```
EEEEEEEEEEEEEEEE  MMM       MMM UUU         UUU LLL                    AAAAAAAA   TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE  MMM       MMM UUU         UUU LLL                    AAAAAAAAA  TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE  MMM       MMM UUU         UUU LLL                    AAAAAAAAA  TTTTTTTTTTTTTTT
EEE               MMMMM   MMMMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMMMMM MMMMMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMMMMM MMMMMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMM MMM   MMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMM  MMM  MMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMM   MMM MMM UUU         UUU LLL                AAA        AAA        TTT
EEEEEEEEEEEEE     MMM       MMM UUU         UUU LLL                AAA        AAA        TTT
EEEEEEEEEEEEE     MMM       MMM UUU         UUU LLL                AAA        AAA        TTT
EEEEEEEEEEEEE     MMM       MMM UUU         UUU LLL                AAAAAAAAAAAAAAA       TTT
EEE               MMM       MMM UUU         UUU LLL                AAAAAAAAAAAAAAA       TTT
EEE               MMM       MMM UUU         UUU LLL                AAAAAAAAAAAAAAA       TTT
EEE               MMM       MMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMM       MMM UUU         UUU LLL                AAA        AAA        TTT
EEE               MMM       MMM UUU         UUU LLL                AAA        AAA        TTT
EEEEEEEEEEEEEEEE  MMM       MMM UUUUUUUUUUUUUUU LLLLLLLLLLLLLLL    AAA        AAA        TTT
EEEEEEEEEEEEEEEE  MMM       MMM UUUUUUUUUUUUUUU LLLLLLLLLLLLLLL    AAA        AAA        TTT
EEEEEEEEEEEEEEEE  MMM       MMM UUUUUUUUUUUUUUU LLLLLLLLLLLLLLL    AAA        AAA        TTT
```

```
VV      VV   AAAAAA    XX    XX  LL                000000    AAAAAA   DDDDDDD
VV      VV   AAAAAA    XX    XX  LL                000000    AAAAAA   DDDDDDD
VV      VV   AA    AA  XX    XX  LL              00    00  AA    AA   DD    DD
VV      VV   AA    AA  XX    XX  LL              00    00  AA    AA   DD    DD
VV      VV   AA    AA   XX  XX   LL              00    00  AA    AA   DD    DD
VV      VV   AA    AA   XX  XX   LL              00    00  AA    AA   DD    DD
VV      VV   AA    AA    XX      LL              00    00  AA    AA   DD    DD
VV      VV   AA    AA    XX      LL              00    00  AA    AA   DD    DD
VV      VV   AAAAAAAAAA  XX XX   LL              00    00  AAAAAAAAAA DD    DD
VV      VV   AAAAAAAAAA  XX XX   LL              00    00  AAAAAAAAAA DD    DD
 VV    VV    AA    AA   XX   XX  LL              00    00  AA    AA   DD    DD   ....
  VV  VV     AA    AA  XX    XX  LL              00    00  AA    AA   DD    DD   ....
   VV        AA    AA  XX    XX  LLLLLLLLL        000000   AA    AA   DDDDDDD    ....
   VV        AA    AA  XX    XX  LLLLLLLLLL       000000   AA    AA   DDDDDDD    ....
```

```
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
LLLLLLLLL         IIIIII      SSSSSSSS
LLLLLLLLLL        IIIIII      SSSSSSSS
```

```
0000      1              .NLIST   CND
0000      5              .TITLE   VAX$LOAD - HEADER FOR LOADABLE CHAR/DECIMAL EMULATION
0000      7              .IDENT   'V04-000'
0000      8
0000      9
0000     10  ;*******************************************************************************
0000     11  ;*                                                                             *
0000     12  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                    *
0000     13  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
0000     14  ;*  ALL RIGHTS RESERVED.                                                       *
0000     15  ;*                                                                             *
0000     16  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED      *
0000     17  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH LICENSE  AND WITH THE       *
0000     18  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER      *
0000     19  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY      *
0000     20  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY      *
0000     21  ;*  TRANSFERRED.                                                               *
0000     22  ;*                                                                             *
0000     23  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE      *
0000     24  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT      *
0000     25  ;*  CORPORATION.                                                               *
0000     26  ;*                                                                             *
0000     27  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS      *
0000     28  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                    *
0000     29  ;*                                                                             *
0000     30  ;*                                                                             *
0000     31  ;*******************************************************************************
0000     32
0000     33  ;++
0000     34  ;
0000     35  ; Facility:
0000     36  ;
0000     37  ;       Instruction Emulator
0000     38  ;
0000     39  ; Abstract:
0000     40  ;
0000     41  ;       This module defines the data structures required for a piece
0000     42  ;       of loadable code.  This includes the pool header and the code
0000     43  ;       needed to hook into the rest of the system.  For the instruction
0000     44  ;       emulation code, the hooks are vectors in the SCB.
