
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP

```
LL              000000      CCCCCCCC  KK      KK  DDDDDDDD    NN      NN
LL              000000      CCCCCCCC  KK      KK  DDDDDDDD    NN      NN
LL            00      00  CC          KK      KK  DD      DD  NN      NN
LL            00      00  CC          KK      KK  DD      DD  NN      NN
LL            00      00  CC          KK    KK    DD      DD  NNNN    NN
LL            00      00  CC          KK    KK    DD      DD  NNNN    NN
LL            00      00  CC          KKKKKK      DD      DD  NN  NN  NN
LL            00      00  CC          KKKKKK      DD      DD  NN  NN  NN
LL            00      00  CC          KK    KK    DD      DD  NN    NNNN
LL            00      00  CC          KK    KK    DD      DD  NN    NNNN
LL            00      00  CC          KK      KK  DD      DD  NN      NN      ....
LL            00      00  CC          KK      KK  DD      DD  NN      NN      ....
LLLLLLLLLL      000000      CCCCCCCC  KK      KK  DDDDDDDD    NN      NN      ....
LLLLLLLLLL      000000      CCCCCCCC  KK      KK  DDDDDDDD    NN      NN      ....


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
```

```
0000     1          .TITLE  LOCKDN - LOCK FCP INTO REAL MEMORY
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ;
0000    31  ; Facility:  F11ACP structure level 1
0000    32  ;
0000    33  ; Abstract:
0000    34  ;
0000    35  ;       This routine touches all of the pages in fcp to bring them
0000    36  ;       into real memory.
0000    37  ;
0000    38  ; Environment:                                                              ; Re
0000    39  ;
0000    40  ;       Starlet operating system, including privileged system services
0000    41  ;       and internal exec routine.                                          ;  A
0000    42  ;
0000    43  ; Author:  Andrew C. Goldstein, Creation Date:  22-DEC-1976  14:46
0000    44  ;
0000    45  ; Modified By:
0000    46  ;
0000    47  ;       V02-002 REFORMAT        Maria del C. Nasr       23-Jul-1980
0000    48  ;
0000    49  ;--
0000    50
0000    51  ;
0000    52  ; Define labels for the start and end of the locked down areas
0000    53  ;
0000    54
00000000    55          .PSECT  $LOCKEDCO$,NOWRT,PAGE
0000    56  LCODE_START:
0000    57
```

```
                00000000     58          .PSECT  $LOCKEDC9$,NOWRT,LONG
                    0000     59 LCODE_END:
                    0000     60
                00000000     61          .PSECT  $LOCKEDD0$,NOEXE,LONG
                    0000     62 LDATA_START:
                    0000     63
                00000000     64          .PSECT  $LOCKEDD9$,NOEXE,LONG
                    0000     65 LDATA_END:
                    0000     66
                00000000     67          .PSECT  $LOCKEDD1$,NOEXE,LONG
                    0000     68
                    0000     69 ;
                    0000     70 ; Own Storage:
                    0000     71 ;
                    0000     72
                    0000     73 WORKING_SET:
                00000004    0000     74          .BLKL   1                       ; space to receive working set size
                    0004     75 SET_SIZE::
                00000040    0004     76          .LONG   64                      ; size to adjust working set to
                    0008     77 LAST_PAGE:
                00000010    0008     78          .BLKL   2                       ; space to receive last page pointers
                    0010     79
                    0010     80 ;
                    0010     81 ; descriptors to lock down the code and data areas that are to be locked into
                    0010     82 ; the working set
                    0010     83 ;
                    0010     84
                00000000     85          .PSECT  $CODE$,NOWRT,LONG
                    0000     86
                    0000     87 LOCKED_CODE:
FFFFFFFF'00000000'  0000     88          .LONG   LCODE_START,LCODE_END-1
                    0008     89 LOCKED_DATA:
FFFFFFFF'00000000'  0008     90          .LONG   LDATA_START,LDATA_END-1
```

```
                                                     ;++
                                                     ;
                                                     ; LOCKDN
                                                     ;       LOCKDOWN
                                                     ;
                                                     ;       This routine touches all of the pages in fcp to bring them
                                                     ;       into real memory.
