



**ATENCIÓN:**

- Lea atentamente todo el enunciado antes de comenzar a contestar.
  - La duración del examen es de 180 minutos.
- 

NOMBRE:

APELLIDOS:

NIA:

**Ejercicio 1:** (1,5 puntos)

Se dispone de un computador con un solo núcleo que ejecuta una aplicación de evaluación de riesgos financieros. Esta aplicación es intensiva en cálculo, a lo que dedica el 90% del tiempo. El 10% restante lo dedica a esperar en operaciones de entrada/salida a disco.

Del tiempo que la aplicación pasa ejecutando instrucciones de cálculo un 75% del tiempo lo pasa ejecutando operaciones en coma flotante y un 25% lo pasa ejecutando otras instrucciones. La ejecución de una instrucción de coma flotante requiere como promedio 12 CPI. El resto de instrucciones requieren como promedio 4 CPI.

Se está valorando la migración de esta aplicación a las siguientes alternativas, que no incorporan ninguna mejora para el tiempo de las operaciones de entrada/salida a disco:

- Alternativa A: Un procesador con un solo núcleo y con una frecuencia de reloj un 50% más alta que la de la máquina original en el que las instrucciones de coma flotante requieren un 10% más de ciclos por instrucción y el resto de instrucciones requieren un 25% más de ciclos por instrucción.
- Alternativa B: Un procesador con cuatro núcleos y con una frecuencia de reloj un 50% más baja que la de la máquina original, en el que las instrucciones de coma flotante requieren un 20% menos de ciclos de reloj y el resto de instrucciones los mismos ciclos de reloj.

Se pide responder de forma justificada a las siguientes cuestiones:

- a) ¿Cuál será la aceleración/deceleración global de la aplicación en el caso A?
- b) ¿Cuál será la aceleración/deceleración global de la aplicación en el caso B si se asume que la parte de cálculo es totalmente paralelizable mientras la entrada/salida no admite ningún tipo de paralelización?

**Ejercicio 2:** (1,5 puntos)

Responda a las siguientes cuestiones:

- a) Diga cuándo un sistema de memoria compartida es coherente (máximo 3-4 líneas).
- b) Enumere las condiciones para la coherencia de memoria.
- c) Enumere las restricciones que garantizan que un sistema ofrece consistencia secuencial.
- d) En un modelo de consistencia de adquisición/liberación ¿Cuál es la semántica de la operación de adquisición (*acquire*)?



Examen

- e) En un modelo de consistencia de adquisición/liberación ¿Cuál es la semántica de la operación de liberación (*release*)?

**Ejercicio 3:** (1 punto):

Explique brevemente las diferencias que hay en OpenMP entre planificación estática, planificación dinámica y planificación guiada.

**Ejercicio 4:** (2 puntos)

En un determinado procesador se pretende ejecutar el siguiente segmento de código:

```
i0: lw $r4, 0($r1)
i1: lw $r5, 0($r2)
i2: add $r4, $r4, $r5
i3: sw $r4, 0($r3)
i4: addi $r1, $r1, 4
i5: addi $r2, $r2, 4
i6: addi $r3, $r3, 4
i7: bne $r3, $r0, i0
```

Asuma que el procesador tiene una arquitectura segmentada de 5 etapas (captación, decodificación, ejecución, memoria y post-escritura) sin envío adelantado (*forwarding*). Todas las operaciones se ejecutan en un ciclo por etapa, excepto:

- Las operaciones de carga y almacenamiento, que requieren un total de dos ciclos para la etapa de memoria (un ciclo adicional).
- Las instrucciones de salto, que requieren un ciclo adicional en la etapa de ejecución. Considere que estas instrucciones no cuentan con ningún tipo de predicción de saltos.

Se pide responder de forma justificada a las siguientes cuestiones:

- a) Determine los riesgos de datos RAW que presenta el código que tienen impacto en la ejecución del código.
- b) Muestre un diagrama de tiempos con las fases de ejecución de cada instrucción para una iteración.
- c) Determine cuantos ciclos requiere la ejecución de una iteración del bucle si no hay ningún tipo de predicción de saltos.
- d) Proponga un desenrollamiento (*loop unrolling*) del bucle asumiendo que bucle se ejecuta 1000 iteraciones. Desenrolle con un factor de cuatro iteraciones.
- e) Determine la aceleración o *speedup* obtenido mediante el desenrollamiento del apartado anterior.

**Ejercicio 5:** (2 puntos)

Sea un multiprocesador con arquitectura de memoria compartida simétrica basado en bus con protocolo de espionaje o *snooping*. Cada procesador tiene una caché privada cuya coherencia se mantiene usando el



Examen

protocolo MSI. Cada caché utiliza correspondencia directa y tiene cuatro bloques cada uno con dos palabras. Esta caché utiliza como campo de etiqueta la dirección de memoria completa.

Las siguientes tablas muestran el estado de cada memoria, con la palabra menos significativa a la izquierda.

**Procesador P0**

Bloque	Estado	Etiqueta	Datos	
B0	I	0x00100700	0x00000000	0x7FAABB11
B1	S	0x00100708	0x00000000	0x00001234
B2	M	0x00100710	0x00000000	0x0077AABB
B3	I	0x00100718	0x00000000	0x7FAABB11

**Procesador P1**

Bloque	Estado	Etiqueta	Datos	
B0	I	0x00100700	0x00000000	0x7FAABB11
B1	M	0x00100728	0x00000000	0xFF000000
B2	I	0x00100710	0x00000000	0xEEEE7777
B3	S	0x00100718	0x00000000	0x7FAABB11

**Procesador P2**

Bloque	Estado	Etiqueta	Datos	
B0	S	0x00100720	0x00000000	0x1111AAAA
B1	S	0x00100708	0x00000000	0X00001234
B2	I	0x00100710	0x00000000	0x7FAABB11
B3	I	0x00100718	0x00001234	0x1111AABB

Para cada uno de los apartados que a continuación se presentan, parta de la situación inicial del problema, sin tener en cuenta los cambios de los apartados anteriores. Indique los cambios que se producen en las cachés. En el caso de las lecturas, indique además cuál es el valor efectivamente leído.

- a) P2: write 0x00100708, 0xFFFFFFFF
- b) P2: read 0x00100708
- c) P2: read 0x00100718

En cada apartado deberá rellenar una tabla con el siguiente formato, justificando la respuesta:

Procesador	Bloque	Estado Anterior	Estado Nuevo	Etiqueta	Datos	



--	--	--	--	--	--	--

### Ejercicio 6: (2 puntos)

Sea la siguiente función:

```
std::mutex m; // mutex global
int contador; // contador global

void f() {
    m.lock();
    ++contador;
    m.unlock();
}
```

Se desea sustituir la variable global **m** y evitar posibles llamadas al sistema operativo, pero al mismo tiempo se desea garantizar la exclusión mutua en el incremento del contador.

Se pide:

- Proponga y codifique una solución que ofrezca consistencia secuencial y no implique llamadas al sistema.
- Proponga y codifique una solución que ofrezca consistencia de adquisición-liberación.
- Proponga y codifique una solución que ofrezca consistencia de adquisición-liberación y sea válida en el caso en que el contador pase a ser una variable de tipo **double**.