0000     45  ;
0000     46  ; Environment: MODE=Kernel
0000     47  ;
0000     48  ; Author:  Kathleen D. Morse, Creation date:  04-May-1983
0000     49  ;
0000     50  ; Modified by:
0000     51  ;
0000     52  ;       V03-004 LJK0028          Lawrence J. Kenah         10-Apr-1984
0000     53  ;               Store base address of emulator image in cell in SYS.EXE
0000     54  ;               set aside for that purpose.
0000     55  ;
0000     56  ;       V03-003 LJK0027          Lawrence J. Kenah         21-Mar-1984
0000     57  ;               Store address of access violation handler into EXE$GL_VAXEXCVEC
0000     58  ;               when loading decimal/string emulator.
0000     59  ;
0000     60  ;       V03-002 LJK0017          Lawrence J. Kenah         17-Jan-1984
0000     61  ;               Make table entries for SCB entries position independent.
```

```
           0000   62 ;                    Change PSECT attributes.
           0000   63 ;
           0000   64 ;        V03-001 WMC0001         Wayne Cardoza              23-Jun-1983
           0000   65 ;                 Fix SLVTAB.
           0000   66 ;
           0000   67 ;--
           0000   68
           0000   69 ;
           0000   70 ; INCLUDE FILES:
           0000   71 ;
           0000   72
           0000   73         $PRTDEF                              ; Define protection codes
           0000   74         $PTEDEF                              ; Define page table entry fields
           0000   75         $VADEF                               ; Define virtual address fields
           0000   76
           0000   77 ; This must be the first program section in the image file.
           0000   78
       00000000   79         .PSECT  $$$$$$BEGIN      PAGE,PIC,USR,CON,REL,GBL,SHR,NOWRT
           0000   80
           0000   81         .ENABLE         LOCAL_BLOCK
           0000   82
           0000   83 10$:
           0000   84
           0000   94 VAX$BEGIN::                                  ; Beginning of string/decimal emulator
           0000   95         SLVTAB  END=VAX$END , -             ; Size of string/decimal ins emulator
           0000   96                 INITRTN=VAX$INIT , -        ; Address of initialization routine
           0000   97                 SUBTYP=DYN$C_NON_PAGED, -   ; Sub-type for data structure
           0000   98                 PROT W=PRT$C_URKW , -       ; Protection on loadable code pages
           0000   99                 FACILITY=VAXEMUL.EXE        ; Name of image loaded
           0024  101
           0024  102
           0024  110 SCB_UVAX:                                    ; Hook for SCB uVAX except
000000C8   0024  111         .LONG   ^XC8                        ;  Offset into SCB
FFFFFE01'  0028  112         .LONG   VAX$EMULATE - VAX$BEGIN_UR  ;  Offset to emulator entry pt
           002C  113 SCB_UVAX_FPD:                                ; Hook for SCB uVAX FPD except
000000CC   002C  114         .LONG   ^XCC                        ;  Offset into SCB
FFFFFE01'  0030  115         .LONG   VAX$EMULATE_FPD - VAX$BEGIN_UR ;  Offset to emulator entry pt
00000000   0034  116         .LONG   0                           ; Empty hook ends table
           0038  118
00000038   0038  119 ...SIZE... = .-10$
           0038  120
           0038  121 ; Insure at least one page before real code begins
           0038  122
           0038  123 SPACE_FILLER1:                               ; This prevents UR access to
000001FF   0038  124         .BLKB   <511 - ...SIZE...>          ;  the pool fragments on either
           01FF  125                                             ;  side of the emulation code.
           01FF  126         .DISABLE        LOCAL_BLOCK
           01FF  127
           01FF  131 VAX$BEGIN_UR::                              ; Starting VA to protect UR
```

VAX$LOAD
V04-000

G 1
- HEADER FOR LOADABLE CHAR/DECIMAL EMULA 16-SEP-1984 01:41:55   VAX/VMS Macro V04-00   Page   3
VAX$INIT - Initialization routine to hoo 5-SEP-1984 00:44:19   [EMULAT.SRC]LOADHDR.MAR;1   (2)

```
                        01FF     137                    .SBTTL  VAX$INIT - Initialization routine to hook into SCB
                        01FF     139
                        01FF     140   ;++
                        01FF     141   ; Functional Description:
                        01FF     142   ;
                        01FF     146   ;        VAX$INIT is linked together with all of the code required for
                        01FF     148   ;        the instruction emulator.  The necessary amount of non-paged pool
                        01FF     149   ;        is allocated and rounded up to page boundary.  Code is then
                        01FF     150   ;        moved into this block of pool.  All of this code must be PIC.
