

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

Organización del Computador 2

Segundo parcial – 19/11/2015

1 (30)	2 (45)	3 (25)	
--------	--------	--------	--

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
1	0x00000000	0x00800	1	1	1	0	0	0	0x8
2	0xF0000000	0xFFFF	1	1	1	0	1	2	0xB
3	0x00120000	0x00000	1	1	1	0	1	3	0x2
4	0x001FFFFFF	0x00100	1	1	1	0	1	2	0x0

Y el siguiente esquema de paginación:

Rango Lineal	Rango Físico	Atributos
0x00000000 - 0x00000FFF	0x00010000 - 0x00010FFF	read only, supervisor
0xF0F55000 - 0xF0F57FFF	0x00022000 - 0x00024FFF	read/write, supervisor
0x00123000 - 0x00124FFF	0x00005000 - 0x00006FFF	read/write, user

(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 los accesos a datos. El tamaño de todas las operaciones es de 2 bytes.

- I - 0x08:0x000008FF - CPL 00 - ejecución
- II - 0x13:0x00020000 - CPL 01 - lectura
- III - 0x1B:0x00005000 - CPL 11 - escritura
- IV - 0x10:0x00F55555 - CPL 10 - lectura
- V - 0x20:0x00004200 - CPL 00 - escritura
- VI - 0x21:0x000EA4E0 - CPL 01 - lectura

recordar:

4KB=0x1000, 64KB=0x10000, 1MB=0x100000, 4MB=0x400000,
1GB=0x40000000, 2GB=0x80000000, 3GB=0xC0000000, 4GB-1=0xFFFFFFFF.

Ej. 2. (45 puntos)

Se desea implementar un sistema multi-tareas semi-colaborativo. El mismo cuenta con un kernel que hace correr concurrentemente a 8 tareas de usuario independientes. Las mismas no deben poder leer ni escribir la memoria de las demás. Las mismas corren hasta que pasen K ciclos o decidan ceder el control. Además, cuando el usuario presione la tecla A se le otorgarán K ciclos extra a la tarea actual (solamente por el período de ejecución actual). Cuando se agota el tiempo, el scheduler pasa a la siguiente tarea. Para ceder el control voluntariamente, en cambio, las tareas utilizarán el syscall 55, indicando en EAX cuál quieren que sea la próxima tarea. Asumir que nunca se generan excepciones y que una tarea no puede cederse el control a sí misma.

- (15p) a. Detallar los campos relevantes de todas las estructuras involucradas en el sistema para manejar segmentación, tareas, interrupciones y privilegios. Instanciar las estructuras con datos y explicar su funcionamiento. Describir el esquema de segmentación que se utilizará así como también el de paginación si es que lo utiliza.
- (20p) b. Escribir el código de las rutinas de atención de interrupciones del reloj y el syscall 55.
- (10p) c. Escribir el código de la rutina de atención de teclado.

Nota: Cualquier código pedido debe estar escrito en C o ASM. Asumir que se dispone de las definiciones de las estructuras del procesador en C. El scancode de la tecla A es 0x1E y el puerto del teclado es el 0x60.

Ej. 3. (25 puntos)

Se tiene un sistema ya funcionando con segmentación *flat* y paginación activada donde corren 8 tareas. Se desea que éstas puedan comunicarse con memoria compartida. Para tal fin, se agregará al sistema el syscall 60, que devuelve a la tarea usuario la dirección de una página para lectura o escritura. Esta syscall recibirá en EAX si es lectura o es escritura y en EBX el ID de la tarea destino, y devolverá en EAX la dirección de la página otorgada. Para implementar el manejo se escribirá una rutina ASM básica que maneje la interrupción y llame a otra de más alto nivel escrita en C que realice lo deseado.

- (4p) a. Describir que hay que agregar o cambiar al sistema para implementar el syscall.
- (6p) b. Escribir el código ASM de manejo del syscall 60.
- (15p) c. Escribir el código de la rutina `void* mapear(uint es_lectura, uint id_destino)`.

Para resolver el ejercicio, se cuenta con algunas rutinas:

- `int tarea_actual()` que devuelve el ID de la tarea actual.
- `int mapear_pagina(uint virtual, uint fisica, uint cr3, uint atributos)`.
- `void* dame_fisica_libre()` que devuelve y asigna una dirección de una pagina libre en memoria fisica.
- `void* dame_virtual_libre(uint tarea)` que devuelve y asigna una dirección de una pagina libre en la memoria virtual de la tarea especificada.

Puede asumir que cuando una tarea pide para lectura de otro proceso, este otro ya pidió para escritura antes. Además, sólo se puede compartir una página por cada par de tareas.