

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

## Organización del Computador 2

### Segundo parcial – 26/06/2012

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.
- No se permite el uso de dispositivos electrónicos que no pertenezcan al siguiente conjunto: { calculadora }.
- 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. (40 puntos)

1. (8 puntos) Describa cómo completaría las primeras entradas de la GDT en función de los segmentos que se detallan en la siguiente tabla. Los valores base y límite deben indicarse en hexadecimal.

Indice	Desde	Hasta	Permisos	Tipo
1	0 Gb	3,5 Gb	00	código kernel - lectura
2	2 Mb	4 Gb	01	datos kernel - no escritura
3	13 Kb	17 Kb	10	código usuario - no lectura
4	3 Mb	5 Mb	11	datos usuario - escritura

2. (12 puntos) Especifique todas las entradas de las estructuras necesarias para construir un esquema de paginación según la siguiente tabla. Suponga que todas las entradas no mencionadas son nulas. Los rangos incluyen el último valor. Los permisos deben definirse como supervisor.

Rango Lineal	Rango físico
0xABCFE000 a 0xABD02FFF	0xBDFFE000 a 0xBE002FFF
0x00000000 a 0x00003FFF	0x033E0000 a 0x033E3FFF

3. (12 puntos) Resolver las siguientes direcciones, de lógica a lineal y a física. Utilizar las estructuras definidas en los ítems anteriores. Si se produjera un error de protección, indicar cuál error y en qué unidad.

- 0x008:0xE0000005 - CPL 00 - ejecución
- 0x009:0xABCFE000 - CPL 00 - lectura
- 0x010:0xABAFE222 - CPL 01 - lectura
- 0x01B:0x00000800 - CPL 11 - lectura
- 0x023:0x00000000 - CPL 10 - escritura
- 0x033:0x00000010 - CPL 11 - ejecución

4. (8 puntos) Suponiendo paginación a 4kb y un CR3 válido. ¿Es posible activar paginación configurando solo una entrada de la tabla de páginas y una entrada del directorio de páginas?. ¿Cuál entrada configuraría?. Justificar.

## Ej. 2. (40 puntos)

En un nuevo y moderno sistema operativo se desea permitir a las tareas trabajar en equipo. En este sistema las tareas son ejecutadas una a una durante intervalos de tiempo fijos (quantum). El quantum es global para todas las tareas y no puede modificarse. En cualquier momento, una tarea puede decidir delegar su tiempo (quantum) restante de procesador a otra tarea, ya sea la siguiente o una tarea determinada. Esta acción se realiza una sola vez. En la siguiente ejecución de la tarea, esta recupera su quantum totalmente. Si lo desea, puede volver a delegarlo.

El sistema permite hacer pedidos a través de la interrupción 66. Los posibles pedidos son:

mensaje	delegar_siguiente	delegar_tarea
código(eax)	101	202
argumento(ebx)		id_tarea

1. (10 puntos) Mencione que estructura de datos utilizaría para almacenar la lista de tareas y que funciones posee para su manipulación.
2. (15 puntos) Escriba en C las funciones **delegar\_tarea(int id)** y **delegar\_siguiente()**, que como su nombre lo indica, delegan el procesador a la tarea **id** o a la siguiente tarea respectivamente.
3. (15 puntos) Indique como se deberá configurar la IDT para cumplir con los requerimientos. Además escriba en ASM la int 66 utilizando las funciones anteriores, **delegar\_tarea(int id)** y **delegar\_siguiente()**.

## Ej. 3. (20 puntos)

En otro sistema operativo, más nuevo y más moderno aún, se ha decidido utilizar un mecanismo de manejo de memoria implícito, también más moderno. Esto quiere decir que, cuando una tarea requiere una página de memoria para utilizarla, elige directamente una dirección de memoria a su gusto y la utiliza, sin avisarle a nadie que lo hará.

Para poder funcionar de esta forma, el sistema operativo utiliza la unidad de paginación que provee la arquitectura. Al detectar la intención de utilizar una página nueva por parte de una tarea, el sistema utiliza el fallo de página (#PF) para realizar las modificaciones correspondientes y mapear una nueva página al lugar solicitado.

1. (8 puntos) Describa la entrada en la IDT necesaria para la rutina de atención de interrupción de fallo de página (#PF).
2. (12 puntos) Escriba el código necesario para la rutina de atención para la interrupción de fallo de página (#PF). En caso de haber memoria física disponible, la interrupción debe mapear una nueva página al lugar solicitado para que la tarea la utilice. Puede asumir que la tarea corre en anillo 0 y siempre hay memoria física disponible.

Notas:

- a) Se puede consultar *6.15 Exception and Interrupt Reference* para saber dónde se guarda la dirección que produjo el fallo y los motivos por los que se produjo.
- b) Se puede asumir que cuenta con las siguientes funciones:
  - **pagina\_libre(void)** que devuelve la dirección física para una nueva página libre.
  - **mapear\_pagina(unsigned int virtual, unsigned int fisica, unsigned int basePD)** que crea un mapeo entre una dirección *virtual* y una *fisica* a partir de *basePD*.
  - **tlbflush()** que realiza un *flush* de la TLB.