

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

Organización del Computador 2

Segundo parcial – 23/06/2015

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. (30 puntos)

- a- (8 puntos) Describir cómo se completarían las entradas de la GDT en función de los segmentos que se detallan en la siguiente tabla. Indicar los campos **Indice**, **Base**, **Limite**, **S**, **G**, **D/B**, **AVL**, **L**, **P**, **DPL** y **Tipo**. Los valores de **Base** y **Limite** deben indicarse en hexadecimal.

Indice	Desde	Tamaño	Permisos	Tipo
1	10 MB	1 MB	nivel 0	Código - sólo ejecución
3	512 MB	16 KB	nivel 3	Código - lectura
4	0 Gb	4 Gb	nivel 1	Datos - sólo lectura
5	2 Gb	1,5 Gb	nivel 3	Datos - lectura/escritura

- b- (8p) Especificar todas las entradas de las estructuras necesarias para construir un esquema de paginación particular según la siguiente tabla. Suponer que todas las entradas no mencionadas son nulas. Los rangos incluyen el último valor.

Rango Lineal	Rango Físico
0x00A00000 a 0x00A03FFF	0x0001A000 a 0x0001DFFF
0xBABA2000 a 0xBABA5FFF	0xF0F03000 a 0xF0F06FFF

- c- (8p) 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. Calcular y definir EPL en todos los casos.

- I - 0x08:0x00000021 - CPL 00 - ejecución
- II - 0x2B:0xBABA2015 - CPL 00 - escritura
- III - 0x12:0xBABA2001 - CPL 11 - lectura
- IV - 0x20:0xBABA6034 - CPL 01 - lectura
- V - 0x0B:0x00000010 - CPL 00 - ejecución
- VI - 0x08:0xBBAF5F5 - CPL 00 - ejecución

- d- (6p) ¿Qué pasaría si en el ítem 1.c.v) el segmento al que se accede fuera de tipo *conforming*?

Ej. 2. (40 puntos)

Sean tres **funciones** denominadas A , B y C que respetan convención C. Estas deben ser ejecutadas como tareas en un sistema de dos niveles de protección. Las funciones serán ejecutadas hasta que finalicen, una tras otra, repetidamente y de forma cíclica. En caso de que se produzca una excepción, la tarea debe ser desalojada y reiniciada desde el principio. Lo mismo debe suceder si la tarea supera el límite de K ticks de reloj como máximo para su ejecución. Tener en cuenta que el código de las tareas debe estar protegido contra escritura y no debe poder ser leído entre tareas.

- a- (15p) Describir todas las estructuras involucradas en el sistema, instanciarlas con datos y explicar su funcionamiento. Explicar paso a paso el funcionamiento del *scheduler* de tareas, desde que se ejecuta una tarea, hasta que ésta finaliza **correctamente**.
- b- (15p) Escribir el código de la rutina de atención de interrupciones del reloj y todo el código que se requiera para completar un ciclo de ejecución de una tarea.
- c- (10p) Escribir el código una rutina de atención para excepciones.

Nota: Cualquier código pedido debe estar escrito en C o ASM. Considerar que se dispone de las definiciones de las estructuras del procesador en C.

Ej. 3. (30 puntos)

Se tiene un sistema con segmentación *flat* y paginación activada. Teniendo en cuenta que un sistema de paginación puede pensarse en general como una **función** de *direcciones lineales a direcciones físicas*:

- a- (15p) Escriba en C un programa que decida, dado un CR3, si la función subyacente a un esquema de paginación dado es inyectiva o no.

La aridad de la función a programar es: `bool esInyectiva(int cr3)`

- b- (15p) Escriba en C un programa que decida, dado un CR3, si la función subyacente a un esquema de paginación dado es sobreyectiva o no.

La aridad de la función a programar es: `bool esSobreyectiva(int cr3)`

Para resolver el ejercicio, es valido usar *page_directory_entry*, *page_table_entry* como estructuras para acceder a cada campo. Además, se cuenta con las siguientes funciones del tipo de datos “conjunto de enteros”.

- `set set()`: que devuelve un conjunto (`set`) vacío de *Ints*.
- `void add(set s, int a)`: que inserta el `int a` en el `set s`.
- `bool is(set s, int a)`: que devuelve `true` si el `int a` pertenece al conjunto `s`.

Nota: Considerar como función subyacente de un esquema de paginación, a las direcciones que son accesibles como lectura/escritura a nivel de usuario.

- Una función de $X \rightarrow Y$ es **inyectiva** si:

$$\forall x_1, x_2 \in X, (x_1 \neq x_2) \Rightarrow (f(x_1) \neq f(x_2))$$
- Una función de $X \rightarrow Y$ es **sobreyectiva** si:

$$\forall y \in Y, \exists x \in X / f(x) = y$$

Hint: Piense qué función auxiliar podría hacer, que se reutilice en a) y en b).