

Organización del Computador II

TP2

13 de junio de 2015

Integrante	LU	Correo electrónico
Federico Beuter	827/13	federicobeuter@gmail.com
Juan Rinaudo	864/13	jangamesdev@gmail.com
Mauro Cherubini	835/13	cheru.mf@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción	3
2. Ejercicio 1	4
3. Ejercicio 2	5
4. Ejercicio 3	6
5. Ejercicio 4	7
6. Ejercicio 5	8
7. Ejercicio 6	9
8. Ejercicio 7	10
9. Ejercicio 8	11

1. Introducción

Aca va la introduccion

2. Ejercicio 1

Para la resolución de este ejercicio modificamos kernel.asm, gdt.c, gdt.h, defines.h y screen.c

(a) Comenzamos desde C creando los defines de las tablas de descriptors en defines.h, como los primeros 8 índices se consideran utilizadas vamos a enumerar nuestros descriptors desde 8 (Como indica el subíndice a), creando los siguientes defines. `GDT_NIVEL0_CODIGO8 GDT_NIVEL0_DATOS9 GDT_NIVEL3_CODIGO10 GDT_NIVEL3_DATOS11`

Luego en el archivo gdt.c agregamos al array de gdt los 4 descriptors, con su respectivo nivel (0 para los descriptors del kernel y 3 para los de usuario) y tipo, con su base en el principio de la memoria y su límite en 500MB (Setiando granularity y el valor de límite), setiando el bit de presente y 32 bits con d/b y no de sistema.

(b) Para el siguiente subíndice (b) pasamos a kernel.asm donde habilitamos A20 (utilizando la función brindada por la cátedra) y luego cargamos la gdt usando LGDT y el puntero al descriptor de gdt `GDT_DESCRIPTOR` (Estructura brindada por la cátedra)

Ahora setiamos los selectores de segmento de datos y de stack, y finalizamos setiando la base(EBP) y el final de la pila(ESP) en 0x27000.

(c) Para este punto volvemos a defines.h y definimos un quinto índice para el descriptor de la memoria de la pantalla. `GDT_PANTALLA12` luego en gdt.c agregamos al array el descriptor que comienza en 0xB8000 (Memoria de video) con un límite de 320x200.

(d) Para este punto vamos a screen.c y completamos las funciones auxiliares `screen_inicializar` (Limpiar la pantalla y escribir los pu

3. Ejercicio 2

4. Ejercicio 3

Para resolver este ejercicio se modificara `screen.c`, `screen.h`, `mmu.c`, `mmu.h`, `kernel.asm` y `defines.h`.

- (a) Para comenzar vamos a modificar `screen.c` y `screen.h`, creamos la funcion en C `screen_refrescarqueeseenrgaradelimpiarlapa`
- (b) Definimos `DIR_PAGINAS_KERNEL` como `0x27000` en `defines.h`, luego vamos a `mmu.h` creamos `mmu_inicializar_dir_kernel`,
- (c) Vamos a `kernel.asm` y llamamos el extern `mmu_inicializar_dir_kernel` para crear el directorio de tablas y las tablas. Luego muevo
- (d) Para terminar en `screen.c/h` creamos la funcion `print_group` que usa `print` para escribir el nombre del grupo (Alineado a la derecha)

5. Ejercicio 4

6. Ejercicio 5

7. Ejercicio 6

8. Ejercicio 7

9. Ejercicio 8