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


LOC  OBJ                  LINE     SOURCE

                             1 +1  $TITLE ('KAOS - MAIN NUCLEUS')
                             2           NAME    NUCLEUS
                             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               ;Offset of the next PCB in whatever
                            17                                     ;queue this PCB is in.
                            18
0002                        19     LAST    DW      0               ;Offset of the preceding PCB in whatever
                            20                                     ;queue this PCB is in.
                            21
0004                        22     TYPEF   DW      0               ;The type field.  This is the constant 17H,
                            23                                     ;and is used by the processing routines
                            24                                     ;to tell what kind of object this is.
                            25                                     ;(note: it is named TYPEF because TYPE
                            26                                     ;is a reserved word in ASM86.)
                            27
0006                        28     QUEUE   DW      0               ;The offset of the header of whatever queue
                            29                                     ;this PCB is in.
                            30
0008                        31     PRI     DW      0               ;The offset of the Ready List Header for
                            32                                     ;the priority of this process.
                            33
000A                        34     STKOFF  DW      0               ;The Segment and Offset of the top of stack
000C                        35     STKSEG  DW      0               ;when this process is not running.
                            36
----                        37     PCB     ENDS
                            38
                            39
                            40
                            41
                            42     ;       SEMAPHORE AND MAILBOX HEADER DEFINITION
                            43
----                        44     HEADER  STRUC
0000                        45     HEAD    DW      0               ;Offset of the first PCB in a queue of PCBs for
                            46                                     ;processes blocked here.
                            47                                     ;IMPORTANT:  Initialized pointing to itself.
                            48
0002                        49     TAIL    DW      0               ;Offset of the last PCB in the queue of PCBs
                            50                                     ;for processes blocked here.
```

```
LOC  OBJ                  LINE     SOURCE

                            51                          ;IMPORTANT: Initialiezed pointing to itself.
                            52
0004                        53     DUMMY   DW      0    ;The TYPEF field, which contains a 14H for
                            54                          ;a semaphore header and a 15H for a mailbox
                            55                          ;header.  It is labeled DUMMY becuase
                            56                          ;ASM86 won't let you have the same name in
                            57                          ;two structures,  but it is really TYPEF.
                            58
0006                        59     COUNT   DW      0    ;If > 0, it is the number of PCBs for
                            60                      ;blocked processed queued here.
                            61                          ;If < 0, it is the negative of the number
                            62                          ;of messages or signals available here.
                            63                          ;If = 0, there are neither messages nor
                            64                          ;nor PCBs here.
                            65                          ;EXCEPTIONS:  On mailboxes there is a transient
                            66                          ;condition between the time a process is made
                            67                          ;ready to run and the time it begins run.
                            68                          ;In this interval, there is actually one
                            69                          ;more message linked to the header than the
                            70                          ;count indicates.  This is corrected when the
                            71                          ;process runs and removes the message from the
                            72                          ;mailbox.  Also, the CLEARLALARM function
                            73                          ;may remove a message (which happens to be
                            74                          ;an Alarm Control Block) during this interval
                            75                          ;causing the system to be short a message
                            76                          ;when the process starts up.  This is corrected
                            77                          ;by Receive, which reissues the WAITSEM
                            78                          ;function if there is no message at the
                            79                          ;mailbox.
                            80
0008                        81     MHEAD   DD      0    ;PRESENT ON MAILBOXES ONLY.  Pointer to the
                            82                          ;first message linked to this mailbox.
                            83
000C                        84     MTAIL   DD      0    ;PRESENT ON MAILBOXES ONLY.  Pointer to the
                            85                          ;last message linked to this mailbox.
                            86
----                        87     HEADER  ENDS
                            88
                            89
                            90
                            91
                            92     ;       MULTIPLE RECEIVE CONTROL BLOCK DEFINITION
                            93     ;
                            94     ;          Since ASM86 doesn't allow negative offsets on structure
                            95     ;       members,  some of the objects here are defined with EQU's
                            96     ;       rather than as structure members.  For efficiency reasons,
                            97     ;       the pointers in the structure point to the NEXT field rather
                            98     ;       than the beginning.
