

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

Organización del Computador 2

Recuperatorio del segundo parcial — 6/12/16

1 (30)	2 (40)	3 (30)	
--------	--------	--------	--

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. (30 puntos)

Se tiene la siguiente tabla GDT:

Indice	Base	Límite	DB	S	P	L	G	DPL	Tipo
1	0x00FAFAFA	0x15000	1	1	1	0	1	11	0x2
2	0x30300000	0x10000	1	1	1	0	0	00	0x8
3	0xBABAF0FA	0x10000	1	1	1	0	1	11	0x2
4	0x30350000	0x0FFFF	1	1	1	0	1	00	0xA

Y el siguiente esquema de paginación:

Rango Lineal	Rango Físico	Atributos
0x00FC4000 - 0x00FC4FFF	0x00001000 - 0x00001FFF	read/write, supervisor
0x30D00000 - 0x30DFFFFF	0x12300000 - 0x123FFFFF	read only, supervisor
0xBABDF000 - 0xBABE0FFF	0x00870000 - 0x00871FFF	read only, user

(10p) a. Especificar todas las entradas de las estructuras necesarias para construir un esquema de paginación. Suponer que todas las entradas no mencionadas son nulas.

(20p) b. Resolver las siguientes direcciones, de lógica a lineal y a física. Utilizar las estructuras definidas y suponer que cualquier otra estructura no lo está. Si se produjera un error de protección, indicar cuál error y en qué unidad. Definir EPL en los accesos a datos. El tamaño de todas las operaciones es de 2 bytes.

- I - 000B:00015100 - CPL 11 - escritura
- II - 0013:00A3F100 - CPL 00 - ejecución
- III - 001B:00030000 - CPL 11 - escritura
- IV - 0020:00002200 - CPL 00 - lectura

Ej. 2. (40 puntos)

Considerar 3 tareas en nivel 3 que corren concurrentemente una a una. Las tres tareas comparten un área de memoria la cual todas las tareas pueden leer y escribir. Para evitar que una tarea escriba sin que otra haya terminado se propone el siguiente mecanismo. Antes de escribir en el área compartida, la tarea llama al servicio `permiso`, que permite que la tarea escriba en dicha área. Luego, cuando finaliza, llama al servicio `listo`, indicando que terminó de escribir en el área compartida. Considerar que si una tarea esta usando el área compartida, la tarea que llame a `permiso` se quedará bloqueada esperando que el área compartida se libere.

- (10p) 1. Proponer una posible implementación del sistema descripto.
- (10p) 2. Implementar en ASM el código de la rutina de atención de interrupciones del reloj.
- (20p) 3. Implementar en ASM el código de los servicios `listo` y `permiso`

Nota: Se pueden hacer funciones auxiliares en C.

Ej. 3. (30 puntos)

En un extraño sistema con 2 niveles de privilegio corriendo en el espacio, dos tareas corren concurrentemente. Las mismas verifican periódicamente el estado de sus registros. En caso de detectar alguna anomalía, la tarea solicita al sistema ser reiniciada al último estado correcto conocido, luego de lo cual debe seguir corriendo. Se utiliza identity mapping para todo el sistema y segmentación flat.

- (5p) 1. Indicar como implementaría el mecanismo descripto.
- (15p) 2. Implementar en ASM el código de la rutina que reinicia la tarea, asumiendo que la memoria no presenta anomalías.
- (10p) 3. Implementar los cambios necesarios para que la rutina escriba 0s en todas las páginas **de la tarea** (salvo en la página `CODE_PAGE`). Se cuenta con la función `write_zeros(int count, void *addr)`.

Nota: Se pueden hacer funciones auxiliares en C.