H 5

```
FFFFFFFFFFFFFFF        111          111     XXX       XXX
FFFFFFFFFFFFFFF        111          111     XXX       XXX
FFFFFFFFFFFFFFF        111          111     XXX       XXX
FFF                 111111       111111     XXX       XXX
FFF                 111111       111111     XXX       XXX
FFF                 111111       111111     XXX       XXX
FFF                    111          111        XXX XXX
FFF                    111          111        XXX XXX
FFF                    111          111        XXX XXX
FFFFFFFF.FFF           111          111          XXX
FFFFFFFFFFFF           111          111          XXX
FFFFFFFFFFFF           111          111          XXX
FFF                    111          111        XXX XXX
FFF                    111          111        XXX XXX
FFF                    111          111        XXX XXX
FFF                    111          111     XXX       XXX
FFF                    111          111     XXX       XXX
FFF                    111          111     XXX       XXX
FFF                 111111111    111111111  XXX       XXX
FFF                 111111111    111111111  XXX       XXX
FFF                 111111111    111111111  XXX       XXX
```

```
_$25

Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP_
MARK
MARK
MARK
MARK
MARK
```

```
  AAAAAA    LL          LL            000000      CCCCCCCC  BBBBBBBB
  AAAAAA    LL          LL            000000      CCCCCCCC  BBBBBBBB
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          BBBBBBBB
AA      AA  LL          LL          00      00  CC          B3BBBBBB
AAAAAAAAAA  LL          LL          00      00  CC          BB      BB
AAAAAAAAAA  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  LLLLLLLLLL  LLLLLLLLLL    000000      CCCCCCCC  BBBBBBBB      ....
AA      AA  LLLLLLLLLL  LLLLLLLLLL    000000      CCCCCCCC  BBBBBBBB      ....

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  ALLOCB - ALLOCATE DYNAMIC MEMORY
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982  '984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION,  .YNARD, 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  ; FACILITY:  F11ACP STRUCTURE LEVEL 2
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;
0000    34  ;       THESE ROUTINES ALLOCATE AND DEALLOCATE SYSTEM
0000    35  ;       DYNAMIC MEMORY FOR FCP CONTROL BLOCKS.
0000    36  ;
0000    37  ; ENVIRONMENT:
0000    38  ;
0000    39  ;       STARLET OPERATING SYSTEM, INCLUDING PRIVILEGED SYSTEM SERVICES
0000    40  ;       AND INTERNAL EXEC ROUTINES. NOTE THAT THIS ROUTINE MUST BE
0000    41  ;       CALLED IN KERNEL MODE.
0000    42  ;
0000    43  ;--
0000    44  ;
0000    45  ; AUTHOR:  ANDREW C. GOLDSTEIN, CREATION DATE:  14-DEC-1976  16:25
0000    46  ;
0000    47  ; MODIFIED BY:
0000    48  ;
0000    49  ;       V03-009 LMP0258         L. Mark Pilant,         25-Jun-1984  9:52
0000    50  ;               Make sure IPL is lowered when leaving the allocation routine.
0000    51  ;
0000    52  ;       V03-008 CDS0003         Christian D. Saether     9-Apr-1984
0000    53  ;               Charge WCBs against BYTLM also.
0000    54  ;
0000    55  ;       V03-007 ACG0408         Andrew C. Goldstein,    23-Mar-1984  10:54
0000    56  ;               Make all of global storage based
0000    57  ;
```

```
0000     58 ;       V03-006 CDS0002         Christian D. Saether     9-Mar-1984
0000     59 ;               Clear type field before deallocating so cleanup can
0000     60 ;               trap double remque attempts.
0000     61 ;
0000     62 ;       V03-005 LMP0192         L. Mark Pilant,          13-Feb-1984  11:36
0000     63 ;               Enter resource wait at IPL SYNCH if allocation of paged pool
0000     64 ;               fails.
0000     65 ;
0000     66 ;       V03-004 LMP0186         L. Mark Pilant,          3-Feb-1984  11:57
0000     67 ;               Add support for CHIP blocks from paged pool.
0000     68 ;
0000     69 ;       V03-003 LMP0036         L. Mark Pilant,          24-Aug-1982  10:52
0000     70 ;               Add support for Access Control List blocks in paged pool.