                            99
-0006                      100     NOBJECTS EQU    -6          ;NUMBER OF OBJECTS IN STRUCTURE
-0004                      101     EVENT   EQU     -4          ;RETURN FOR WHICH EVENT OCCURED
-0002                      102     PCBP    EQU     -2          ;OFFSET OF PCB
                           103     ;NEXT   ---
                           104     ;LAST   ---
                           105     ;TYPEF  ---
```

```
LOC  OBJ                  LINE     SOURCE

                           106     ;QUEUE  ---
  0008                     107     LINKBACK EQU    8               ;LINK BACK TO "NEXT" FIELD OF FIRST EVENT
                           108
                           109
                           110     ;       DEFINITION OF OBJECT TYPES FOR TYPEF FIELD
                           111
  0014                     112     SMPH    EQU     14H             ;SEMAPHORE TYPE
  0015                     113     MBX     EQU     15H             ;MAILBOX TYPE
  0016                     114     RLH     EQU     16H             ;READY LIST HEADER TYPE
  0017                     115     PCBT    EQU     17H             ;PROCESS CONTROL BLOCK TYPE
  0018                     116     WMCB    EQU     18H             ;WAIT MULTIPLE CONTROL BLOCK
                           117
                           118
                           119     ;       GROUP AND EXTERNAL DEFINITIONS
                           120
                           121     DGROUP  GROUP   DATA
----                       122     DATA    SEGMENT byte PUBLIC 'DATA'
                           123
                           124
                           125
                           126     ;;;     THE DATA BASE
                           127
                           128             EXTRN   READYLIST:WORD  ;The ready list headers, one per process.
                           129
                           130             PUBLIC  CP,CPRI,TPRI
0000 (1                    131     CP      DW      1 DUP (?)       ;Current Process -- The offset of the control
     ????
     )
                           132                                     ;block for the process which is currently
                           133                                     ;running.
                           134
0002 (1                    135     CPRI    DW      1 DUP (?)       ;Current Process -- The offset of the ready
     ????
     )
                           136                                     ;list header for the priority of the the
                           137                                     ;currently running process.
                           138
0004 (1                    139     TPRI    DW      1 DUP (?)       ;Top Priority -- The offset of the ready list
     ????
     )
                           140                                     ;header of the highest priority ready process.
                           141                                     ;This is >= CPRI;  equal if CP is the highest
                           142                                     ;priority process, greater is there is a higher
                           143                                     ;priority process which will run when SCHEDULE
                           144                                     ;is called.
                           145
----                       146     DATA    ENDS
                           147
                           148     CGROUP  GROUP   CODE
----                       149     CODE    SEGMENT byte PUBLIC 'CODE'
                           150     ASSUME  CS:CGROUP,DS:DGROUP
                           151             EXTRN   HALTANDCATCHFIRE:NEAR
                           152
                           153 +1  $ EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           154     ;;;     SEND (MAILBOX$O, MESSAGE$P) - SEND MESSAGE TO MAILBOX.
                           155     ;
                           156     ;       PARAMETERS:
                           157     ;          MAILBOX$O - OFFSET OF THE MAILBOX TO SEND TO.
                           158     ;          MESSAGE$P - POINTER TO THE MESSAGE TO BE SENT.
