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 cómo 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 sera pasada como parámetro (indicada por su `task_sel` deben realizar la ejecucion de la pagina 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

Teóricos

Ejercicio 2 - System Programming - (10 puntos)

1. A continuación se tiene las entradas de una TLB correspondientes a una misma tarea que ejecuta en un sistema basado en un procesador x86. Las entradas están en el orden en el que se han ido generando las traducciones. Para dicha tarea el registro CR3 = 0x000E4000, y las tablas de página correspondientes a las direcciones en uso están inmediatamente contiguas al DTP, en el orden en el que se han ido requiriendo.

#	Nro.Pag.	Descriptor	Ctrl
1	7C047	0EF00121	ccc
2	7EEF0	1EF01067	ccc
3	EC004	001F0163	ccc
4	EC005	001F7123	ccc
5	46104	1F011005	ccc
6	46109	1F010027	ccc

Se pide:

(a) Para las entradas 1 y 2 de la TLB, escribir el contenido de sus correspondientes descriptores en cada nivel de la jerarquía de tablas de traducción a direcciones físicas, considerando que se trata de las dos primeras páginas requeridas al ejecutar la tarea.

(b) Especificar para cada PDE que valor deben tener los bits U/S y R/W, para ser consistentes con el contenido de la TLB. Respuestas posibles en cada caso: 0, 1, o X (indistinto).

(c) Suponiendo que las seis entradas están siendo utilizadas por una misma tarea, y por cada task switch se modifica CR3. ¿Que ocurre con cada una al momento del task switch?. ¿Cual es el tratamiento para aquellas que se han modificado?

(d) ¿Cual es la función dentro de la TLB de los bits identificados genéricamente como Ctrl?.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address of page directory[1]																				Ignored							PCD	PWT	Ignored			CR3
Bits 31:22 of address of 4MB page frame										Reserved (must be 0)						Bits 39:32 of address[2]			PAT	Ignored			G	1	D	A	PCD	PWT	U/S	R/W	1	PDE: 4MB page
Address of page table																				Ignored				0	Ign	A	PCD	PWT	U/S	R/W	1	PDE: page table
Ignored																															0	PDE: not present
Address of 4KB page frame																				Ignored			G	PAT	D	A	PCD	PWT	U/S	R/W	1	PTE: 4KB page
Ignored																															0	PTE: not present

2. ¿Porqué un sistema que utilice dos o más niveles de privilegio diferentes debe usar una TSS aun cuando la conmutación de tareas es manual?

Ejercicio 3 - Micro Arquitectura - (15 puntos)

1. Se tiene un sistema SMP. Cada procesador tiene su propio controlador Cache. Utiliza para mantener coherente los sub sistemas de memoria cache y la DRAM el protocolo MESI

   (a) ¿Cuál es el recurso de hardware mediante el cual cada Controlador Cache detecta las transacciones que los demás Cores cursan con la memoria del sistema (DRAM)?. Indicar el nombre del recurso, y a que líneas del bus se conecta.

   (b) Explicar si M es un estado preciso o impreciso. Justificar

   (c) En caso que otro procesador inicie una transferencia de lectura sobre un dato que éste procesador tiene en una línea, cuyo estado es **M**, se desea saber:

      i. ¿Están coherentes las diferentes copias del dato requerido?. Justificar.

      ii. ¿Cómo se asegura la coherencia de la operación?. Justificar. Detallar el handshake de hardware describiendo las líneas involucradas si corresponde.

      iii. ¿Cuál es la política de escritura aplicada por el procesador antes de la transacción analizada y después de la misma?

2. Considerar el siguiente código:

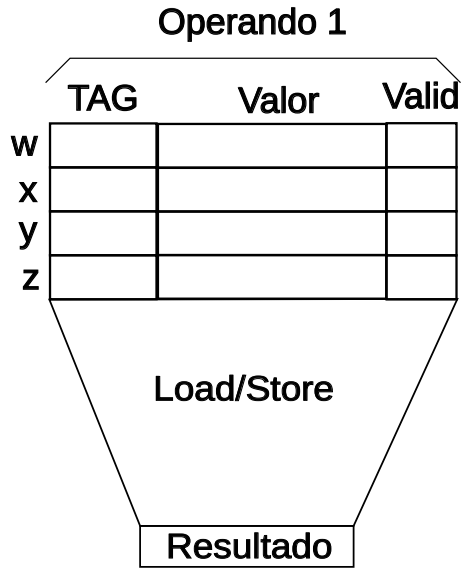
```
1  #define count 1024
2
3  section .data
4  buffer:     dq  0x4523, 0xFFFF5666723, ... ; 'count' values
5  result:     resq 2
6
7  section .text
8  mov     RSI,buffer
9  mov     RCX,count
10 mov     RDI,result
11 lods
12 add     RBX,RAX ; Estado luego de ejecutada esta instrucción!
13 mov     RAX,RBX
14 idiv    RCX
15 mov     [RDI],RAX
16 mov     [RDI+8],RDX
```

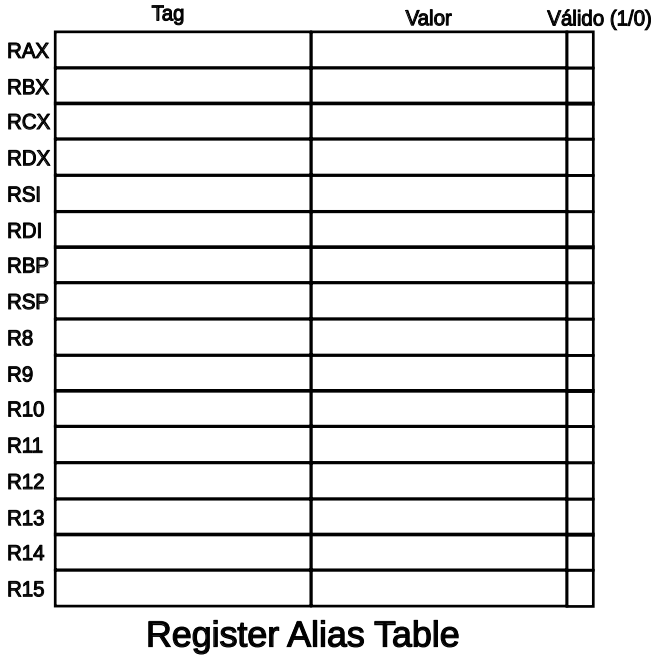
   Se pide:

   (a) Enumerar los pares de instrucciones con hazards debidos a que un procesador de Intel ejecuta fuera de orden. Explicar que significa cada uno de los hazards.

   (b) Ubicar los valores apropiados en cada TAG, VALUE, y V, para cada elemento involucrado en el algoritmo anterior para el modelo de Tomasulo que se representa a continuación, cuando se termina de ejecutar la instrucción de la línea de código **12** del listado anterior. Se pretende tener el estado de máquina terminada dicha instrucción.

(c) Explicar como se modifica el estado definido en el ítem anterior con la entrada al esquema de la figura de la instrucción de la línea **13**.





Register Alias Table