F990    CF              03  FB 00464  22$:    CALLS   #3, FAO_BUFFER                        ;
                                          50  DD 00469          PUSHL   R0                                    ;
                  0000'   DF              01  FB 0046B          CALLS   #1, @$SCR$PUT_SCREEN                  ;
             A9           52              55  F2 00470  23$:    AOBLSS  R5, LINE, 18$                         ; 0794
                                          03  DD 00474          PUSHL   #3                                    ; 0809
                                 0000'    CF  DD 00476          PUSHL   MAX_ROW                               ;
                  0000'   DF              02  FB 0047A          CALLS   #2, @$SCR$SET_CURSOR                  ;
                                 0000'    CF  9F 0047F          PUSHAB  P.ADE                                 ;
                  F971    CF              01  FB 00483          CALLS   #1, FAO_BUFFER                        ;
                                          50  DD 00488          PUSHL   R0                                    ;
                  0000'   DF              01  FB 0048A          CALLS   #1, @$SCR$PUT_SCREEN                  ;
                                          11  DD 0048F          PUSHL   #17                                   ; 0810
                                 0000'    CF  DD 00491          PUSHL   MAX_ROW                               ;
                  0000'   DF              02  FB 00495          CALLS   #2, @$SCR$SET_CURSOR                  ;
                                 0000'    CF  9F 0049A          PUSHAB  P.ADG                                 ;
                  F956    CF              01  FB 0049E  24$:    CALLS   #1, FAO_BUFFER                        ;
                                          50  DD 004A3          PUSHL   R0                                    ;
                  0000'   DF              01  FB 004A5          CALLS   #1, @$SCR$PUT_SCREEN                  ;
                    04    AE              01  D0 004AA          MOVL    #1, POS                               ; 0813
                                  02E4    CE  D4 004AE          CLRL    IMAGE_DESC                            ; 0814
                          5B        63    8F  9A 004B2          MOVZBL  #99, COUNT                            ; 0815
             50   0000'   CF              01  C3 004B6          SUBL3   #1, MAX_ROW, R0                       ; 0816
                          50              59  C4 004BC          MULL2   VPN_PER_COL, R0                       ;
     50      00           6E              00  2C 004BF          MOVC5   #0, (SP), #0, R0, VMAP                ;
                                 0000'    CF     004C4                                                        ;
                  0000'   CF              01  CE 004C7          MNEGL   #1, GRP                               ; 0817
                  0000'   CF              01  CE 004CC          MNEGL   #1, MEM                               ; 0818
                  0000'   CF              01  CE 004D1          MNEGL   #1, STATE                             ; 0819
                  0000'   CF              01  CE 004D6          MNEGL   #1, PRI                               ; 0820
                  0000'   CF              01  CE 004DB          MNEGL   #1, PRIB                              ; 0821
                  0000'   CF              01  CE 004E0          MNEGL   #1, PC                                ; 0822
                  0000'   CF              01  CE 004E5          MNEGL   #1, SP                                ; 0823
                  0000'   CF              01  CE 004EA          MNEGL   #1, PPGCNT                            ; 0824
                  0000'   CF              01  CE 004EF          MNEGL   #1, GPGCNT                            ; 0825
```

```
                                                                   B 16
SHOW$PROCESS_CO                                                  16-Sep-1984 00:05:41    VAX-11 Bliss-32 V4.0-742              Page  39
V04-000                                                          14-Sep-1984 12:08:33    DISK$VMSMASTER:[CLIUTL.SRC]INFO.B32;1   (8)