0000     71 ;
0000     72 ;       V03-002 CDS0001         C Saether                 2-Aug-1982
0000     73 ;               Make PCB address calculation PIC.
0000     74 ;
0000     75 ;       V03-001 LMP0017         L. Mark Pilant,          29-Mar-1982  9:45
0000     76 ;               Put back in the quota checking.
0000     77 ;
0000     78 ;       V02-002 LMP0011         L. Mark Pilant,          23-Feb-1982 15:00
0000     79 ;               Remove the quota checking inserted in LMP0003.  This is for
0000     80 ;               the V3 FT2 release only.
0000     81 ;
0000     82 ;       V02-002 LMP0003         L. Mark Pilant,          4-Dec-1981  09:05
0000     83 ;               Charge the user for any windows created.
0000     84 ;
0000     85 ;       A0101   ACG0081         Andrew C. Goldstein,     6-Nov-1979  21:40
0000     86 ;               Add cache block for quota cacheing
0000     87 ;
0000     88 ;       A0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  20:04
0000     89 ;               Previous revision history moved to F11A.REV
0000     90 ;**
0000     91
0000     92 ;
0000     93 ; INCLUDE FILES:
0000     94 ;
0000     95 ;       INCLUDE "FCPDEF.MAR"
0000     96
0000     97 ;
0000     98 ; EQUATED SYMBOLS:
0000     99 ;
0000    100 ; ARG LIST OFFSETS
0000    101 ;
00000004 0000    102 BYTES   = 4                         ; BYTE COUNT DESIRED
00000008 0000    103 TYPE    = 8                         ; BLOCK TYPE BEING ALLOCATED
00000004 0000    104 ADDRESS = 4                         ; ADDRESS OF BLOCK BEING DEALLOCATED
0000    105
0000    106         $DYNDEF                             ; DEFINE STRUCTURE TYPE CODES
0000    107         $IPLDEF                             ; DEFINE SYSTEM IPL NAMES
0000    108         $IRPDEF                             ; DEFINE I/O PACKET OFFSETS
0000    109         $JIBDEF                             ; DEFINE JOB INFORMATION BLOCK FORMAT
0000    110         $PCBDEF                             ; DEFINE PROCESS CONTROL BLOCK FORMAT
0000    111         $PRDEF                              ; DEFINE PROCESSOR REGISTER NAMES
0000    112         $RSNDEF                             ; DEFINE RESOURCE NAMES
0000    113         $WCBDEF                             ; DEFINE WINDOW BLOCK FORMAT
0000    114                                             ; USED ONLY FOR TAGS TO THE BLOCK TYPE
```

```
                     0000   115                                             ; AND SIZE FIELDS
                     0000   116
                 00000000   117          .PSECT  $LOCKEDC1$,NOWRT,LONG
                     0000   118 ;
                     0000   119 ; OWN STORAGE:
                     0000   120 ;
                     0000   121 BLOCK_TYPE:                                 ; LIST OF SYSTEM TYPE CODES,
                  07 0000   122          .BYTE   DYN$C_FCB,-                ; INDEXED BY FCP TYPE CODE
                     0001   123                  DYN$C_WCB,-
                  12 0001   124                  DYN$C_VCB,-
                  11 0002   125                  DYN$C_RVT,-                ; RELATIVE VOLUME TABLE CONTROL BLOCK
                  0E 0003   126                  DYN$C_MVL,-                ; MAGNETIC TAPE VOLUME LIST
                  16 0004   127                  DYN$C_AQB,-                ; ACP QUEUE CONTROL BLOCK
                  03 0005   128                  DYN$C_VCA,-                ; ACP CACHE DATA BLOCK
                  32 0006   129                  DYN$C_ACL,-                ; ACCESS CONTROL LIST BLOCK
               48 3F 0007   130                  DYN$C_CHIP                 ; CHIP BLOCK
                     0009   131
                     0009   132          .ALIGN  2
```

```
                     000C   134 ;++
                     000C   135 ;
                     000C   136 ; FUNCTIONAL DESCRIPTION:
                     000C   137 ;
                     000C   138 ; THIS ROUTINE ALLOCATES THE REQUESTED BLOCK SIZE FROM SYSTEM
                     000C   139 ; NON-PAGED DYNAMIC MEMORY. THE BLOCK IS CLEARED, AND THE STANDARD
                     000C   140 ; SIZE AND TYPE DATA IS INSERTED.
