



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Programación SIMD

Organización del Computador II
Primer Cuatrimestre de 2014

Integrante	LU	Correo electrónico
Barrios, Leandro Ezequiel	404/11	ezequiel.barrios@gmail.com
Benegas, Gonzalo Segundo	958/12	gsbenegas@gmail.com



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	1
	a. GDT Básica	1
	b. Modo Protegido	2
	c. GDT - Área de Memoria	2
	d. Limpiar la pantalla	2
2.	Ejercicio 2	3
	a. replace-me-with-a-descriptive-text	3
	b. replace-me-with-a-descriptive-text	3
3.	Ejercicio 3	4
	a. replace-me-with-a-descriptive-text	4
	b. replace-me-with-a-descriptive-text	4
	c. replace-me-with-a-descriptive-text	4
	d. replace-me-with-a-descriptive-text	4
4.	Ejercicio 4	5
	a. replace-me-with-a-descriptive-text	5
	b. replace-me-with-a-descriptive-text	5
	c. replace-me-with-a-descriptive-text	5
	d. replace-me-with-a-descriptive-text	5
5.	Ejercicio 5	7
	a. replace-me-with-a-descriptive-text	7
	b. replace-me-with-a-descriptive-text	7
	c. replace-me-with-a-descriptive-text	7
	d. replace-me-with-a-descriptive-text	7
6.	Ejercicio 6	8
	a. replace-me-with-a-descriptive-text	8
	b. replace-me-with-a-descriptive-text	8
	c. replace-me-with-a-descriptive-text	8

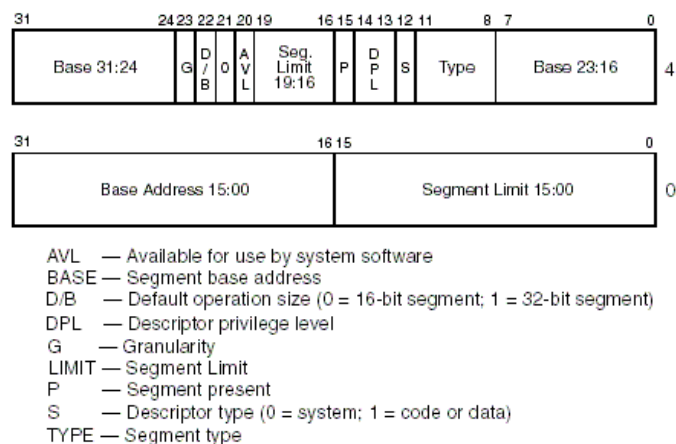
	d.	replace-me-with-a-descriptive-text	8
	e.	replace-me-with-a-descriptive-text	9
	f.	replace-me-with-a-descriptive-text	9
	g.	replace-me-with-a-descriptive-text	9
7.	Ejercicio 7		10
	a.	replace-me-with-a-descriptive-text	10
	b.	replace-me-with-a-descriptive-text	10
	c.	replace-me-with-a-descriptive-text	10
	d.	replace-me-with-a-descriptive-text	10
	e.	replace-me-with-a-descriptive-text	10
	f.	replace-me-with-a-descriptive-text	11
8.	Ejercicio 8 (optativo)		12
	a.	replace-me-with-a-descriptive-text	12
	b.	replace-me-with-a-descriptive-text	12
	c.	replace-me-with-a-descriptive-text	12

Ejercicios

1) Ejercicio 1

a) **GDT Básica** Completar la Tabla de Descriptores Globales (GDT) con 4 segmentos, dos para código de nivel 0 y 3; y otros dos para datos de nivel 0 y 3. Estos segmentos deben direccionar los primeros 733MB de memoria. En la gdt, por restricción del trabajo práctico, las primeras 8 posiciones se consideran utilizadas y no deben utilizarse. El primer índice que deben usar para declarar los segmentos, es el 9 (contando desde cero).

