Babystep7

From OSDev Wiki

Unreal Mode

a.k.a Big Real or voodoo mode

Babystep7	
Tutorial	
Previous	**Next**
Babystep6	Babystep8

While this code is largely just a party trick, understanding it gives a gentle intro to protected mode concepts and possibly avoids some headaches later on 'cause you skipped over this kind of stuff.

The single descriptor in the global descriptor table at the bottom is laid out to match Babystep6. The 'size' given is 1 MB, the base address is 0x0, and the bit fields you can work out yourself.

The reason for doing this is to enable 32-bit offsets in real mode. However, you won't be able to go past 1 meg quite yet.

In protected mode, the bits 3-15 in the segment register are an index into the descriptor table. That's why in this code 0x08 = 1000b gets you the 1 entry.

When this register given a "selector", a "segment descriptor cache register" is filled with the descriptor values, including the size (or limit). After the switch back to real mode, these values are not modified, regardless of what value is in the 16-bit segment register. So the 64k limit is no longer valid and 32-bit offsets can be used with the real-mode addressing rules (i.e. shift segment 4 bits, then add offset).

Finally, note that IP is unaffected by all this, so the code itself is still limited to 64k.

```
AsmExample:

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

[ORG 0x7c00]      ; add to offsets

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

   cli      ; no interrupt
   push ds      ; save real mode
```

```
   lgdt [gdtinfo]   ; load gdt register

   mov  eax, cr0   ; switch to pmode by
   or al,1         ; set pmode bit
   mov  cr0, eax

   mov  bx, 0x08   ; select descriptor 1
   mov  ds, bx   ; 8h = 1000b

   and al,0xFE     ; back to realmode
   mov  cr0, eax   ; by toggling bit again

   pop ds      ; get back old segment
   sti

   mov bx, 0x0f01   ; attrib/char of smiley
   mov eax, 0x0b8000 ; note 32 bit offset
   mov word [ds:eax], bx

   jmp $      ; loop forever

gdtinfo:
   dw gdt_end - gdt - 1   ;last byte in table
   dd gdt         ;start of table

gdt        dd 0,0  ; entry 0 is always unused
flatdesc    db 0xff, 0xff, 0, 0, 0, 10010010b, 11001111b, 0
gdt_end:

   times 510-($-$$) db 0  ; fill sector w/ 0's
   db 0x55           ; req'd by some BIOSes
   db 0xAA
;=========================================
```

See Also

Articles

- Unreal Mode

Retrieved from "http://wiki.osdev.org/index.php?title=Babystep7&oldid=10711"
Category: Babystep