                           159
                           160
                           161             PUBLIC  SEND,ISEND
0000 B80100                162     SEND1:  MOV     AX,1            ;ABORT IF NOT A MAILBOX
0003 50                    163             PUSH    AX
0004 E80000          E     164             CALL    HALTANDCATCHFIRE
0007                       165     ISEND:
0007 58                    166     SEND:   POP     AX              ;POP RETURN ADDRESS
0008 5E                    167             POP     SI              ;POP MESSAGE ADDRESS
0009 07                    168             POP     ES
000A 5B                    169             POP     BX              ;POP MAILBOX ADDRESS'
000B 837F0415              170             CMP     [BX].TYPEF,MBX  ;TEST IF MAILBOX TYPE
000F 75EF                  171             JNZ     SEND1
0011 9C                    172             PUSHF                   ;SAVE INTERRPUT FLAG
0012 FA                    173             CLI                     ;DISABLE INTERRPUTS
0013 26C704FFFF            174             MOV     ES:[SI],OFFFFH  ;LINK AT TAIL OF LIST
0018 8CC1                  175             MOV     CX,ES           ;PUT MESSAGE BASE IN CX
001A 817F08FFFF            176             CMP     WORD PTR [BX].MHEAD,OFFFFH
001F 7508                  177             JNZ     SEND2           ;IF ITEMS ALREADY IN LIST
0021 897708                178             MOV     WORD PTR [BX].MHEAD,SI  ;MOVE APPROPRIATE POINTER TO MHEAD
0024 894F0A                179             MOV     WORD PTR [BX].MHEAD+2,CX
0027 EB0A                  180             JMP     SHORT SEND3     ;ELSE,
0029 C47F0C                181     SEND2:  LES     DI,[BX].MTAIL   ;LINK TO END OF LIST
002C 268935                182             MOV     ES:[DI],SI
002F 26894D02              183             MOV     ES:[DI+2],CX
0033 897708                184     SEND3:  MOV     WORD PTR [BX].MTAIL,SI  ;SET UP MTAIL
0036 894F0E                185             MOV     WORD PTR [BX].MTAIL+2,CX
                           186
0039 FF4F06                187             DEC     [BX].COUNT      ;CHECK SEMAPHORE
003C 7D1C                  188             JGE     SIGS            ;SIGNAL SEMAPHORE IF NEEDED
003E 9D                    189             POPF                    ;RESTORE INTERRUPT FLAG
003F FFE0                  190             JMP     AX              ;RETURN
                           191
                           192 +1  $ EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           193     ;;;     SIGNAL (SEMAPHORE$O) - SIGNAL SEMAPHORE.
                           194     ;
                           195     ;       PARAMETER:
                           196     ;          SEMAPHORE$O - OFFSET OF THE SEMAPHORE HEADER TO SIGNAL
                           197
                           198
                           199             PUBLIC  SIGNAL,ISIGNAL
0041 B80200                200     SIG1:   MOV     AX,2            ;ABORT IF NOT SEMAPHORE
0044 50                    201             PUSH    AX
0045 E80000         E      202             CALL    HALTANDCATCHFIRE
0048                       203     ISIGNAL:
0048 58                    204     SIGNAL: POP     AX              ;POP RETURN ADDRESS
0049 5B                    205             POP     BX              ;POP SEMAPHORE TOKEN
004A 837F0414              206             CMP     [BX].TYPEF,SMPH ;TEST FOR SEMAPHORE TYPE
004E 75F1                  207             JNZ     SIG1
0050 9C                    208             PUSHF                   ;RUN WITH INTERRUPTS OFF
0051 FA                    209             CLI
                           210
0052 FF4F06                211             DEC     [BX].COUNT      ;REDUCE NUMBER WAITING
0055 7D03                  212             JGE     SIGS            ;IF SOMEONE WAITING
0057 9D                    213             POPF                    ;RETURN TO CALLER
0058 FFE0                  214             JMP     AX
                           215
                           216     ;  REMOVE PROCESS FROM SEMAPHORE QUEUE AND LINK INTO READY LIST