La GDT es una tabla de descriptores de segmento, en donde cada entrada describe un segmento particular, y tiene la siguiente forma:



Entrada 1

- $.base_{015} = 0$,
- $.base_{2316} = 0$,
- $.base_{3124} = 0$, // Hay que direccionar 187648 bloques de 4kbytes o sea $0x2DCFF$. $limit_{015} = 0xDCFF$,
- $.limit_{1619} = 0x2$, $.type = 8$, // 1000 en binario, Code Execute – Only
- $.s = 1$, // Código/Datos
- $.dpl = 0$, // Nivel de privilegio 0
- $.p = 1$, // 1 = Presente

- `.avl = 0, //` A disposición del programador
- `.l = 0, //` Va en 0 cuando estás en 32 bits
- `.db = 1, //` 0=16bits, 1=32bits
- `.g = 1 //` Granularidad, bloques de 4k

Entrada 2

Entrada 3

Entrada 4 Editamos el archivo gdt.c.

b) **Modo Protegido** Completar el código necesario para pasar a modo protegido y setear la pila del kernel en la dirección 0x27000.

FIXME: Respuesta

c) **GDT - Área de Memoria** Declarar un segmento adicional que describa el área de la pantalla en memoria que pueda ser utilizado sólo por el kernel.

FIXME: Respuesta

d) **Limpiar la pantalla** Escribir una rutina que se encargue de limpiar la pantalla y pintar el area de el mapa un fondo de color (sugerido verde). Para este ejercicio se debe escribir en la pantalla usando el segmento declarado en el punto anterior (para los próximos ejercicios se accederá a la memoria de vídeo por medio del segmento de datos de 773MB).

Nota: La GDT es un arreglo de gdt entry declarado sólo una vez como gdt. El descriptor de la GDT en el código se llama GDT DESC.

FIXME: Respuesta

2) Ejercicio 2

a) **replace-me-with-a-descriptive-text** Completar las entradas necesarias en la IDT para asociar diferentes rutinas a todas las excepciones del procesador. Cada rutina de excepción debe indicar en pantalla qué problema se produjo e interrumpir la ejecución. Posteriormente se modificarán estas rutinas para que se continúe la ejecución, resolviendo el problema y desalojando a la tarea que lo produjo.

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Hacer lo necesario para que el procesador utilice la IDT creada anteriormente. Generar una excepción para probarla.

Nota: La IDT es un arreglo de idt entry declarado solo una vez como idt. El descriptor de la IDT en el código se llama IDT DESC. Para inicializar la IDT se debe invocar la función idt inicializar.

FIXME: Respuesta

3) Ejercicio 3

a) **replace-me-with-a-descriptive-text** Escribir una rutina que se encargue de limpiar el buffer de vídeo y pintarlo como indica la figura 8. Tener en cuenta que deben ser escritos de forma genérica para posteriormente ser completados con información del sistema. Además considerar estas imágenes como sugerencias, ya que pueden ser modificadas a gusto según cada grupo mostrando siempre la misma información.

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Escribir las rutinas encargadas de inicializar el directorio y tablas de páginas para el kernel (mmu inicializar dir kernel). Se debe generar un directorio de páginas que mapee, usando identity mapping, las direcciones 0x00000000 a 0x00DC3FFF, como ilustra la figura 5. Además, esta función debe inicializar el directorio de páginas en la dirección 0x27000 y las tablas de páginas según muestra la figura 1.

FIXME: Respuesta

c) **replace-me-with-a-descriptive-text** Completar el código necesario para activar paginación.

FIXME: Respuesta

d) **replace-me-with-a-descriptive-text** Escribir una rutina que imprima el nombre del grupo en pantalla. Debe estar ubicado en la primer línea de la pantalla alineado a derecha.

FIXME: Respuesta

4) Ejercicio 4

a) **replace-me-with-a-descriptive-text** Escribir una rutina (inicializar mmu) que se encargue de inicializar las estructuras necesarias para administrar la memoria en el area libre.

