

ExamenTeoriajunio2020.pdf



MrBrown2



Programación Concurrente y de Tiempo Real



2º Grado en Ingeniería Informática



**Escuela Superior de Ingeniería
Universidad de Cádiz**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL

[Página Principal](#) / [Mis cursos](#) / [GRADO EN INGENIERÍA INFORMÁTICA](#) / [21714020_19_20_01](#) / [General](#)
/ [Cuestionario \(TEST\) de Examen de Teoría](#)

Comenzado el lunes, 15 de junio de 2020, 10:30

Estado Finalizado

Finalizado en lunes, 15 de junio de 2020, 10:52

Tiempo empleado 22 minutos 39 segundos

Pregunta **1**

Finalizado

Puntúa como
1,00

El uso de la interfaz Future junto con tareas Callable

Seleccione una:

- ☐ es igual que con la interfaz Runnable, cambiando el método run() por el método call().
- ☐ permite el acceso a los datos de retorno del método call() bajo bloqueos explícitos, que el programador debe definir escogiendo alguna técnica de control adecuada.
- ☐ permita procesar a esas tareas dándolas como parámetros a objetos de la clase Thread.
- ☒ permite el acceso a los datos de retorno del método call() de manera segura, cuando la elección y gestión de las estructuras de datos se hace adecuadamente.

La respuesta correcta es: permite el acceso a los datos de retorno del método call() de manera segura, cuando la elección y gestión de las estructuras de datos se hace adecuadamente.

Pregunta **2**

Finalizado

Puntúa como
1,00

Considere un sistema que controla el acceso concurrente a información compartida entre tareas. Sabemos que el acceso a dicha información implica un número de operaciones de escrituras concurrentes muy alto frente al número de lecturas concurrentes, e igualmente sabemos que la carga de tareas será elevada. En esas condiciones, usted como diseñador de una solución eficiente para el sistema

Seleccione una:

- ☒ descarta de inmediato una solución basada en memoria transaccional software.
- ☐ considera que merece la pena implementar soluciones con memoria transaccional software y bloqueos estándares y desarrollar un "benchmarking" para estimar cuál de ellas le ofrece mejor rendimiento.
- ☐ dado el desbalance entre el número de operaciones de lectura y escritura, estima que la solución basada en memoria transaccional software y opta por ella.
- ☐ considera que una solución con memoria transaccional que utilice Java sobre Clojure es la mejor.

La respuesta correcta es: descarta de inmediato una solución basada en memoria transaccional software.

Pregunta **3**

Finalizado

Puntúa como
1,00

Usted dispone de un servidor remoto soportado mediante el framework RMI. Dicho servidor define entre otros, un recurso que debe gestionarse bajo control de exclusión mutua. Su equipo de programadores controla dicha gestión mediante un método `void myControl()` que protege el recurso mediante el API de la clase `Semaphore`. Sus programadores crean el semáforo necesario mediante la instrucción `Semaphore mySem = new Semaphore(2)`. El objeto servidor se registra, y queda en espera de peticiones sobre el puerto 2001. Múltiples clientes situados en máquinas virtuales distintas piden a través sus referencias remotas la ejecución del método `myControl()`, y la arquitectura remota:

Seleccione una:

- ☐ Procesa secuencialmente esas peticiones, obligada por el sistema de gestión de la concurrencia del framework RMI.
- ☐ Sus programadores han desarrollado un código que es seguro pero carece de vivacidad, y usted los abronca.
- ☒ Sus programadores han desarrollado un código que no es seguro, y usted los despidе.
- ☐ La arquitectura no tiene problema alguno bajo los supuestos planteados, y funciona normalmente preservando la integridad de los datos. Usted felicita a sus programadores.

La respuesta correcta es: Sus programadores han desarrollado un código que no es seguro, y usted los despidе.

Pregunta **4**

Finalizado

Puntúa como
1,00

Desde el punto de vista de las primitivas teóricas de control de la concurrencia, una región crítica

Seleccione una:

- ☐ puede simularse únicamente con semáforos.
- ☒ puede simularse con semáforos y también con monitores.
- ☐ puede simularse únicamente con monitores.
- ☐ las primitivas de control teórico de la concurrencia no pueden simularse entre sí.

La respuesta correcta es: puede simularse con semáforos y también con monitores.

Pregunta **5**

Finalizado

Puntúa como
1,00

El uso de variables `volatile` en el ámbito de la concurrencia

Seleccione una:

- ☐ nos garantiza que los datos almacenados en ellas son accedidos de forma segura.
- ☐ permite garantizar código parcialmente correcto, pero no totalmente correcto.
- ☐ sirven para optimizar el rendimiento de la aplicación.
- ☒ fuerza a que la lectura/escritura de estas variables se haga en memoria principal.

