

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

Organización del Computador 2

Recuperatorio del segundo parcial – 7 de Julio de 2015

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
0x01	0x003F0000	0x0001D	1	1	1	1	1	0	0x8
0x07	0x00000000	0x00402	1	1	1	1	1	3	0xA
0x10	0x00020000	0x01000	1	1	1	1	1	3	0x2
0x11	0x00300000	0x0FFFF	1	1	1	1	0	0	0x2

Y el siguiente esquema de paginación:

Rango Lineal	Rango Físico	Atributos
0x003FF000 a 0x000400FFF	0x00021000 a 0x00022FFF	read/write, level 0
0x00000000 a 0x00001FFF	0x00021000 a 0x00022FFF	read/write, level 3
0x00023000 a 0x00024FFF	0x003FE000 a 0x003FFFFF	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 los accesos a datos. El tamaño de todas las operaciones es de 2 bytes.

- I - 0x08:0x10000FAD - CPL 00 - ejecución
- II - 0x38:0x003F4DAD - CPL 11 - lectura
- III - 0x8B:0x00003FA6 - CPL 11 - escritura
- IV - 0x89:0x00003BAD - CPL 00 - escritura
- V - 0x83:0x000035AD - CPL 11 - ejecución
- VI - 0x80:0x002206A1 - CPL 00 - lectura

recordar:

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

Ej. 2. (50 puntos)

En un complejo sistema se desea ejecutar K tareas de usuario concurrentemente. Existe una política de prioridades basada en el uso del teclado. Inicialmente las tareas corren con un quantum determinado. Las tareas pueden preguntar cuál fue la última tecla presionada mediante la interrupción `0x72`, que devuelve el resultado en el registro `eax`. Un intento de lectura es fallido si no se presionó ninguna tecla o si ésta ya fue leída, y exitoso en caso contrario. Los intentos fallidos causan el desalojo de la tarea e inmediato salto a la próxima. Si el intento es exitoso se continúa la ejecución de la tarea que interrumpió. Por cada intento de lectura fallido se le restará 1 al quantum de la tarea. Por cada intento exitoso se le sumará 1. Para los intentos fallidos se devuelve 0.

- (10p) (a) Describir todas las estructuras involucradas en el sistema. Indicar cómo deben inicializarse todas las estructuras del procesador y las variables del kernel para poner en funcionamiento este esquema. Explicar el funcionamiento general del sistema. Explicar paso a paso el funcionamiento del *scheduler*.
- (10p) (b) Escribir el código de la rutina de atención de interrupciones del reloj.
- (10p) (c) Escribir el código de la rutina de atención de interrupciones del teclado.
- (20p) (d) Escribir el código de la `_isr72`.

Nota: Pueden disminuir los quantums hasta que sólo 1 tarea quede ejecutando, pero no menos.

Ej. 3. (20 puntos)

Nos informan desde la NASA que a un procesador de la EEI¹ se le ha roto la instrucción `ltr`. Se desea reemplazar a la misma por una rutina equivalente a esta instrucción. Sabemos que el sistema usa segmentación flat e identity mapping de toda la memoria. No deben agregarse ni quitarse tareas. No se cuenta con las direcciones de variables globales como `gdt[]`, `tsss[]`, etc. Usando las funciones auxiliares descritas abajo se pide:

- (6p) (a) Escribir en ASM la función `int direccion_gdt()` que devuelve la dirección lineal de la GDT, y en C la función `void copiar_tss(ushort selector, tss* buffer)` que copia el contenido de la TSS indicada por selector al buffer.
- (14p) (b) Escribir una sección de código ASM que sea análoga a la instrucción `ltr selector`. Usar la pila como buffer para guardar y restablecer las TSS. Al finalizar el código de la rutina las TSS origen y destino deben quedar exactamente igual que si se hubiera ejecutado `ltr`. Los registros `EAX`, `ECX` y `EDX` pueden modificarse a gusto.
- `memcpy(void *dst, void *src, int size)`,
 - `void tocar_tss(ushort selector, uint nuevo_eip, uint nuevo_esp)` que dado un selector de TSS cambia el contenido de su EIP y ESP,
 - `void guardar_estado_actual(tss_t *tss)` que copia el estado actual de los registros a una TSS.

¹Estacion Espacial Internacional