                     000C   141 ;
                     000C   142 ; CALLING SEQUENCE:
                     000C   143 ;       CALL    ALLOCATE (ARG1, ARG2)
                     000C   144 ;
                     000C   145 ; INPUT PARAMETERS:
                     000C   146 ;       ARG1: NUMBER OF BYTES TO ALLOCATE
                     000C   147 ;       ARG2: TYPE OF BLOCK
                     000C   148 ;
                     000C   149 ; IMPLICIT INPUTS:
                     000C   150 ;       NONE
                     000C   151 ;
                     000C   152 ; OUTPUT PARAMETERS:
                     000C   153 ;       NONE
                     000C   154 ;
                     000C   155 ; IMPLICIT OUTPUTS.
                     000C   156 ;       NONE
                     000C   157 ;
                     000C   158 ; ROUTINE VALUE:
                     000C   159 ;       ADDRESS OF BLOCK
                     000C   160 ;
                     000C   161 ; SIDE EFFECTS:
                     000C   162 ;       BLOCK ALLOCATED
                     000C   163 ;
                     000C   164 ;--
                     000C   165
                     000C   166 ALLOCATE::
                 003C  000C   167                .WORD   ^M<R2,R3,R4,R5>         ; SAVE THE USUAL REGISTERS
      51    04 AC    D0  000E   168 10$:            MOVL    BYTES(AP),R1            ; GET SIZE ARGUMENT
         51    0F    C0  0012   169                 ADDL2   #15,R1                  ; ASSUME 16 BYTE GRANULARITY
         51    0F    CA  0015   170                 BICL2   #15,R1
                         0018   171                 SETIPL  #IPL$_SYNCH             ; RAISE IPL TO SYNCHRONIZE
      01    08 AC    D1  001B   172                 CMPL    TYPE(AP),#WCB_TYPE      ; IS THE BLOCK TO BE A WCB?
               39    12  001F   173                 BNEQ    30$                     ; XFER IF NOT, NO FURTHER CHECKS NEEDED
            0000'CA  D5  0021   174                 TSTL    W^CONTEXT_SAVE(R10)     ; ELSE CHECK FOR SECONDARY CONTEXT
               33    12  0025   175                 BNEQ    30$                     ; XFER IF IN SECONDARY CONTEXT, NO CHARGE
      50    0000'CA  D0  0027   176                 MOVL    W^CURRENT_WINDOW(R10),R0 ; GET CURRENT WINDOW ADDRESS
               05    13  002C   177                 BEQL    20$                     ; XFER IF NONE, NO SHARING POSSIBLE
            0C A0    B5  002E   178                 TSTW    WCB$L_PID(R0)           ; ELSE CHECK FOR SHARED WINDOWS
               27    13  0031   179                 BEQL    30$                     ; XFER IF SHARED, NO CHARGE
      50    0000'CA  D0  0033   180 20$:           MOVL    W^IO_PACKET(R10),R0     ; ELSE GET ADDRESS OF THE IRP
        50    0C A0    3C  0038   181                 MOVZWL  IRP$L_PID(R0),R0        ; GET PROCESS INDEX
     00000000'GF    DD  003C   182                 PUSHL   G^SCH$GL_PCBVEC         ; GET ADDRESS OF PCB VECTOR
           50  9E40    D0  0042   183                 MOVL    @(SP)+[R0],R0           ; GET PCB ADDRESS
      50    0080 C0    D0  0046   184                 MOVL    PCB$L_JIB(R0),R0        ; GET JIB ADDRESS
52    20 A0    51    C3  004B   185                 SUBL3   R1,JIB$L_BYTCNT(R0),R2  ; CALCULATE NEW BYTE COUNT
               4D    15  0050   186                 BLEQ    60$                     ; XFER IF NOT ENOUGH QUOTA
         20 A0    52    D0  0052   187                 MOVL    R2,JIB$L_BYTCNT(R0)     ; SET NEW BYTE COUNT