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Escribir una rutina (mmu inicializar dir tarea) encargada de inicializar un directorio de páginas y tablas de páginas para una tarea, respetando la figura 5. La rutina debe copiar el código de la tarea a su área asignada, es decir sus dos páginas de código dentro de el mapa y mapear dichas páginas a partir de la dirección virtual 0x08000000(128MB).

FIXME: Respuesta

c) **replace-me-with-a-descriptive-text** Escribir dos rutinas encargadas de mapear y desmapear páginas de memoria.

I- mmu mapear pagina(unsigned int virtual, unsigned int cr3, unsigned int fisica)
Permite mapear la página física correspondiente a fisica en la dirección virtual virtual utilizando cr3.

II- mmu unmapear pagina(unsigned int virtual, unsigned int cr3). Borra el mapeo creado en la dirección virtual virtual utilizando cr3.

FIXME: Respuesta

d) **replace-me-with-a-descriptive-text** Construir un mapa de memoria para tareas e intercambiarlo con el del kernel, luego cambiar el color del fondo del primer caracter de la pantalla y volver a la normalidad.

Nota: Por la construcción del kernel, las direcciones de los los mapas de memoria (page directory y page table) están mapeadas con identity mapping. En los ejercicios en donde se modifica el directorio o tabla de páginas, hay que llamar a la función tlbflush para que se invalide la cache de traducción de direcciones.

FIXME: Respuesta

5) Ejercicio 5

a) **replace-me-with-a-descriptive-text** Completar las entradas necesarias en la IDT para asociar una rutina a la interrupción del reloj, otra a la interrupción de teclado y por último una a la interrupción de software 0x52.

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Escribir la rutina asociada a la interrupción del reloj, para que por cada tick llame a la función screen proximo reloj. La misma se encarga de mostrar cada vez que se llame, la animación de un cursor rotando en la esquina inferior derecha de la pantalla. La función proximo reloj está definida en isr.asm.

FIXME: Respuesta

c) **replace-me-with-a-descriptive-text** Escribir la rutina asociada a la interrupción de teclado de forma que si se presiona cualquier número, se presente el mismo en la esquina superior derecha de la pantalla. El número debe ser escrito en color blanco con fondo de color aleatorio por cada tecla que sea presionada¹.

FIXME: Respuesta

d) **replace-me-with-a-descriptive-text** Escribir la rutina asociada a la interrupción 0x52 para que modifique el valor de eax por 0x42. Posteriormente este comportamiento va a ser modificado para atender los servicios del sistema.

FIXME: Respuesta

¹[http://wiki.osdev.org/Text UI](http://wiki.osdev.org/Text_UI)

6) Ejercicio 6

a) **replace-me-with-a-descriptive-text** Definir 3 entradas en la GDT para ser usadas como descriptores de TSS. Una será reservada para la tarea inicial y otras dos para realizar el intercambio entre tareas, denominadas TSS1 y TSS2 respectivamente.

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Completar la entrada de la TSS1 con la información de la tarea Idle. Esta información se encuentra en el archivo TSS.C. La tarea Idle se encuentra en la dirección 0x00020000. La pila se alojará en la misma dirección que la pila del kernel y será mapeada con identity mapping. Esta tarea ocupa 2 paginas de 4KB y debe ser mapeada con identity mapping. Además la misma debe compartir el mismo CR3 que el kernel.

FIXME: Respuesta

c) **replace-me-with-a-descriptive-text** Completar el resto de la información correspondiente a cada tarea en la estructura auxiliar de contextos. El código de las tareas se encuentra a partir de la dirección 0x00010000 ocupando dos páginas de 4kb cada una. El mismo debe ser mapeado a partir de la dirección 0x08000000. Para la dirección de la pila se debe utilizar el mismo espacio de la tarea, la misma crecerá desde la base de la tarea. Para el mapa de memoria se debe construir uno nuevo para cada tarea utilizando la función mmu inicializar dir usuario. Además, tener en cuenta que cada tarea utilizará una pila distinta de nivel 0, para esto se debe pedir una nueva pagina libre a tal fin.