              0000'  CF            01  CE 004F4         MNEGL   #1, CPUTIME                                 ; 0826
              0000'  CF            01  CE 004F9         MNEGL   #1, DIRIO                                   ; 0827
              0000'  CF            01  CE 004FE         MNEGL   #1, BUFIO                                   ; 0828
              0000'  CF            01  CE 00503         MNEGL   #1, PAGEFLTS                                ; 0829
              0000'  CF            01  CE 00508         MNEGL   #1, LOCEVFLO                                ; 0830
              0000'  CF            01  CE 0050D         MNEGL   #1, LOCEVFL1                                ; 0831
              0000'  CF            01  CE 00512         MNEGL   #1, VAUSE                                   ; 0832
              0000'  CF            01  CE 00517         MNEGL   #1, PSL                                     ; 0833
                        00000000G  00  B4 0051C         CLRW    PROC_A_DESC                                 ; 0834
                                   51  D4 00522  25$:   CLRL    R1                                          ; 0841
              0000'  CF     0000'  CF  D1 00524         CMPL    PREV_PC, PC                                 ;
                                   02  12 0052B         BNEQ    26$                                         ;
                                   51  D6 0052D         INCL    R1                                          ;
                                   50  D4 0052F  26$:   CLRL    R0                                          ; 0842
              0000'  CF     0000'  CF  D1 00531         CMPL    PREV_CPUTIME, CPUTIME                       ;
                                   02  12 00538         BNEQ    27$                                         ;
                                   50  D6 0053A         INCL    R0                                          ;
                     57            51  D2 0053C  27$:   MCOML   R1, STATUS                                  ;
       57            50            57  CB 0053F         BICL3   STATUS, R0, STATUS                          ;
                     58 00000000G  00  3C 00543         MOVZWL  PROC_A_DESC, R8                             ; 0844
                     50 00000000G  00  D0 0054A         MOVL    PROC_A_DESC+4, R0                           ;
0000'  CF            60            58  28 00551         MOVC3   R8, (R0), PREV_PRCNAM                       ;
              0000'  CF            58  B0 00557         MOVW    R8, PREV_DESC                               ; 0846
              0000'  CF     0000'  CF  9E 0055C         MOVAB   PREV_PRCNAM, PREV_DESC+4                    ; 0847
0000'  CF     02E8   DE     02E4   CE  28 00563         MOVC3   IMAGE_DESC, @IMAGE_DESC+4, PREV_IMAGE       ; 0848
              0000'  CF     02E4   CE  D0 0056D         MOVL    IMAGE_DESC, PREV_IMGDESC                    ; 0849
              0000'  CF     0000'  CF  9E 00574         MOVAB   PREV_IMAGE, PREV_IMGDESC+4                  ; 0850
              0000'  CF     0000'  CF  7D 0057B         MOVQ    STATE, PREV_STATE                           ; 0851
              0000'  CF     0000'  CF  D0 00582         MOVL    PRIB, PREV_PRIB                             ; 0853
              0000'  CF     0000'  CF  7D 00589         MOVQ    GRP, PREV_GRP                               ; 0854
              0000'  CF     0000'  CF  D0 00590         MOVL    PSL, PREV_PSL                               ; 0857
              0000'  CF     0000'  CF  7D 00597         MOVQ    PC, PREV_PC                                 ; 0856
              0000'  CF     0000'  CF  7D 0059E         MOVQ    PPGCNT, PREV_PPGCNT                         ; 0859
              0000'  CF     0000'  CF  7D 005A5         MOVQ    CPUTIME, PREV_CPUTIME                       ; 0861
              0000'  CF     0000'  CF  7D 005AC         MOVQ    BUFIO, PREV_BUFIO                           ; 0863
              0000'  CF     0000'  CF  7D 005B3         MOVQ    LOCEVFLO, PREV_LOCEVFLO                     ; 0865
              0000'  CF     0000'  CF  D0 005BA         MOVL    VAUSE, PREV_VAUSE                           ; 0867
       50     0000'  CF            01  C3 005C1         SUBL3   #1, MAX_ROW, R0                             ; 0868
                     50            59  C4 005C7         MULL2   VPN_PER_COL, R0                             ;
0000'  DF     0000'  CF            50  28 005CA         MOVC3   R0, VMAP, @PREV_VMAP                        ;
                     10     0000'  CF  E9 005D2         BLBC    DEV_FLAGS, 29$                              ; 0875
                     07            57  E9 005D7         BLBC    STATUS, 28$                                 ; 0876
                     5A      02EE  8F  3C 005DA         MOVZWL  #750, MSEC                                  ;
                     0B            11  11 005DF         BRB     30$                                         ;
                     5A        64  8F  9A 005E1  28$:   MOVZBL  #100, MSEC                                  ;
                     05            11  11 005E5         BRB     30$                                         ;
                     5A      07D0  8F  3C 005E7  29$:   MOVZWL  #2000, MSEC                                 ; 0875
03EC   CE            5A FFFFD8F0   8F  C5 005EC  30$:   MULL3   #-10000, MSEC, QUAD_TIME                    ; 0880
                                   7E  7C 005F6         CLRQ    -(SP)                                       ; 0882
                             03F4  CE  9F 005F8         PUSHAB  QUAD_TIME                                   ;
                                   01  DD 005FC         PUSHL   #1                                          ;
          00000000G  00            04  FB 005FE         CALLS   #4, SYS$SETIMR                              ;
                     57            50  D0 00605         MOVL    R0, STATUS                                  ;
                     77            57  E9 00608         BLBC    STATUS, 31$                                 ;
                                   01  DD 0060B         PUSHL   #1                                          ; 0883
          00000000G  00            01  FB 0060D         CALLS   #1, SYS$WAITFR                              ;
                     57            50  D0 00614         MOVL    R0, STATUS                                  ;
```

```
                            68           57 E9 00617       BLBC    STATUS, 31$                                  ;
                                         7E D4 0061A       CLRL    -(SP)                                        ; 0884
                                    F839 CF 9F 0061C       PUSHAB  KERNEL_GET_INFO                              ;
                   00000000G 00          02 FB 00620       CALLS   #2, SYS$CMKRNL                               ;
                             57          50 D0 00627       MOVL    R0, STATUS                                   ;
                             55          57 E9 0062A       BLBC    STATUS, 31$                                  ;
                                         5B D6 0062D       INCL    COUNT                                        ; 0886
                             05          5B D1 0062F       CMPL    COUNT, #5                                    ; 0887
                                         61 19 00632       BLSS    35$                                          ;
                                         5B D4 00634       CLRL    COUNT                                        ; 0894
                        02E8 CE     02EC CE 9E 00636       MOVAB   IMAGE, IMAGE_DESC+4                          ; 0895
                          74 AE 02070080 8F D0 0063D       MOVL    #34013312, ITEM_LIST                         ; 0896
                          78 AE     02E8 CE D0 00645       MOVL    IMAGE_DESC+4, ITEM_LIST+4                    ; 0898
                          7C AE     02E4 CE 9E 0064B       MOVAB   IMAGE_DESC, ITEM_LIST+8                      ; 0899
                                    0080 CE D4 00651       CLRL    ITEM_LIST+12                                 ; 0900
                                         7E 7C 00655       CLRQ    -(SP)                                        ; 0905
                                    008C CE 9F 00657       PUSHAB  IOSB                                         ;
                                    0080 CE 9F 0065B       PUSHAB  ITEM_LIST                                    ;
                                         7E D4 0065F       CLRL    -(SP)                                        ;
                               00000000G 00 9F 00661       PUSHAB  PROC_L_PID                                   ;
                                         7E D4 00667       CLRL    -(SP)                                        ;
                   00000000G 00          07 FB 00669       CALLS   #7, SYS$GETJPI                               ;
                             1E          50 E9 00670       BLBC    R0, 34$                                      ;
                                         7E D4 00673       CLRL    -(SP)                                        ; 0908
                   00000000G 00          01 FB 00675       CALLS   #1, SYS$WAITFR                               ;
                             57          50 D0 0067C       MOVL    R0, STATUS                                   ;
                             0A          57 E8 0067F       BLBS    STATUS, 33$                                  ;
                                         57 DD 00682  31$: PUSHL   STATUS                                       ;
                   00000000G 00          01 FB 00684  32$: CALLS   #1, LIB$SIGNAL                               ;
                                            04 0068B       RET                                                  ;
                             04     0084 CE E8 0068C  33$: BLBS    IOSB, 35$                                    ; 0910
                                    02E4 CE D4 00691  34$: CLRL    IMAGE_DESC                                   ; 0915
                                   0000' CF D5 00695  35$: TSTL    DISPLAY_MODE                                 ; 0918
                                         03 13 00699       BEQL    36$                                          ;
                                      0490 31 0069B        BRW     59$                                          ;
                             7E       41 8F 9A 0069E  36$: MOVZBL  #65, -(SP)                                   ; 0920
                                   0000' CF DD 006A2       PUSHL   SPACING                                      ;
                       0000' DF          02 FB 006A6       CALLS   #2, @SCR$SET_CURSOR                          ;
                                         7E D4 006AB       CLRL    -(SP)                                        ;
                                   0000' CF 9F 006AD       PUSHAB  P.ADI                                        ;
                        F743 CF          02 FB 006B1       CALLS   #2, FAO_BUFFER                               ;
                                         50 DD 006B6       PUSHL   R0                                           ;
                       0000' DF          01 FB 006B8       CALLS   #1, @SCR$PUT_SCREEN                          ;
                             50 00000000G 00 D0 006BD      MOVL    PROC_A_DESC+4, R0                            ; 0922
00000000G 00       20  0000' DF    0000' CF 2D 006C4       CMPC5   PREV_DESC, @PREV_DESC+4, #32, PROC_A_DESC, -; 0921
                                         60    006D1               (R0)                                         ;
                                         28 13 006D2       BEQL    37$                                          ;
                                         26 DD 006D4       PUSHL   #38                                          ; 0923
                                   0000' CF DD 006D6       PUSHL   SPACING                                      ;
                       0000' DF          02 FB 006DA       CALLS   #2, @SCR$SET_CURSOR                          ;
                               00000000G 00 DD 006DF       PUSHL   PROC_A_DESC+4                                ; 0924
                             7E 00000000G 00 3C 006E5      MOVZWL  PROC_A_DESC, -(SP)                           ;
                                   0000' CF 9F 006EC       PUSHAB  P.ADR                                        ;
                        F704 CF          03 FB 006F0       CALLS   #3, FAO_BUFFER                               ;
                                         50 DD 006F5       PUSHL   R0                                           ;
                       0000' DF          01 FB 006F7       CALLS   #1, @SCR$PUT_SCREEN                          ;
     02E4 CE       20  0000' DF    0000' CF 2D 006FC  37$: CMPC5   PREV_IMGDESC, @PREV_IMGDESC+4, #32, -        ; 0926
```