                           217
005A 8B37                  218     SIGS:   MOV     SI,[BX].HEAD    ;UNLINK FROM SEMAPHORE QUEUE
005C 837C0417              219             CMP     [SI].TYPEF,PCBT ;CHECK IF REMOVED A PCB
0060 7429                  220             JZ      SIG4            ;IF A PCB, THEN GO TO PCB PROCESSING
                           221
                           222     ;       PROCESS MULTIPLE RECEIVE
                           223
0062 FF4706                224             INC     [BX].COUNT      ;INC COUNT BECAUSE I WILL DEC AGAIN LATER
0065 8B7C08                225             MOV     DI,[SI+LINKBACK] ;FIND HEAD OF WAIT LIST
0068 895DFC                226             MOV     [DI+EVENT],BX   ;SAVE OBJECT POINTER
006B 8B4DFA                227             MOV     CX,[DI+NOBJECTS] ;CX = COUNT OF OBJECTS BEING WAITED UPON
                           228
006E 8B1D                  229     SIG3:   MOV     BX,[DI].NEXT    ;BX -> NEXT ITEM
0070 8B7502                230             MOV     SI,[DI].LAST    ;SI -> LAST ITEM
0073 891C                  231             MOV     [SI].NEXT,BX    ;CLOSE LINKS
0075 897702                232             MOV     [BX].LAST,SI
0078 8B5D06                233             MOV     BX,[DI].QUEUE   ;DECREMENT NUMBER WAITING
007B FF4F06                234             DEC     [BX].COUNT
007E 83C70A                235             ADD     DI,10           ;ADVANCE TO NEXT OBJECT
0081 E2EB                  236             LOOP    SIG3            ;LOOP IF STILL OTHER OBJECTS
                           237
0083 8B7DFE                238             MOV     DI,[DI+LINKBACK-10] ;FIND TOP OF LIST
0086 8B75FE                239             MOV     SI,[DI+PCBP]    ;PUT PCB POINTER IN SI FOR REST OF PROCESSING
0089 EB07                  240             JMP     SHORT SIG5
                           241
                           242     ;       UNLINK PCB IF A PCB
                           243
008B 8B3C                  244     SIG4:   MOV     DI,[SI].NEXT
008D 893F                  245             MOV     [BX].HEAD,DI
008F 895D02                246             MOV     [DI].LAST,BX
                           247
```

```
LOC  OBJ                  LINE     SOURCE

                           248     ;       LINK INTO READY LIST
                           249
0092 8B5C08                250     SIG5:   MOV     BX,[SI].PRI     ;BX = PRI
0095 8B7F02                251             MOV     DI,[BX].TAIL    ;DI = TAIL
0098 897702                252             MOV     [BX].TAIL,SI    ;NEW TAIL
009B 8935                  253             MOV     [DI].NEXT,SI    ;LINK OF OLD TAIL
009D 891C                  254             MOV     [SI].NEXT,BX    ;FIELDS OF NEW TAIL
009F 897C02                255             MOV     [SI].LAST,DI
00A2 895C06                256             MOV     [SI].QUEUE,BX
                           257
                           258     ;  CHECK PRIORITY
                           259
00A5 3B1E040C        R     260             CMP     BX,TPRI
00A9 7D04                  261             JGE     SIG6            ;IF LOWER PRIORITY (HIGHER VALUE)
00AB 891E040C        R     262             MOV     TPRI,BX         ;NEW HIGHEST PRIORITY
00AF 9D                    263     SIG6:   POPF                    ;RESTORE FLAGS
00B0 FFE0                  264             JMP     AX              ;RETURN TO CALLER
                           265
                           266 +1  $EJECT
```

```
LOC  OBJ                  LINE     SCURCE

                           267     ;;;     MRECEIVE (BLOCK$O) - MULTIPLE RECEIVE.
                           268     ;
                           269     ;       PARAMETER:
                           270     ;          BLCCK$C - OFFSET OF A MULTIPLE RECEIVE CONTROL BLOCK (MRCB).
                           271     ;
                           272     ;       RETURNS:
                           273     ;          POINTER TO MESSAGE IF EVENT OCCURRED AT MAILBOX.
                           274     ;          UNDEFINED IF EVENT OCCURRED AT SEMAPHORE.
