



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Programacion SIMD

Organización del Computador II
Segundo Cuatrimestre de 2015

Integrante	LU	Correo electrónico
Leandro Raffo		
Maximiliano Fernández Wortman		
Uriel Rozenberg		



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Ejercicio 1.	3
2. Ejercicio 2.	3
3. Ejercicio 3.	3
4. Ejercicio 4.	3
5. Ejercicio 5.	3

1. Ejercicio 1.

Básicamente armamos la tabla de gdt, empezando en la posición 8 del array, con 4 segmentos, dos de código nivel 0 y 3 y dos de datos nivel 0 y nivel 3, llamados respectivamente:

```
GDT_IDX_CS_CERO_DESC  
GDT_IDX_CS_TRES_DESC  
GDT_IDX_DS_CERO_DESC  
GDT_IDX_DS_TRES_DESC
```

Excepto el dpl y el tipo todos estos segmentos tienen un límite de 0x1E400, con base en 0x0000, ya que esto nos da 0x1F400 que son 500mb de memoria, esto es porque tenemos la granularidad en 1 es decir contamos de a 4k.

Para pasar a modo protegido lo que hacemos es

```
    jmp 0x40:modoprotegido  
BITS 32  
modoprotegido:  
    mov eax, 0x50  
    mov ss, ax  
    mov ds, ax  
    mov gs, ax  
    mov fs, ax  
    mov es, ax  
    mov ebp, 0x1337  
    mov esp, 0x28000
```

Hacemos un far jump con el offset para CS tenga el valor de nuestro código en nivel 0 que es $8 \ll 3 = 0x40$ y ponemos el comando BITS 32 para que el ensamblador sepa que las próximas instrucciones sean generadas para un procesador de 32 bits, seteamos todos los registros de segmentos con el de datos de nivel 0 ($10 \ll 3 = 0x50$) y definimos el stack esp en 0x28000 como pide el enunciado.

2. Ejercicio 2.

3. Ejercicio 3.

4. Ejercicio 4.

5. Ejercicio 5.