

(24)

#24

Segundo Parcial  
Segundo Cuatrimestre 2022

### Normas generales

- El parcial es INDIVIDUAL
- Puede disponer de la bibliografía de la materia y acceder al repositorio de código del taller de system programming, desarrollado durante la cursada
- Las resoluciones que incluyan código, pueden usar assembly o C. No es necesario que el código compile correctamente, pero debe tener un nivel de detalle adecuado para lo pedido por el ejercicio.
- Numere las hojas entregadas. Complete en la primera hoja la cantidad de hojas entregadas
- Entregue esta hoja junto al examen. La misma no se incluye en el total de hojas entregadas.
- Los ejercicios teóricos deberán entregarlos en hojas separadas del Ejercicio 1
- Luego de la entrega habrá una instancia coloquial de defensa del examen

### Régimen de Aprobación

- Para aprobar el examen es necesario obtener como mínimo **60 puntos**.
- Para promocionar es condición suficiente y necesaria obtener como mínimo **80 puntos** tanto en este examen como en el primer parcial

### Enunciado - Tema 1

#### Ejercicio 1 - (75 puntos)

**NOTA: Lea el enunciado del ejercicio hasta el final, antes de comenzar a resolverlo**

En un sistema similar al que implementamos en los talleres del curso (modo protegido con paginación activada), se quiere implementar un servicio tal que cualquier tarea del sistema lo pueda invocar mediante la siguiente instrucción:

```
int 100
```

Recibirá los siguientes parámetros (en ese orden):

- `uint32_t virt`, una dirección de página virtual
- `uint32_t phy`, una dirección de página física
- `uint16_t task_sel`, un selector de segmento que apunta a un descriptor de TSS en la GDT.

Para pasar los parámetros a este servicio, se puede escoger una convención arbitraria.

El servicio en cuestión forzará la ejecución de código comenzando en la dirección física `phy`, mapeado en `virt`. Tanto la tarea actual como la tarea que será pasada como parámetro (indicada por su `task_sel`) deben realizar la ejecución de la página física en cuestión.

Para eso, dicho servicio deberá:

- Realizar los mapeos necesarios
- Modificar los campos necesarios para que la tarea determinada por `task_sel`, retome su ejecución en la posición establecida la próxima vez que se conmute a ella.
- Modificar los campos necesarios para que la tarea actual, retome su ejecución en la posición establecida una vez completada la llamada

*Se recomienda organizar la resolución del ejercicio realizando paso a paso los items mencionados anteriormente.*

### **Se pide:**

- a. Definir o modificar las estructuras de sistema necesarias para que dicho servicio pueda ser invocado
- b. Implementar dicho servicio (pseudocódigo)
- c. Dar un ejemplo de invocación de dicho servicio

### **Detalles de implementación:**

- El código en cuestión a donde se salta es de nivel 3.
- Los punteros a las pilas de nivel 3 de ambas tareas y el puntero a la pila de nivel 0 de la tarea pasada por parámetro, deberán ser reinicializados a la base de la pila, teniendo en cuenta que las mismas comienzan al final de la página y no se extienden más que 4096 bytes.
- Asumir que todas las tareas ya fueron alojadas al menos una vez y que el cambio de tareas se hace en la rutina de interrupción de reloj, como en el taller

### **A tener en cuenta para la entrega:**

- Está permitido utilizar las funciones desarrolladas en el taller
- Es necesario que se incluya una explicación con sus palabras de la idea general de la solución
- Es necesaria la entrega de código o pseudocódigo que implemente la solución