                           275     ;          MRCB.EVENT = OFFSET OF OBJECT AT WHICH EVENT OCCURRED.
                           276
                           277
                           278             PUBLIC  MRECEIVE
00B2                       279     MRECEIVE:
00B2 58                    280             PCP     AX              ;POP RETURN ADDRESS
00B3 5B                    281             PCP     BX              ;POP CONTROL BLOCK POINTER
00B4 50                    282             PUSH    AX              ;RESTORE RETURN ADDRESS
00B5 9C                    283             PUSHF                   ;SAVE FLAGS (ESP. INTERRUPT)
                           284
00B6 83C306                285             ADD     BX,6            ;POINT BX TO FIRST WAIT BLOCK
00B9 8B4FFA                286             MOV     CX,[BX+NOBJECTS] ;CX GETS REPEAT COUNT
00BC 8BD3                  287             MOV     DX,BX           ;SAVE BX
                           288
00BE C747041800            289     MRX1:   MOV     [BX].TYPEF,WMCB ;INITIALIZE CONSTANT PARTS OF STRUCTURE
00C3 895708                290             MOV     [BX+LINKBACK],DX
00C6 83C30A                291             ADD     BX,10
00C9 E2F3                  292             LOOP    MRX1
                           293
                           294     ;       LOOK IF AN EVENT ALREADY AVAILABLE
                           295
00CB 8BDA                  296     MRX2:   MOV     BX,DX           ;REINITIALIZE FOR LOOP UNDER INTERRUPT DISABLE
00CD 8B4FFA                297             MOV     CX,[BX+NOBJECTS]
00D0 FA                    298             CLI                     ;RUN WITH INTERRUPTS OFF
00D1 8B7706                299     MRX3:   MOV     SI,[BX].QUEUE   ;CHECK IF SIGNAL AVAILABLE AT QUEUE
00D4 837C060C              300             CMP     [SI].COUNT,0
00D8 7C5F                  301             JL      MRX7            ;IF EVENT FOUND
00DA 83C30A                302             ADD     BX,10
00DD E2F2                  303             LOOP    MRX3
                           304
                           305     ;       MUST WAIT -- LINK MRCB INTO SPECIFIED SEMAPHORE AND MAILBOX QUEUES
                           306
00DF 8BDA                  307             MOV     BX,DX           ;REINITIALIZE LOOP - WE KNOW WE MUST WAIT
00E1 8B4FFA                308             MOV     CX,[BX+NOBJECTS]
00E4 8B7706                309     MRX4:   MOV     SI,[BX].QUEUE   ;ENTER WAIT BLOCK INTO EACH QUEUE
00E7 FF4406                310             INC     [SI].COUNT      ;MARK ONE MORE WAITING
00EA 8B7C02                311             MOV     DI,[SI].TAIL    ;LINK WAIT BLOCK INTO QUEUE
00ED 895C02                312             MOV     [SI].TAIL,BX
00F0 891D                  313             MOV     [DI].NEXT,BX
00F2 8937                  314             MOV     [BX].NEXT,SI
00F4 897F02                315             MOV     [BX].LAST,DI
00F7 83C30A                316             ADD     BX,10
00FA E2E8                  317             LOOP    MRX4
                           318
                           319     ;       MOVE CURRENT PROCESS TO MRCB AND DISABLE
                           320
00FC 8B3E000C        R     321             MOV     DI,CP           ;PUT PCB POINTER IN MRCB
```

```
LOC  OBJ                  LINE     SOURCE

C100 8BDA                  322              MOV     BX,DX
C102 897FFE                323              MOV     [BX+PCBP],DI
                           324
C105 8B35                  325              MOV     SI,[DI].NEXT    ;UNLINK PCB
C107 8B5D02                326              MOV     BX,[DI].LAST    ;CLOSE LINKS
C10A 895C02                327              MOV     [SI].LAST,BX
C10D 8937                  328              MOV     [BX].NEXT,SI
                           329
C10F 52                    330              PUSH    DX              ;CALL SCHEDULER
C110 9C                    331              PUSHF
C111 0E                    332              PUSH    CS
C112 E8A000                333              CALL    SCHED
                           334
                           335      ;       PROCESS RETURNED EVENT INDICATION
                           336
C115 5B                    337              POP     BX              ;PUT WMCB POINTER INTO BX
C116 8BD3                  338              MOV     DX,BX           ;COPY IN CASE WE HAVE TO REPEAT WAIT
C118 8B77FC                339              MOV     SI,[BX+EVENT]
