

Prueba objetiva 1 (4F2M) - Clave a
Concurrencia
2010-2011 - Segundo semestre
Dpto. de Lenguajes, Sistemas Informáticos e Ingeniería de Software

Normas

Este es un cuestionario que consta de **3 preguntas de respuesta simple** y **una pregunta de desarrollo** en **4 páginas**. La puntuación total del examen es de **10 puntos**. La duración total es de **una hora**. El examen debe contestarse en las **mismas hojas**. No olvidéis rellenar **apellidos, nombre y número de matrícula** en cada hoja.

Sólo hay una respuesta válida a cada pregunta de respuesta simple. Toda pregunta en que se marque más de una respuesta se considerará incorrectamente contestada. Toda pregunta incorrectamente contestada restará del examen una cantidad de puntos igual a la puntuación de la pregunta dividido por el número de alternativas ofrecidas en la misma.

Cuestionario

(2 puntos) 1. Dado el siguiente **CTAD** (sólo se muestran las partes necesarias):

TIPO: $LE = (l : \mathbb{Z} \times e : \mathbb{Z})$

INVARIANTE: $\forall r \in LE \bullet r.e \geq 0 \wedge r.l \geq 0 \wedge r.e \leq 1 \wedge ((r.e > 0 \Rightarrow r.l = 0) \wedge (r.l > 0 \Rightarrow r.e = 0))$

INICIAL(r): $r = (0, 0)$

CPRE: $r.e = 0$

IL(r)

POST: $r = (r^{pre}.l + 1, r^{pre}.e)$

CPRE: $r.e = 0 \wedge r.l = 0$

IE(r)

POST: $r = (r^{pre}.l, r^{pre}.e + 1)$

CPRE: *cierto*

FL(r)

POST: $r = (r^{pre}.l - 1, r^{pre}.e)$

CPRE: *cierto*

FE(r)

POST: $r = (r^{pre}.l, r^{pre}.e - 1)$

Se pide marcar la afirmación correcta:

- (a) El recurso no puede pasar por más de tres estados.
- (b) Sin restricciones sobre los esquemas de llamadas de los procesos sobre un recurso r el recurso no violará la invariante de r .
- (c) Si los procesos respetan los esquemas de llamadas $IL(r)$; ...; $FL(r)$ y $IE(r)$; ...; $FE(r)$ no se violará la invariante de r .
- (d) Ninguna de las otras respuestas es correcta.

PISTA: Dibujar el grafo de estados en este hueco:

- (2 puntos) 2. Dado el siguiente programa concurrente en el que `Dato` es una clase cuyos atributos internos son de tipo `int` y `m` un método de los objetos de dicha clase que sólo trabaja con dichos atributos:

<pre>static Dato x = new Dato();</pre>	
<pre>static class T extends Thread { private Dato y; public T (Dato y) { this.y = y; } }</pre>	<pre>public void run() { Dato z = new Dato(); x.m(); y.m(); z.m(); }</pre>
<pre>// Programa principal Dato w = new Dato(); Thread[] t = new Thread[] {new T(w), new T(w)}; t[0].start(); t[1].start(); t[0].join(); t[1].join();</pre>	

Se pide marcar la afirmación correcta.

- (a) `x.m()` no es una sección crítica.
- (b) `y.m()` no es una sección crítica.
- (c) `z.m()` no es una sección crítica.
- (d) Ninguna de las otras respuestas es correcta.

- (2 puntos) 3. Dadas las clases de threads `T_1`, `T_2` y `T_3`, que comparten una variable `t`, se lanzan sendas instancias `t_1`, `t_2` y `t_3`:

<pre>static int t = 3; static Semaphore s_1 = new Semaphore(1); static Semaphore s_2 = new Semaphore(0); static Semaphore s_3 = new Semaphore(0); static class T_1 extends Thread { public void run() { s_1.acquire(); s_2.acquire(); t = t + 2; s_1.release(); } }</pre>	<pre>static class T_2 extends Thread { public void run() { s_2.acquire(); s_3.acquire(); t = t * 2; s_2.release(); } } static class T_3 extends Thread { public void run() { s_3.release(); s_1.acquire(); t = t * t; s_1.release(); s_2.release(); } }</pre>
--	--

Se pide marcar la respuesta correcta:

- (a) No está garantizada la exclusión mutua en el acceso a la variable `t`.
- (b) Está asegurada la terminación de la tarea `t_3`.
- (c) Tras terminar las tres tareas, la variable `t` puede tomar tanto el valor 20 como el 22.
- (d) No está garantizada la terminación de las tres tareas.

- (4 puntos) 4. Se quiere especificar un *buffer* con capacidad para almacenar un dato de tipo *A* y un dato de tipo *B*, es decir, en el *buffer* puede haber simultáneamente un dato de cada tipo y a lo sumo un dato de cada tipo. Dicho recurso lo utilizarán procesos productores que almacenan datos de tipo *A* o de tipo *B* y procesos que consumidores que querrán extraer o bien datos de tipo *A* o bien datos de tipo *B* o bien datos de cualquiera de los tipos. El interfaz es el siguiente:

C-TAD BufferAB**OPERACIONES**

ACCIÓN AlmacenarA: $BufferAB[io] \times A[i]$

ACCIÓN AlmacenarB: $BufferAB[io] \times B[i]$

ACCIÓN ExtraerA: $BufferAB[io] \times A[o]$

ACCIÓN ExtraerB: $BufferAB[io] \times B[o]$

ACCIÓN ExtraerAoB: $BufferAB[io] \times \mathbb{B}[o] \times A[o] \times B[o]$

Se pide especificar en esta página el dominio (tipos e invariante), estado inicial y las CPREs y POSTs de las operaciones *AlmacenarA*, *ExtraerB* y *ExtraerAoB*. donde el segundo parámetro de la operación *ExtraerAoB* indica cuando es Cierto que el dato extraído es de tipo *A* (en el tercer parámetro) y cuando es Falso que el dato extraído es de tipo *B* (cuarto parámetro).

(Utiliza esta página para trabajar en sucio si lo necesitas.)