

Recuperatorio del Segundo Parcial

Primer 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 - (70 puntos)

En un sistema similar al que implementamos en los talleres del curso (modo protegido con paginación activada) se pide:

- Implementar la *Syscall exit* que al ser llamada por una tarea, inactiva dicha tarea y pone a correr la siguiente (segun indique el sistema de prioridad utilizado). Mostrar el código.
- ¿Cómo modificarías el punto anterior para que exit (además de lo que hace normalmente) guarde el ID de quién la llamó en el EAX de próxima tarea a ejecutar? Mostrar código.
- ¿Y si ahora no es la *Syscall exit* la que modifica el EAX de nivel 3 de la tarea que va a ser ejecutada luego de la llamada a la *Syscall* sino la interrupción de reloj? Como deberías modificar el código de la interrupción de reloj?. Mostrar el código y explicar todo lo que agregues al sistema.

- d. ¿Consideras que es una buena práctica que las tareas se comuniquen sobre-escribiendo los registros de propósito general de otra tarea? ¿Qué problemas pueden surgir? Teniendo en cuenta lo visto en la materia, ¿de que otra forma se podrían pasar mensajes las tareas entre sí? Hacer un esquema que muestre los mapeos de memoria virtual a física de dos tareas que usen la forma de comunicarse propuesta por ustedes. Realizá un diagrama que muestre la interacción del mecanismo con el scheduler y/o los mapeos de memoria.

Se recomienda organizar la resolución del ejercicio realizando paso a paso los items mencionados anteriormente y explicar las decisiones que toman.

Ejercicio 2 - (30 puntos)

El ENTEL575 fué un microprocesador compatible con los Intel i686. Este procesador incluía varias funcionalidades extra que nunca fueron replicadas por Intel. Una de ellas es la operación RSTLOOP, la cual escribe un cero en ECX 'reiniciando' el contador de vueltas. RSTLOOP se codifica con la secuencia de bytes 0x0F 0x0B.

Tenemos un montón de software escrito para el ENTEL575 pero lamentablemente no poseemos hardware que lo pueda correr. ¿Podrías desarrollar un sistema que nos permita hacerlo?

Para ello respondé los siguientes puntos:

- ¿Qué excepción ocurre cuándo un procesador x86 intenta ejecutar una instrucción no soportada?
- Realice un diagrama de pila que muestre el estado de la pila del kernel luego de que una aplicación de usuario intentó ejecutar RSTLOOP.
- ¿Qué dirección de retorno se encuentra en la pila al atender la excepción?
- Describa una posible implementación de RSTLOOP utilizando el mecanismo descrito en (a) y (b).
 - El mecanismo propuesto sólo debe actuar cuándo la instrucción no soportada es RSTLOOP.
 - Si la instrucción que generó la excepción no es RSTLOOP la tarea debe ser deshabilitada y la ejecución debe saltar a la tarea idle.
 - Si la instrucción que generó la excepción es RSTLOOP adecúe la dirección de retorno de manera que permita a la tarea continuar la ejecución sin problemas.
- ¿Qué ocurriría si no se adecuara la dirección de retorno luego de simular RSTLOOP?
- Detalle los cambios a las estructuras del sistema visto en el taller que haría para realizar la implementación descrita en (d).
- Muestre código para la rutina de atención de interrupciones descrita en (d) y todo otro cambio de comportamiento que haya visto necesario.

Recomendaciones:

- Lea el capítulo del manual sobre interrupciones y excepciones
- Revise con sumo cuidado el *Exception and interrupt reference*
- Repase el mecanismo de cambio de pila
- Recuerde los mecanismos que el procesador le ofrece para realizar cambios de tareas

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 que implemente las soluciones.