

Babystep5

From OSDev Wiki

This code is meant to show how the hardware interrupt generated when you press a key can be handled by replacing the `seg:offset` specified in the IVT (interrupt vector table). This normally points to a BIOS routine. To find the entry in the IVT, multiply the interrupt number by 4 (which is the size of each entry).

This key handler just displays the scan code without conversion to ASCII, buffering, or handling extended keys. The reason for doing this is to not muddle up the basic idea, which is to provide input, as well as output, in its most simple form.

I will not go into the hows and whys of reading the ports involved in a key press. Suffice it to say that you are communicating with actual chips (or parts of chips), not some software intermediary. I personally feel it is good to remember that, no matter what level of abstraction you work at, you are ultimately telling hardware what to do.

I will point out the turning the keyboard on/off through port 0x61 is given in its complete form, some of which might not be needed, depending on the system.

Babystep5

Tutorial

Previous

Babystep4

Next

Babystep6

```
;=====
; nasmw boot.asm -f bin -o boot.bin
; partcopy boot.bin 0 200 -f0

[ORG 0x7c00]      ; add to offsets
    jmp start

%include "print.inc"

start:    xor ax, ax    ; make it zero
    mov ds, ax        ; DS=0
    mov ss, ax        ; stack starts at 0
    mov sp, 0x9c00     ; 200h past code start

    mov ax, 0xb800     ; text video memory
    mov es, ax

    cli                ;no interruptions
    mov bx, 0x09       ;hardware interrupt #
    shl bx, 2          ;multiply by 4
```

```

xor ax, ax
mov gs, ax    ;start of memory
mov [gs:bx], word keyhandler
mov [gs:bx+2], ds ; segment
sti

jmp $        ; loop forever

```

keyhandler:

```

in al, 0x60    ; get key data
mov bl, al     ; save it
mov byte [port60], al

in al, 0x61    ; keybrd control
mov ah, al
or al, 0x80    ; disable bit 7
out 0x61, al   ; send it back
xchg ah, al   ; get original
out 0x61, al   ; send that back

mov al, 0x20   ; End of Interrupt
out 0x20, al   ;

and bl, 0x80   ; key released
jnz done      ; don't repeat

mov ax, [port60]
mov word [reg16], ax
call printreg16

```

done:

```

iret

```

```

port60    dw 0

```

```

    times 510-($-$$) db 0 ; fill sector w/ 0's
    dw 0xAA55           ; req'd by some BIOSes
;=====

```

See Also

External Links

- hardware fun <http://chip.ms.mff.cuni.cz/pcguts/>
- Intel's Summer Reading List <http://developer.intel.com/vtune/cbts/refman.htm>
- John Fine links to hardware programming
<http://www.geocities.com/SiliconValley/Peaks/8600/device.html>

Retrieved from "<http://wiki.osdev.org/index.php?title=Babystep5&oldid=7947>"

Category: Babystep

- This page was last modified on 13 May 2009, at 08:28.
- This page has been accessed 38,261 times.