                                                     ;
                                                     ; Calling sequence:
                                                     ;       call lockdown ()
                                                     ;       none
                                                     ;
                                                     ; Input Parameters:
                                                     ;       none
                                                     ;
                                                     ; Implicit Inputs:
                                                     ;       none
                                                     ;
                                                     ; Output Parameters:
                                                     ;       none
                                                     ;
                                                     ; Implicit Outputs:
                                                     ;       none
                                                     ;
                                                     ; Routine Value:
                                                     ;       none
                                                     ;
                                                     ; Side Effects:
                                                     ;       all of fcp residing in real memory
                                                     ;
                                                     ;--

                                                     LOCKDOWN::
                                0000                 .WORD   ^M<>

                                                     ;
                                                     ; adjust the working set size to a suitable value
                                                     ;

50  0004'CF   0000'CF   C3      $ADJWSL_S       #0,W^WORKING_SET; read current size
                                SUBL3   W^WORKING_SET,W^SET_SIZE,R0; compute increment
                                $ADJWSL_S       R0,W^WORKING_SET; and set to desired size

                                                     ;
                                                     ; lock into the working set the code and data areas that should be.
                                                     ;

                                $LKWSET_S       LOCKED_CODE
                                $LKWSET_S       LOCKED_DATA

                                                     ;
                                                     ; expand the program region by one page to get the address of the top.
                                                     ; this page will never be touched and will therefore remain demand zero.
                                                     ;

                                $EXPREG_S       #1,LAST_PAGE,REGION=#0
```

```
                                0063   149
                                0063   150 ;
                                0063   151 ; now touch all pages up to the one created.
                                0063   152 ;
                                0063   153
            50   00000200 9F     9E  0063   154          MOVAB   @#^X200,R0              ; start with page 1
                        60     D5  006A   155 10$:        TSTL    (R0)                    ; touch it
50  00000200 8F   00000008'EF     F1  006C   156          ACBL    LAST_PAGE,#^X200,R0,10$ ; next page and loop
                              FFF0  0078
                                04  007A   157          RET
                                007B   158
                                007B   159
                                007B   160
                                007B   161          .END
```

```
$$T1            = 00000000
AQB_TYPE        = 00000005
FCB_TYPE        = 00000000
LAST_PAGE         00000008 R      05
LCODE_END         00000000 R      02
LCODE_START       00000000 R      01
LDATA_END         00000000 R      04
LDATA_START       00000000 R      03
LOCKDOWN          00000010 RG     06
LOCKED_CODE       00000000 R      06
LOCKED_DATA       00000008 R      06
MVL_TYPE        = 00000004
RVT_TYPE        = 00000003
SET_SIZE          00000004 RG     05
SYS$ADJWSL        ******** GX     06
SYS$EXPREG        ******** GX     06
SYS$LKWSET        ******** GX     06
VCB_TYPE        = 00000002
WCB_TYPE        = 00000001
WORKING_SET       00000000 R      05
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$LOCKEDC0$	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC PAGE
$LOCKEDC9$	00000000 (0.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
$LOCKEDD0$	00000000 (0.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
$LOCKEDD9$	00000000 (0.)	04 (4.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
$LOCKEDD1$	00000010 (16.)	05 (5.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
$CODE$	0000007B (123.)	06 (6.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.84
Command processing	111	00:00:00.67	00:00:06.56
Pass 1	125	00:00:01.17	00:00:05.77
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	42	00:00:00.60	00:00:02.02
Symbol table output	3	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	315	00:00:02.60	00:00:15.26

```
The working set limit was 900 pages.
4069 bytes (8 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 20 non-local and 1 local symbols.
344 source lines were read in Pass 1, producing 21 object records in Pass 2.
12 pages of virtual memory were used to define 11 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	5

41 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:LOCKDN/OBJ=OBJ$:LOCKDN MSRC$:MTADEF1/UPDATE=(ENH$:MTADEF1)+MSRC$:LOCKDN/UPDATE=(ENH$:LOCKDN)+EXECML$/LIB
```

LOCKDB LIS

LOGIO LIS

MATCHNAME LIS

IODONE LIS

LOCKDN LIS

NXTVOL LIS

MAIL LIS

MOUVOL LIS

OPRCOM LIS

INIMTA LIS

MODIFY LIS

PARSE LIS