```
                                      02E8  DE    00707            IMAGE_DESC, @IMAGE_DESC+4                     ;
                                            35 13 0070A            BEQL    38$                                   ;
                                            05 DD 0070C            PUSHL   #5                                    ; 0929
                   7E      0000'   CF       0B C5 0070E            MULL3   #11, SPACING, -(SP)                   ;
                           0000'   DF       02 FB 00714            CALLS   #2, @SCR$SET_CURSOR                   ;
                                      02E4  CE 9F 00719            PUSHAB  IMAGE_DESC                            ;
                                      0000' CF 9F 0071D            PUSHAB  P.ADM                                 ;
                           F6D3    CF       02 FB 00721            CALLS   #2, FAO_BUFFER                        ;
                                            50 DD 00726            PUSHL   R0                                    ;
                           0000'   DF       01 FB 00728            CALLS   #1, @SCR$PUT_SCREEN                   ;
                           0000'   DF       00 FB 0072D            CALLS   #0, @SCR$ERASE_LINE                   ;
                                   01 0000' CF D1 00732            CMPL    SPACING, #1                           ; 0930
                                            08 12 00737            BNEQ    38$                                   ;
           0000'   CF      0000'   CF       01 C1 00739            ADDL3   #1, LOCEVFL1, PREV_LOCEVFL1           ;
                           0000'   CF 0000' CF D1 00741 38$:       CMPL    PREV_STATE, STATE                     ; 0932
                                            2C 13 00748            BEQL    39$                                   ;
                                            19 DD 0074A            PUSHL   #25                                   ; 0933
                   7E      0000'   CF       03 C5 0074C            MULL3   #3, SPACING, -(SP)                    ;
                           0000'   DF       02 FB 00752            CALLS   #2, @SCR$SET_CURSOR                   ;
                                      0000' CF 9F 00757            PUSHAB  STATE_TABLE                           ;
                                      0000' CF DD 0075B            PUSHL   STATE                                 ;
                           F6BA    CF       02 FB 0075F            CALLS   #2, TRANSLATE_VALUE                   ;
                                            50 DD 00764            PUSHL   R0                                    ;
                                      0000' CF 9F 00766            PUSHAB  P.ADO                                 ;
                           F68A    CF       02 FB 0076A            CALLS   #2, FAO_BUFFER                        ;
                                            50 DD 0076F            PUSHL   R0                                    ;
                           0000'   DF       01 FB 00771            CALLS   #1, @SCR$PUT_SCREEN                   ;
                           0000'   CF 0000' CF D1 00776 39$:       CMPL    PREV_PRI, PRI                         ; 0935
                                            21 13 0077D            BEQL    40$                                   ;
                                            18 DD 0077F            PUSHL   #24                                   ; 0936
                   7E      0000'   CF       02 78 00781            ASHL    #2, SPACING, -(SP)                    ;
                           0000'   DF       02 FB 00787            CALLS   #2, @SCR$SET_CURSOR                   ;
                                      0000' CF DD 0078C            PUSHL   PRI                                   ;
                                      0000' CF 9F 00790            PUSHAB  P.ADQ                                 ;
                           F660    CF       02 FB 00794            CALLS   #2, FAO_BUFFER                        ;
                                            50 DD 00799            PUSHL   R0                                    ;
                           0000'   DF       01 FB 0079B            CALLS   #1, @SCR$PUT_SCREEN                   ;
                           0000'   CF 0000' CF D1 007A0 40$:       CMPL    PREV_PRIB, PRIB                       ; 0938
                                            21 13 007A7            BEQL    41$                                   ;
                                            1B DD 007A9            PUSHL   #27                                   ; 0939
                   7E      0000'   CF       02 78 007AB            ASHL    #2, SPACING, -(SP)                    ;
                           0000'   DF       02 FB 007B1            CALLS   #2, @SCR$SET_CURSOR                   ;
                                      0000' CF DD 007B6            PUSHL   PRIB                                  ;
                                      0000' CF 9F 007BA            PUSHAB  P.ADS                                 ;
                           F636    CF       02 FB 007BE            CALLS   #2, FAO_BUFFER                        ;
                                            50 DD 007C3            PUSHL   R0                                    ;
                           0000'   DF       01 FB 007C5            CALLS   #1, @SCR$PUT_SCREEN                   ;
                           0000'   CF 0000' CF D1 007CA 41$:       CMPL    PREV_PC, PC                           ; 0941
                                            21 13 007D1            BEQL    42$                                   ;
                                            19 DD 007D3            PUSHL   #25                                   ; 0942
                   7E      0000'   CF       05 C5 007D5            MULL3   #5, SPACING, -(SP)                    ;
                           0000'   DF       02 FB 007DB            CALLS   #2, @SCR$SET_CURSOR                   ;
                                      0000' CF DD 007E0            PUSHL   PC                                    ;
                                      0000' CF 9F 007E4            PUSHAB  P.ADU                                 ;
                           F60C    CF       02 FB 007E8            CALLS   #2, FAO_BUFFER                        ;
                                            50 DD 007ED            PUSHL   R0                                    ;
                           0000'   DF       01 FB 007EF            CALLS   #1, @SCR$PUT_SCREEN                   ;
```