C11B 837C0415              340      MRX5:   CMP     [SI].TYPEF,MBX  ;CHECK TYPE
C11F 7516                  341              JNZ     MRX6            ;IF NOT A MAILBOX
C121 C45C08                342              LES     BX,[SI].MHEAD
C124 81FBFFFF              343              CMP     BX,0FFFFH       ;MAKE SURE A MESSAGE IS HERE (FOR CLEARALARM)
C128 74A1                  344              JZ      MRX2            ;REPEAT WAIT IF NO MESSAGE
C12A 268B3F                345              MOV     DI,ES:[BX]
C12D 897C08                346              MOV     WORD PTR [SI].MHEAD,DI
C130 268B7F02              347              MOV     DI,ES:[BX+2]
C134 897C0A                348              MOV     WORD PTR [SI].MHEAD+2,DI
C137 9D                    349      MRX6:   POPF                    ;RESTORE FLAGS
C138 C3                    350              RET
                           351
                           352      ;       EVENT WAS FOUND WHILE SEARCHING
                           353
C139 FF4406                354      MRX7:   INC     [SI].COUNT      ;INCREMENT COUNT
C13C 8BDA                  355              MOV     BX,DX           ;RESET BX
C13E 8977FC                356              MOV     [BX+EVENT],SI   ;MARK WHICH EVENT HAPPENED
C141 EBD8                  357              JMP     MRX5
                           358
                           359 +1   $ EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           360     ;;;     RECEIVE (MAILBOX$O) - RECEIVE MESSAGE FROM MAILBOX.
                           361     ;
                           362     ;       PARAMETER:
                           363     ;          MAILBOX$O - OFFSET OF MAILBOX HEADER.
                           364     ;
                           365     ;       RETURNS:
                           366     ;          POINTER TO MESSAGE.
                           367
                           368
                           369             PUBLIC  RECEIVE
0143 B80300                370     REC1:   MOV     AX,3            ;ABORT IF NOT A MAILBOX
0146 50                    371             PUSH    AX
0147 E80000          E     372             CALL    HALTANDCATCHFIRE
014A                       373     RECEIVE:
014A 58                    374             POP     AX              ;GET MAILBOX ADDRESS OFF STACK
014B 5E                    375             POP     SI
014C 837C0415              376             CMP     [SI].TYPEF,MBX  ;CHECK FOR MAILBOX TYPE
0150 75F1                  377             JNZ     REC1
0152 50                    378             PUSH    AX              ;RESTORE RETURN ADDRESS
0153 9C                    379             PUSHF                   ;SAVE INTERRUPT FLAG
0154 FA                    380             CLI                     ;DISABLE INTERRUPTS
                           381
0155 FF4406                382     REC2:   INC     [SI].COUNT      ;CHECK SEMAPHORE
0158 7E07                  383             JLE     REC3            ;IF A MESSAGE AVAILABLE
                           384
015A 56                    385             PUSH    SI              ;SAVE MAILBOX ADDRESS
015B 9C                    386             PUSHF                   ;FORMAT STACK FOR WAITING ON SEMAPHORE
015C 0E                    387             PUSH    CS
015D E83400                388             CALL    WAITR           ;EXECUTE WAIT CODE
0160 5E                    389             POP     SI              ;RESTORE MAILBOX ADDRESS
                           390
0161 C45C08                391     REC3:   LES     BX,[SI].MHEAD   ;UNLINK MESSAGE FROM QUEUE
0164 81FBFFFF              392             CMP     BX,0FFFFH       ;MAKE SURE A MESSAGE IS HERE (FOR CLEARALARM)
0168 74EB                  393             JZ      REC2            ;REPEAT RECEIVE IF NO MESSAGE
016A 268B3F                394             MOV     DI,ES:[BX]
016D 897C08                395             MOV     WORD PTR [SI].MHEAD,DI
0170 268B7F02              396             MOV     DI,ES:[BX+2]
0174 897C0A                397             MOV     WORD PTR [SI].MHEAD+2,DI
0177 9D                    398             POPF                    ;RESTORE INTERRUPT FLAG
0178 C3                    399             RET                     ;RETURN
                           400
                           401 +1  $ EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          402      ;;;     WAITSEM (SEMAPHORE$O) - WAIT FOR EVENT AT SEMAPHORE.
