

1) Modo protegido: estado nativo del procesador

Modo real: modo en el que arrancan todos los x86 , despues de un set y reset

El real trabaja con 16 bits y podemos direccionar hasta 1MB de memoria. Los modos de direccionamiento son más limitados que en modo protegido. No hay protección de memoria ni privilegio

En el modo protegido se trabaja con 4GB y hay 4 niveles de protección. Tiene privilegios

2) No podemos usar el so en modo real porque es inseguro. Además, tiene muy poca memoria.

3) El GDT es una estructura de datos que administra los descriptores de segmento(lo que hay dentro de cada direccion de memoria en GDT) en el modo protegido del procesador. Los primeros 8 bits van a estar dedicados a la base. Luego en el proximo bit se va a llamar accedido, que se setea cada vez que se va a acceder a una direccion en el segmento. Permite al sistema operativo contabilizar los accesos para poder hacer estadísticas de uso que permiten identificar cual es el segmento a ser desalojado llegado el momento. r es para codigo, w es para data. Expand down es para data, conforming es para codigo. bit 10 duda(ED/C).

Si el bit 11 es 1 es segmento de codigo. Si el bit 11 es 0 es segmento de datos.Luego viene un bit dedicado a system,que es un activo bajo que permite administrar en la tabla de descriptores dos clases de segmentos: segmentos de codigo o datos y segmentos de sistema.El proximo bit va a ser el DPL, que debe ser el nivel de segmento que contiene el codigo para acceder a este.

El proximo bit va a mostrar si esta presente el segmento correspondiente en memoria o no. Luego van a haber 4 bits que muestran el limite, es decir, el tamaño del segmento - 1.

4) 1010 en la practica esta la tablita.

10)

```
extern gdt_entry_t gdt[]; // devuelve la str_gdt_entry
extern gdt_descriptor_t GDT_DESC; // devuelve una estructura
descriptor_t con su longitud y su direccion
```

LGDT carga los registros de GDT en donde arranca y devuelve la longitud . En el extern GDT descriptor. En el gdt.h.

13)Sí, hay que modificarlo. Está hecho en el código del taller.

15)Usamos el selector de nivel 0 de código y lo shifteamos 3 bits.

20)50x80x2bytes

21) screen_draw_box pinta un cuadrado relleno de un mismo caracter y atributo de dimensiones pasadas por parametro. Usa una matriz para representar cada celda. Cada celda ocupa 2 bytes (1 byte c y 1 byte a)