CD Examen Teoría y Práctica **16 de septiembro de 2005**

Dr. Arno Formella, Dra. María José Lado Touriño

Apellidos, Nombre: _____

D.N.I: _____

Firma: _____

Prácticas presentadas: _____

_____

1	2	3	4	5	6	7	8	9	10	Suma
(4)	(4)	(6)	(4)	(4)	(8)	(4)	(6)	(4)	(6)	(50)

Pregunta 1: [4 Puntos]

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

Pregunta 2: [4 Puntos]

¿Porqué se dice que la "inanición" de procesos es una propiedad de "vivacidad" que se puede convertir en una propiedad de "seguridad"? (No te olvides aclarar brevemente los términos que usas en tu explicación.)

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.)

Pregunta 4: [4 Puntos]

¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos?

Pregunta 5: [4 Puntos]

¿Cómo se define formalmente que un programa concurrente es correcto? ¿Por qué se distingue entre corrección parcial y total?

Pregunta 6: [8 Puntos]

Razona sobre las propiedades (política de justicia) del siguiente sistema que controla el acceso a un recurso compartido. Cada proceso que quiere acceso al recurso está añadido a una cola; el primer proceso en la cola obtiene acceso al recurso (con exclusión mutua) cuando el recurso esté disponible y se elimina de la cola. Mientras los demás procesos están esperando en la cola, cualquier de ellos puede cambiar su sitio con el actualmente primero en la cola, siempre cuando dicho primer proceso está de acuerdo. Obviamente la "justicia" depende del protocolo como "los procesos se ponen de acuerdo". Considera los casos extremos y da ideas de posibles protocolos "útiles".

Pregunta 7: [4 Puntos]

Describe brevemente el patrón de diseño *aviso de hecho* (o *double–checked locking*).

Pregunta 8: [6 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: [4 Puntos]

Queremos simular un semáforo general con unos semáforos binarios. Detalla el código y razona porqué tu solución funciona.

Pregunta 10: [6 Puntos]

¿Cuál es una estrategia aplicable para que no se produzca ningún bloqueo en una lista concurrente que tenga la propiedad que varios hilos pueden manipular la lista concurrentemente en zonas suficientemente separadas?