```
ASM80 :f1:intask.m80


ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         INTASK   PAGE    1
MIP-ISIS INTASK

  LOC  OBJ          LINE        SOURCE STATEMENT

                       1 $TITLE ('MIP-ISIS INTASK')
                       2 ;ISIS$Intask:
                       3 ;do;
                       4 ;
                       5 ; THIS ROUTINE REMOVES COMMANDS AND RESPONSES FROM THE
                       6 ; REQUEST QUEUE TO THIS DEVICE. IT IS CALLED BY MIPSND
                       7 ; AND MIPRCV WHEN THEY ARE LOOKING FOR INPUT.
                       8 ;
                       9
                      10          NAME    INTASK
                      11          PUBLIC  INTASK
                      12          EXTRN   OUTRQD,INRQD,WREPLY,PTOMB
                      13          EXTRN   RQTPTR,RQGPTR,RLTPTR,RLGPTR,TRQGPT
                      14
                      15 $INCLUDE(:F1:MIP.EQU)
                    =  16 ;
                    =  17 ; DEFINE RQD RESULTS
                    =  18 ;
  0001              =  19 GERROR  EQU     1H
  0004              =  20 GBUSY   EQU     4H
  0008              =  21 FIRSTG  EQU     8H
  0010              =  22 GDISAB  EQU     10H
  0020              =  23 GFULL   EQU     20H
  0040              =  24 DISABT  EQU     40H
  0080              =  25 FULLF   EQU     80H
                    =  26
  0001              =  27 TERROR  EQU     1H
  0004              =  28 TBUSY   EQU     4H
  0008              =  29 FIRSTT  EQU     8H
  0010              =  30 TDISAB  EQU     10H
  0020              =  31 TEMPTY  EQU     20H
  0040              =  32 DISABG  EQU     40H
  0080              =  33 EMPTYF  EQU     80H
                    =  34 ;
                    =  35 ; DEFINE MIP CMDS AND RESPONSES
                    =  36 ;
  0070              =  37 CSEND   EQU     70H
  0080              =  38 SENTOK  EQU     80H
  0081              =  39 UNKNP   EQU     81H
  0083              =  40 ACTIVP  EQU     83H
  0085              =  41 INSUFM  EQU     85H
  0087              =  42 INACTP  EQU     87H
  0089              =  43 DEADP   EQU     89H
                    =  44 ;
                    =  45 ; DEFINE MIP-ISIS PARAMETERS
                    =  46 ;
  0000              =  47 MYIDS   EQU     0
  0003              =  48 THIDEV  EQU     3
                       49          CSEG
                       50 ;declare Out$RQD word external,
                       51 ;        In$RQD word  external,
                       52 ;        W$reply byte external,
                       53
```

```
 LOC  OBJ          LINE        SOURCE STATEMENT

                     55
                     56 ;Gen$int:
                     57 ;  procedure external;
                     58 ;end Gen$int;
                     59
                     60
                     61 ;/*************************************************************/
                     62
                     63 ;In$task:
                     64 ;  procedure  public;
                     65
                     66 ;  declare Msgptr word,
                     67 ;          Msg based Msgptr structure(Mip$msg$format),
                     68 ;          G$RQEptr word,
                     69 ;          G$RQEntry based G$RQEptr structure(Rqentry$format),
                     70 ;          T$RQEptr word,
                     71 ;          T$RQEntry based T$RQEptr structure(Rqentry$format),
                     72 ;          Give$state byte,
                     73 ;          Take$state byte,
                     74 ;          Port byte,
                     75 ;          Cmd byte,
                     76 ;          Given boolean,
                     77 ;          Listbase word,
                     78 ;          Listlink based Listbase word,
                     79 ;          T$result byte;
                     80 INTASK:
                     81 ;       dc while not (Take$state:=Rqt$ptr(.In$RQD));
0000 CD0000    E     82 @9:      CALL    RQTPTR
0003 1F              83          RAR
0004 D8              84          RC
                     85 ;         /*
                     86 ;           there is something in the queue, take it
