

SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE CREATE
 OBJECT MODULE PLACED IN :F1:CREATE.OBJ
 INVOCATION LINE CONTROLS: DEBUG

```

LOC  OBJ          LINE  SOURCE
                1 +1  $TITLE ('KAOS - OBJECT CREATION ROUTINES')
                2      NAME      CREATE
                3
                4      ;;;      Intel Corporation Proprietary Information. This
                5      ;        listing is supplied under the terms of a license
                6      ;        agreement with Intel Corporation and may not be copied
                7      ;        nor disclosed except in accordance with the terms of
                8      ;        the agreement.
                9
               10
               11      ;        DATA STRUCTURE DEFINITIONS
               12
               13      ;        PROCESS CONTROL BLOCK (PCB) DEFINITION
               14
-----
               15      PCB      STRUC
0000           16      NEXT     DW      0
0002           17      LAST     DW      0
0004           18      TYPEF    DW      0
0006           19      QUEUE    DW      0
0008           20      PRI      DW      0
000A           21      STKOFF   DW      0
000C           22      STKSEG   DW      0
-----
               23      PCB      ENDS
               24
               25      ;        SEMAPHORE AND MAILBOX HEADER DEFINITION
               26
-----
               27      HEADER   STRUC
0000           28      HEAD     DW      0
0002           29      TAIL     DW      0
0004           30      DUMMY    DW      0          ;TYPE BUT CAN'T REDECLARE
0006           31      COUNT    DW      0
0008           32      MHEAD    DC      0          ;MAILBOXES ONLY
000C           33      MTAIL    DC      0          ;MAILBOXES ONLY
-----
               34      HEADER   ENDS
               35
               36      ;        DEFINITION OF OBJECT TYPES FOR TYPEF FIELD
               37
0014           38      SMPH     EQU      14H          ;SEMAPHORE TYPE
0015           39      MBX      EQU      15H          ;MAILBOX TYPE
0016           40      RLH      EQU      16H          ;READY LIST HEADER TYPE
0017           41      PCBT     EQU      17H          ;PROCESS CONTROL BLOCK TYPE
               42
               43      ;        GROUP AND EXTERNAL DEFINITIONS
               44
               45      DGROUP   GROUP   DATA
-----
               46      DATA   SEGMENT byte PUBLIC 'DATA'
               47      EXTRN   CP:WORD
               48      EXTRN   CPRI:WORD
               49      EXTRN   TPRI:WORD
               50      EXTRN   READYLIST:WORD

