```
TITLE BIOS - I/O ROUTINES
NAME BIOS
;DC Pankratz (All Rights reserved)

COMMENT !
        REMARKS:
                - Active page BH is always set to 0.

                - All routines are written assuming SCREEN TYPE
                  is type 3 = 25x80 color text. Set this by calling VIDEOTYPE.

                - To use this module, you must define your code segment
                  as  CSEG   SEGMENT  PUBLIC

                  and declare as EXTRN NEAR all procedures you want from
                  this library within CSEG
!
;*****************************************************************************
;                       V I D E O   I N T   1 0
;*****************************************************************************
CSEG    SEGMENT  BYTE  PUBLIC
        Assume   CS:CSEG

        PUBLIC  LOCATE, WHERE, WRITEATTR, WRITE, READATTR
        PUBLIC  SCROLLATTR, CLRSCR, CLRSCRATTR
        PUBLIC  CURSORON, CURSOROFF  ; 2001
        PUBLIC  GETCURSOR, SETCURSOR ; 2011

Comment!
                - The ATTRIBUTE byte for color settings is defined as:
                  BIT PATTERN  xbbb ifff
                  where  ifff = foreground (i = intensity)
                         xbbb = background (x = intensity in Win mode)
                                           (x = blinking in DOS mode)
                    000 = black
                    001 = blue
                    010 = green
                    100 = red
                    011 = green/blue (cyan)
                    101 = red/blue (magenta)
                    110 = red/green (brown)
                    111 = white

                - Active page BH is always set to 0.
        !

;*****************************************************************************
LOCATE          PROC
;        Positions the cursor on the screen
;        IN:   DH    row number (0-25 decimal)
;              DL    col number (0-79 decimal)
;        OUT:  none

        push    bx
        push    ax
        mov     bh,0    ; page 0
        mov     ah,2    ; service
        int     10h
        pop     ax
        pop     bx
        ret
LOCATE          ENDP

;*****************************************************************************
SETCURSOR       PROC
; Same as LOCATE
        call LOCATE
        ret
SETCURSOR       ENDP
```

```
70
71  ;*************************************************************************
72  WHERE           PROC
73  COMMENT !
74                  Reads the cursor position on the screen.
75                  IN:     none
76                  OUT:    DH   row number (0-18h)
77                          DL   col number (0-4Fh)
78                  ******  Remove the PUSH and POP CX if you want to read the
79                          cursor mode (CH start line - CL end line)
80          !
81          push    cx      ; remove for cursor mode
82          push    bx      ; page 0
83          push    ax
84          mov     bh,0
85          mov     ah,3
86          int     10h
87          pop     ax
88          pop     bx
89          pop     cx      ; remove for cursor mode
90          ret
91  WHERE           ENDP
92
93  ;*************************************************************************
94  GETCURSOR       PROC
95            ; same as WHERE
96            call WHERE
97            ret
98  GETCURSOR       ENDP
99
100 ;*************************************************************************
101 CURSOROFF        PROC
102     ; turns the cursor off by resizing it
103     push ax
104     mov AH, 1
105     mov CH, 20h
106     mov CL, 0
107     int 10h
108     pop ax
109     Ret
110 CURSOROFF       ENDP
111
112 ;*************************************************************************
113 CURSORON      PROC
114      ; turns cursor on by resizing it
115      push ax
116      mov  ah,1
117      mov  CH, 0Dh
118      mov  CL, 0Eh
119      int 10h
120      pop ax
121      ret
122 CURSORON      ENDP
123
124 ;*************************************************************************
125 WRITEATTR              PROC
126 COMMENT !
127                 Displays a character and an attribute w/o advance.
128                 IN:  AL    character
129                      CX    number of times to repeat character
130                      BL    attribute
131
132                 OUT: none
133         !
134         push    ax
135         push    bx
136         mov     bh,0    ; page
137         mov     ah,9
138         int     10h
```

```
        pop     bx
        pop     ax
        ret
WRITEATTR               ENDP

;*************************************************************************
WRITE                   PROC
COMMENT !
                Displays a character using the current attribute w/o advance.
