

Recuperatorio del Segundo Parcial

Segundo Cuatrimestre 2023

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.
- Luego de la entrega habrá una instancia coloquial de defensa del examen

Régimen de Aprobación

- Para aprobar el examen es necesario obtener cómo mínimo **60 puntos**.

NOTA: Lea el enunciado del parcial hasta el final, antes de comenzar a resolverlo.

Enunciado

Ejercicio 1 - (60 puntos)

En un sistema como el desarrollado en los talleres se desea modificar la forma en que las tareas se muestran en la pantalla. En lugar de realizar la syscall `draw` (previamente implementada como `int 88`) se quiere que **las tareas tengan acceso directo a la memoria de video**:

- la memoria física de video (`0xB8000-0xB9FFF`) es la que se refleja en la pantalla real.
- Sólo puede haber una única tarea en un determinado momento con la memoria física de video (`0xB8000-0xB9FFF`) mapeada.
- El resto de las tareas tendrá asignadas una pantalla *dummy* (falsa) en la región física (`0x1E000-0x1FFFF`).
- La memoria de video de una tarea se mapeará **siempre** en el rango virtual `0x08004000-0x08005FFF`, independientemente si tiene acceso a la pantalla real o no.

Con el diseño propuesto hay una única tarea “actualmente en pantalla” (con acceso a la memoria física de video). Las tareas que no se encuentren en pantalla podrán escribir a las direcciones virtuales de video pero verán sus escrituras reflejadas en la “pantalla virtual compartida”(dummy).

Soltar la tecla TAB¹ cambiará la tarea “actualmente en pantalla”. Los cambios de tarea en pantalla se realizarán de manera cíclica (T1-T2-T3-T4-T1-T2-...).

Se solicita describir los cambios requeridos para implementar esta nueva característica.

Para ello respondé los siguientes puntos:

- Dibuje el esquema de memoria virtual de las tareas del nuevo sistema.
- Describa los cambios al proceso de creación de tareas (`init_task_dir`, `create_task`, `sched_add_task`, etc...). Muestre código y pseudocódigo.
- Explique qué mecanismo usará para que el sistema sepa “a qué tarea le toca” el acceso a la pantalla.
- Describa los cambios necesarios para realizar el “cambio de pantalla” al **soltar** la tecla TAB. Proponga una implementación posible mostrando código y pseudocódigo.
- En el mecanismo propuesto las tareas no tienen forma sencilla de saber si “es su turno” de usar la pantalla. Proponga una solución. *No se pide código ni pseudocódigo, sólo la idea.*
- En el mecanismo propuesto la tarea debe redibujar toda su pantalla cuando logra conseguir acceso a la misma. ¿Cómo podría evitarse eso? *No se pide código ni pseudocódigo, sólo la idea.*

Ejercicio 2 - (40 puntos)

Sea un sistema similar al utilizado en los talleres de la materia que ejecuta concurrentemente 4 tareas independientes. El mapa de memoria virtual es el mismo para todas las tareas y cada una tiene asignado 1MB de memoria para datos. La tarea con el ID 1 es una tarea maliciosa que puede “robarle” información a otras tareas utilizando el servicio `CopiarPagina` que recibe un ID de otra tarea en EDI y una dirección de memoria virtual en ESI. Este servicio se encarga de producir una copia de la página indicada por ESI de la tarea pasada por parámetro EDI en la misma página de la tarea llamadora. Es decir, que si los parámetros pasados fueron ID: 3 y posición de memoria virtual: 0x7000000, al finalizar el servicio la tarea llamadora tendrá en la página indicada por la posición de memoria virtual 0x7000000 una copia de lo que tenía la tarea 3 en esta dirección.

- Dibujar un esquema que muestre una posible distribución de memoria para este sistema.
- Implementar el servicio pedido. Definir para esto cualquier variable global utilizada. Indicar todo lo que se asume para resolver este ejercicio.
- Para que el servicio pedido funcione, ¿deben modificar la interrupción de reloj?. Si la respuesta es sí, como deben modificarla?. Si la respuesta es no, explicar por qué.

¹el scancode al apretar TAB es 0x0F. El scancode al soltarla es 0x8F.

A tener en cuenta para la entrega (para todos los ejercicios):

- Está permitido utilizar las funciones desarrolladas en los talleres.
- Es necesario que se incluya una explicación con sus palabras de la idea general de las soluciones.
- Es necesario escribir todas las asunciones que haga sobre el sistema.
- Es necesaria la entrega de código o pseudocódigo que implemente las soluciones.