

Relación de problemas 4 (temas 4 y 5)

1. Indica cuál sería el proceso de ejecución de las siguientes sentencias y el resultado final sobre el átomo *a* según:

- el algoritmo de ordenación parcial modificado y
 - el algoritmo de ordenación de lecturas frente a escrituras
- suponiendo que las referencias de átomo se encuentran inicialmente a 0.

a=3, read(*T*₅,*a*), read(*T*₃,*a*), write(*T*₄,*a*=8), write(*T*₇,*a*=10), write(*T*₆,*a*=12)

2. Indica el mecanismo de recuperación de un sistema ante las siguientes situaciones:

- Fallo en (2)
- Fallo en (1) con savepoint en (2)

T_id	Estado	Dato	V. anterior	V. nuevo
T ₁	<i>starts</i>			
T ₁	<i>update</i>	S	60	61
T ₁	<i>commits</i>			
T ₂	<i>starts</i>			
T ₂	<i>update</i>	A	61	62
T ₃	<i>starts</i>			
T ₃	<i>update</i>	B	20	21
T ₄	<i>starts</i>			
T ₃	<i>update</i>	D	40	41
T ₄	<i>update</i>	F	70	71
T ₃	<i>commits</i>			
T ₂	<i>update</i>	E	50	51
(2)				
T ₂	<i>commits</i>			
(1)				
T ₄	<i>update</i>	B	21	22
T ₄	<i>commits</i>			

3. Dadas las referencias de átomo siguientes RR(*a*)=5 y WR(*a*)=8, y suponiendo que el algoritmo de control de concurrencia es ordenación parcial modificado, di qué sucede ante cada una de las siguientes operaciones independientemente:

- lee (*T*₆, *a*)
- escribe (*T*₆, *a*)
- lee (*T*₈, *a*)
- escribe (*T*₉, *a*)

4. Suponiendo el siguiente orden de ejecución de T1, T2, T3 y T4 sobre los átomos *a*, *b*, *c* y *d*:

lee (T1, a), lee (T1, b), lee (T2, a), escribe (T3, a), lee (T3, b),
lee (T4, c), lee (T4, d), escribe (T4, d), lee (T2, d), lee (T1, c),
escribe (T2, c), escribe (T3, b), escribe (T1, c), lee (T4, a),
escribe (T1, d), escribe (T4, a)

- Dibujar el grafo de precedencia
 - Determinar la serializabilidad del plan y, en su caso, dar la secuencia de transacciones.
5. Sea *P* el plan de ejecución para cuatro transacciones: T1, T2, T3, T4 que se detalla a continuación. En él se muestran las operaciones de lectura y escritura llevadas a cabo por cada transacción.

lee (T1, c), lee (T2, b), lee(T1, d), escribe(T1, c), escribe(T1, d),
lee(T3, c), escribe(T3, c), lee(T2, d), lee(T3, a), lee(T4, c),
escribe(T2, d), escribe(T2, b) , lee(T4, d), escribe(T4, d)

- Dibuja el grafo de precedencia del plan. Justifica cada enlace.
 - Indica si es serializable o no; en caso afirmativo, muestra algún plan en serie equivalente. Justifica la respuesta.
6. Indica cómo sería la ejecución de las siguientes transacciones sobre el átomo *a* según el algoritmo de ordenación parcial modificado, teniendo en cuenta que cada escritura multiplica por 2 el valor anterior y que $a=2$ inicialmente.
- lee (T2, a), escribe(T3, a), lee (T4, a), escribe(T2, a), escribe(T4, a),
escribe(T5, a), lee(T6, a), escribe(T7, a), escribe(T6, a)

7. Para el siguiente plan de ejecución, construye el grafo de precedencia e indica si la ejecución es o no serializable. En caso afirmativo, dar alguna secuencia de transacciones equivalente.

lee(T1,C), escribe(T1,A), lee(T2,C), escribe(T2,C), lee(T3,A),
escribe(T1,C), lee(T4,C), lee(T5,B), escribe(T4,B), escribe(T5,D),
escribe(T8,A), lee(T7,D), lee(T6,D), escribe(T7,B), lee(T8,D)

8. Sean dos transacciones

T1 : lee(*w*, *n*); lee(*x*, *t*), $t := t + n$, escribe(*x*, *t*); lee(*y*, *t*);
 $t := t - n$; escribe(*y*, *t*)

T2 : lee(*w*, *n*); lee(*x*, *t*); $t := t - n$; escribe(*x*, *t*); lee(*y*, *t*);
 $t := t + n$; escribe(*y*, *t*)

¿Conoces alguna técnica para el control de concurrencia que pueda garantizar la serialidad de cualquier plan de ejecución? Razona la respuesta. Muestra un plan de ejecución con T1 y T2 utilizando el método de bloqueo en dos fases, que sea serializable y en las que sus instrucciones estén entremezcladas.