



```

AAAAAA LL LL 000000 CCCCCCCC BBBB88888
AAAAAA LL LL 000000 CCCCCCCC BBBB88888
AA AA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AAAAAAAA LL LL 00 00 CC BB BB
AAAAAAAA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AA AA LL LL 00 00 CC BB BB
AA AA LL LLLLLLLLLL LLLLLLLLLL 000000 CCCCCCCC BBBB88888
AA AA LL LLLLLLLLLL LLLLLLLLLL 000000 CCCCCCCC BBBB88888

```

```

LL I I I I I S S S S S S S
LL I I I I I S S S S S S S
LL I I S S
LL I I S S
LL I I S S
LL I I S S S S S S
LL I I S S S S S S
LL I I S S
LL I I S S
LL I I S S
LL I I S S
LLLLLLLLLLLL I I I I I S S S S S S S
LLLLLLLLLLLL I I I I I S S S S S S S

```

```

0000 1      .TITLE  ALLOCB - ALLOCATE DYNAMIC 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:  mtaacp
0000 32
0000 33     Abstract:
0000 34
0000 35         these routines allocate and deallocate system non-paged
0000 36         dynamic memory for acp control blocks.
0000 37
0000 38     Environment:
0000 39
0000 40         starlet operating system, including privileged system services
0000 41         and internal exec routines. note that this routine must be
0000 42         called in kernel mode.
0000 43
0000 44     Author:  Andrew C. Goldstein, Creation Date:  14-DEC-1976  16:25
0000 45
0000 46     Modified By:
0000 47
0000 48         V02-002 REFORMAT          R Schaefer          23-Jul-1980    15:53
0000 49         Reformat the source.
0000 50     --
0000 51
0000 52
0000 53
0000 54     Include Files:
0000 55
0000 56
0000 57

```

```

0000 58 : Equated Symbols:
0000 59 :
0000 60 : arg list offsets
0000 61 :
0000 62 :
00000004 0000 63 BYTES = 4 ; byte count desired
00000004 0000 64 ADDRESS = 4 ; address of block being deallocated
0000 65 :
0000 66 $IPLDEF ; define system ipl names
0000 67 $PRDEF ; define processor register names
0000 68 $RSNDEF ; define resource names
0000 69 $SWCBDEF ; define window block format
0000 70 : used only for tags to the block type
0000 71 : and size fields
0000 72 :
00000000 73 .PSECT $LOCKEDC1$,NOWRT,LONG
0000 74 :
0000 75 :
0000 76 : Own Storage:
0000 77 :
0000 78 :
0000 79 .ALIGN 2

```

<

```

0000 81
0000 82 :++
0000 83 :
0000 84 : ALLOCATE:      Allocates the requested block size from system
0000 85 :                  non-paged dynamic memory. the block is cleared, and
0000 86 :                  the standard size.
0000 87 :
0000 88 : Calling sequence:
0000 89 :     CALL      ALLOCATE (ARG1)
0000 90 :
0000 91 : Input Parameters:
0000 92 :     ARG1: number of bytes to allocate
0000 93 :
0000 94 : Implicit Inputs:
0000 95 :     none
0000 96 :
0000 97 : Output Parameters:
0000 98 :     none
0000 99 :
0000 100 : Implicit Outputs:
0000 101 :     none
0000 102 :
0000 103 : Routine Value:
0000 104 :     address of block
0000 105 :
0000 106 : Side Effects:
0000 107 :     block allocated
0000 108 :
0000 109 :--
0000 110 :
0000 111 : ALLOCATE::
0000 112 : .WORD      ^M<R2,R3,R4,R5>          ; save the usual registers
51 04 AC 003C 0002 113 10$:  MOVL      BYTES(AP),R1          ; get size argument
0000 114 :     MOVPSL  -(SP)                    ; save the psl for wait call below
0000 115 :     DSBINT  #IPL$ SYNCH,R2          ; raise ipl to synchronize
0000 116 :     JSB     @#EXE$ALONONPAGED       ; get block from exec
0000 117 :     BLBC    R0,20$                  ; branch on failure
0000 118 :     ENBINT  #0                      ; restore ipl
0000 119 :     MOVL    R1,(SP)                 ; clean psl off stack and
62 51 00 62 00 2C 001A 120 :     PUSHL   R2                      ; save returned byte count
0000 121 :     MOVCS   #0,(R2),#0,R1,(R2)     ; and address
0000 122 :     MOVCS   #0,(R2),#0,R1,(R2)     ; zero out the block
0000 123 :
0000 124 :     MOVL    (SP)+,R0                ; get block address
0000 125 :     CVTLW   (SP)+,WCBSW_SIZE(R0)    ; put in size word
0000 126 :     RET                                ; and return
0000 127 :
0000 128 :
0000 129 : : we get here if memory is not available
0000 130 :
0000 131 :
0000 132 : 20$:  MOVZWL  #RSNS NPDYNMEM,R0      ; get appropriate resource code
54 00000000'9F 03 3C 002D 133 :     MOVL    @#SCH$GL CURPCB,R4      ; and process pcb address
0000 134 :     JSB     @#SCH$R$WAIT            ; and wait for pool to appear
0000 135 :     BRB     10$                    ; try again