                        01FF     151   ;        This code is then re-protected so that it can be executed from
                        01FF     152   ;        user mode.  A page is allocated on either side of the emulator
                        01FF     153   ;        to serve as buffers, because the code is not loaded on a page
                        01FF     154   ;        boundary and pool cannot be protected UR for security reasons.
                        01FF     155   ;
                        01FF     160   ;        The vectors for instruction emulation and instruction emulation
                        01FF     161   ;        first-part-done are then connected to the emulation code.
                        01FF     163   ;
                        01FF     164   ; Calling Sequence:
                        01FF     165   ;
                        01FF     169   ;        JSB     VAX$INIT
                        01FF     171   ;
                        01FF     172   ; Input Parameters:
                        01FF     173   ;
                        01FF     174   ;        None
                        01FF     175   ;
                        01FF     176   ;--
                        01FF     177
                        01FF     178   ; This PSECT holds the init routines.
                        01FF     179
                    00000000     180                    .PSECT  ____INITHK        BYTE,PIC,USR,CON,REL,GBL,SHR,NOWRT
                        0000     181
                        0000     182                    .ENABLE         LOCAL_BLOCK
                        0000     183
                        0000     184   10$:
                        0000     189   VAX$INIT::                                          ; Hook in emulation code
                        0000     190   VAX$END_UR::                                        ; Also ending VA to protect UR
              7E   50   7D  0000  192                    MOVQ    R0,-(SP)                  ; Save registers
              7E   52   7D  0003  193                    MOVQ    R2,-(SP)                  ; Save registers
                        0006     194
                        0006     195   ;
                        0006     196   ; Now reset the protection on the non-paged pool to be
                        0006     197   ; user-read, so that the emulation code can be accessed from
                        0006     198   ; all modes.  Make it kernel-write so that breakpoints can be
                        0006     199   ; set in the emulation code with XDELTA.
                        0006     200   ;
         51   01FF'CF   9E  0006  204                    MOVAB   W^VAX$BEGIN_UR,R1         ; Get starting VA to protect URKW
   51   51   15   09   EF  000B  206                    EXTZV   #VA$V_VPN,#VA$S_VPN,R1,R1 ; Make address into VPN
         51   51   02   78  0010  207                    ASHL    #2,R1,R1                  ; Make into byte index into SPT
         52   FFE8 CF   9E  0014  211                    MOVAB   W^VAX$END_UR,R2           ; Get ending address to protect URKW
   52   52   15   09   EF  0019  213                    EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R2 ; Make address into VPN
         52   52   02   78  001E  214                    ASHL    #2,R2,R2                  ; Make into byte index into SPT
              50   0E   9A  0022  215                    MOVZBL  #PRT$C_URKW,R0            ; New protection for emulation code
   53   00000000'GF   9E  0025  216                    MOVAB   G^MMG$GL_SPTBASE,R3        ; Get address of system page table
                        002C     217   20$:
00 B341  04   1B   50   F0  002C  218                    INSV    R0,#PTE$V_PROT,#PTE$S_PROT,@(R3)[R1] ; Set new
   FFF3  51   04   52   F1  0033  219                    ACBL    R2,#4,R1,20$              ; protection for each page
                        0039     220                    INVALID                           ; Invalidate the translation buffer
```

VAX$LOAD
V04-000

H 1
- HEADER FOR LOADABLE CHAR/DECIMAL EMULA 16-SEP-1984 01:41:55   VAX/VMS Macro V04-00   Page   4
VAX$INIT - Initialization routine to hoo 5-SEP-1984 00:44:19   [EMULAT.SRC]LOADHDR.MAR;1   (2)

```
                                  003C    221
                                  003C    222
                                  003C    223 ; Now connect the emulation code to the system control block.
                                  003C    224 ;
       53   00000000'GF    DO     003C    225          MOVL      G^EXE$GL_SCB,R3              ; Base address of SCB
 00C8 C3      0000'CF    9E       0043    231          MOVAB     W^VAX$EMULATE,^XC8(R3)      ; Set SCB to point to emulator code
 00CC C3      0000'CF    9E       004A    232          MOVAB     W^VAX$EMULATE_FPD,^XCC(R3)  ; Set SCB to point to emulator code
              0000'CF    9E       0051    233          MOVAB     W^VAX$MODIFY_EXCEPTION,-    ; Store address of access violation
          00000000'GF             0055    234                    G^EXE$GL_VAXEXCVEC         ;  handler
              0000'CF    9E       005A    235          MOVAB     W^VAX$BEGIN,-
          00000000'GF             005E    236                    G^MMG$GL_VAXEMUL_BASE      ; Store base address of image
              52   8E   7D        0063    238          MOVQ      (SP)+,R2                    ; Restore registers
              50   8E   7D        0066    239          MOVQ      (SP)+,R0                    ; Restore registers
                        05        0069    240          RSB                                   ; and return
                                  006A    241
                      0000006A    006A    242 ...INIT_SIZE... = .-10$
                                  006A    243
                                  006A    244          .DISABLE       LOCAL_BLOCK
                                  006A    245
                                  006A    246 ; This must be the last program section in the image
                                  006A    247
                      00000000    0000    248          .PSECT     _____END        BYTE,PIC,USR,CON,REL,GBL,SHR,NOWRT
                                  0000    249
                                  0000    250 ; Insure at least one page at the end of the image, too
                                  0000    251
                                  0000    252 SPACE_FILLER2:                                 ; This prevents UR access to
                      00000195    0000    253          .BLKB    <511 - ...INIT_SIZE...>       ;  the pool fragments on either
                                  0195    254                                                 ;  side of the emulation code.