                     87 ;         */
                     88 ;         if (T$RQEntry.Request$id = CSEND then
0005 7E              89          MOV     A,M             ; GET REQUESTID
0006 FE70            90          CPI     CSEND
0008 C25D00    C     91          JNZ     @7
                     92 ;           do;  /* CSEND */
                     93 ;             /*
                     94 ;               see if socket is open and this device can get to the
                     95 ;               buffer
                     96 ;             */
                     97 ;               Port = T$RQEntry.Dest$port$id;
000B 23              98          INX     H       ; INDEX TO REQUEST ID
000C 56              99          MOV     D,M     ; SAVE SRC REQ ID
000D 23             100          INX     H       ; INDEX TO DEST DEVICE
000E 23             101          INX     H       ; AND SKIP TO PORT
000F 7E             102          MOV     A,M     ; LOAD PORT
0010 4F             103          MOV     C,A     ; SAVE PORT IN C
0011 23             104          INX     H       ; INDEX TO SRC DEV AND SAVE IT
0012 5E             105          MOV     E,M
0013 D5             106          PUSH    D       ; SAVE SRC REQUEST ID AND SRC DEVICE
                    107 ;             if  Port < Max$no$ports then
0014 FE07           108          CPI     7H
```

```
  LOC  OBJ          LINE        SOURCE STATEMENT

                     110 ;                /*
                     111 ;                  the socket exists
                     112 ;                */
                     113 ;                Msgptr = TSRQEntry.bufbaseadr(0);
  0019 0680          114          MVI     B,SENTOK          ; SET UP RESPONSE
  001B C5            115          PUSH    B                 ; SAVE TRESULT/PORT
  001C 23            116          INX     H                 ; get past srcDEV
                     117
  001D 4E            118          MOV     C,M               ; LOAD MSGPTR
  001E 23            119          INX     H
  001F 46            120          MOV     B,M               ; MSGPTR IN B/C
  0020 23            121          INX     H                 ; to bufbaseadr(1)
                     122 ;                /*
                     123 ;                  send it off to the user
                     124 ;                */
                     125 ;                Msg.link(0) = 0;
  0021 EB            126          XCHG
  0022 D1            127          POP     D                 ; GET PORT
  0023 D5            128          PUSH    D
                     129
  0024 AF            130          XRA     A
  0025 02            131          STAX    B
  0026 03            132          INX     B                 ; FILL LINK FIELD WITH ZEROES
  0027 02            133          STAX    B                 ; TO MAKE IT THE LAST ON THE PORT
                     134 ;                Listbase = .Port$to$mailbox(Port);
  0028 1600          135          MVI     D,0
  002A 210000    E   136          LXI     H,PTOMB
  002D 19            137          DAD     C
  002E 19            138          DAD     D
                     139
                     140 ;                do while (Listlink <> 0);
  002F 7E            141 @11:     MOV     A,M
  0030 23            142          INX     H                 ; SEE IF PTS TO ZERO (EOL)
  0031 B6            143          ORA     M
  0032 CA3C0C    C   144          JZ      @12
                     145 ;                  Listbase = Listlink;
  0035 56            146          MOV     D,M
  0036 2B            147          DCX     H
  0037 5E            148          MOV     E,M
  0038 EB            149          XCHG
                     150 ;                end;
  0039 C32F0C    C   151          JMP     @11
                     152 ;                Listlink = Msgptr;
  003C 0B            153 @12:     DCX     B       ; GET BACK TO MSGPTR
  003D 70            154          MOV     M,B
  003E 2B            155          DCX     H
  003F 71            156          MOV     M,C
                     157 ;              end;
  0040 C3460C    C   158          JMP     @13
                     159 ;           else TSresult = UNKNP
  0043 0681          160 @3:      MVI     B,UNKNP
