

Nº Orden	Apellido y nombre	L.U.	Cantidad de hojas

Organización del Computador 2

Segundo parcial – 24/11/2011

1 (40)	2 (40)	3 (20)	
--------	--------	--------	--

Normas generales

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Entregue esta hoja junto al examen, la misma **no** se incluye en la cantidad total de hojas entregadas.
- Está permitido tener los manuales y los apuntes con las listas de instrucciones en el examen. Está prohibido compartir manuales o apuntes entre alumnos durante el examen.
- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con nombre, apellido y LU.
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Los parciales tienen tres notas: I (Insuficiente): 0 a 59 pts, A- (Aprobado condicional): 60 a 64 pts y A (Aprobado): 65 a 100 pts. No se puede aprobar con A- ambos parciales. Los recuperatorios tienen dos notas: I: 0 a 64 pts y A: 65 a 100 pts.

Ej. 1. (40 puntos)

1. (10 puntos) Describa cómo completaría las primeras entradas de la GDT en función de los segmentos que se detallan en la siguiente tabla. Los valores base y límite deben indicarse en hexadecimal.

	Comienzo	Tamaño	Tipo
1	0 Gb	1 Gb	datos usuario - escritura
2	512 Mb	16 Kb	código usuario - lectura
3	2 Gb	2 Gb	código kernel - no lectura
4	2.5 Gb	512 Mb	datos kernel - no escritura

2. (10 puntos) Especifique todas las entradas de las estructuras necesarias para construir un esquema de paginación según la siguiente tabla. Suponga que todas las entradas no mencionadas son nulas. Los rangos incluyen el último valor. Los permisos deben definirse como supervisor.

Rango virtual	Rango físico
0x90000000 a 0x90003FFF	0x80001000 a 0x80004FFF
0xA0000000 a 0xA0004FFF	0xE0002000 a 0xE0006FFF

3. (15 puntos) Resolver las siguientes direcciones, de lógica a lineal y a física. Utilizar las estructuras definidas en los ítems anteriores. Si se produjera un error de protección, indicar cuál error y en qué unidad.

- 0x008:0xA000010A
- 0x010:0x00000256
- 0x018:0x10003072
- 0x020:0x00002A00

4. (5 puntos) Se tiene implementado un sistema con segmentación sin solapamiento y sin paginación. Responder si la siguiente afirmación es verdadera o falsa.

- “Es posible modificar código o ejecutar datos.”

Justificar, indicar cómo es posible o argumentar como se impide.

Ej. 2. (40 puntos)

Se cuenta con un sistema operativo básico al cual se le desea agregar una llamada al sistema que permita crear subtareas. La misma se la quiere implementar a través de la **interrupt gate** número **60**. La llamada toma como parámetro la posición de memoria a partir de donde se va a empezar a

ejecutar la subtarea una vez creada. También toma como parámetro un dato de 32 bits que se le va a pasar a la subtarea. Estos parámetros se pasan a través de los registros `eax` y `ebx`, respectivamente. El prototipo de la llamada es el siguiente:

```
void crear_subtarea(unsigned int eip, unsigned int dato)
```

Ejemplo de uso:

```
int main() {
    ...
    crear_subtarea(imprimir_valor, 0xBABA)
    ...
}

void imprimir_valor(int valor)
{
    printf("valor: %d\n", valor);

    exit(0);
}
```

Se pide:

- (10 puntos) Describir los datos correspondientes de la entrada en la IDT para esta nueva interrupción. La misma se debe poder acceder desde el anillo de protección 3.
- (30 puntos) Programar en C la función `crear_subtarea` para ser llamada desde la nueva interrupción. La subtarea se debe ejecutar en el anillo de protección 3, con el mismo mapa de memoria que la tarea padre. La pila será una página libre que se debe mapear con identity mapping. Cuenta con la definición de las estructuras de `gdt_entry`, `tss` y las siguientes funciones:

- `dame_tss()`: Retorna el puntero a una TSS vacía.
- `dame_cr3()`: Retorna el `cr3` de la tarea actual.
- `dame_gdt()`: Retorna el índice de una entrada libre en la GDT.
- `dame_base_gdt()`: Retorna la dirección base de la GDT.
- `dame_pagina_libre_usuario()`: Retorna la dirección física de una página libre de usuario.
- `mapear_pagina(dir_virtual, dir_fisica, cr3, permisos)`: Realiza el mapeo entre la dirección virtual y la dirección física. Además, setea los permisos (bits 0 a 2).
- `desmapear_pagina(dir_virtual, cr3)`: Elimina el mapeo entre la dirección virtual y la dirección física.
- `dame_segmento_codigo_usuario`: Retorna índice en la GDT donde se encuentra el segmento de código de usuario.
- `dame_segmento_datos_usuario`: Retorna índice en la GDT donde se encuentra el segmento de código de datos.
- `agregar_a_scheduler(indice_gdt)`: Agrega una tarea, identificada por el índice en la GDT de su TSS, al scheduler.

Nota:

- La rutina creada se ejecuta en nivel 3 y SOLO puede ser interrumpida por rutinas a nivel 1 y 0.
- Todos los selectores de segmento de datos se setean con el mismo segmento de datos.

Ej. 3. (20 puntos)

Dada una tabla GDT con las siguientes **únicas entradas válidas**:

Indice	Base	Limite	DPL	G	Tipo
5	0x00005000	0x0005	10	1	datos - solo lectura
8	0x00004444	0x1111	00	0	código - solo ejecución
9	0x00008000	0xFFFF	11	1	código - solo ejecución

Determinar para cada una de siguientes direcciones lógicas a que direcciones físicas se accede. Indicar además cual es el nivel de protección efectivo para cada acceso. En el caso que se produzca un error, indicar cual fue y por qué se generó.

- 0x03:0x000001FF - CPL=11 - lectura
- 0x30:0x00005555 - CPL=00 - escritura
- 0x43:0x00008220 - CPL=11 - ejecución
- 0x40:0x00005378 - CPL=00 - escritura
- 0x48:0x00010001 - CPL=01 - ejecución
- 0x29:0x10101010 - CPL=10 - lectura