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

Organización del Computador 2
Recuperatorio del segundo parcial – 3/12/2019

1 (24)	2 (40)	3 (36)	

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. (24 puntos)

(24p) a. Considerando la siguiente tabla de traducciones de direcciones por segmentación y paginación. De ser posible, dar un conjunto de descriptores de segmento, directorio de paginas y tablas de paginas que cumplan con todas las traducciones **simultáneamente**. Detallar los campos de todas las estructuras involucradas. Además indicar si la traducción es *identity mapping* (tanto en segmentación como paginación) y en el caso de que alguna traducción no sea posible explicar por qué.

Lógica	Lineal	Física	Características
`0x0192:0x00001235`	`0x010433F0`	`0x00A233F0`	Escritura de 1MB, como nivel 2
`0x0223:0x00000543`	`0x00100A98`	`0x00100A98`	Lectura de 8KB, como nivel 3
`0x0098:0x00450011`	`0x00450011`	`0x04328011`	Escritura de 16KB, como nivel 0
`0x0223:0x001F3422`	`0x002F3977`	`0x0AA45977`	Ejecución de 8KB, como nivel 3

Ej. 2. (40 puntos)

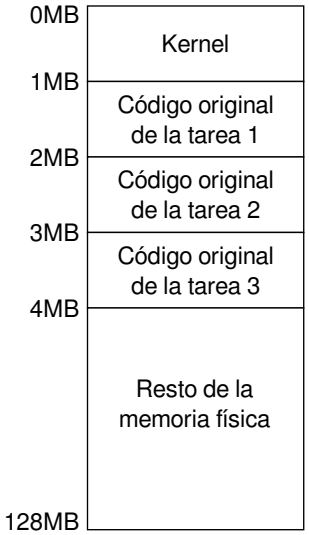
Se cuenta con un sistema de navegación de un avión caza Lockheed F-117 Nighthawk corriendo sobre un procesador Intel 80386.

El mismo cuenta con tres tareas denominadas A, B y C[1]. Las tareas ocupan 1MB de memoria en total entre código, datos y pila. Las mismas se deben ejecutar de forma independiente en dos niveles de protección. Además, el sistema debe poder mapear las tareas a partir de los 10MB, en cualquier posición de memoria virtual. Originalmente el mapa de memoria física es el indicado por la figura.

Como es un sistema crítico este no puede fallar, y por está razón las tareas deben ser reiniciadas en el caso de que alguna genere una excepción. Para ello, se debe copiar el código original de la tarea, y ejecutarlo inmediatamente. Además, una vez reiniciada, la tarea debe conocer la cantidad de veces que fue reiniciada. Tener en cuenta que las tareas deben comenzar desde un estado conocido, ya que cualquier registro puede estar roto al intentar reiniciarlas.

Las tareas entre sí son ejecutadas concurrentemente, pero pueden requerir que otra tarea sea puesta a ejecutar de inmediato. Para esto, se cuenta con un servicio que permite que una tarea le indique al sistema que debe ejecutar otra tarea de inmediato. El servicio se denomina `sys_runNow`, que toma el identificador de la tarea a ser ejecutada y se encarga de ejecutar de inmediato dicha tarea.

[1] Los nombres no son revelados por el estricto secreto en el desarrollo del sistema.



(10p) a. Suponiendo segmentación *flat*, especificar un posible mapa de memoria para paginación. Definir las interrupciones necesarias y los campos en la tabla `IDT`. Explicar cualquier consideración especial a tener en cuenta sobre el sistema.

(15p) b. Implementar en ASM/C la rutina de atención de interrupciones del reloj y el llamado al sistema `sys_runNow`.

(15p) c. Implementar en ASM/C una de las rutinas de atención de excepciones. Describir cuál es el mecanismo utilizado para informar a la tarea la cantidad de veces que fue reiniciada.

Ej. 3. (36 puntos)

Suponer un sistema tipo que ejecuta tareas concurrentemente en dos niveles de protección. Se pide construir un servicio del sistema (*syscall*) que provea la extraña funcionalidad de conocer cuantos KB de memoria de una tarea cualquiera están "sucios". Esto quiere decir que en paginación las paginas con nivel de usuario tengan su bit *dirty* encendido.

Se provee la función `tss* getTSSByTaskID(int taskID)` que dado un identificador de tarea obtiene el puntero a la *TSS* de la tarea.

El servicio debe recibir a través del registro `eax` el identificador de la tarea, y retornará en el mismo registro el valor calculado.

(20p) a. Implementar en ASM/C el servicio del sistema, suponer que el mismo debe atenderse en la interrupción `0x77`.

(16p) b. Modificar la implementación anterior para obligar al sistema a marcar como *dirty* todas las paginas de usuario (No cambiar el bit manualmente).

Nota: Se consideran paginas de usuario aquellas que el usuario pueda acceder.
Nota^2: No se puede alterar de forma persistente la memoria de ninguna tarea.