

Apellidos, Nombre: _____

D.N.I: _____

Firma: _____

Prácticas presentadas: _____

1	2	3	4	5	6	7	8	Suma
(6)	(6)	(5)	(8)	(12)	(8)	(5)	(10)	(50)

Pregunta 1: [6 Puntos]

Ya se venden ordenadores de mesa con la tecnología “dual-core”, es decir, se encuentran dos CPUs en un solo chip del procesador. Da argumentos qué ventajas un usuario (tanto como usuario final de aplicaciones como usuario programador) tiene si se compra tal equipo. También discuta posibles desventajas comparando el dual-core con la tecnología clásica (ten en cuenta que los dual core no suelen alcanzar la misma frecuencia del reloj). Es decir, pon las ventajas y desventajas de la programación concurrente que vimos en clase en el contexto concreto de dichos procesadores y sus usuarios.

Pregunta 2: [6 Puntos]

¿Qué se entiende con “espera activa” de un proceso? ¿Cuáles son métodos para evitar la “espera activa”? ¿Cuáles son situaciones en las cuales no se puede evitar una “espera activa”?

Pregunta 3: [4 Puntos]

Describe brevemente la posibilidad usando el “compare-and-swap” como ayuda en hardware para implementar acceso con exclusión mutua a la memoria y destaca su diferencia con una implementación usando la operación “test-and-set”.

Pregunta 4: [8 Puntos]

¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos? Describe los tres métodos disponibles para solventar el problema del bloqueo. Razona brevemente sobre sus eficiencias en entornos distribuidos.

Pregunta 5: [12 Puntos]

Asumimos que un cruce en una carretera es un recurso y que los coches que quieren pasar de alguna manera por dicho cruce son los procesos. Con la llegada de los coches a las diferentes entradas del cruce se forman colas de espera. Asumimos que es cola única per entrada y que naturalmente hay que avanzar primero en dicha cola hasta llegar al cruce. Existen diferentes reglas que regulan el tráfico en los cruces. Considera las siguientes 5 reglas de tráfico:

1. En los EE.UU. existe la regla: cada una de las cuatro entradas en el cruce tiene una señal “stop”, un conductor llegando al cruce tiene que dejar pasar a todos aquellos otros coches que ya están parados como primeros en su entrada cuando él mismo acaba de parar. Después le toca a él, independientemente de su trayectoria.
2. Existen cruces sin ninguna señalización donde hay que ceder paso al que llegá desde la derecha y él que dobla a la izquierda tiene que ceder paso a los que vienen en frente.
3. Existen cruces donde dos entradas tienen “ceder paso” (o “stop”) y dos no la tienen (y también él que dobla a la izquierda tiene que ceder paso a los que vienen en frente).
4. Existen cruces reguladas por semáforos.
5. Existen cruces con rotondas donde tiene preferencia el coche que ya está en el círculo y los demás tienen que ceder el paso.

Razona sobre las políticas de justicia de los diferentes tipos de cruces. Ten en cuenta posibles situaciones de bloqueo y inanición. Discuta brevemente la eficiencia de los cinco tipos (respecto a fluidez del tráfico según su densidad.)

Pregunta 6: [8 Puntos]

Describe brevemente el patrón de diseño *acceptor* y *conector*.

Pregunta 7: [4 Puntos]

¿Cómo se puede conseguir que solo un hilo construye un objeto en Java (es decir, un segundo hilo que lo intente ya encuentra el objeto construido y lo nota)?

Pregunta para las prácticas:

Pregunta 8: [12 Puntos]

En las prácticas hemos estudiado la implementación de una lista concurrente que disponía de las operaciones *insertar*, *borrar*, e *iterar*.

Describe brevemente cuales fueran las técnicas usadas para que varios hilos pueden manipular la lista concurrentemente (por lo menos en aquellas situaciones en las cuales las modificaciones tienen lugar en posiciones suficientemente separadas en la lista) sin que exista la posibilidad de un bloqueo mutuo.

¿Cuáles consideras las principales desventajas de dicha solución?