

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

Organización del Computador 2

Recuperatorio del segundo parcial – 10 de julio de 2014

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)

Considerando el siguiente modelo para la administración de memoria, Segmentación:

Indice	Desde	Tamaño	G	Permisos	Tipo
7	10 Kb	1Gb	1	nivel 0	Código - lectura
18	20 Kb	1025 bytes	0	nivel 3	Código - lectura
21	50 Kb	512 bytes	0	nivel 0	Datos - lectura/escritura
33	16 Kb	4 Kb	1	nivel 3	Datos - lectura/escritura

Paginación:

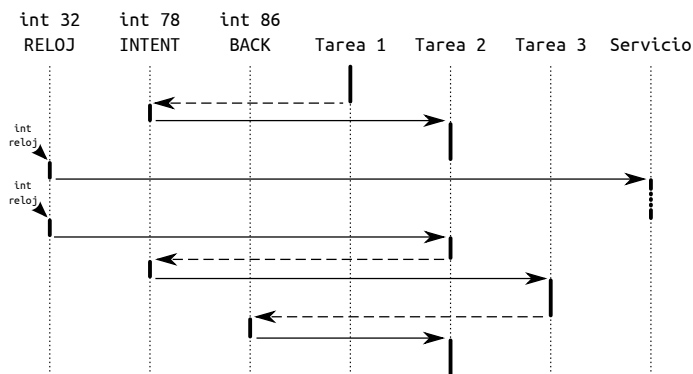
Rango Lineal	Rango Físico	permisos
0x0000C000 a 0x0000CFFF	0x01010000 a 0x01010FFF	root y rw
0x00002000 a 0x00005FFF	0x00FFF000 a 0x01002FFF	user y rw

- a) (10p) Describir cómo se completarían las entradas de la GDT. Indicar los campos Índice, Base, Limite, S, D/B, AVL, L, P, DPL y Tipo. Los valores de Base y Limite deben indicarse en hexadecimal.
- b) (10 puntos) Especificar todas las entradas de las estructuras necesarias para construir el esquema de paginación según presentado. Suponer que todas las entradas no mencionadas son nulas. Los rangos incluyen el último valor.
- c) (10 puntos) Resolver las siguientes direcciones, de lógica a lineal y a física. Utilizar las estructuras definidas en los ítems anteriores y suponer que cualquier otra estructura no está definida. Si se produjera un error de protección, indicar cuál error y en qué unidad. Definir EPL en todos los casos.

- I - 0x0038:0x00000135 - CPL 00 - ejecución
- II - 0x010F:0x00000AAA - CPL 11 - escritura
- III - 0x0FF1:0x60606060 - CPL 10 - ejecución
- IV - 0x0108:0x00000AAA - CPL 00 - lectura
- V - 0x0090:0x00000402 - CPL 00 - escritura

Ej. 2. (40 puntos)

Considerar un sistema capaz de ejecutar las siguientes 4 tareas, Tarea_1, Tarea_2, Tarea_3 y Tarea_Servicio. Por cada ciclo de reloj se ejecutará *SOLO* una de las tres tareas o el servicio. Las tareas podrán ejecutar dos servicios del sistema, “intent” en la interrupción 78 y “back” en la interrupción 86. El servicio “intent” permite saltar a la siguiente tarea de la lista de tareas, por ejemplo, si se esta ejecutando la tarea 2, se pasará a ejecutar la tarea 3. El servicio “back” produce el efecto contrario, de volver a la tarea desde donde fue ejecutado el “intent”. El orden de las tareas es fijo y corresponde a los numeros de las mismas. El sistema comienza ejecutando la tarea 1 y el servicio.



En la figura se muestra un ejemplo donde se ejecutan dos “intent” para saltar de la tarea 1 a la 2 y de la tarea 2 a la 3. Además, se ejecuta concurrentemente el servicio durante un ciclo de reloj.

- (5p) Complete las entradas en la IDT para interrupción 32, 78 y 86
- (10p) Considerando que las tareas que corren en nivel 3 y el servicio en nivel 0, describir las estructuras de datos necesarias para poder implementar el sistema, teniendo en cuenta que se utiliza además paginación y segmentación *flat*.
- (10p) Escribir el código ASM de la interrupción de reloj.
- (15p) Escribir el código ASM de las interrupciones 78 y 86.

Nota: de ser necesario es posible implementar funciones auxiliares en C.

Ej. 3. (30 puntos)

Un sistema operativo tiene un extraño sistema de comunicación entre tareas para un aun más extraño servicio a implementar. Cada tarea tiene mapeada una página especial como solo lectura, cuando esta página intenta ser escrita por la tarea se produce una *page fault*. Como resultado de esta interrupción, el sistema debe escribir simultáneamente en las paginas especiales de todas las tareas un byte (0x55) en el *offset* correspondiente a la posición que produjo la falla. En la tarea que se produjo la falla no se debe escribir ningún byte.

- (7p) Implementar en C la función `mapear_pagina`, que dado un CR3, una dirección virtual y una física, mapea esta ultima a la dirección virtual sobre el CR3 dado.
- (7p) Implementar una función en C que dada una dirección virtual y un CR3 resuelva la dirección física correspondiente a la dirección virtual sobre el CR3 indicado.
- (16p) Implementar en ASM el *handler* de la interrupción de *page fault*, considerando que el sistema tiene **solo 2 tareas**, donde:
 - Tarea 1 : dirección virtual de la página especial = `ADDR_PAGE_1` y `CR3 = CR3_1`
 - Tarea 2 : dirección virtual de la página especial = `ADDR_PAGE_2` y `CR3 = CR3_2`

Nota: Las tareas escriben en su página especial ejecutando una instrucción de `SIZE_WRITE` cantidad de bytes. Se cuenta con un par de funciones que permiten obtener el valor del registro CR3 y CR2. Además de dos funciones para obtener paginas libres, virtuales y físicas. **Es válido hacer funciones auxiliares en C.**