```
          0000'  CF      0000'  CF  D1 007F4 42$:     CMPL    PREV_PSL, PSL                                 ; 0944
                                21  13 007FB          BEQL    43$                                           ;
                                19  DD 007FD          PUSHL   #25                                           ; 0945
    7E    0000'  CF                 06  C5 007FF          MULL3   #6, SPACING, -(SP)                            ;
          0000'  DF                 02  FB 00805          CALLS   #2, @$SCR$SET_CURSOR                          ;
                         0000'  CF  DD 0080A          PUSHL   PSL                                           ;
                         0000'  CF  9F 0080E          PUSHAB  P.ADW                                         ;
          F5E2   CF                 02  FB 00812          CALLS   #2, FAO_BUFFER                                ;
                                50  DD 00817          PUSHL   R0                                            ;
          0000'  DF                 01  FB 00819          CALLS   #', @$SCR$PUT_SCREEN                          ;
          0000'  CF      0000'  CF  D1 0081E 43$:     CMPL    PREV_SP, SP                                   ; 0947
                                21  13 00825          BEQL    44$                                           ;
                                19  DD 00827          PUSHL   #25                                           ; 0948
    7E    0000'  CF                 07  C5 00829          MULL3   #7, SPACING, -(SP)                            ;
          0000'  DF                 02  FB 0082F          CALLS   #2, @$SCR$SET_CURSOR                          ;
                         0000'  CF  DD 00834          PUSHL   SP                                            ;
                         0000'  CF  9F 00838          PUSHAB  P.ADY                                         ;
          F5B8   CF                 02  FB 0083C          CALLS   #2, FAO_BUFFER                                ;
                                50  DD 00841          PUSHL   R0                                            ;
          0000'  DF                 01  FB 00843          CALLS   #1, @$SCR$PUT_SCREEN                          ;
          0000'  CF      0000'  CF  D1 00848 44$:     CMPL    PREV_GRP, GRP                                 ; 0950
                                0C  12 0084F          BNEQ    45$                                           ;
          0000'  CF      0000'  CF  D1 00851          CMPL    PREV_MEM, MEM                                 ;
                                03  12 00858          BNEQ    45$                                           ;
                              0113  31 0085A          BRW     50$                                           ;
                 50      0000'  CF  3C 0085D 45$:     MOVZWL  GRP, R0                                       ; 0954
    50           50             10  78 00862          ASHL    #16, R0, R0                                   ;
                 52      0000'  CF  3C 00866          MOVZWL  MEM, UIC                                      ;
                 52             50  C8 0086B          BISL2   R0, UIC                                       ;
                                52  DD 0086E          PUSHL   UIC                                           ; 0955
                         0000'  CF  9F 00870          PUSHAB  P.AEA                                         ;
          F580   CF                 02  FB 00874          CALLS   #2, FAO_BUFFER                                ;
                 6E             50  D0 00879          MOVL    R0, CONVERTED_UIC                             ;
                 23         00  BE  D1 0087C          CMPL    @CONVERTED_UIC, #35                           ; 0956
                                3A  18 00880          BGEQ    47$                                           ;
    50    0000'  CF                 09  C5 00882          MULL3   #9, SPACING, R0                               ; 0960
                 13         00  BE  D1 00888          CMPL    @CONVERTED_UIC, #19                           ; 0959
                                12  14 0088C          BGTR    46$                                           ;
                                19  DD 0088E          PUSHL   #25                                           ; 0960
                                50  DD 00890          PUSHL   R0                                            ;
          0000'  DF                 02  FB 00892          CALLS   #2, @$SCR$SET_CURSOR                          ;
                                52  DD 00897          PUSHL   UIC                                           ;
                         0000'  CF  9F 00899          PUSHAB  P.AEC                                         ;
                              00C4  31 0089D          BRW     48$                                           ;
                                09  DD 008A0 46$:     PUSHL   #9                                            ; 0961
                                50  DD 008A2          PUSHL   R0                                            ;
          0000'  DF                 02  FB 008A4          CALLS   #2, @$SCR$SET_CURSOR                          ;
                                52  DD 008A9          PUSHL   UIC                                           ;
    7E           23         04  BE  C3 008AB          SUBL3   @CONVERTED_UIC, #35, -(SP)                    ;
                         0000'  CF  9F 008B0          PUSHAB  P.AEE                                         ;
          F540   CF                 03  FB 008B4          CALLS   #3, FAO_BUFFER                                ;
                              00AD  31 008B9          BRW     49$                                           ;
          00D4   CE                 20  D0 008BC 47$:     MOVL    #32, GRP_DESC                                 ; 0965
          00D8   CE      00B4   CE  9E 008C1          MOVAB   GRP_NAME, GRP_DESC+4                          ; 0966
          00AC   CE                 20  D0 008C8          MOVL    #32, MEM_DESC                                 ; 0967
          00B0   CE      008C   CE  9E 008CD          MOVAB   MEM_NAME, MEM_DESC+4                          ; 0968
                                7E  7C 008D4          CLRQ    -(SP)                                         ; 0971
```