La respuesta correcta es: fuerza a que la lectura/escritura de estas variables se haga en memoria principal.

Pregunta **6**

Finalizado

Puntúa como
1,00

El uso de un enfoque paralelo en la resolución de un problema da lugar a

Seleccione una:

- ☐ una mejora del rendimiento de la solución paralela frente a la secuencial, siempre.
- ☐ no mejorar nunca el rendimiento.
- ☒ una mejora del rendimiento dependiente de la naturaleza del problema.
- ☐ una mejora del rendimiento dependiente, exclusivamente, del número de cores de la máquina.

La respuesta correcta es: una mejora del rendimiento dependiente de la naturaleza del problema.

Pregunta **7**

Finalizado

Puntúa como
1,00

Una hebra soportada en Java mediante una lambda-expresión

Seleccione una:

- ☐ Puede estar detenida en un punto de su código por un bloqueo de exclusión mutua, mientras ejecuta código de una zona diferente.
- ☐ Puede acceder a varios objetos diferentes de forma concurrente.
- ☒ Puede ser procesada a través de un ejecutor.
- ☐ tiene un ciclo de vida distinto a una hebra soportada mediante Runnable.

La respuesta correcta es: Puede ser procesada a través de un ejecutor.

Pregunta **8**

Sin contestar

Puntúa como
1,00

Considere un problema de naturaleza numérica que se ha resuelto mediante procesamiento paralelo utilizando un procesador multicore de 16 cores, y donde no se han identificado latencias conocidas; se sabe que la paralelización ha sido bien realizada, y las pruebas realizadas para un rango de tareas paralelas de 2 a 32 ilustran un speedup máximo de 0.98 para 2 tareas. Usted deduce:

Seleccione una:

- ☐ que el coeficiente de bloqueo del problema está próximo a uno.
- ☐ que el problema es inherentemente no paralelizable.
- ☐ que el análisis debería haber probado un rango de tareas superior a 32.
- ☐ nada de lo anterior es cierto.

La respuesta correcta es: que el problema es inherentemente no paralelizable.

Pregunta **9**

Finalizado

Puntúa como
1,00

Usted debe desarrollar una aplicación bajo el framework RMI que controla un sistema de reserva de billetes de avión, y sabe:

Seleccione una:

- ☐ Que puede situar DNS y objetos servidores en máquinas diferentes.
- ☒ Que habrá que utilizar necesariamente control de exclusión mutua explícito en el lado del servidor.
- ☐ Que deberá crear una hebra de servicio por cada petición procedente de un cliente.
- ☐ Que utilizando generación dinámica de resguardos puede prescindir de utilizar un DNS local a la máquina que ejecuta el servidor.

La respuesta correcta es: Que habrá que utilizar necesariamente control de exclusión mutua explícito en el lado del servidor.

