Segmentación

Los selectores de segmento son:

- 0x20 = 00100000b
- 0x30 = 00110000b
- 0x40 = 01000000b
- 0x50 = 01010000b

De los cuales los últimos 2 bits describen el RPL (Requested Privilege Level), el antepenúltimo describe el Table Indicator (si se trata de una tabla de descriptores locales o globales) y el resto de los bits denotan el índice dentro de la tabla. Dicho esto, podemos interpratarlos como

- 0x20 = dpl=0, ti=0, indice = 4
- 0x30 = dpl=0, ti=0, indice = 6
- 0x40 = dpl=0, ti=0, indice = 8
- 0x50 = dpl=0, ti=0, indice = 10

Como dice la página 89 del manual de System Programming de Intel, “The base address plus the offset thus forms a linear address in the processor’s linear address space”. Por lo tanto, podemos calcular el campo Base Address del descriptor de segmento como la dirección lineal - el offset.
Luego, citando de vuelta al manual, “For expand-up segments, the offset in a logical address can range from 0 to the segment limit” (página 98). Luego, como la consigna especifica que la acción abarca 4 bytes, concluímos que el límite de cada descriptor será su offset + 3 bytes, ya que si se intentara acceder a [offset + 4] estaríamos leyendo el 5to byte desde offset y el límite describe el último byte al que deberíamos poder acceder

Índice 4 de la GDT:

Segment limit: 0x003993AA + 0x3 = **0x3993AD**
Base address: 0x003993AA - 0x003993AA = **0x0**
Type: **0x2** = Data, Read/Write (no hay un tipo que implique sólo escritura)
S: **0x1** (es un segmento de código/datos, no de sistema)
DPL: **0x0** por consigna
P: **0x1** (presente)
D/B: **0x1** (código/datos de 32 bits)
G: **0x0** alcanza con interpretar el límite en bytes para describir el alcance del segmento
L: **0x0** no estamos en 64 bits

Índice 6 de la GDT:

Segment limit: 0x0000011A + 0x3 = **0x0000011D**
Base address: 0x00999FFF - 0x0000011A = **0x999EE5**
Type: **0x0** = Data, Read only
S: **0x1** (es un segmento de código/datos, no de sistema)
DPL: **0x0** por consigna
P: **0x1** (presente)

D/B: **0x1** (código/datos de 32 bits)
G: **0x0** alcanza con interpretar el límite en bytes para describir el alcance del segmento
L: **0x0** no estamos en 64 bits

Índice 8 de la GDT:

Segment limit: 0x9993252A + 0x3 = **0x9993252D**
Base address: 0x9993252A - 0x9993252A = **0x0**
Type: **0xA** = Code, Execute
S: **0x1** (es un segmento de código/datos, no de sistema)
DPL: **0x0** por consigna
P: **0x1** (presente)
D/B: **0x1** (código/datos de 32 bits)
G: **0x0** alcanza con interpretar el límite en bytes para describir el alcance del segmento
L: **0x0** no estamos en 64 bits

Índice 10 de la GDT:

Segment limit: 0x0043534A + 0x3 = **0x0043534D**
Base address: 0x99912311 - 0x0043534A = **0x994DCFC7**
Type: **0x0** = Data, Read only
S: **0x1** (es un segmento de código/datos, no de sistema)
DPL: **0x0** por consigna
P: **0x1** (presente)
D/B: **0x1** (código/datos de 32 bits)
G: **0x0** alcanza con interpretar el límite en bytes para describir el alcance del segmento
L: **0x0** no estamos en 64 bits

Paginación

Dada una dirección lineal, para poder interpretarla correctamente en el contexto de paginación debemos separar sus bits en 12 bits para el offset, 10 bits para el índice dentro de una Page Table y 10 bits para el índice dentro de una Page Directory

- 0x99912311 = 0x266 | 0x112 | 0x311
- 0x9993252A = 0x266 | 0x132 | 0x52A
- 0x00999FFF = 0x002 | 0x199 | 0xFFF
- 0x003993AA = 0x000 | 0x399 | 0x3AA

Directorio de páginas

En el índice 0x0 habrá una PDE con la siguiente información:

- User/Supervisor: **0x0** (supervisor)
- Read/Write: **0x1** (escritura habilitada)
- Present: **0x1**
- Table: Una tabla tal que en su índice 0x399 habrá una PTE con la siguiente info:
    - User/Supervisor: **0x0** (supervisor)
    - Read/Write: **0x1** (escritura habilitada)
    - Present: **0x1**
    - Physical addres: 0x0

En el índice 0x2 habrá una PDE con la siguiente información:

- User/Supervisor: **0x0** (supervisor)
- Read/Write: **0x0** (sólo lectura)
- Present: **0x1**
- Table: Una tabla tal que en su índice 0x199 habrá una PTE con la siguiente info:
    - User/Supervisor: **0x0** (supervisor)
    - Read/Write: **0x0** (sólo lectura)
    - Present: **0x1**
    - Physical addres: 0x88462

En el índice 0x266 habrá una PDE con la siguiente información:

- User/Supervisor: **0x0** (supervisor)
- Read/Write: **0x0** (sólo lectura)
- Present: **0x1**
- Table: Una tabla tal que en su índice 0x112 habrá una PTE con la siguiente info:
    - User/Supervisor: **0x0** (supervisor)
    - Read/Write: **0x0** (sólo lectura)
    - Present: **0x1**
    - Physical addres: 0x99912
      Y en su índice 0x132 habrá una PTE con la siguiente info:
    - User/Supervisor: **0x0** (supervisor)
    - Read/Write: **0x0** (sólo lectura)
    - Present: **0x1**
    - Physical addres: 0x0

Cuando digo “una tabla tal que” en realidad me refiero a un puntero que apunta a un Page Table