```
                                   7E  D4 008D6          CLRL    -(SP)                                    ;
                         00E0      CE  9F 008D8          PUSHAB  GRP_DESC                                 ;
                         00E4      CE  9F 008DC          PUSHAB  GRP_DESC                                 ;
   50      0000'   CF              10  78 008E0          ASHL    #16, GRP, R0                             ;
   7E              50 0000FFFF     8F  C9 008E6          BISL3   #65535, R0, -(SP)                        ;
      00000000G    00              06  FB 008EE          CALLS   #6, SYS$IDTOASC                          ;
                                   7E  7C 008F5          CLRQ    -(SP)                                    ; 0974
                                   7E  D4 008F7          CLRL    -(SP)                                    ;
                         00B8      CE  9F 008F9          PUSHAB  MEM_DESC                                 ;
                         00BC      CE  9F 008FD          PUSHAB  MEM_DESC                                 ;
                                   52  DD 00901          PUSHL   UIC                                      ;
      00000000G    00              06  FB 00903          CALLS   #6, SYS$IDTOASC                          ;
                                   0A  DD 0090A          PUSHL   #10                                      ; 0975
   7E      0000'   CF              09  C5 0090C          MULL3   #9, SPACING, -(SP)                       ;
           0000'   DF              02  FB 00912          CALLS   #2, @SCR$SET_CURSOR                      ;
                         00D4      CE  9F 00917          PUSHAB  GRP_DESC                                 ;
                         0000'     CF  9F 0091B          PUSHAB  P.AEG                                    ;
           F4D5    CF              02  FB 0091F          CALLS   #2, FAO_BUFFER                           ;
                                   50  DD 00924          PUSHL   R0                                       ;
           0000'   DF              01  FB 00926          CALLS   #1, @SCR$PUT_SCREEN                      ;
   7E      00D4    CE              0B  C1 0092B          ADDL3   #11, GRP_DESC, -(SP)                     ; 0976
   7E      0000'   CF              09  C5 00931          MULL3   #9, SPACING, -(SP)                       ;
           0000'   DF              02  FB 00937          CALLS   #2, @SCR$SET_CURSOR                      ;
                         0000'     CF  9F 0093C          PUSHAB  P.AEI                                    ;
           F4B4    CF              01  FB 00940          CALLS   #1, FAO_BUFFER                           ;
                                   50  DD 00945          PUSHL   R0                                       ;
           0000'   DF              01  FB 00947          CALLS   #1, @SCR$PUT_SCREEN                      ;
                                   0B  DD 0094C          PUSHL   #11                                      ; 0977
   50      0000'   CF              09  C5 0094E          MULL3   #9, SPACING, R0                          ;
                           01      A0  9F 00954          PUSHAB  1(R0)                                    ;
           0000'   DF              02  FB 00957          CALLS   #2, @SCR$SET_CURSOR                      ;
                         00AC      CE  9F 0095C          PUSHAB  MEM_DESC                                 ;
                         0000'     CF  9F 00960          PUSHAB  P.AEK                                    ;
           F490    CF              02  FB 00964 48$:     CALLS   #2, FAO_BUFFER                           ;
                                   50  DD 00969 49$:     PUSHL   R0                                       ;
           0000'   DF              01  FB 0096B          CALLS   #1, @SCR$PUT_SCREEN                      ;
   51      0000'   CF        0000' CF  C1 00970 50$:     ADDL3   PREV_GPGCNT, PREV_PPGCNT, R1             ; 0980
   50      0000'   CF        0000' CF  C1 00978          ADDL3   GPGCNT, PPGCNT, R0                       ;
                   50              51  D1 00980          CMPL    R1, R0                                   ;
                                   27  13 00983          BEQL    51$                                      ;
                   7E        43    8F  9A 00985          MOVZBL  #67, -(SP)                               ; 0981
   7E      0000'   CF              03  C5 00989          MULL3   #3, SPACING, -(SP)                       ;
           0000'   DF              02  FB 0098F          CALLS   #2, @SCR$SET_CURSOR                      ;
   7E      0000'   CF        0000' CF  C1 00994          ADDL3   GPGCNT, PPGCNT, -(SP)                    ;
                             0000' CF  9F 0099C          PUSHAB  P.AEM                                    ;
           F454    CF              02  FB 009A0          CALLS   #2, FAO_BUFFER                           ;
                                   50  DD 009A5          PUSHL   R0                                       ;
           0000'   DF              01  FB 009A7          CALLS   #1, @SCR$PUT_SCREEN                      ;
           0000'   CF        0000' CF  D1 009AC 51$:     CMPL    PREV_VAUSE, VAUSE                        ; 0983
                                   29  13 009B3          BEQL    52$                                      ;
                   7E        42    8F  9A 009B5          MOVZBL  #66, -(SP)                               ; 0984
   7E      0000'   CF              02  78 009B9          ASHL    #2, SPACING, -(SP)                       ;
           0000'   DF              02  FB 009BF          CALLS   #2, @SCR$SET_CURSOR                      ;
   7E      0000'   CF 00000200     8F  C7 009C4          DIVL3   #512, VAUSE, -(SP)                       ;
                             0000' CF  9F 009CE          PUSHAB  P.AEO                                    ;
           F422    CF              02  FB 009D2          CALLS   #2, FAO_BUFFER                           ;
                                   50  DD 009D7          PUSHL   R0                                       ;
```

