

EJERCICIO 1: En un multiprocesador NUMA con 16 nodos, 4GBytes por nodo, y líneas de cache de 64Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

SOLUCIÓN: $2^2 * 2^{30} / 2^6 = 2^{26}$ entradas

EJERCICIO 2: 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?

SOLUCIÓN: $16+1=17$ bits

EJERCICIO 3: En un multiprocesador NUMA con 8 nodos, 16GBytes por nodo, y líneas de cache de 64Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

SOLUCIÓN: $16 \text{ GBytes} = 2^{34} \text{ Bytes} ; 2^{34} / 2^6 = 2^{28}$ líneas

EJERCICIO 4: 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 en un protocolo MSI con directorio?

SOLUCIÓN: $8+1 = 9$ bits

EJERCICIO 5: En el mismo multiprocesador NUMA anterior con el mismo protocolo de coherencia de cache, se puede tener el estado 1 1 0 ... 0 V (1: hay copia del bloque en la cache del nodo correspondiente al bit; 0: no hay copia en la cache del nodo correspondiente al bit; V: bloque válido en memoria principal) en alguna de las entradas de alguno de los directorios.

RESPUESTA: VERDADERO

EJERCICIO 6: En un multiprocesador NUMA con 8 nodos, 8GBytes por nodo, y líneas de cache de 128Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

SOLUCIÓN: $8 \text{ GBytes} = 2^{33} \text{ Bytes} ; 2^{33} / 2^7 = 2^{26}$ líneas

EJERCICIO 7: 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 en un protocolo MSI con directorio?

SOLUCIÓN: $8 + 1 = 9$ bits

EJERCICIO 8: En el mismo multiprocesador NUMA anterior con el mismo protocolo de coherencia de cache, se puede tener el estado 1 1 0 ... 0 I (1: hay copia del bloque en la cache del nodo correspondiente al bit; 0: no hay copia en la cache del nodo correspondiente al bit; I: bloque no válido en memoria principal).

RESPUESTA: FALSO

EJERCICIO 9: En un multiprocesador NUMA con 64 nodos, 8GBytes por nodo, y líneas de cache de 128Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

SOLUCIÓN: $2^3 * 2^{30} / 2^7 = 2^{26}$ entradas

EJERCICIO 10: 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?

SOLUCIÓN: $64 + 1 = 65$ (un bit por nodo más un bit de validez)

EJERCICIO 11: En un multiprocesador NUMA con 32 nodos, 16GBytes por nodo, y líneas de cache de 128Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

SOLUCIÓN: $2^4 * 2^{30} / 2^7 = 2^{27}$ entradas

EJERCICIO 12: 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?

SOLUCIÓN: $32 + 1 = 33$ (un bit por nodo más un bit de validez)

EJERCICIO 13: ¿Qué valores se puede observar en R si el modelo de consistencia de memoria del computador donde están los procesadores que ejecutan estos códigos no respeta el orden W->W (sí respeta los demás) e inicialmente X=Y=0?

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

SOLUCIÓN: R=0; R=1; R=2

Si respeta el order W->W : R=1; R=2

EJERCICIO 14: ¿Qué valor deben r1,r2 y r3 para que la secuencia de instrucciones siguiente implemente un cerrojo lock(k) en el que k=1 significa que el cerrojo está cerrado y 0 que está abierto?

```

b=r1;
do
    compare&swap(r2,b,k); // si r2==k y b se intercambian
while (b==r3);
    
```

SOLUCIÓN: r1=1; r2=0; r3=1

EJERCICIO 15: Si el modelo de consistencia de memoria de un multiprocesador NO respeta el orden R->W (Sí respeta todos los demás), e inicialmente X=Y=Z=r1=0 (donde X,Y,Z son variables en memoria compartida y r1 es un registro de P1), al final se podría tener Y=0.

P1: (1) while (Z==0) {}; (2) r1=X; (3) Y=r1;	P2: (a) X=1; (b) Y=2; (c) Z=1;
--	--

SOLUCIÓN: En P2 siempre se producen en Escrituras mientras que en P1 se produce un RAW entre (2) y (3). Como se respeta W->W, entonces las escrituras de P2 se hacen en orden, por tanto X no puede valer nunca 0, y en consecuencia, sólo podría darse que X=1.

RESPUESTA: FALSO

EJERCICIO 16: Si el modelo de consistencia de memoria de un multiprocesador NO respeta el orden R->W (SÍ respeta todos los demás), e inicialmente $X=Y=Z=r1=0$ (donde X,Y,Z son variables en memoria compartida y $r1$ es un registro de P1), al final SÓLO se podría tener $Y=1$.

P1: (1) while ($Z==0$) {}; (2) $r1=X$; (3) $Y=r1$;	P2: (a) $X=1$; (b) $Y=2$; (c) $Z=1$;
--	---

SOLUCIÓN: Sucede igual que en el ejercicio anterior.

RESPUESTA: VERDADERO

EJERCICIO 17: ¿Qué valores se puede observar en R si el modelo de consistencia de memoria del computador donde están los procesadores que ejecutan estos códigos tienen un modelo de consistencia secuencial, e inicialmente $X=Y=0$? (R es un registro del procesador donde se ejecuta P2 y X e Y son direcciones de memoria compartida).

P1: $X=2$ $Y=1$	P2: $R=1$; if($Y==1$) $R=X$;
----------------------------------	---

SOLUCIÓN: $R=1$ ó $R=2$.

EJERCICIO 18: ¿Qué valores se observarían en el registro R del problema anterior si no se garantiza el orden W->R?

SOLUCIÓN: Lo mismo (no cambia el orden de las escrituras en P1) $R=1$ ó $R=2$

EJERCICIO 19: ¿Qué valores se observarían en el registro R del problema anterior si no se garantiza el orden W->W?

SOLUCIÓN: $R=1$; $R=2$ ó $R=0$.

EJERCICIO 20: ¿Qué valor pondría en a para que la secuencia de instrucciones siguiente implemente un cerrojo lock(k) en el que $k=1$ significa que el cerrojo está cerrado y 0 que está abierto?

```

b=1;
//... colocar a
do
    compare&swap(a,b,k); //lect-mod-escritura atómica
while(b==1);

```

SOLUCIÓN: $a=0$;

EJERCICIO 21: ¿Cómo implementaría un cerrojo (lock(k)) en el que $k=1$ significa que el cerrojo está cerrado y 0 que está abierto con una primitiva del tipo test_&_set?

SOLUCIÓN: while(test_&_set(k)==1){}; //k se inicializa a 0.

EJERCICIO 22: ¿Qué valor deben tener r1, r2 y r3 tener para que la secuencia de instrucciones siguiente implemente un cerrojo lock(k) en el que k=1 significa que el cerrojo está cerrado y 0 que está abierto?

```
b=r1;  
do  
    compare&swap(r2,b,k); //Si r2==k -> k y b se intercambian  
while(b==r3);
```

SOLUCIÓN: r1=1; r2=0; r3=1.