                IN:  AL    character
                     CX    number of times to repeat character

                OUT: none
        !
        push    ax
        push    bx
        mov     bh,0
        mov     ah,0ah
        int     10h
        pop     bx
        pop     ax
        ret
WRITE                   ENDP

;*************************************************************************
READATTR                PROC
COMMENT   !
                Returns the character at the current position of the cursor on
                the screen along with the ATTRIBUTE
                IN:   none
                OUT:  AL    character
                      AH    attribute
          !
        push    bx
        mov     bh,0
        mov     ah,8
        int     10h
        pop     bx
        ret
READATTR                ENDP

;*************************************************************************
SCROLLATTR              PROC

COMMENT !
                Scrolls the window up.  Input lines inserted at bottom of
                window.   Set CX = 0000 and DX = 184FH for the entire screen.
                IN:  AL   number of lines (AL = 0 blanks the entire window)
                     CH,CL row,column of upper left hand corner of window
                     DH,DL row,column of lower right hand corner of window
                     BH    ATTRIBUTE for blank lines
                OUT:    none
        !
        push    ax
        mov     ah,6
        int     10h
        pop     ax
        ret

SCROLLATTR              ENDP

;*************************************************************************
CLRSCR                  PROC
COMMENT !
                Scrolls the screen up so that it is blank
                Uses normal, non-blinking, low-intensity as the attribute.
                Positions the cursor in upper left hand corner.
                IN:  none
                OUT: none
```

```
208           !
209           push    ax
210           push    bx
211           push    cx
212           push    dx
213           mov     cx,0000 ; upper left hand corner
214           mov     dx,184fh   ; lower right hand corner
215           mov     bh,07h     ; normal,non-b,non-i
216           mov     al,0       ; blank the entire window
217           mov     ah,6       ; scroll up
218           int     10h
219           mov     dx,0
220           mov     bh,0       ; page 0,
221           mov     ah,2       ; home cursor
222           int     10h
223           pop     dx
224           pop     cx
225           pop     bx
226           pop     ax
227           ret
228   CLRSCR              ENDP
229
230   ;****************************************************************************
231   CLRSCRATTR                  PROC
232   COMMENT !
233                   Scrolls the screen up so that it is blank
234                   and sets a new attribute for the screen.
235                   Positions the cursor in upper left hand corner.
236                   IN:  BH  new attribute
237                   OUT: none
238           !
239           push    ax
240           push    bx
241           push    dx
242           mov     cx,0000 ; upper left hand corner
243           mov     dx,184fh   ; lower right hand corner
244           mov     al, 0      ; 1997 fixed
245           mov     ah,6       ; scroll up
246           int     10h
247           mov     dx,0
248           mov     bh,0       ; page 0,
249           mov     ah,2       ; home cursor
250           int     10h
251           pop     dx
252           pop     bx
253           pop     ax
254           ret
255   CLRSCRATTR                ENDP
256
257   ;****************************************************************************
258   CSEG          ENDS
259
260
261   CSEG          SEGMENT  BYTE   PUBLIC
262                 ASSUME   CS:CSEG
263   ;****************************************************************************
264   ;                         K E Y B D   1 6
265   ;****************************************************************************
266
267                   PUBLIC  getch, kbhit
268   COMMENT !
269           The hardware int 9 inserts data to the memory keyboard buffer every
270           time a key is pressed. If an int 9 occured, each of these
271           procedures will return the data in AX.
272           AL = ascii char or zero if there is no ascii for the key
273           AH = scan code    !
274
275   ;****************************************************************************
276   getch           Proc
```

```
;         Extracts data from the buffer. Waits if buffer is empty.
;        Does not echo the key on screen
         Mov Ah,0
         Int 16h
         Ret
getch    Endp

;*************************************************************************
kbhit          Proc
Comment !
         Examines the buffer to see if a key was entered. AX has a copy of the
         key, but it still remains in the buffer.
         If buffer non empty, then  ZF is set off = 0.
         If buffer empty,  then  ZF is set on = 1.
         For example, JZ NOKEY  will branch to the label NOKEY if no key was pressed.
         !
           Mov Ah, 1
           Int 16h
           Ret
kbhit    Endp

CSEG       ENDS
                END
```