

## Guía de ejercicios de concurrencia y recuperación

1. Defina protocolo de dos fases y sus variantes:
  - a) Básico
  - b) Conservador
  - c) Estricto
  - d) Riguroso
  
2. Ordene los siguientes modos de aislamiento según su nivel de aislamiento: read committed, serializable, repeatable read, read uncommitted.
  
3. El protocolo de bloqueo de dos fases conservador evita cualquier posibilidad de candado mortal (deadlock), pero no hambruna (starvation). Discuta alternativas de solución.
  
4. El protocolo de bloqueo de dos fases básico no evita candado mortal (deadlock) ni hambruna (starvation). Estudie cómo mejorar esta situación a través del uso de prioridades y/u órdenes en la toma de bloqueos. Tenga en consideración cómo definir las prioridades y qué hacer con distintos niveles de granularidad.
  
5. El control de concurrencia optimista transforma las transacciones de manera que cada una realiza primero las lecturas, luego ve si tiene conflictos con otras transacciones. Si no los tiene, escribe y hace commit; de lo contrario aborta. Con las operaciones  $a=a+b$  y  $b=b+a$ , pruebe que el control de concurrencia optimista puede provocar hambruna (starvation).
  
6. Para cada uno de los siguientes planes, indique si es serializable por conflicto y/o vista:

a)

T1	T2
W(A)	
	R(B)
R(B)	
	W(B)
R(C)	
	W(C)
W(B)	
W(C)	

b)

T1	T2
R(A)	
	R(C)
R(B)	
	R(B)
R(C)	
	R(A)
W(C)	
	W(A)

c)

T1	T2	T3	T4
R(A)			
		R(C)	
	R(B)		
W(A)			
			R(D)
		W(C)	
	W(B)		
			W(D)

d)

T1	T2	T3	T4
	R(A)		
	W(B)		
			R(B)
			W(B)
		R(C)	
		W(B)	
R(A)			
R(B)			
R(D)			
W(A)			
W(D)			

e)

T1	T2	T3	T4
R(A)			
		R(A)	
R(B)			
	R(A)		
W(A)			
			R(B)
W(B)			
	W(C)		
		R(C)	
			R(D)
		R(D)	
			W(D)
		W(E)	

f)

T1	T2	T3	T4
R(A)			
	R(A)		
			R(C)
		R(D)	
W(A)			
	W(B)		
			W(C)
		R(C)	
		W(D)	

7. Organice las siguientes transacciones usando bloqueo de dos fases básico y estricto, con bloqueos que pueden ser exclusivos (escritura) o compartidos (lectura), minimizando el tiempo de atención (o inicio) promedio de cada transacción:

- a)  $T1=(R(A),R(B),W(A),R(C),W(C)),$   
 $T2=(R(B),R(C),W(C),W(B))$
- b)  $T1=(R(A),W(A),R(B),W(B),R(C),W(C)),$   
 $T2=(R(B),W(B),R(C),W(C),R(A),W(A)),$   
 $T3=(R(C),W(C),R(D),W(D),R(B),W(B))$
- c)  $T1=(R(A),R(B),W(B),W(C)),$   
 $T2=(R(D),R(E),W(E),W(D)),$   
 $T3=(R(B),R(D),W(B),W(D)),$   
 $T4=(W(A),W(B),W(C),W(D))$

- d)  $T1=(R(A),R(B),W(A),W(B)),$   
 $T2=(R(B),R(A),W(A),W(B)),$   
 $T3=(R(A),R(B),W(B),W(A)),$   
 $T4=(R(B),R(A),W(B),W(A))$

- e)  $T1=(R(A),W(A),W(B),R(C),W(C)),$   
 $T2=(R(C),W(C)),$   
 $T3=(R(B),W(B),W(C),R(D),W(D))$

- f)  $T1=(R(A),R(B),R(C),W(B),W(C)),$   
 $T2=(R(A),R(B),R(C),W(B),W(D)),$   
 $T3=(R(C),R(D),W(C),W(D)),$   
 $T4=(R(A),R(E),W(A),W(D))$

- g)  $T1=(R(A),R(B),R(C),W(A),W(B),W(C)),$   
 $T2=(R(A),R(B),W(C)),$   
 $T3=(R(A),R(C),W(C)),$   
 $T4=(R(A),R(B),W(B))$

8. Organice las siguientes transacciones usando bloqueo de dos fases básico, distinguiendo entre bloqueos compartidos y exclusivos, de manera de generar candados mortales (deadlocks):