```
                      0000'  DF              01  FB 009D9         CALLS    #1, @$SCR$PUT_SCREEN                          ;
                      0000'  CF        0000' CF  D1 009DE 52$:    CMPL     PREV_CPUTIME, CPUTIME                         ; 0986
                                             65  13 009E5         BEQL     53$                                           ;
                                             3E  DD 009E7         PUSHL    #62                                           ; 0987
               7E     0000'  CF              05  C5 009E9         MULL3    #5, SPACING, -(SP)                            ;
                      0000'  DF              02  FB 009EF         CALLS    #2, @$SCR$SET_CURSOR                          ;
                             50        0000' CF  D0 009F4         MOVL     CPUTIME, R0                                   ; 0991
       7E      00            50              01  7A 009F9         EMUL     #1, R0, #0, -(SP)                             ;
       6E      7E            8E 00000064     8F  7B 009FE         EDIV     #100, (SP)+, -(SP), (SP)                      ;
               51            50 00000064     8F  C7 00A07         DIVL3    #100, R0, R1                                  ;
       7E      00            51              01  7A 00A0F         EMUL     #1, R1, #0, -(SP)                             ;
       6E      7E            8E              3C  7B 00A14         EDIV     #60, (SP)+, -(SP), (SP)                       ;
               51            50 00001770     8F  C7 00A19         DIVL3    #6000, R0, R1                                 ;
       7E      00            51              01  7A 00A21         EMUL     #1, R1, #0, -(SP)                             ;
       6E      7E            8E              3C  7B 00A26         EDIV     #60, (SP)+, -(SP), (SP)                       ;
                             50 00057E40     8F  C6 00A2B         DIVL2    #360000, R0                                   ;
       7E      00            50              01  7A 00A32         EMUL     #1, R0, #0, -(SP)                             ;
       6E      7E            8E              18  7B 00A37         EDIV     #24, (SP)+, -(SP), (SP)                       ;
                                       0000' CF  9F 00A3C         PUSHAB   P.AEQ                                         ;
                      F3B4   CF              05  FB 00A40         CALLS    #5, FAO_BUFFER                                ;
                                             50  DD 00A45         PUSHL    R0                                            ;
                      0000'  DF              01  FB 00A47         CALLS    #1, @$SCR$PUT_SCREEN                          ;
                      0000'  CF        0000' CF  D1 00A4C 53$:    CMPL     PREV_DIRIO, DIRIO                             ; 0993
                                             23  13 00A53         BEQL     54$                                           ;
                             7E           41 8F  9A 00A55         MOVZBL   #65, -(SP)                                    ; 0994
               7E     0000'  CF              06  C5 00A59         MULL3    #6, SPACING, -(SP)                            ;
                      0000'  DF              02  FB 00A5F         CALLS    #2, @$SCR$SET_CURSOR                          ;
                                       0000' CF  DD 00A64         PUSHL    DIRIO                                         ;
                                       0000' CF  9F 00A68         PUSHAB   P.AES                                         ;
                      F388   CF              02  FB 00A6C         CALLS    #2, FAO_BUFFER                                ;
                                             50  DD 00A71         PUSHL    R0                                            ;
                      0000'  DF              01  FB 00A73         CALLS    #1, @$SCR$PUT_SCREEN                          ;
                      0000'  CF        0000' CF  D1 00A78 54$:    CMPL     PREV_BUFIO, BUFIO                             ; 0996
                                             23  13 00A7F         BEQL     55$                                           ;
                             7E           41 8F  9A 00A81         MOVZBL   #65, -(SP)                                    ; 0997
               7E     0000'  CF              07  C5 00A85         MULL3    #7, SPACING, -(SP)                            ;
                      0000'  DF              02  FB 00A8B         CALLS    #2, @$SCR$SET_CURSOR                          ;
                                       0000' CF  DD 00A90         PUSHL    BUFIO                                         ;
                                       0000' CF  9F 00A94         PUSHAB   P.AEU                                         ;
                      F35C   CF              02  FB 00A98         CALLS    #2, FAO_BUFFER                                ;
                                             50  DD 00A9D         PUSHL    R0                                            ;
                      0000'  DF              01  FB 00A9F         CALLS    #1, @$SCR$PUT_SCREEN                          ;
                      0000'  CF        0000' CF  D1 00AA4 55$:    CMPL     PREV_PAGEFLTS, PAGEFLTS                       ; 0999
                                             23  13 00AAB         BEQL     56$                                           ;
                             7E           41 8F  9A 00AAD         MOVZBL   #65, -(SP)                                    ; 1000
               7E     0000'  CF              03  78 00AB1         ASHL     #3, SPACING, -(SP)                            ;
                      0000'  DF              02  FB 00AB7         CALLS    #2, @$SCR$SET_CURSOR                          ;
                                       0000' CF  DD 00ABC         PUSHL    PAGEFLTS                                      ;
                                       0000' CF  9F 00AC0         PUSHAB   P.AEW                                         ;
                      F330   CF              02  FB 00AC4         CALLS    #2, FAO_BUFFER                                ;
                                             50  DD 00AC9         PUSHL    R0                                            ;
                      0000'  DF              01  FB 00ACB         CALLS    #1, @$SCR$PUT_SCREEN                          ;
                      0000'  CF        0000' CF  D1 00AD0 56$:    CMPL     PREV_LOCEVFLO, LOCEVFLO                       ; 1002
                                             23  13 00AD7         BEQL     57$                                           ;
                             7E           41 8F  9A 00AD9         MOVZBL   #65, -(SP)                                    ; 1003
               7E     0000'  CF              09  C5 00ADD         MULL3    #9, SPACING, -(SP)                            ;
                      0000'  DF              02  FB 00AE3         CALLS    #2, @$SCR$SET_CURSOR                          ;
```

