

# ARQUITECTURA DE COMPUTADORES

## Prueba del TEMA 3

Estudiante:

1. Un microprocesador multi-núcleo puede incluir núcleos SMT (Multihebra simultánea) que, en un ciclo SOLO pueden enviar a ejecutar varias instrucciones si pertenecen a hebras diferentes (F)

2. En el protocolo MSI para mantener la coherencia de cache, si una línea está en el estado M en la cache del nodo N1, la línea pasaría al estado S (shared) cuando el procesador del nodo N1 intenta leer un dato de esa línea (F)

3. En el protocolo MESI, si en la cache de un nodo N1 hay un bloque en estado E (Exclusivo), y ese nodo detecta que otro nodo N2 intenta leer un dato que está en ese bloque, el bloque pasa al estado S en los dos nodos N1 y N2 (V)

4. En un multiprocesador NUMA con 16 nodos, 4 GBytes por nodo, y líneas de cache de 128 Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la cache en un protocolo MSI sin difusión?  $2^{32} \text{ Bytes} / 2^{17} \text{ Bytes/line} \rightarrow 2^{15} \text{ entradas}$

5. En el multiprocesador NUMA descrito en la pregunta anterior ¿Cuántos bits tiene cada una de las entradas del directorio que se utiliza para mantener la coherencia de cache?  $16 + 1 = 17 \text{ bits}$

6. En el mismo multiprocesador NUMA, con protocolo MSI de coherencia de cache, podría ocurrir que alguna de las entradas de alguno de los directorios esté en el estado 1 1 0 ... 0 V (1: copia en cache del nodo correspondiente al bit; 0: no hay copia en la cache del nodo correspondiente al bit; V: línea válida en memoria principal) (V)

7. En un multiprocesador con un modelo de consistencia que no garantiza el orden  $W \rightarrow R$  (garantiza todos los demás) se ha ejecutado el siguiente código, donde X e Y son variables en memoria, inicialmente a 0, y reg1 y reg2 son registros del núcleo P1:

Accesos a mem. P1	Código P1	Accesos a mem. P2	Código P2
(1.1)W(X)	X=1;	(2.1)W(Y)	Y=2
(1.2)R(X)	reg1=X;	(2.2)W(X)	X=2
(1.3)R(Y)	reg2=Y;		

El acceso a memoria (1.2) podría adelantar al (1.1) y el (1.3) no adelantaría nunca al (1.2) (F)

8. ¿Qué combinaciones de valores (R1,R2) se pueden observar al final de P2 si los procesadores P1 y P2 implementan un modelo de consistencia en el que NO se respeta  $W \rightarrow R$  (los demás sí se respetan), e inicialmente  $X=Y=R1=R2=0$ ?

P1:	P2:
X=2;	R2=Y;
Y=1;	if (Y==1) R1=X;

$R1=2 \ R2=1$   
 $R1=2 \ R2=0$   
 $R1=0 \ R2=0$

9. El código siguiente permite implementar un cerrojo (lock(k)) en el que k=1 significa que el cerrojo está cerrado y 0 que está abierto

```
b=0; k=1;
while (fetch_&_or(k,b)==1) {};
```

10. Suponiendo que sum es una variable compartida y suml una variable local de la hebra que ejecuta el código. ¿Cómo escribiría el código siguiente con compare\_&\_swap?

```
fetch_&_add(sum,suml);
```

$\rightarrow$  do  
 $a = \text{sum};$   
 $b = a + \text{suml};$   
 $\text{compare \& swap}(a, b, \text{sum});$   
while (a != b)

# ARQUITECTURA DE COMPUTADORES

## Prueba del TEMA 3

Estudiante:

1. Un microprocesador multi-núcleo puede incluir núcleos SMT (Multihebra simultánea) que, en un ciclo, pueden enviar a ejecutar varias instrucciones de la misma o de distintas hebras (V)
2. En el protocolo MESI para mantener la coherencia de cache, si una línea está en el estado M en la cache del nodo N1, la línea pasaría al estado I (no válido) cuando el procesador del nodo N2 intenta leer un dato de esa línea (F)
3. En el protocolo MESI, si en la cache de un nodo N1 hay una línea en estado E (Exclusivo), SOLO saldrá de ese estado cuando el procesador de OTRO nodo intente leer/escribir un dato de esa línea (F)
4. En un multiprocesador NUMA con 8 nodos, 8 GBytes por nodo, y líneas de cache de 64 Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la cache en un protocolo MSI sin difusión?  
 $2^{33} \text{ Bytes} / 2^5 \text{ Bytes/línea} \rightarrow 2^{27} \text{ entradas}$
5. En el multiprocesador NUMA descrito en la pregunta anterior ¿Cuántos bits tienen cada una de las entradas del directorio que se utiliza para mantener la coherencia de cache?  
 $8 \text{ nodos} + 1 \text{ V/F} = 9 \text{ bits}$
6. En el mismo multiprocesador NUMA, con protocolo MSI de coherencia de cache, podría ocurrir que alguna de las entradas de alguno de los directorios esté en el estado 0... 0 1 0 ... 0 1 (1: copia en cache del nodo correspondiente al bit; 0: no hay copia en la cache del nodo correspondiente al bit; I: línea no-válida en memoria principal) (V)
7. En un multiprocesador con un modelo de consistencia que no garantiza el orden  $W \rightarrow R$  ni el  $W \rightarrow W$  (garantiza todos los demás) se ha ejecutado el siguiente código, donde X e Y son variables en memoria, inicialmente a 0, y reg1 y reg2 son registros del núcleo P1:

Accesos a mem. P1	Código P1	Accesos a mem. P2	Código P2
(1.1)W(X)	X=1;	(2.1)W(Y)	Y=2
(1.2)R(X)	reg1=X;	(2.2)W(X)	X=2
(1.3)R(Y)	reg2=Y;		

El acceso a memoria (1.2) no podría adelantar al (1.1) y el (2.2) podría adelantar al (2.1)

8. ¿Qué combinaciones de valores (R1,R2) se pueden observar al final de P2 si los procesadores P1 y P2 implementan un modelo de consistencia en el que NO se respeta  $W \rightarrow R$  (los demás sí se respetan), e inicialmente  $X=Y=R1=R2=0$ ?  
 $R1=2 \quad R2=2$   
 $R1=0 \quad R2=0$   
 $R1=2 \quad R2=0$

P1:	P2:
Y=1;	R2=X;
X=2;	if (Y==1) R1=X;

9. El código siguiente permite implementar un cerrojo (lock(k)) en el que k=1 significa que el cerrojo está cerrado y 0 que está abierto

```
b=1;
do compare&swap(1,b,k); // lectura-mod-escritura atómica
while (b==1);
```

10. Suponiendo que sum y rest son variables compartidas y suml y restl variables locales de la hebra que ejecuta el código. ¿Cómo escribiría el código siguiente si solo tuviera primitivas fetch\_&\_or para la sincronización (y modelo de consistencia de memoria secuencial)?

```
fetch_&_add(sum,suml);
fetch_&_add(rest, - restl);
```

$\text{while ( fetch\_or (k,1) = 1 ) \{$   
 $\text{sum} = \text{sum} + \text{suml};$   
 $\text{rest} = \text{rest} - \text{restl};$   
 $\text{k} = 0;$