```

LOC	OBJ	LINE	SCURCE
		51	EXTRN READYLISTSIZE:ABS
0000	(1	52	PTRSAVE DD 1 DUP (?)
	????????		
)		
----		53	DATA ENDS
		54	
		55	CGROUP GROUP CCDE
----		56	CCDE SEGMENT byte PUBLIC 'CODE'
		57	ASSUME CS:CGROUP,DS:DGROUP
		58	EXTRN CREATEALARM:NEAR
		59	
		60 +1	\$ EJECT

```

LOC  OBJ          LINE   SOURCE
                                61   ;;;   CREATSEMAPHORE (SEMAPHORE$, INITIAL) - CREATE SEMAPHORE.
                                62   ;
                                63   ;   PARAMETERS:
                                64   ;   SEMAPHORE$ - OFFSET OF AN 8 BYTE BLOCK OF MEMORY
                                65   ;   IN WHICH TO CREATE THE SEMAPHORE.
                                66   ;   INITIAL - WORD COUNT OF NUMBER OF SIGNALS TO INITIALIZE
                                67   ;   THE SEMAPHORE TO.
                                68
                                69
                                70   PUBLIC CREATSEMAPHORE
0000          71   CREATSEMAPHORE:
0000 58          72   PCP     AX             ;POP RETURN ADDRESS
0001 59          73   PCP     CX             ;POP INITIAP
0002 5B          74   PCP     BX             ;POP BLOCK POINTER
0003 891F        75   MCV     [BX].HEAD,BX      ;INITIALIZE HEAD POINTER
0005 895F02      76   MCV     [BX].TAIL,BX      ; AND TAIL POINTER
0008 C747041400  77   MOV     [BX].TYPEF,SMPH ;INITIALIZE TYPEF
000D F7D9        78   NEG     CX             ;INITIALIZE COUNT
000F 894F06      79   MCV     [BX].COUNT,CX
0012 FFED        80   JMP     AX             ;RETURN
                                81
                                82
                                83
                                84
                                85   ;;;   CREAT$MAILBOX (MAILBOX$) - CREATE MAILBOX.
                                86   ;
                                87   ;   PARAMETER:
                                88   ;   MAILBOX$ - OFFSET OF A 16 BYTE BLOCK OF MEMORY
                                89   ;   IN WHICH TO CREATE THE MAILBOX.
                                90
                                91
                                92   PUBLIC CREATMAILBOX
0014          93   CREATMAILBOX:
0014 58          94   POP     AX             ;POP RETURN ADDRESS
0015 5B          95   PCP     BX             ;POP BLOCK POINTER
0016 891F        96   MCV     [BX].HEAD,BX      ;INITIALIZE HEAD POINTER
0018 895F02      97   MOV     [BX].TAIL,BX      ; AND TAIL POINTER
001B C747041500  98   MOV     [BX].TYPEF,MBX ;INITIALIZE TYPEF
0020 C747060C00  99   MCV     [BX].COUNT,0     ;INITIALIZE COUNT
0025 C74708FFFF 100  MCV     WORD PTR [BX].MHEAD,OFFFHH ;INITIALIZE MHEAD
002A FFED        101  JMP     AX             ;RETURN
                                102
003 +1 SEJECT

```

```

LOC  OBJ          LINE    SOURCE
                                104    ;;;    CREATE$PROCESS (PCB$0, PRI, ENTRO$0, STACK$0) - CREATE PROCESS.
                                105    ;
                                106    ;    PARAMETERS:
                                107    ;    PCB$0 - OFFSET OF A 14 BYTE BLOCK OF MEMORY IN WHICH
                                108    ;    TO CREATE THE PCB OF THE NEW PROCESS.
                                109    ;    PRI - WORD PRIORITY OF THE NEW PROCESS. MUST BE >= 0 AND
                                110    ;    < THE NUMBER OF READY LIST HEADERS IN THE SYSTEM.
                                111    ;    ENTRY$C - OFFSET RELATIVE TO CS OF THE FIRST BYTE
                                112    ;    OF THE PROCESS TO BE EXECUTED.
                                113    ;    STACK$C - OFFSET RELATIVE TO DS OF THE TOP OF STACK
                                114    ;    OF THE NEW PROCESS.
                                115
                                116
                                117    PUBLIC  CREATEPROCESS
002C          002C 58      118    CREATEPROCESS:
002D          002D 5B      119    PCP    AX          ;POP RETURN ADDRESS
002E          002E C747FE0002 120    POP    BX          ;POP NEW STACK POINTER
0033          0033 8C4FFC      121    MOV    [BX-2],200H ;FLAGS
0036          0036 59      122    MOV    [BX-4],CS   ;STORE CS
0037          0037 894FFA      123    POP    CX          ;POP NEW IP
003A          003A 8C5FF8      124    MOV    [BX-6],CX   ;STORE NEW IP
003D          003D 83EBOA      125    MOV    [BX-8],DS   ;SAVE DS
0040          0040 59      126    SUB    BX,10       ;UPDATE NEW STACK POINTER (INC SPACE FOR BP)
0041          0041 5F      127    POP    CX          ;POP "PRI"
0042          0042 C7450417C0 128    POP    DI          ;POP PCB POINTER
0047          0047 895DOA      129    MOV    [DI].TYPEF,PCBT ;INITIALIZE TYPEF FIELD
004A          004A 8C5DOC      130    MOV    [DI].STKOFF,BX ;SET UP STACK POINTER
004D          004D D1E1      131    MOV    [DI].STKSEG,DS
004F          004F 8BD9      132    SHL    CX,1        ;COMPUTE ADDRESS OF RL HEADER (X 6)
0051          0051 D1E1      133    MOV    BX,CX
0053          0053 03D9      134    SHL    CX,1
0055          0055 81C30000      135    ADD    BX,CX
0059          0059 891D      136    ADD    BX,OFFSET DGROUP:READYLIST
005B          005B 895D08      137    MOV    [DI].NEXT,BX ;INITIALIZE NEXT FIELD
005E          005E 895D06      138    MOV    [DI].PRI,BX  ;INITIALIZE PRI AND QUEUE FIELDS
                                139    MOV    [DI].QUEUE,BX
                                140
0061          0061 9C      141    PUSHF             ;TURN OFF INTERRUPTS WHILE UPDATING READY LIST
0062          0062 FA      142    CLI
0063          0063 8B7702      143    MOV    SI,[BX].TAIL ;SI = TAIL
0066          0066 897F02      144    MOV    [BX].TAIL,DI ;NEW TAIL
0069          0069 893C      145    MOV    [SI].NEXT,DI ;LINK OF OLD TAIL
006B          006B 897502      146    MOV    [DI].LAST,SI ;POINTER TO LAST PCB
006E          006E 891D      147    MOV    [DI].NEXT,BX
                                148
0070          0070 3B1E0000      149    CMP    BX,TPRI     ;CHECK IF HIGHER THAN TOP PRIORITY
0074          0074 7F04      150    JG    CRP1        ;IF LOWER PRIORITY (HIGHER NUMBER)
0076          0076 891E0000      151    MOV    TPRI,BX
007A          007A 9D      152    CRP1: PCPF
007B          007B FFEO      153    JMP    AX          ;RETURN TO CALLER
                                154
                                155    +1    $EJECT