Pregunta **10**

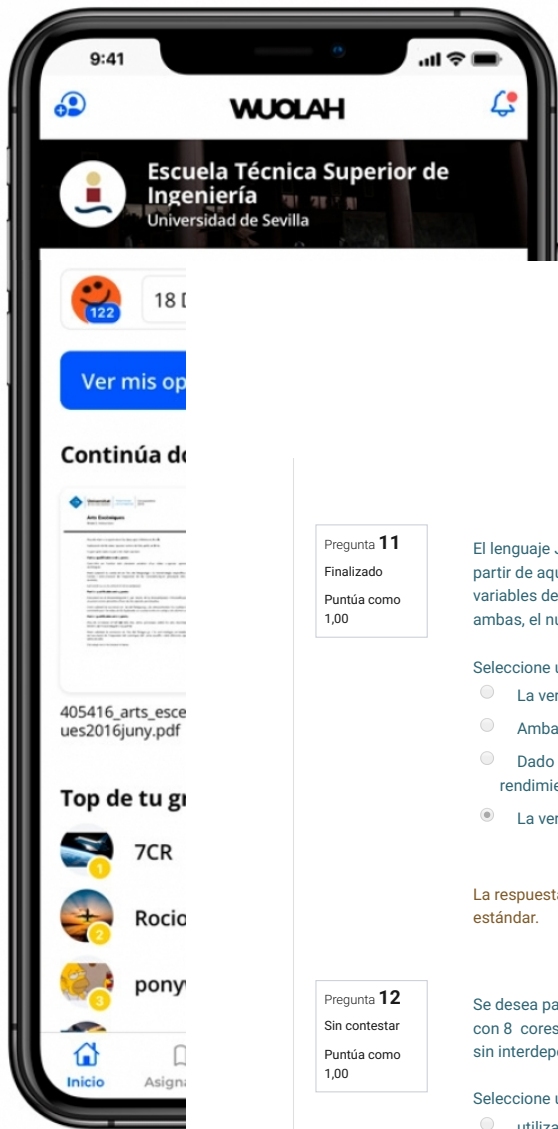
Finalizado

Puntúa como
1,00

Se programa en Java una clase M que consta de 5 métodos de los cuales 3 de ellos están sincronizados mediante un cerrojo de clase `ReentrantLock`, y dos no. La clase define un conjunto de atributos. A continuación se escribe un programa que instancia un objeto O de clase M, y crea hebras que comparten el acceso a O mediante variables de referencia. En esas condiciones:

Seleccione una:

- ☐ El objeto O implementa de manera adecuada el concepto teórico de monitor.
- ☐ Existe exclusión mutua sobre los datos que el objeto O encapsula.
- ☒ Es posible tener en tiempo t múltiples hebras accediendo al objeto O.
- ☐ No existe reentrancia para aquellas hebras que acceden al objeto O a través de método que están sincronizados.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Pregunta 11

Finalizado

Puntúa como
1,00

El lenguaje Java soporta la implementación del concepto teórico de monitor mediante el los API estándar y de alto nivel. A partir de aquí, considere un problema concurrente resuelto mediante un monitor teórico que requiere la presencia de cinco variables de condición. A continuación se escriben implementaciones con Java de este monitor con ambos API, y para ambas, el número de tareas que utilizan ambas implementaciones es elevado. Entonces

Seleccione una:

- ☐ La versión del monitor con el API de alto nivel resultará menos eficiente que la versión con el API estándar.
- ☐ Ambas versiones tendrá una eficiencia equivalente.
- ☐ Dado que con ambas API, en tiempo solo hay una hebra activa en el monitor, no tiene sentido especular sobre el rendimiento.
- ☒ La versión del monitor con el API de alto nivel resultará más eficiente que la versión con el API estándar.

La respuesta correcta es: La versión del monitor con el API de alto nivel resultará más eficiente que la versión con el API estándar.

Pregunta 12

Sin contestar

Puntúa como
1,00

Se desea paralelizar un algoritmo para el que se ha determinado que $C_b=0.8$ y que tiene que trabajar sobre una máquina con 8 cores lógicos, para procesar 60 tareas con apertura de sockets a un servidor remoto y posterior descarga de datos, sin interdependencias conocidas entre ellas. Con estos datos, usted decide:

Seleccione una:

- ☐ utilizar un contenedor sincronizado de tareas para optimizar el rendimiento.
- ☐ utilizar 60 tareas y procesarlas utilizando un objeto de clase `ThreadPoolExecutor` con `corePoolSize=4`.
- ☐ utilizar 60 tareas y procesarlas utilizando un objeto de clase `ThreadPoolExecutor` con `corePoolSize=40`.
- ☐ utilizar 60 tareas y procesarlas utilizando un objeto de clase `ThreadPoolExecutor` con `corePoolSize=16`.

La respuesta correcta es: utilizar 60 tareas y procesarlas utilizando un objeto de clase `ThreadPoolExecutor` con `corePoolSize=40`.

Pregunta 13

Finalizado

Puntúa como
1,00

Para programar una solución a un problema con coeficiente de bloqueo $C_b=0$ utilizando el lenguaje Python usted escogería una aproximación

Seleccione una:

- ☐ de paralelismo con procesos, compartiendo memoria.
- ☒ de paralelismo con procesos, comunicándolos mediante mensajes.
- ☐ de paralelismo con hebras, compartiendo memoria y utilizando bloqueos de exclusión mutua cuando sea necesario.
- ☐ el Global Interpreter Lock (GIL) impide que Python desarrolle un paralelismo real de ninguna forma.

La respuesta correcta es: de paralelismo con procesos, comunicándolos mediante mensajes.

Pregunta 14

Finalizado

Puntúa como
1,00

La técnica de paralelismo de datos mediante división del dominio de datos entre múltiples tareas paralelas puede desarrollarse adecuadamente

Seleccione una:

- ☒ con los lenguajes Java, C++, Python y el API MPJ-Express para Java.
- ☐ con Python utilizando tareas soportadas mediante objetos de la clase `Thread`.
- ☐ con los lenguajes Java, C++, Python pero no con el API MPJ-Express para Java.
- ☐ únicamente con Python cuando se dispone de una GPU.

