

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

Organización del Computador 2

Recuperatorio del Segundo parcial – 02-12-2014

1 (30)	2 (50)	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. (30 puntos)

Se tiene la siguiente tabla GDT:

Indice	Base	Límite	DB	S	P	L	G	DPL	Tipo
0x24	0x00010000	0x0000F	1	1	1	1	1	3	0xA
0x36	0x00020000	0x0000E	1	1	1	1	1	3	0x2
0x42	0x00030000	0x0000D	1	1	1	1	1	0	0x8
0x53	0x00040000	0xFFFFF	1	1	1	1	0	0	0x2

Y el siguiente esquema de paginación:

Rango Lineal	Rango Físico	Atributos
0x00014000 a 0x00015FFF	0x09002000 a 0x09003FFF	read/write, level 0
0x00023000 a 0x00024FFF	0xBB006000 a 0xBB007FFF	read/write, level 3
0x11041000 a 0x11041FFF	0xBB007000 a 0xBB007FFF	read/write, level 3

(12p) (a) Especificar todas las entradas de las estructuras necesarias para construir un esquema de paginación. Suponer que todas las entradas no mencionadas son nulas.

(18p) (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 todos los casos. El tamaño de todas las operaciones es de 2 bytes.

- I - 0x0123:0x00004DAD - CPL 11 - lectura
- II - 0x01B3:0x00003BAD - CPL 11 - lectura
- III - 0x0210:0x10000FAD - CPL 00 - ejecución
- IV - 0x0298:0x11001FA6 - CPL 00 - escritura
- V - 0x01B3:0x000035AD - CPL 11 - ejecución
- VI - 0x0210:0x100006A1 - CPL 00 - lectura

recordar:

64B=0x40, 4KB=0x1000, 65536B=64KB, 1MB=0x100000,
 1GB=0x40000000, 2GB=0x80000000, 3GB=0xC0000000, 4GB=0x100000000.

Ej. 2. (50 puntos)

A diferencia de los siete enanitos, Blancanieves tiene piernas más largas, por lo que logra caminar más rápido. Suponiendo que tenemos 8 tareas, una denominada Blancanieves y el resto como los enanitos, se pide desarrollar un sistema minimal que ejecute tareas concurrentemente en dos niveles de protección. La tarea Blancanieves (por tener piernas mas largas) requiere de la mitad de tiempo que los enanitos, es decir, a cada tarea enanito se le asigna el doble de tiempo que a Blancanieves. Cada tanto un enanito se cansa y pide que lo esperen. Para esto, la tarea llama a la interrupción `_esperame` que se encarga de darle a la tarea algo más de tiempo de procesador. Pero ojo! la tarea enanito solo puede llamar a esta interrupción una vez durante su turno asignado por el procesador. De hacerlo mas veces, no surgirá ningún efecto.

- (5p) (a) Indicar todas las estructuras de datos y de sistema necesarias para construir lo pedido, justificando el uso de cada una.
- (10p) (b) Instanciar en un caso particular todas las estructuras mencionadas en el punto anterior. Se debe completar con valores numéricos o indicando referencias. Recuerde indicar en detalle el nivel de protección de cada estructura.
- (20p) (c) Implementar en (C/ASM) la rutina de la interrupción de reloj en ASM/C.
- (15p) (d) Implementar en (C/ASM) la rutina de la interrupción `_esperame` en ASM/C.

Ej. 3. (20 puntos)

Se tiene un sistema en dos niveles de privilegio con segmentación y con paginación no activada, que puede ejecutar concurrentemente un conjunto limitado de tareas. Estas se encuentran declaradas en un arreglo de *TSS* definidas en la GDT.

- (10p) (a) ¿Es posible determinar en cualquier momento, con la información provista por las estructuras de datos del sistema, la cantidad de tareas que ejecutan su código en nivel cero? Si es posible, construya una función que lo determine, en caso contrario justifique detalladamente.
- (10p) (b) Suponiendo que el rango de direcciones del kernel es el correspondiente al primer MB de memoria, construir una función que dado el puntero a la GDT y un índice, determine si el descriptor de la TSS corresponde a un descriptor de TSS donde su próxima instrucción es de kernel.