         24 A0    51    C2  0056   188                 SUBL2   R1,JIB$L_BYTLM(R0)      ; REDUCE BYTLM - THIS IS LONG-LIVED ALLOCATI
               7E    DC  005A   189 30$:            MOVPSL  -(SP)                   ; SAVE THE PSL FOR WAIT CALL BELOW
     00000000'9F    16  005C   190                 JSB     @#EXE$ALONONPAGED       ; GET BLOCK FROM EXEC
```

```
                     21 50    E9  0062   191          BLBC    R0,50$                      ; BRANCH ON FAILURE
                                 0065   192          SETIPL  #0                          ; RESTORE IPL
                                 0068   193                                              ; CLEAN PSL OFF STACK AND
                        6E    51 D0    0068   194          MOVL    R1,(SP)                     ; SAVE RETURNED BYTE COUNT
                              52 DD    006B   195          PUSHL   R2                          ; AND ADDRESS
          62    51    00    62 00    2C  006D   196          MOVC5   #0,(R2),#0,R1,(R2)          ; ZERO OUT THE BLOCK
                                 0073   197
                        50    8E D0    0073   198          MOVL    (SP)+,R0                    ; GET BLOCK ADDRESS
                     08 A0    8E F7    0076   199          CVTLW   (SP)+,WCB$W_SIZE(R0)        ; PUT IN SIZE WORD
                     51    08 AC D0    007A   200          MOVL    TYPE(AP),R1                 ; GET BLOCK TYPE ARG
                0A A0    FF7D CF41    90  007E   201          MOVB    BLOCK_TYPE[R1],WCB$B_TYPE(R0)
                                 04     0085   202          RET                                 ; AND RETURN
                                        0086   203 ;
                                        0086   204 ; WE GET HERE IF MEMORY IS NOT AVAILABLE
                                        0086   205 ;
                     06    08 AC    D1  0086   206 50$:     CMPL    TYPE(AP),#CACHE_TYPE        ; SEE IF THIS IS A CACHE BLOCK
                              13    13  008A   207          BEQL    60$                         ; BRANCH IF SO - DON'T WAIT
                        50    03 3C    008C   208          MOVZWL  #RSN$_NPDYNMEM,R0           ; GET APPROPRIATE RESOURCE CODE
                54    00000000'9F    D0  008F   209          MOVL    @#SCH$GL_CURPCB,R4          ; AND PROCESS PCB ADDRESS
                      00000000'9F    16  0096   210          JSB     @#SCH$RWAIT                 ; AND WAIT FOR POOL TO APPEAR
                           FF6F      31  009C   211          BRW     10$                         ; TRY AGAIN
                                        009F   212
                                        009F   213 ; UPON FAILURE, LOWER IPL AND LEAVE.
                                        009F   214
                              50    D4  009F   215 60$:     CLRL    R0                          ; NOTE FAILURE
                                        00A1   216          SETIPL  #0                          ; LOWER IPL
                                 04     00A4   217          RET                                 ; AND RETURN
```

```
                        00A5   219 ;++
                        00A5   220 ;
                        00A5   221 ; FUNCTIONAL DESCRIPTION:
                        00A5   222 ;
                        00A5   223 ; THIS ROUTINE ALLOCATES THE REQUESTED BLOCK SIZE FROM SYSTEM
                        00A5   224 ; PAGED DYNAMIC MEMORY. THE BLOCK IS CLEARED, AND THE STANDARD
                        00A5   225 ; SIZE AND TYPE DATA IS INSERTED.
