

I dug the following code out of my archive (it's dated 24/06/1995);
 this is the keyboard handler for a program which I wrote which blanks the screen if both shift keys are pressed.

```

kb_int    proc    far
          pushf                    ;Save FLAGS
          push  ax                  ;Save AX
          in   al,60h               ;Read the scan code
          mov  cs:[scancode],al     ;Save it
          pop  ax                  ;Restore AX
          popf                      ;Restore FLAGS
          pushf                    ;Push FLAGS
          call cs:[int09h]         ;Call previous handler
          sti                             ;Enable interrupts

          test cs:[scancode],80h   ;Exit if high bit of scan
          jnz kb_exit              ; code is set

          push ax                  ;Save AX and ES
          push es
          mov  ax,40h              ;Point ES to the BIOS
          mov  es,ax              ; Data Area
          mov  al,es:[17h]         ;Get keyboard flags
          and  al,03h             ;Zero the upper 6 bits
          cmp  al,03h             ;Are the Shift keys pressed?
          pop  es                  ;Restore AX and ES
          pop  ax
          jne  kb2                 ;Branch if they're not
          call disable_video       ;Blank the screen
          iret                      ;Return from interrupt
kb2:      push  cs:[time]          ;Reset countdown timer
          pop  cs:[count]
          cmp  cs:[blanked],0     ;Is the screen blanked?
          je   kb_exit            ;If not, then exit
          call enable_video       ;Unblank the screen
kb_exit:  iret                      ;Return from interrupt
kb_int    endp

```

And here is the code to hook the interrupt - this runs at the beginning of the program

```

          mov  ax,3509h           ;Hook interrupt 09H
          int  21h
          mov  word ptr int09h,bx
          mov  word ptr int09h[2],es
          mov  ax,2509h
          mov  dx,offset kb_int
          int  21h

```

The entire program is too long to post here - 33KB. But then, you only want to see an example of what to do....

Here's another example which checks for various alt/ctrl/key functions

```

even
New_09    proc    far
          sti
          pushf
          push  ax
          mov  ah, 2                ; get shift key status
          int  16h
          and  al, 0Fh
          cmp  al, 12                ; alt/ctrl?
          jne  @@0                  ; no
          in  al, 60h
          cmp  al, 19                ; 'R'?
          je   @@1                  ; yes
          cmp  al, 31                ; 'S'
          je   @@1                  ; yes
          cmp  al, 16                ; 'Q'
          je   @@1                  ; yes
@@0:      pop  ax                    ; exit if not my hotkey
          popf
          jmp  cs:old_09

@@1:      push  bp
          mov  bp, ax                ; save scan code
          in  al, 61h                ; reset keyboard
          mov  ah, al
          or   al, 80h
          out  61h, al
          mov  al, ah
          out  61h, al
          cli
          mov  al, 20h
          out  20h, al
          sti

          mov  ax, bp                ; restore scan code
          cmp  al, 16                ; was it Q?
          jne  @@GetMode

```

I don't remember now why all the ins and outs were needed (this code is from 14/05/1992 - twenty years ago!!!).