La respuesta correcta es: con los lenguajes Java, C++, Python pero no con el API MPJ-Express para Java.

Pregunta **15**
Sin contestar
Puntúa como
1,00

El multiprocesamiento simétrico que utiliza paralelismo mediante división del dominio de datos entre las tareas

Seleccione una:

- ☐ es propio de y limitado al lenguaje Java.
- ☐ es un modelo de programación paralela.
- ☐ solo tiene sentido en lenguajes de programación sin orientación a objetos.
- ☐ tiene sentido en problemas con coeficiente de bloqueo con un valor próximo a la unidad.

La respuesta correcta es: es un modelo de programación paralela.

Pregunta **16**
Finalizado
Puntúa como
1,00

Los monitores escritos en lenguaje C++

Seleccione una:

- ☒ permiten varias colas de espera por condición.
- ☐ permiten únicamente una cola de espera por condición, con señalización SU.
- ☐ funcionan únicamente para proteger recursos compartidos bajo control de exclusión mutua, igual que una región crítica.
- ☐ funcionan igual que los monitores del lenguaje Python.

La respuesta correcta es: permiten varias colas de espera por condición.

Pregunta **17**
Finalizado
Puntúa como
1,00

Cuando se utiliza el modelo de memoria transaccional software en Java sobre Clojure:

Seleccione una:

- ☐ el delimitador sintáctico `dosync` gestiona el acceso a regiones críticas bajo transacciones gestionadas por el manejador de transacciones.
- ☐ los interbloqueos entre tareas concurrentes siguen sin poder evitarse.
- ☐ puede haber inconsistencias de los datos compartidos entre tareas que están protegidos bajo transacciones.
- ☒ es necesario utilizar a nivel sintáctico, el delimitador `LockingTransaction.runInTransaction()`.

La respuesta correcta es: es necesario utilizar a nivel sintáctico, el delimitador `LockingTransaction.runInTransaction()`.

Pregunta **18**
Finalizado
Puntúa como
1,00

Cuando se programa el control de la exclusión mutua con las primitivas disponibles para ello en los lenguajes de programación Java, C++ y Python, la característica de "reentrancia"

Seleccione una:

- ☐ es inherente a todas las primitivas para los tres lenguajes.
- ☒ no es necesariamente inherente a todas las primitivas de los tres lenguajes.
- ☐ depende de cómo estén soportados las hebras en los diferentes lenguajes.
- ☐ depende del modelo de memoria que cada lenguaje implementa.

La respuesta correcta es: no es necesariamente inherente a todas las primitivas de los tres lenguajes.

Pregunta **19**

Finalizado

Puntúa como
1,00

Se desea paralelizar un algoritmo para el que se ha determinado un coeficiente de bloqueo $C_b=0$, y que tiene que trabajar sobre una máquina con 8 cores lógicos, para procesar un vector de 10^6 componentes sin interdependencia conocidas entre ellas. Con estos datos, usted decide:

Seleccione una:

- ☐ utilizar un contenedor sincronizado para modelar el vector y optimizar el rendimiento.
- ☐ utilizar 16 tareas con 16 sub-vectores disjuntos y procesarlas utilizando un objeto de clase `ThreadPoolExecutor` con `corePoolSize=4`.
- ☐ utilizar utilizar 16 tareas con 16 subvectores y procesarlas utilizando un ejecutor de tamaño fijo mediante `newFixedThreadPool(16)`.
- ☒ ninguna de las respuestas anteriores es correcta-

La respuesta correcta es: ninguna de las respuestas anteriores es correcta-

Pregunta **20**

Finalizado

Puntúa como
1,00

La planificación mediante ejecutivo cíclico en un sistema de tiempo real

Seleccione una:

- ☐ se realiza on-line de forma dinámica durante la ejecución del sistema
- ☒ es completamente segura frente a la concurrencia sin necesidad de programar un control explícito de exclusión mutua.
- ☐ tiene un hiperperiodo $T_m = mcd(T_1, T_2, \dots, T_m)$
- ☐ aprovecha todos los cores disponibles de la plataforma.

La respuesta correcta es: es completamente segura frente a la concurrencia sin necesidad de programar un control explícito de exclusión mutua.

Ir a...

Correo →

¿NECESITAS AYUDA?



[CAU Campus Virtual](#)



campus.virtual@uca.es



Horario de 9:00 a 14:00 L-V
956 01 67 55



[Canal YouTube](#)

WUOLAH

Descarga la app de Wuolah desde tu store favorita