```
                                         0000'  CF   DD 00AE8         PUSHL    LOCEVFL0                               :
                                         0000'  CF   9F 00AEC         PUSHAB   P.AEY                                  :
                     F304    CF                 02   FB 00AF0         CALLS    #2, FAO_BUFFER                         :
                                                50   DD 00AF5         PUSHL    R0                                     :
                     0000'   DF                 01   FB 00AF7         CALLS    #1, @$SCR$PUT_SCREEN                   :
                     0000'   CF          0000'  CF   D1 00AFC  57$:   CMPL     PREV_LOCEVFL1, LOCEVFL1                : 1005
                                                26   13 00B03         BEQL     58$                                    :
                             7E           40    8F   9A 00B05         MOVZBL   #64, -(SP)                             : 1006
             50      0000'   CF                 09   C5 00B09         MULL3    #9, SPACING, R0                        :
                                          01    A0   9F 00B0F         PUSHAB   1(R0)                                  :
                     0000'   DF                 02   FB 00B12         CALLS    #2, @$SCR$SET_CURSOR                   :
                                         0000'  CF   DD 00B17         PUSHL    LOCEVFL1                               :
                                         0000'  CF   9F 00B1B         PUSHAB   P.AFA                                  :
                     F2D5    CF                 02   FB 00B1F         CALLS    #2, FAO_BUFFER                         :
                                                50   DD 00B24         PUSHL    R0                                     :
                     0000'   DF                 01   FB 00B26         CALLS    #1, @$SCR$PUT_SCREEN                   :
                                              016F   31 00B2B  58$:   BRW      77$                                    : 0918
                             54                 02   CE 00B2E  59$:   MNEGL    #2, LAST_AT                            : 1012
             55      0000'   CF                 01   C3 00B31         SUBL3    #1, MAX_ROW, R5                        : 1013
                             55                 59   C4 00B37         MULL2    VPN_PER_COL, R5                        :
   0000'     DF      0000'   CF                 55   29 00B3A         CMPC3    R5, VMAP, @PREV_VMAP                   :
                                                03   12 00B42         BNEQ     60$                                    :
                                              00C7   31 00B44         BRW      74$                                    :
                             52                 01   CE 00B47  60$:   MNEGL    #1, I                                  : 1016
                                                53   11 00B4A         BRB      64$                                    :
                     0000'DF42           0000'CF42   91 00B4C  61$:   CMPB     VMAP[I], @PREV_VMAP[I]                 : 1018
                                                48   13 00B55         BEQL     64$                                    :
                             50           01    A4   9E 00B57         MOVAB    1(R4), R0                              : 1021
                             50                 52   D1 00B5B         CMPL     I, R0                                  :
                                                0E   12 00B5E         BNEQ     62$                                    :
   7E        00              52                 01   7A 00B60         EMUL     #1, I, #0, -(SP)                       : 1022
   50        50              8E                 59   7B 00B65         EDIV     VPN_PER_COL, (SP)+, R0, R0             :
                                                50   D5 00B6A         TSTL     R0                                     :
                                                19   12 00B6C         BNEQ     63$                                    :
   7E        00              52                 01   7A 00B6E  62$:   EMUL     #1, I, #0, -(SP)                       : 1024
   50        50              8E                 59   7B 00B73         EDIV     VPN_PER_COL, (SP)+, R0, R0             :
                                              6640   9F 00B78         PUSHAB   (VPN_1ST_COL)[R0]                      :
             50              52                 59   C7 00B7B         DIVL3    VPN_PER_COL, I, R0                     :
                                          01    A0   9F 00B7F         PUSHAB   1(R0)                                  :
                     0000'   DF                 02   FB 00B82         CALLS    #2, @$SCR$SET_CURSOR                   :
                             54                 52   D0 00B87  63$:   MOVL     I, LAST_AT                             : 1025
                             53          0000'CF42   90 00B8A         MOVB     VMAP[I], BITS                          : 1026
                                                0F   18 00B90         BGEQ     65$                                    :
                                                01   DD 00B92         PUSHL    #1                                     :
                                                7E   7C 00B94         CLRQ     -(SP)                                  :
                                         0000'  CF   9F 00B96         PUSHAB   P.AFC                                  :
                     0000'   DF                 04   FB 00B9A         CALLS    #4, @$SCR$PUT_SCREEN                   :
                                                64   11 00B9F  64$:   BRB      72$                                    :
   50        53              01                 05   EF 00BA1  65$:   EXTZV    #5, #1, BITS, R0                       :
   51        53              01                 04   EF 00BA6         EXTZV    #4, #1, BITS, R1                       :
                             50                 51   C8 00BAB         BISL2    R1, R0                                 :
   58        53              01                 00   EF 00BAE         EXTZV    #0, #1, BITS, R8                       :
                             58                 58   D2 00BB3         MCOML    R8, R8                                 :
                             50                 58   CA 00BB6         BICL2    R8, R0                                 :
                             01                 50   D1 00BB9         CMPL     R0, #1                                 :
                                                06   12 00BBC         BNEQ     66$                                    :
                                         0000'  CF   9F 00BBE         PUSHAB   P.AFE                                  :
                                              3C  11 00BC2         BRB     71$                               :
                                              51  D4 00BC4 66$:    CLRL    R1                                :
      03         53         03                01  ED 00BC6         CMPZV   #1, #3, BITS, #3                  :
                                              02  12 00BCB         BNEQ    67$                               :
                                              51  D6 00BCD         INCL    R1                                :
                                              50  D4 00BCF 67$:    CLRL    R0                                :
      02         53         03                01  ED 00BD1         CMPZV   #1, #3, BITS, #2                  :
                                              02  12 00BD6         BNEQ    68$                               :
                                              50  D6 00BD8         INCL    R0                                :
                            50                51  C8 00BDA 68$:    BISL2   R1, R0                            :
      58         53         01                00  EF 00BDD         EXTZV   #0, #1, BITS, R8                  :
                            58                58  D2 00BE2         MCOML   R8, R8                            :
                            50                58  CA 00BE5         BICL2   R8, R0                            :
                            01                50  D1 00BE8         CMPL    R0, #1                            :
                                              06  12 00BEB         BNEQ    69$                               :
                                     0000'    CF  9F 00BED         PUSHAB  P.AFG                             :
                                              0D  11 00BF1         BRB     71$                               :
                            06                53  E9 00BF3 69$:    BLBC    BITS, 70$                         :
                                     0000'    CF  9F 00BF6         PUSHAB  P.AFI                             :
                                              04  11 00BFA         BRB     71$                               :
                                     0000'    CF  9F 00BFC 70$:    PUSHAB  P.AFK                             :
                      0000'   DF              01  FB 00C00 71$:    CALLS   #1, @SCR$PUT_SCREEN               :
                 02         52                55  F2 00C05 72$:    AOBLSS  R5, I, 73$                        : 1018
                                              03  11 00C09         BRB     74$                               :
                                            FF3E  31 00C0B 73$:    BRW     61$                               :
                      0000'   CF     0000'    CF  D1 00C0E 74$:    CMPL    PREV_PC, PC                       : 1033
                                              1F  13 00C15         BEQL    75$                               :
                                              07  DD 00C17         PUSHL   #7                                : 1034
                                     0000'    CF  DD 00C19         PUSHL   MAX_ROW                           :
                      0000'   DF              02  FB 00C1D         CALLS   #2, @SCR$SET_CURSOR               :
                                     0000'    CF  DD 00C22         PUSHL   PC                                :
                                     0000'    CF  9F 00C26         PUSHAB  P.AFM                             :
                      F1CA    CF              02  FB 00C2A         CALLS   #2, FAO_BUFFER                    :
                                              50  DD 00C2F         PUSHL   R0                                :
                      0000'   DF              01  FB 00C31         CALLS   #1, @SCR$PUT_SCREEN               :
                      0000'   CF     0000'    CF  D1 00C36 75$:    CMPL    PREV_STATE, STATE                 : 1036
                                              2A  13 00C3D         BEQL    76$                               :
                                              18  DD 00C3F         PUSHL   #24                               : 1037
                                     0000'    CF  DD 00C41         PUSHL   MAX_ROW                           :
                      0000'   DF              02  FB 00C45         CALLS   #2, @SCR$SET_CURSOR               :
                                     0000'    CF  9F 00C4A         PUSHAB  STATE_TABLE                       :
                                     0000'    CF  DD 00C4E         PUSHL   STATE                             :
                      F1C7    CF              02  FB 00C52         CALLS   #2, TRANSLATE_VALUE               :
                                              50  DD 00C57         PUSHL   R0                                :
                                     0000'    CF  9F 00C59         PUSHAB  P.AFO                             :
                      F197    CF              02  FB 00C5D         CALLS   #2, FAO_BUFFER                    :
                                              50  DD 00C62         PUSHL   R0                                :
                      0000'   DF              01  FB 00C64         CALLS   #1, @SCR$PUT_SCREEN               :
 02E4   CE       20   0000'   DF     0000'    CF  2D 00C69 76$:    CMPC5   PREV_IMGDESC, @PREV_IMGDESC+4, #32, -   : 1039
                                     02E8     DE     00C74                 IMAGE_DESC, @IMAGE_DESC+4         :
                                              24  13 00C77         BEQL    77$                               :
                                              20  DD 00C79         PUSHL   #32                               : 1042
                                     0000'    CF  DD 00C7B         PUSHL   MAX_ROW                           :
                      0000'   DF              02  FB 00C7F         CALLS   #2, @SCR$SET_CURSOR               :
                                     02E4     CE  9F 00C84         PUSHAB  IMAGE_DESC                        :
                                     0000'    CF  9F 00C88         PUSHAB  P.AFQ                             :
                      F168    CF              02  FB 00C8C         CALLS   #2, FAO_BUFFER                    :
```