                        00A5   226 ;
                        00A5   227 ; CALLING SEQUENCE:
                        00A5   228 ;       CALL    ALLOC_PAGED (ARG1, ARG2)
                        00A5   229 ;
                        00A5   230 ; INPUT PARAMETERS:
                        00A5   231 ;       ARG1: NUMBER OF BYTES TO ALLOCATE
                        00A5   232 ;       ARG2: TYPE OF BLOCK
                        00A5   233 ;
                        00A5   234 ; IMPLICIT INPUTS:
                        00A5   235 ;       NONE
                        00A5   236 ;
                        00A5   237 ; OUTPUT PARAMETERS:
                        00A5   238 ;       NONE
                        00A5   239 ;
                        00A5   240 ; IMPLICIT OUTPUTS:
                        00A5   241 ;       NONE
                        00A5   242 ;
                        00A5   243 ; ROUTINE VALUE:
                        00A5   244 ;       ADDRESS OF BLOCK
                        00A5   245 ;
                        00A5   246 ; SIDE EFFECTS:
                        00A5   247 ;       BLOCK ALLOCATED
                        00A5   248 ;
                        00A5   249 ;--
                        00A5   250
                        00A5   251 ALLOC_PAGED::
                   003C 00A5   252         .WORD   ^M<R2,R3,R4,R5>         ; SAVE THE USUAL REGISTERS
          51   04 AC   D0 00A7   253 10$:    MOVL    BYTES(AP),R1            ; GET SIZE ARGUMENT
             51   0F   C0 00AB   254         ADDL2   #15,R1                  ; ASSUME 16 BYTE GRANULARITY
             51   0F   CA 00AE   255         BICL2   #15,R1
                  7E   DC 00B1   256         MOVPSL  -(SP)                   ; SAVE THE PSL FOR WAIT CALL BELOW
        00000000'9F   16 00B3   257         JSB     @#EXE$ALOPAGED          ; GET BLOCK FROM EXEC
              1E 50   E9 00B9   258         BLBC    R0,50$                  ; BRANCH ON FAILURE
                      00BC   259                                             ; CLEAN PSL OFF STACK AND
             6E   51   D0 00BC   260         MOVL    R1,(SP)                 ; SAVE RETURNED BYTE COUNT
                  52   DD 00BF   261         PUSHL   R2                      ; AND ADDRESS
62   51   00   62   00   2C 00C1   262         MOVC5   #0,(R2),#0,R1,(R2)      ; ZERO OUT THE BLOCK
                      00C7   263
             50   8E   D0 00C7   264         MOVL    (SP)+,R0                ; GET BLOCK ADDRESS
          08 A0   8E   F7 00CA   265         CVTLW   (SP)+,WCB$W_SIZE(R0)    ; PUT IN SIZE WORD
          51   08 AC   D0 00CE   266         MOVL    TYPE(AP),R1             ; GET BLOCK TYPE ARG
      0A A0   FF29 CF41   90 00D2   267         MOVB    BLOCK_TYPE[R1],WCB$B_TYPE(R0)
                      04 00D9   268 40$:    RET                             ; AND RETURN
                      00DA   269 ;
                      00DA   270 ; WE GET HERE IF MEMORY IS NOT AVAILABLE
                      00DA   271 ;
                      00DA   272 50$:    SETIPL  #IPL$_SYNCH             ; RAISE IPL TO SYNCHRONIZE
             50   05   3C 00DD   273         MOVZWL  #RSN$_PGDYNMEM,R0       ; GET APPROPRIATE RESOURCE CODE
       54 00000000'9F   D0 00E0   274         MOVL    @#SCH$GL_CURPCB,R4      ; AND PROCESS PCB ADDRESS
        00000000'9F   16 00E7   275         JSB     @#SCH$RWAIT             ; AND WAIT FOR POOL TO APPEAR
```

```
                 00ED   276          SETIPL  #0                      ; RESTORE IPL
FFB4    31       00F0   277          BRW     10$                     ; TRY AGAIN
```

```
                        00F3   279 ;++
                        00F3   280 ;
                        00F3   281 ; FUNCTIONAL DESCRIPTION:
                        00F3   282 ;
                        00F3   283 ;       THIS ROUTINE DEALLOCATES THE INDICATED BLOCK OF MEMORY BACK
                        00F3   284 ;       TO THE SYSTEM POOL OF NON-PAGED DYNAMIC MEMORY.