FIXME: Respuesta

d) **replace-me-with-a-descriptive-text** Completar la entrada de la GDT correspondiente a la tarea inicial.

FIXME: Respuesta

e) **replace-me-with-a-descriptive-text** Completar la entrada de la GDT correspondiente a la TSS1, que contiene la información de la tarea Idle.

FIXME: Respuesta

f) **replace-me-with-a-descriptive-text** Completar la entrada de la GDT correspondiente a la TSS2.

FIXME: Respuesta

g) **replace-me-with-a-descriptive-text** Escribir el código necesario para ejecutar la tarea Idle, es decir, saltar intercambiando las TSS, entre la tarea inicial y la tarea Idle.

Nota: En tss.c están definidas las tss como estructuras TSS. Trabajar en tss.c y kernel.asm.

FIXME: Respuesta

7) Ejercicio 7

a) **replace-me-with-a-descriptive-text** Construir una función para inicializar las estructuras de datos del scheduler.

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Crear la función sched proximo indice() que devuelve el índice en la GDT de la próxima tarea a ser ejecutada. Construir la rutina de forma devuelva el indice de la TSS1 y luego el de la TSS2 de forma intercalada, para dos tareas fijas.

FIXME: Respuesta

c) **replace-me-with-a-descriptive-text** Modificar la rutina de la interrupción 0x52, para que implemente los tres servicios del sistema según se indica en la sección 3.1.1.

FIXME: Respuesta

d) **replace-me-with-a-descriptive-text** Modificar el código necesario para que se realice el intercambio de tareas por cada ciclo de reloj. El intercambio se realizará según indique la función sched proximo indice().

FIXME: Respuesta

e) **replace-me-with-a-descriptive-text** Modificar la función sched proximo indice() de forma que ejecute todas las tareas según se describe en la sección 3.2.

FIXME: Respuesta

f) **replace-me-with-a-descriptive-text** Modificar las rutinas de excepciones del procesador para que impriman el problema que se produjo en pantalla, desalojen a la tarea que estaba corriendo y corran la próxima, indicando en pantalla porque razón fue desalojada la tarea en cuestión. Nota: Se recomienda construir funciones en C que ayuden a resolver problemas como convertir direcciones de el mapa a direcciones físicas.

FIXME: Respuesta

8) Ejercicio 8 (optativo)

a) **replace-me-with-a-descriptive-text** Crear un tanque (tarea) propio que mapee paginas a muerte contra otros intrepidos tanques. Para esto pueden editar el código del primer tanque a gusto. El tanque debe tener las siguientes características,

- No ocupar más de 8 kb (tener en cuenta la pila).
- Tener como punto de entrada la dirección cero.
- Estar compilada para correr desde la dirección 0x08000000.
- Utilizar solo los servicios presentados en el trabajo práctico.

Explicar en pocas palabras qué estrategia utilizaron en su tanque en términos de «defensa» y «ataque».

FIXME: Respuesta

b) **replace-me-with-a-descriptive-text** Si consideran que su tanque es capaz de enfrentarse contra los tanques del resto de sus compañeros, pueden enviar el binario a la lista de docentes indicando los siguientes datos,

- Nombre del tanque, ej: *T-90S Bhishma*
- Características letales, ej: *Cañón 2A46M de 125 mm con cargador automático*
- Sistema de defensa, ej: *1.350 mm en blindaje laminado, blindaje reactivo...*

Se realizará una competencia a fin de cuatrimestre con premios en/de chocolate para los primeros puestos.

FIXME: Respuesta

c) **replace-me-with-a-descriptive-text** Mencionar la mayor cantidad de características del tanque montado en el jardín del edificio Libertador en la ciudad de Buenos Aires.

FIXME: Respuesta