```

```
LOC  OBJ          LINE  SOURCE
                                156  ;;;  CREATE_RL - CREATE READY LIST.
                                157  ;
                                158  ;  PARAMETERS:
                                159  ;  NONE:
                                160
                                161
C07D          162  CREATE_RL:
C07D 33C0          163  XOR    AX,AX          ;INITIALIZE STATE VARIABLES
C07F A30000      E  164  MOV    CP,AX
0082 A30000      E  165  MOV    CPRI,AX
C085 A30000      E  166  MOV    TPRI,AX
                                167
0088 B80000      E  168  MOV    BX,OFFSET DGROUP:READYLIST ;INITIALIZE READYLIST
008B B90C00      E  169  MOV    CX,READYLISTSIZE
                                170
008E 891F        171  CRR1:  MOV    [BX].HEAD,BX
0090 895F02      172  MOV    [BX].TAIL,BX
0093 C747041600  173  MOV    [BX].TYPEF,RLH
0098 83C306      174  ADD    BX,6
009B E2F1        175  LOOP  CRR1
009D C3          176  RET
                                177
                                178
                                179  +1  $ EJECT
```

```

LOC 03J          LINE    SOURCE
                180      ;;;  CREATE$LIST(LIST$PTR) - CREATE OBJECTS FROM A LIST.
                181      ;
                182      ;  PARAMETER:
                183      ;      LIST$PTR - 32-BIT POINTER TO A LIST OF OBJECTS TO BE CREATED.
                184
                185
009E 0400        186      CRLA   DW      4          ;NUMBER OF WORDS OF PARAMETERS IN CALL
00A0 0200        187          DW      2
00A2 0100        188          DW      1
00A4 0200        189          DW      2
00A6 0000        190          DW      0
00A8 2C00        191      CRLB   DW      OFFSET CGROUP:CREATEPROCESS ;ROUTINE TO CALL
00AA 0000        192          DW      OFFSET CGROUP:CREATESEMAPHORE
00AC 1400        193          DW      OFFSET CGROUP:CREATEMAILBOX
00AE 0000        194          DW      OFFSET CGROUP:CREATEALARM
00B0 7D00        195          DW      OFFSET CGROUP:CREATE_RL
                196
                197
                198          PUBLIC  CREATELIST
00B2            199      CREATELIST:
00B2 58          200          PCP      AX          ;MOVE PARAMETER TO ES:SI
00B3 5E          201          PCP      SI
00B4 07          202          PCP      ES
00B5 50          203          PUSH     AX
                204
00B6 268A1C      205      CRL1:   MCV      BL,ES:[SI]   ;READ TYPE
00B9 46          206          INC      SI
00BA 80FB00      207          CMP      BL,0          ;CHECK FOR END OF LIST
00BD 7501        208          JNZ      CRL2          ;IF NOT AT END OF LIST
00BF C3          209          RET              ;RETURN IF AT END OF LIST
00C0 B700        210      CRL2:   MCV      BH,0
00C2 2E8B8F9C00  211          MOV      CX,CRLA[BX-2] ;READ PARAMETER COUNT
00C7 E309        212          JCXZ     CRL4
                213
00C9 268B04      214      CRL3:   MOV      AX,ES:[SI]   ;COPY PARAMETERS TO STACK
00CC 83C602      215          ADD      SI,2
00CF 50          216          PUSH     AX
00D0 E2F7        217          LOOP    CRL3
                218
00D2 89360000    219      CRL4:   MCV      WORD PTR PTRSAVE,SI ;SAVE POINTER
00D6 8C060200    220          MOV      WORD PTR PTRSAVE+2,ES
00DA 2EFF97A600  221          CALL   WORD PTR CRLB[BX-2] ;CALL PROCESSING ROUTINE
00DF C4360000    222          LES      SI,PTRSAVE   ;RESTORE POINTER
00E3 EBD1        223          JMP      CRL1          ;LOOP
                224
                225
-----        226      CODE   ENDS
                227
                228 +1  $TITLE ('KAOS - CONDITIONAL OPERATIONS')
                229 +1  $EJECT