  0045 C5            161 @6:      PUSH    B                 ; TRESULT SOULD BE ON TOS
                     162 ;          end;
                     163 ;          Given = FALSE;
```

```
  LOC  OBJ          LINE        SOURCE STATEMENT

                     165 ;            /*
                     166 ;              now send the cmd or ack/nak to the device
                     167
                     168 ;              see if we can put the response into the RQ
                     169 ;            */
                     170 ;            if (Rqg$ptr(.Out$RQD) and GERROR) = 0 then
  0046 CD0000    E   171 @13:    CALL    TRQGPT
  0049 1F            172         RAR
  004A 3E89          173         MVI     A,DEADP         ; IF HAS ERROR THEN PROBLEM
  004C C1            174         POP     B               ; GET TRESULT AND PORT
  004D D1            175         POP     D               ; GET REQID AND SRC DEVICE
  004E DA5D00    C   176         JC      @7
                     177 ;            do;
                     178 ;              /*
                     179 ;                there is space in the RQ
                     180 ;              */
                     181 ;              G$RQEntry.Request$id = T$result;
  0051 70            182         MOV     M,B             ; PUT TRESULT AWAY
                     183 ;              G$RQEntry.SRC$REQUEST$ID  = T$RQEntry.Src$Request$id;
  0052 23            184         INX     H
  0053 72            185         MOV     M,D
                     186 ;              G$RQEntry.DestDev$id = T$RQEntry.Src$Dev$id;
                     187
  0054 23            188         INX     H
  0055 73            189         MOV     M,E
  0056 13            190         INX     D
                     191 ;              Give$state = Rlg$ptr(.Out$RQD);
  0057 CD0000    E   192         CALL    RLGPTR
                     193 ;              Given = TRUE;
  005A C36000    C   194         JMP     @8
                     195 ;            end;
                     196 ;          end; /* given loop */
                     197 ;         end;  /* cmd processing */
                     198 ;         else
                     199 ;         do;
                     200 ;           /*
                     201 ;             the received item is a response, so do what needs to
                     202 ;             be done.
                     203 ;           */
                     204 ;           W$reply = T$RQEntry.Request$id;
  005D 320000    E   205 @7:     STA     WREPLY
                     206 ;         end; /*response processing*/
                     207 ;         /*
                     208 ;           we have completed processing on this taken item, release
                     209 ;           the RQ. Signal the device.
                     210 ;         */
                     211 ;         Take$state = Rlt$ptr(.In$RQD);
  0060 CD0000    E   212 @8:     CALL    RLTPTR
                     213 ;       end;  /* while loop */
  0063 C30000    C   214         JMP     @9
                     215 ;       */
                     216 ;end IN$task;
                     217
                     218 ;end ISIS$In$task;
```

```
  LOC  OBJ          LINE        SOURCE STATEMENT


PUBLIC SYMBOLS
INTASK C 0000

EXTERNAL SYMBOLS
INRQD  E 000C    OUTRQD E 00C0    PTCMB  E 0C00    RLGPTR E 0000    RLTPTR E 0000    RQGPTR E 00C0    RQTPTR E 0C00
TRQGPT E 0000    WREPLY E 0000

USER SYMBOLS
@11    C 002F    @12    C 003C    @13    C 0046    @3     C 0043    @6     C 0045    @7     C 005D    @8     C 0060
@9     C 000C    ACTIVP A 0083    CSEND  A 0C70    DEADP  A 0089    DISABG A 0040    DISABT A 0040    EMPTYF A 0080
FIRSTG A 0008    FIRSTT A 0008    FULLF  A 0080    GBUSY  A 0004    GDISAB A 0010    GERROR A 0001    GFULL  A 0C20
INACTP A 0087    INRQD  E 00C0    INSUFM A 0085    INTASK C 0000    MYIDS  A 0000    OUTRQD E 0000    PTCMB  E 0000
RLGPTR E 000C    RLTPTR E 000C    RQGPTR E 0C00    RQTPTR E 0000    SENTOK A 0080    TBUSY  A 0004    TDISAB A 0010
TEMPTY A 002C    TERROR A 0001    THIDEV A 0C03    TRQGPT E 0000    UNKNP  A 0081    WREPLY E 0000

ASSEMBLY COMPLETE,   NO ERRORS
```