                                  0195    258 VAX$END::                                       ;
                                  0195    260
                                  0195    261          .END
```

```
...INIT_SIZE...                = 0000006A
...SIZE...                     = 00000038
DYN$C_LOADCODE                 = 00000062
DYN$C_NON_PAGED                = 00000001
EXE$GC_SCB                       ********   X    03
EXE$GL_VAXEXCVEC                 ********   X    03
MMG$GL_SPTBASE                   ********   X    03
MMG$GL_VAXEMUL_BASE              ********   X    03
PR$_TBIA                         ********   X    03
PRT$C_ER                       = 00000007
PRT$C_URKW                     = 0000000E
PTE$S_PROT                     = 00000004
PTE$V_PROT                     = 0000001B
SCB_UVAX                         00000024  R    02
SCB_UVAX_FPD                     0000002C  R    02
SPACE_FILLER1                    00000038  R    02
SPACE_FILLER2                    00000000  R    04
VA$S_VPN                       = 00000015
VA$V_VPN                       = 00000009
VAX$BEGIN                        00000000  RG   02
VAX$BEGIN_UR                     000001FF  RG   02
VAX$EMULATE                      ********   X    02
VAX$EMULATE_FPD                  ********   X    02
VAX$END                          00000195  RG   04
VAX$END_UR                       00000000  RG   03
VAX$INIT                         00000000  RG   03
VAX$MODIFY_EXCEPTION             ********   X    03
```

```
                              +-----------------+
                              ! Psect synopsis  !
                              +-----------------+
```

PSECT name	Allocation		PSECT No.	Attributes										
. ABS .	00000000 (	0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS	00000000 (	0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
$$$$$$BEGIN	000001FF (	511.)	02 (2.)	PIC	USR	CON	REL	GBL	SHR	EXE	RD	NOWRT	NOVEC	PAGE
____INITHK	0000006A (	106.)	03 (3.)	PIC	USR	CON	REL	GBL	SHR	EXE	RD	NOWRT	NOVEC	BYTE
_____END	00000195 (	405.)	04 (4.)	PIC	USR	CON	REL	GBL	SHR	EXE	RD	NOWRT	NOVEC	BYTE

```
                              +-----------------------+
                              ! Performance indicators !
                              +-----------------------+
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	18	00:00:00.05	00:00:01.56
Command processing	86	00:00:00.48	00:00:05.97
Pass 1	127	00:00:03.01	00:00:11.34
Symbol table sort	0	00:00:00.27	00:00:00.46
Pass 2	41	00:00:00.69	00:00:04.21
Symbol table output	4	00:00:00.04	00:00:00.04
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	278	00:00:04.56	00:00:23.61

The working set limit was 900 pages.

VAX$LOAD   - HEADER FOR LOADABLE CHAR/DECIMAL EMULA 16-SEP-1984 01:41:55 VAX/VMS Macro V04-00   Page 6
VAX-11 Macro Run Statistics   5-SEP-1984 00:44:19 [EMULAT.SRC]LOADHDR.MAR;1   (2)

16957 bytes (34 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 250 non-local and 5 local symbols.
261 source lines were read in Pass 1, producing 17 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   5
-255DUA28:[SYSLIB]STARLET.MLB;2   4
TOTALS (all libraries)   9

355 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:VAXLOAD/OBJ=OBJ$:VAXLOAD MSRC$:LOADHDR/UPDATE=(ENH$:LOADHDR)+EXECML$/LIB

VAXLOAD
LIS

ERFBRIEF
MAP

ERFPROC1
MAP

ERFDISK
MAP

ERFBUS
MAP

ERFINICOM
MAP

ERFCOMMON
MAP

VAXSTATUS
LIS

ENCRYP ERF

ENCSTUBS ERF
LIS   MAP

ERFPROC2
MAP

VAXSTRING
LIS