                          403      ;
                          404      ;       PARAMETER:
                          405      ;          SEMAPHORE$O - OFFSET OF SEMAPHORE HEADER TO WAIT AT.
                          406
                          407
                          408              PUBLIC  WAITSEM
0179 B80400               409      WAIT1:  MOV     AX,4            ;ABORT IF NOT A SEMAPHORE
017C 50                   410              PUSH    AX
017D E80000         E     411              CALL    HALTANDCATCHFIRE
0180                      412      WAITSEM:
0180 58                   413              POP     AX              ;POP RETURN ADDRESS
0181 5E                   414              POP     SI              ;POP SEMAPHORE TOKEN
0182 837C0414             415              CMP     [SI].TYPEF,SMPH ;CHECK FOR SEMAPHORE TYPE
0186 75F1                 416              JNZ     WAIT1
0188 9C                   417              PUSHF                   ;RUN WITH INTERRUPTS OFF
0189 FA                   418              CLI
                          419
018A FF4406               420              INC     [SI].COUNT      ;REDUCE NUMBER WAITING
018D 7F03                 421              JG      WAIT2           ;IF NO SIGNAL AVAILABLE
018F 9D                   422              POPF                    ;RETURN IF SIGNAL AVAILABLE
0190 FFE0                 423              JMP     AX
                          424
                          425      ;  REMOVE PROCESS FROM READY LIST AND LINK INTO SEMAPHORE LIST
                          426
0192 0E                   427      WAIT2:  PUSH    CS              ;SET UP PROPER STACK FORMAT
0193 50                   428              PUSH    AX
0194 8BC6                 429      WAITR:  MOV     AX,SI           ;MUST USE SI REG, SO SAVE SEMAPHORE PTR IN AX
0196 8B3E020C       R     430              MOV     DI,CPRI         ;UNLINK ;DI = RL HEAD
019A 8B35                 431              MOV     SI,[DI].HEAD    ;SI = PROC
019C 8B1C                 432              MOV     BX,[SI].NEXT    ;CLOSE LINKS
019E 891D                 433              MOV     [DI].HEAD,BX
01A0 897F02               434              MOV     [BX].LAST,DI
01A3 8BD8                 435              MOV     BX,AX           ;RESTORE SEMAPHORE POINTER TO BX
                          436
01A5 8B7F02               437              MOV     DI,[BX].TAIL    ;LINK ;DI = TAIL OF SEMAPHORE LIST
01A8 897702               438              MOV     [BX].TAIL,SI    ;NEW TAIL
01AB 8935                 439              MOV     [DI].NEXT,SI    ;LINK OF OLD TAIL
01AD 891C                 440              MOV     [SI].NEXT,BX    ;FILL IN NEW TAIL
01AF 897C02               441              MOV     [SI].LAST,DI
01B2 895C06               442              MOV     [SI].QUEUE,BX
                          443      ;;;;    JMP     SCHED ---       ;JUMP TO SCHEDULER (NEXT ROUTINE)
                          444
                          445 +1   $ EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           446     ;;      SCHED - INTERNAL SCHEDULER.