                        00F3   285 ;
                        00F3   286 ; CALLING SEQUENCE:
                        00F3   287 ;       CALL    DEALLOCATE (ARG1)
                        00F3   288 ;
                        00F3   289 ; INPUT PARAMETERS:
                        00F3   290 ;       ARG1: ADDRESS OF BLOCK BEING DEALLOCATED
                        00F3   291 ;
                        00F3   292 ; IMPLICIT INPUTS:
                        00F3   293 ;       NONE
                        00F3   294 ;
                        00F3   295 ; OUTPUT PARAMETERS:
                        00F3   296 ;       NONE
                        00F3   297 ;
                        00F3   298 ; IMPLICIT OUTPUTS:
                        00F3   299 ;       NONE
                        00F3   300 ;
                        00F3   301 ; ROUTINE VALUE:
                        00F3   302 ;       NONE
                        00F3   303 ;
                        00F3   304 ; SIDE EFFECTS:
                        00F3   305 ;       BLOCK DEALLOCATED
                        00F3   306 ;
                        00F3   307 ;--
                        00F3   308
                        00F3   309 DEALLOCATE::
                003C    00F3   310          .WORD   ^M<R2,R3,R4,R5>         ; SAVE REGISTERS
  50    04 AC   D0      00F5   311          MOVL    ADDRESS(AP),R0          ; GET ADDRESS OF BLOCK
  51    08 A0   3C      00F9   312          MOVZWL  WCB$W_SIZE(R0),R1       ; GET BLOCK SIZE
     51    0F   C0      00FD   313          ADDL2   #15,R1                  ; ASSUME 16 BYTE GRANULARITY
     51    0F   CA      0100   314          BICL2   #15,R1
                        0103   315          SETIPL  #IPL$_SYNCH             ; RAISE IPL TO SYNCHRONIZE
  12    0A A0   91      0106   316          CMPB    WCB$B_TYPE(R0),#DYN$C_WCB       ; IS THE BLOCK A WCB?
              2B   12   010A   317          BNEQ    10$                     ; XFER IF NOT, NO FURTHER CHECKS
           0C A0   B5   010C   318          TSTW    WCB$L_PID(R0)           ; ELSE CHECK FOR SHARED WINDOWS
              26   13   010F   319          BEQL    10$                     ; XFER IF NOT A SHARED WINDOW
        0000'CA   D5   0111   320          TSTL    W^CONTEXT_SAVE(R10)     ; ELSE SEE IF IN SECONDARY CONTEXT
              20   12   0115   321          BNEQ    10$                     ; XFER IF SO, NO CREDIT
52      0000'CA   D0   0117   322          MOVL    W^IO_PACKET(R10),R2     ; ELSE GET IRP ADDRESS
  52    0C A2   3C     011C   323          MOVZWL  IRP$L_PID(R2),R2        ; GET PROCESS INDEX
 00000000'GF    DD     0120   324          PUSHL   G^SCH$GL_PCBVEC         ; PUSH PCB VECTOR ADDRESS
   52   9E42    D0     0126   325          MOVL    @(SP)+[R2],R2           ; GET PCB ADDRESS
52    0080 C2   D0     012A   326          MOVL    PCB$L_JIB(R2),R2        ; GET JIB ADDRESS
  20 A2   51    C0     012F   327          ADDL2   R1,JIB$L_BYTCNT(R2)     ; CREDIT THE USER
  24 A2   51    C0     0133   328          ADDL2   R1,JIB$L_BYTLM(R2)      ; RESTORE BYTLM ALSO.
        0A A0   94     0137   329 10$:      CLRB    WCB$B_TYPE (R0)         ; CLEAR SO WE CAN CATCH DOUBLE REMQUE
 00000000'9F    16     013A   330          JSB     @#EXE$DEANONPAGED       ; AND DEALLOCATE THRU EXEC
                       0140   331          SETIPL  #0                      ; RESTORE IPL
                    04 0143   332          RET
```

```
                     0144    334 ;++
                     0144    335 ;
                     0144    336 ; FUNCTIONAL DESCRIPTION:
                     0144    337 ;
                     0144    338 ;       THIS ROUTINE DEALLOCATES THE INDICATED BLOCK OF MEMORY BACK
                     0144    339 ;       TO THE SYSTEM POOL OF PAGED DYNAMIC MEMORY.