```
                                        50  DD 00C91          PUSHL   R0                                  ;
                0000'   DF              01  FB 00C93          CALLS   #1, @$SCR$PUT_SCREEN                ;
                0000'   DF              00  FB 00C98          CALLS   #0, @$SCR$ERASE_LINE                ;
                                        01  DD 00C9D 77$:     PUSHL   #1                                  ; 1046
                               0000'    CF  DD 00C9F          PUSHL   MAX_ROW                             ;
                0000'   DF              02  FB 00CA3          CALLS   #2, @$SCR$SET_CURSOR                ;
                0000'   DF              00  FB 00CA8          CALLS   #0, @$SCR$PUT_BUFFER                ; 1047
                               00DC     CE  9F 00CAD          PUSHAB  BUFDESC                             ; 1048
                0000'   DF              01  FB 00CB1          CALLS   #1, @$SCR$SET_BUFFER                ;
                        03     0000'    CF  E9 00CB6          BLBC    KEEP_GOING, 78$                     ; 1050
                                      F562  31 00CBB          BRW     14$                                 ;
                                            04 00CBE 78$:     RET                                         ; 1053

; Routine Size:  3263 bytes,    Routine Base:  INFO_CODE + 021A
; 1062          1054  1 END
; 1063          1055  0 ELUDOM



                                                                       .EXTRN  LIB$SIGNAL

;                                   PSECT SUMMARY
;
;      Name                      Bytes                      Attributes
;
;  $OWN$                              28  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  INFO_OWN                         3532  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  INFO_CODE                        3801  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  INFO_PLIT                        1256  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                              Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;      File                                Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619        79         0      1000          00:01.8



;                                   COMMAND QUALIFIERS

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

; Size:          3794 code + 4823 data bytes
; Run Time:         00:59.5
; Elapsed Time:     03:21.9
; Lines/CPU Min:     1064
; Lexemes/CPU-Min: 19229
; Memory Used:  719 pages
; Compilation Complete
```

0049 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

JBCPRSDEF
REQ

CNVCLIATB
LIS

INFO
LIS

TYPE
REQ

CHRSUB
LIS

CNVCLINUM
LIS

SHODEVDEF
REQ

CLIMAC
MAR

CNVCLIFRM
LIS

DIGRAMS
LIS

CALCMAX
LIS

JBCCMDPRS
LIS

CLIUTLMAC
MAR

CVTTIME
LIS

SHOWDEF
REQ

CREATE
LIS