                           447     ;
                           448     ;          SCHED SEARCHES THE READY LIST HEADER LIST TO
                           449     ;       FIND THE HIGHEST PRIORITY READY PROCESS.  IT STARTS
                           450     ;       SEARCHING AT TPRI TO REDUCE SEARCH TIME IF A HIGH
                           451     ;       PRIORITY PROCESS HAS JUST BEEN MADE READY.
                           452
                           453
01B5 8B1E0400        R     454     SCHED:  MOV     BX,TPRI
01B9 EB03                  455             JMP     SHORT SCHED2
01BB 83C306                456     SCHED1: ADD     BX,6
01BE 391F                  457     SCHED2: CMP     [BX].HEAD,BX
01C0 74F9                  458             JE      SCHED1
01C2 8B37                  459             MOV     SI,[BX].HEAD
                           460
01C4 1E                    461             PUSH    DS              ;SAVE SMALL STATE
01C5 55                    462             PUSH    BP
01C6 8B3E0000        R     463             MOV     DI,CP           ;SAVE STACK POINTER IN PCB
01CA 8C550C                464             MOV     [DI].STKSEG,SS
01CD 89650A                465             MOV     [DI].STKOFF,SP
                           466
01D0 89360000        R     467     LOAD:   MOV     CP,SI           ;SET UP NEW STATE
01D4 891E0400        R     468             MOV     TPRI,BX
01D8 891E0200        R     469             MOV     CPRI,BX
01DC 8E540C                470     LOAD1:  MOV     SS,[SI].STKSEG
01DF 8B640A                471             MOV     SP,[SI].STKOFF
01E2 5D                    472     LOAD2:  POP     BP
01E3 1F                    473             POP     DS
01E4 CF                    474             IRET
                           475
                           476 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          477      ;;;     START - START KAOS NUCLEUS.
                          478      ;
                          479      ;       PARAMETERS:
                          480      ;          NONE.
                          481      ;
                          482      ;       DOES NOT RETURN.  EVER.
                          483
                          484
                          485              EXTRN   TIMERINIT:NEAR
                          486              PUBLIC  START
01E5 E80000         E     487      START:  CALL    TIMERINIT       ;INITIALIZE TIMERS
01E8 BB0000         E     488              MOV     BX,OFFSET DGROUP:READYLIST  ;START FIRST PROCESS
01EB EB03                 489              JMP     SHORT START2
01ED 83C306               490      START1: ADD     BX,6
01F0 391F                 491      START2: CMP     [BX].HEAD,BX
01F2 74F9                 492              JE      START1
01F4 8B37                 493              MOV     SI,[BX].HEAD
01F6 EBD8                 494              JMP     SHORT LOAD
                          495
                          496
                          497
                          498
                          499      ;;;     SCHEDULE - START HIGHER PRIORITY PROCESS IF ONE READY.
                          500      ;
                          501      ;       PARAMETERS: NONE.
                          502      ;
                          503      ;       RETURNS: NONE.
                          504
                          505
                          506              PUBLIC  SCHEDULE
01F8                      507      SCHEDULE:
01F8 58                   508              POP     AX              ;REFORMAT STACK FOR IRET
01F9 9C                   509              PUSHF
01FA 0E                   510              PUSH    CS
01FB 50                   511              PUSH    AX
01FC FA                   512              CLI                     ;NO INTERRUPTS FOR INDIVISIBILITY
01FD 8B1E0400       R     513              MOV     BX,TPRI         ;TEST PRIORITY
0201 3B1E0200       R     514              CMP     BX,CPRI
0205 7CB7                 515              JL      SCHED2          ;IF HIGHER PRIORITY THAN CURRENT
0207 CF                   516              IRET
                          517
                          518
----                      519      CODE    ENDS
                          520              END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```