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

Organización del Computador 2
Segundo parcial – 21/06/18

1 (30)	2 (40)	3 (30)	

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
28	`0x30100123`	`0x23F01`	1	1	1	0	1	11	`0xA`
37	`0x4A1F0102`	`0x12222`	1	1	1	0	1	10	`0x8`
39	`0x222F22F2`	`0x40DDD`	1	1	1	0	1	00	`0x2`
43	`0x1D900D00`	`0x5FFFF`	1	1	1	0	1	00	`0x0`

Y el siguiente esquema de paginación:

Rango Lineal	Rango Físico	Atributos
`0x4CA40000 - 0x4CA55FFF`	`0xD549C000 - 0xD54B1FFF`	read only, supervisor
`0x55040000 - 0x5504FFFF`	`0x9A88F000 - 0x9A89EFFF`	read/write, supervisor
`0x5EDF2000 - 0x5EFF2FFF`	`0x934F3000 - 0x936F3FFF`	read only, user

(10p) a. Especificar todas las entradas de las estructuras necesarias para construir un esquema de paginación. Suponer que todas las entradas no mencionadas son nulas.

(20p) 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 - `00E3:1C9500FF` - CPL `11` - lectura

II - `012A:0BB50088` - CPL `10` - ejecución

III - `013A:3CB00000` - CPL `00` - escritura

IV - `015A:3E25FFFF` - CPL `00` - lectura

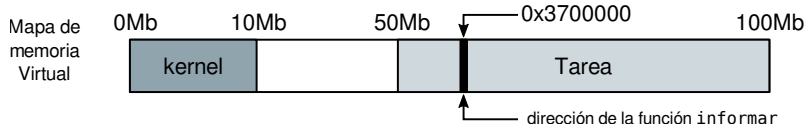
Ej. 2. (40 puntos)

Considerar un sistema en dos niveles de protección con paginación activa. Este ejecuta concurrentemente 15 tareas una despues de la otra.

Una vez ejecutandose, las tareas pueden generar cualquier tipo de excepción, en cuyo caso deben ser reiniciadas. Antes de reiniciar una tarea, se debe informar a todas las otras tareas que una tarea esta siendo reiniciada. Para esto se debe ejecutar la función `void informar(int tarea)` en el contexto

de cada tarea a fin de informar a la misma de dicho suceso. Cada tarea implementa su propia función informar, alojando en `0x3700000` la dirección de la función `informar`. Una vez que termina de ejecutar, la función llama al syscall `finalizar` en la interrupción `0x54`.

Para reiniciar una tarea se debe llamar a la función `void limpiarMemoria(int tarea)`, que toma un índice que identifica la tarea y pisa toda la memoria de la tarea por una copia limpia de la misma.



(10p) a. Describir los campos relevantes de todas las estructuras involucradas en el sistema para administrar segmentación, paginación, tareas, interrupciones y privilegios.

(30p) b. Escribir en ASM/C la rutina de atención de interrupciones del Reloj, una rutina de atención de excepciones y la syscall `finalizar`.

Nota: Suponer que las funciones `informar` no producen excepciones.

Ej. 3. (30 puntos)

Se tiene un sistema tipo con segmentación flat y paginación activa que ejecuta concurrentemente una cantidad variable de tareas. Se desea implementar un mecanismo que permita contar, en paginación, a cuantas paginas distintas de usuario se accede entre cambios de contexto por parte de una tarea. El sistema generará un *log* para cada tarea con esta información utilizando la función `void saveCount(int count)` que almacena la cuenta de la tarea actual.

(10p) a. Idear una solución al mecanismo pedido. Explicar detalladamente su funcionamiento.

(20p) b. Implementar en ASM/C todas las rutinas necesarias para implementar el mecanismo.