

Apellidos, Nombre: _____

D.N.I: _____

Firma: _____

Prácticas presentadas: _____

1	2	3	4	5	6	7	8	9	Suma
(4)	(6)	(6)	(8)	(4)	(6)	(4)	(4)	(8)	(50)

Pregunta 1: [4 Puntos]

Los procesadores modernos para ordenadores de mesa ya incluyen tecnología para la ejecución concurrente de varios procesos/hilos. Incluso existen ya placas bases con varios procesadores en paralelo. Razona brevemente sobre las ventajas y desventajas de desarrollar aplicaciones para dichos sistemas.

Pregunta 2: [6 Puntos]

Razona como están relacionadas la distribución del trabajo con la distribución de los datos en una aplicación concurrente en un sistema distribuido.

Pregunta 3: [6 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 el algoritmo de Dekker que solamente trabaja con load y store. Describe también dicho algoritmo de Dekker. Destaca las ventajas y desventajas de ambos métodos.

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 su eficiencia en entornos distribuidos.

Pregunta 5: [4 Puntos]

Describe brevemente el patrón de diseño *proactor*.

Pregunta 6: [6 Puntos]

Describe un protocolo de entrada y salida para varios hilos que garantiza exclusión mutua sin espera activa.

Pregunta 7: [4 Puntos]

¿Cómo se puede conseguir que el código de un constructor se ejecute con exclusión mutua en Java?

Pregunta 8: [4 Puntos]

Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes. ¿Cómo evita dicha semántica introducida en la versión 1.5 de Java que el optimizador haga reordenaciones del código inesperadas por el usuario?

Pregunta 9: [8 Puntos]

Describe brevemente pero de forma precisa como se puede implementar las operaciones *insertar*, *borrar*, e *iterar* sobre una lista concurrente que tenga la propiedad que varios usando los `ReadersWritersLock` de Java 1.5. Asuma que los hilos manipulan la mayoría del tiempo la lista en zonas suficientemente separadas.