

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

## Organización del Computador 2

### Segundo parcial – 18-11-2014

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
0x04	0x0000000	0xFFFFFFF	1	1	1	0	0	0	0xA
0x07	0x0000000	0x00A00	1	1	1	0	1	0	0x2
0x10	0x0100000	0x00A00	1	1	1	0	1	3	0x8
0x31	0x0100000	0x00A00	1	1	1	0	1	3	0x2

Y el siguiente esquema de paginación:

Rango Lineal	Rango Físico	Atributos
0x00041000 a 0x00046FFF	0x05002000 a 0x05007FFF	read/write, level 0
0x001A0000 a 0x001A3FFF	0xA0001000 a 0xA0004FFF	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 - 0x0020:0x00043721 - CPL 00 - lectura
- II - 0x0038:0x00041FAA - CPL 00 - lectura
- III - 0x0083:0x000A2133 - CPL 11 - ejecución
- IV - 0x018B:0x000A1FFF - CPL 11 - escritura
- V - 0x0080:0x000A3133 - CPL 00 - ejecución
- VI - 0x003B:0x00041FAA - CPL 11 - lectura

**recordar:**

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

## Ej. 2. (50 puntos)

El sistema de navegación principal de la *International Space Station* (ISS) opera bajo un *80386DX*. Este sistema debe responder a tres funciones básicas (*ccg*, *svi* y *car*) que se implementan como tareas de nivel 3. Las tres tareas serán ejecutadas concurrentemente asignando la misma cantidad de tiempo a cada una. Como cada una de estas tareas son vitales para la supervivencia en el espacio, cualquier excepción debe ser capturada y la tarea debe ser reiniciada. Además, existe un conjunto de interrupciones **externas** que obliga al sistema a ejecutar alguna de las tres tareas de forma inmediata. Estas interrupciones son *\_ccg*(36), *\_svi*(37) y *\_car*(38).

- (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 numericos o indicando referencias. Recuerde indicar en detalle el nivel de protección de cada estructura.
- (25) (c) Implementar en (C/ASM) la rutina de atención de la interrupción de reloj y también la rutina de atención de una excepción cualquiera.
- (10) (d) Implementar en (C/ASM) la rutina de la interrupción *\_ccg*.

Se cuenta con la función `resetTask(tss* t, int n)` que resetea el estado de la *tss* pasada por parámetro al estado de la tarea indicada por *n*. Estos pueden ser *ccg*= 1, *svi*= 2 y *car*= 3.

## Ej. 3. (20 puntos)

Se tiene un sistema en dos niveles de privilegio con segmentación flat y con paginación no activada.

- (5p) (a) Construir una función en ASM que obtenga el puntero a la IDT y el puntero a la GDT.
- (15p) (b) Construir en ASM o C una función `existe()` que determine si existe alguna interrupción que sólo pueda ser llamada desde nivel supervisor pero que se ejecuta en nivel de usuario. Considere que la IDT contiene solamente entradas de tipo interrupt gate.