```

LOC OBJ

LINE SOURCE

```
230 ; GROUP AND EXTERNAL DEFINITIONS
231
---- 232 CODE SEGMENT BYTE PUBLIC 'CODE'
233 ASSUME CS:CGROUP
234 EXTRN HALTANDCATCHFIRE:NEAR
235
236 +1 $ EJECT
```

```

LOC  OBJ          LINE  SOURCE
      237      ;;;  CRECEIVE (MAILBOX$0) - RECEIVE MESSAGE FROM MAILBOX IF ONE IS
      238      ;    AVAILABLE.
      239      ;
      240      ;    PARAMETER:
      241      ;    MAILBOX$0 - OFFSET OF MAILBOX TO RECEIVE FROM.
      242      ;
      243      ;    RETURNS:
      244      ;    POINTER TO MESSAGE IF ONE IS AVAILABLE.
      245      ;    OFFSET OF OFFFFH, FRAME UNDEFINED IF NO MESSAGE IS AVAILABLE.
      246
      247
      248      PUBLIC  CRECEIVE, ICRECEIVE
00E5  B80500      249      CREC1:  MOV     AX,5           ;ABORT IF NOT A MAILBOX
00E8  50          250      PUSH    AX
00E9  E80000      251      CALL  HALTANDCATCHFIRE
COEC          252      ICRECEIVE:
00EC          253      CRECEIVE:
COEC  59          254      POP     CX           ;POP RETURN ADDRESS
COED  5E          255      POP     SI           ;POP SEMAPHORE POINTER
00EE  837C0415    256      CMP     [SI].TYPEF,MBX ;CHECK IF MAILBOX TYPE
00F2  75F1        257      JNZ    CREC1
00F4  9C          258      PUSHF           ;SAVE FLAGS
00F5  FA          259      CLI           ;RUN WITH INTERRUPTS OFF
00F6  FF4406      260      INC     [SI].COUNT ;TEST COUNT
00F9  7F13        261      JG     CREC2       ;IF NO MESSAGES AVAILABLE
00FB  C45C08      262      LES    BX,[SI].MHEAD ;READ MESSAGE POINTER
00FE  268B3F      263      MCV    DI,ES:[BX]   ;READ LINK TO NEXT
0101  897C08      264      MCV    WORD PTR [SI].MHEAD,DI ;UPDATE MHEAD POINTER
0104  268B7F02    265      MOV    DI,ES:[BX+2]
0108  897C0A      266      MCV    WORD PTR [SI].MHEAD+2,DI
010B  9D          267      PCPF
010C  FFE1        268      JMP     CX           ;RETURN
      269
010E  FF4C06      270      CREC2:  DEC     [SI].COUNT ;RESTORE COUNT
0111  BBFFFF      271      MCV    BX,OFFFH    ;RETURN OFFFH
0114  9D          272      POPF
0115  FFE1        273      JMP     CX
      274
      275 +1  $EJECT