```

```

003F 137
003F 138 :++
003F 139
003F 140 : DEALLOCATE: Deallocates the indicated block of memory back
003F 141 : to the system pool of non-paged dynamic memory.
003F 142
003F 143 : Calling sequence:
003F 144 : CALL DEALLOCATE (ARG1)
003F 145
003F 146 : Input Parameters:
003F 147 : ARG1: address of block being deallocated
003F 148
003F 149 : Implicit Inputs:
003F 150 : none
003F 151
003F 152 : Output Parameters:
003F 153 : none
003F 154
003F 155 : Implicit Outputs:
003F 156 : none
003F 157
003F 158 : Routine Value:
003F 159 : none
003F 160
003F 161 : Side Effects:
003F 162 : block deallocated
003F 163
003F 164 :--
003F 165
003F 166 DEALLOCATE::
50 04 AC 003C 003F 167 .WORD ^M<R2,R3,R4,R5> ; save registers
51 08 A0 3C 0041 168 MOVL ADDRESS(AP),R0 ; get address of block
00000000'9F 16 0045 169 MOVZWL WCB$W_SIZE(R0),R1 ; get block size
0049 170 DSBINT #IPL$-SYNCH,R2 ; raise ipl to synchronize
004F 171 JSB @#EXE$DEANONPAGED ; and deallocate thru exec
0055 172 ENBINT #0 ; restore ipl
04 0058 173 RET
0059 174
0059 175
0059 176 .END

```

B  
S  
A  
B  
C  
C  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

ALLOCB  
Symbol table

- ALLOCATE DYNAMIC MEMORY

J 12

16-SEP-1984 02:02:23  
5-SEP-1984 02:00:22

VAX/VMS Macro V04-00  
[MTAACP.SRC]ALLOCB.MAR;1

Page 5  
(5)

```

ADDRESS = 00000004
ALLOCATE = 00000000 RG 02
AQB_TYPE = 00000005
BYTES = 00000004
DEALLOCATE = 0000003F RG 02
EXESALONONPAGED ***** X 02
EXESDEANONPAGED ***** X 02
FCB_TYPE = 00000000
IPL$ SYNCH = 00000008
MVL_TYPE = 00000004
PRS IPL = 00000012
RSNS NPDYMEM = 00000003
RVT_TYPE = 00000003
SCH$GL CURPCB ***** X 02
SCH$RWAIT ***** X 02
VCB_TYPE = 00000002
WCB$W SIZE = 00000008
WCB_TYPE = 00000001
  
```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$LOCKEDC1\$	00000059 ( 89.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.09	00:00:00.53
Command processing	165	00:00:00.79	00:00:05.14
Pass 1	170	00:00:02.71	00:00:12.11
Symbol table sort	0	00:00:00.22	00:00:00.68
Pass 2	44	00:00:00.79	00:00:03.60
Symbol table output	4	00:00:00.03	00:00:00.49
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	422	00:00:04.66	00:00:22.58

The working set limit was 1200 pages.  
 13510 bytes (27 pages) of virtual memory were used to buffer the intermediate code.  
 There were 10 pages of symbol table space allocated to hold 201 non-local and 2 local symbols.  
 359 source lines were read in Pass 1, producing 13 object records in Pass 2.  
 20 pages of virtual memory were used to define 18 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
-----	-----
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	9

277 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:ALLOCB/OBJ=OBJ\$:ALLOCB MSRC\$:MTADEF1/UPDATE=(ENH\$:MTADEF1)+MSRC\$:ALLOCB/UPDATE=(ENH\$:ALLOCB)+EXECMLS/LIB