                     0144    340 ;
                     0144    341 ; CALLING SEQUENCE:
                     0144    342 ;       CALL    DALLOC_PAGED (ARG1)
                     0144    343 ;
                     0144    344 ; INPUT PARAMETERS:
                     0144    345 ;       ARG1: ADDRESS OF BLOCK BEING DEALLOCATED
                     0144    346 ;
                     0144    347 ; IMPLICIT INPUTS:
                     0144    348 ;       NONE
                     0144    349 ;
                     0144    350 ; OUTPUT PARAMETERS:
                     0144    351 ;       NONE
                     0144    352 ;
                     0144    353 ; IMPLICIT OUTPUTS:
                     0144    354 ;       NONE
                     0144    355 ;
                     0144    356 ; ROUTINE VALUE:
                     0144    357 ;       NONE
                     0144    358 ;
                     0144    359 ; SIDE EFFECTS:
                     0144    360 ;       BLOCK DEALLOCATED
                     0144    361 ;
                     0144    362 ;--
                     0144    363
                     0144    364 DALLOC_PAGED::
                003C 0144    365         .WORD   ^M<R2,R3,R4,R5>         ; SAVE REGISTERS
  50   04 AC   D0   0146    366         MOVL    ADDRESS(AP),R0          ; GET ADDRESS OF BLOCK
  51   08 A0   3C   014A    367         MOVZWL  WCB$W_SIZE(R0),R1       ; GET BLOCK SIZE
     51   0F   C0   014E    368         ADDL2   #15,R1                  ; ASSUME 16 BYTE GRANULARITY
     51   0F   CA   0151    369         BICL2   #15,R1
00000000'9F   16   0154    370         JSB     @#EXE$DEAPAGED          ; AND DEALLOCATE THRU EXEC
                04   015A    371         RET
                     015B    372
                     015B    373
                     015B    374
                     015B    375         .END
```

```
ACL_TYPE                        = 00000007
ADDRESS                         = 00000004
ALLOCATE                          0000000C RG    02
ALLOC_PAGED                       000000A5 RG    02
AQB_TYPE                        = 00000005
BITMAP_TYPE                     = 00000001
BLOCK_TYPE                        00000000 R     02
BYTES                           = 00000004
CACHE_TYPE                      = 00000006
CHIP_TYPE                       = 00000008
CONTEXT_SAVE                      ********   X   02
CURRENT_WINDOW                    ********   X   02
DALLOC_PAGED                      00000144 RG    02
DATA_TYPE                       = 00000004
DEALLOCATE                        000000F3 RG    02
DIRECTORY_TYPE                  = 00000002
DYN$C_ACL                       = 0000003F
DYN$C_AQB                       = 00000003
DYN$C_CHIP                      = 00000048
DYN$C_FCB                       = 00000007
DYN$C_MVL                       = 00000016
DYN$C_RVT                       = 0000000E
DYN$C_VCA                       = 00000032
DYN$C_VCB                       = 00000011
DYN$C_WCB                       = 00000012
EXE$ALONONPAGED                   ********   X   02
EXE$ALOPAGED                      ********   X   02
EXE$DEANONPAGED                   ********   X   02
EXE$DEAPAGED                      ********   X   02
FCB_TYPE                        = 00000000
HEADER_TYPE                     = 00000000
INDEX_TYPE                      = 00000003
IO_PACKET                         ********   X   02
IPL$_SYNCH                      = 00000008
IRP$L_PID                       = 0000000C
JIB$L_BYTCNT                    = 00000020
JIB$L_BYTLM                     = 00000024
MVL_TYPE                        = 00000004
PCB$L_JIB                       = 00000080
PR$_IPL                         = 00000012
QUOTA_TYPE                      = 00000005
RSN$_NPDYNMEM                   = 00000003
RSN$_PGDYNMEM                   = 00000005
RVT_TYPE                        = 00000003
SCH$GL_CURPCB                     ********   X   02
SCH$GL_PCBVEC                     ********   X   02
SCH$RWAIT                         ********   X   02
TYPE                            = 00000008
VCB_TYPE                        = 00000002
WCB$B_TYPE                      = 0000000A
WCB$L_PID                       = 0000000C
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
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$LOCKEDC1$	0000015B (347.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.12	00:00:00.82
Command processing	150	00:00:00.77	00:00:03.97
Pass 1	249	00:00:06.69	00:00:15.37
Symbol table sort	0	00:00:00.93	00:00:02.03
Pass 2	77	00:00:01.73	00:00:04.27
Symbol table output	8	00:00:00.07	00:00:00.07
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	518	00:00:10.34	00:00:26.58

The working set limit was 1350 pages.
38421 bytes (76 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 686 non-local and 9 local symbols.
476 source lines were read in Pass 1, producing 14 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	8
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	12

747 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

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

CHKDMO
LIS

CPYNAM
LIS

CLENUP
LIS

CHKPRO
LIS

CHARGEQ
LIS

COMMON
LIS

CREATE
LIS

BADSCN
LIS

CHKHD2
LIS

CHKSUM
LIS

ALLOCB
LIS