```



```

LOC  OBJ          LINE    SOURCE
                276      ;;;      CWAIT (SEMAPHORE$0) - ACCEPT SIGNAL FROM SEMAPHORE ON THE CONDITION
                277      ;          THAT A SIGNAL IS ALREADY AVAILABLE.
                278      ;
                279      ;      PARAMETER:
                280      ;          SEMAPHORE$0 - OFFSET OF THE SEMAPHORE TO CHECK.
                281      ;
                282      ;      RETURNS:
                283      ;          TRUE - A SIGNAL WAS AVAILABLE AND ACCEPTED.
                284      ;          FALSE - NO SIGNAL WAS AVAILABLE.
                285
                286
                287      PUBLIC  CWAIT,ICWAIT
C117  B30600      288      CWAIT1: MOV     AX,6          ;ABORT IF NOT A SEMAPHORE
C11A  50          289      PUSH    AX
C11B  E80000      290      CALL   HALTANDCATCHFIRE
C11E                    291      ICWAIT:
C11E                    292      CWAIT:
C11E  59          293      POP     CX          ;POP RETURN ADDRESS
C11F  5B          294      POP     BX
O120  837F0414    295      CMP     [BX].TYPEF,SMPH ;CHECK IF A SEMAPHORE
O124  75F1        296      JNZ     CWAIT1
O126  9C          297      PUSHF          ;SAVE INTERRUPT FLAG
C127  FA          298      CLI          ;RUN WITH INTERRUPTS OFF
O128  FF4706      299      INC     [BX].COUNT ;TEST SEMAPHORE
C12B  7F05        300      JG      CWAIT2     ;IF NO SIGNALS AVAILABLE
O12D  80FF        301      MCV    AL,OFFH    ;RETURN TRUE
O12F  9D          302      POPF
C130  FFE1        303      JMP     CX          ;RETURN
                304
O132  FF4F06      305      CWAIT2: DEC     [BX].COUNT ;RESTORE VALUE
C135  33C0        306      XOR     AX,AX      ;RETURN FALSE
C137  9D          307      POPF
C138  FFE1        308      JMP     CX          ;RETURN
                309
----            310      CODE   ENDS
                311
                312      END

```

ASSEMBLY COMPLETE, NO ERRORS FOUND

-submit :f1:gen3

-delete :f1:*.bak,:f1:*.obj,:f1:*.lst

:f1:*.BAK, NO SUCH FILE

:f1:PUCONF.CBJ, DELETED

:f1:DLL.OBJ, DELETED

:f1:DLL.LST, DELETED

:f1:GEN3.LST, CAN'T DELETE OPEN FILE

-run

ISIS-II RUN 8086, V1.1

>:f2:asm86 :f1:puconf.com debug

SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND

>:F2:ASM86 :f1:dll.src debug

SERIES-III 8086/8087/8088 MACRO ASSEMBLER, V1.0

ASSEMBLY COMPLETE, NO ERRORS FOUND

>exit

TO SOURCE DESTROY F1:GEN3 C:\F2\ASM86\GEN3.LST 09/01/80