

Babystep4

From OSDev Wiki

I know this is starting to look like a half-baked tutorial in assembly, but there's actually a reason behind my madness. Namely, solving as many problems as possible before switching to Protected mode etc. will lessen the confusion a great deal.

This example prints a string and the contents of a memory location (which is the first letter of the string in video memory). It is meant to demonstrate printing to screen in text mode without using BIOS, as well as converting hex so it can be displayed -- so we can check register and memory values.

A stack is included, but left unused.

Babystep4	
Tutorial	
Previous	Next
Babystep3	Babystep5

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

[ORG 0x7c00]      ; add to offsets
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

mov si, msg       ; show text string
call sprint

mov ax, 0xb800    ; look at video mem
mov gs, ax
mov bx, 0x0000    ; 'W'=57 attrib=0F
mov ax, [gs:bx]

mov word [reg16], ax ;look at register
call printreg16

hang:
jmp hang
```

```

;-----
dochar:    call cprint          ; print one character
sprint:    lodsb                ; string char to AL
    cmp al, 0
    jne dochar    ; else, we're done
    add byte [ypos], 1    ;down one row
    mov byte [xpos], 0    ;back to left
    ret

cprint:    mov ah, 0x0F        ; attrib = white on black
    mov cx, ax                ; save char/attribute
    movzx ax, byte [ypos]
    mov dx, 160                ; 2 bytes (char/attrib)
    mul dx                    ; for 80 columns
    movzx bx, byte [xpos]
    shl bx, 1                ; times 2 to skip attrib

    mov di, 0                ; start of video memory
    add di, ax                ; add y offset
    add di, bx                ; add x offset

    mov ax, cx                ; restore char/attribute
    stosw                    ; write char/attribute
    add byte [xpos], 1        ; advance to right

    ret

;-----

printreg16:
    mov di, outstr16
    mov ax, [reg16]
    mov si, hexstr
    mov cx, 4                ;four places
hexloop:
    rol ax, 4                ;leftmost will
    mov bx, ax                ; become
    and bx, 0x0f              ; rightmost
    mov bl, [si + bx];index into hexstr
    mov [di], bl
    inc di
    dec cx
    jnz hexloop

    mov si, outstr16
    call sprint

    ret

```

```

;-----

xpos    db 0
ypos    db 0
hexstr   db '0123456789ABCDEF'
outstr16 db '0000', 0 ;register value string
reg16    dw 0 ; pass values to printreg16
msg      db "What are you doing, Dave?", 0
times 510-($-$$) db 0
db 0x55
db 0xAA
;=====

```

Retrieved from "<http://wiki.osdev.org/index.php?title=Babystep4&oldid=7901>"

Category: Babystep

-
- This page was last modified on 9 May 2009, at 20:32.
 - This page has been accessed 44,514 times.