- a)  $T1=(R(A),R(B),R(C),W(A)),$   
 $T2=(R(A),R(B),R(C),W(B))$

- b)  $T1=(R(A),W(B)),$   
 $T2=(R(A),W(C)),$   
 $T3=(R(D),W(A)),$   
 $T4=(R(A),W(D))$

- c)  $T1=(R(A),R(B),R(C),W(D)),$   
 $T2=(R(A),R(B),R(C),W(E)),$   
 $T3=(R(D),R(E),R(F),W(F))$

- d)  $T1=(R(A),W(B)),$   
 $T2=(R(B),W(A)),$   
 $T3=(R(C),W(C))$

- e)  $T1=(R(A),W(B)),$   
 $T2=(R(B),W(C)),$   
 $T3=(R(C),W(A))$

- f) T1 ejecuta  $x:=x+1,$   
T2 ejecuta  $y:=y+1,$   
T3 compara  $x==y$

- g) T1 ejecuta `SELECT * FROM autos;`  
T2 ejecuta `UPDATE autos SET ano=ano+1;`  
(Traducir a la notación tradicional y resolver)

9. Indique la relación de equivalencia entre S y S', y la seriabilidad de cada uno, en cada uno de los siguientes casos:

- a) S es equivalente por vista a T, T es equivalente por conflicto a V, V es equivalente por conflicto a S'
- b) S y S' son serializables por conflicto; S es equivalente por conflicto a K, S' es equivalente por conflicto a K', y K y K' son equivalentes por vista.
- c) S y S' son planes seriales.
- d) S y S' son equivalentes por conflicto a K y K', los que son equivalentes por conflicto a J y J', donde J y J' son equivalentes a todos los planes seriales.
- e) S y S' son equivalentes por conflicto a K y K'. T es una concatenación de los planes K y K' ( T=K.K' ), y es serializable por conflicto.
- f) S es serial y equivalente por vista a J. J es serializable por conflicto y equivalente por vista a K. T es un plan que consiste en sólo una transacción. S' es la concatenación de K y T.
- g) El grafo de conflictos de S es idéntico al de S', salvo que tiene todos los arcos apuntando en la dirección contraria. S es serializable por conflicto.
- h) T es serializable por conflicto, y consiste de los planes S y S' intercalados: la primera operación de T es la primera de S, la segunda es la primera de S', la tercera es la segunda de S, la cuarta es la segunda de S' y así sucesivamente.

10. Una base de datos con escritura forzada, esto es, que actualiza los registros de las tablas inmediatamente cuando una transacción escribe, se ha caído. Indique los valores de a, b y c luego de la recuperación, para las siguientes bitácoras (logs):

- a) /\*a=10;b=5;c=15\*/  
checkpoint  
t1:escribe(a=15)  
t2:lee(X=b)  
t1:escribe(b=8)  
t2:escribe(a=X)  
t1:commit

- b) /\*a=1;b=2;c=3\*/  
checkpoint  
t1:lee(X=a)  
t2:lee(Y=a)  
t3:lee(Z=c)  
t1:escribe(c=a)  
t2:escribe(a=Y+1)  
t3:abort  
t4:lee(Z=c)  
t2:lee(Y=b)  
t1:commit

- c) /\*a=5;b=10;c=15\*/  
checkpoint  
t1:escribe(a=10)  
t2:lee(X=a)  
t1:escribe(b=10)  
t2:escribe(a=20-X)  
t2:commit

- d) /\*a=15;b=15;c=15\*/  
checkpoint  
t1:escribe(a=15)  
t2:lee(X=b)  
t1:escribe(Y=a)  
t2:escribe(a=X)  
t3:lee(Z=a)  
t1:abort  
t2:escribe(c=Y)

- e) /\*a=0;b=0;c=0\*/  
checkpoint  
t1:lee(X=a)  
t2:lee(Y=b)  
t1:escribe(b=X)  
t2:escribe(a=Y)  
t1:abort

- f) /\*a=15;b=10;c=15\*/  
checkpoint  
t1:escribe(a=7)  
t2:lee(X=b)  
t2:escribe(a=X)  
t1:escribe(b=8)  
t1:commit  
t2:escribe(c=X+8)  
t3:lee(Z=c)  
t2:abort

11. ¿Qué tipos de serializabilidad asegura el protocolo de dos fases? Comente para los distintos tipos de dos fases. ¿Cuándo aseguran recuperabilidad?

12. Un procesador con ejecución fuera de orden puede organizar las operaciones de manera de maximizar el paralelismo de sus instrucciones (pipelining). Si soporta renombrado de registros (register renaming) puede aumentar aún más el paralelismo al guardar los valores nuevos y antiguos de un registro en registros distintos. Compare ejecución fuera de orden con marcas de tiempo simple (timestamps) y multiversion. ¿Por qué ejecución fuera de orden es mucho menos flexible que marcas de tiempo? O, para ser más precisos, ¿qué tipo de serializabilidad exige ejecución fuera de orden?

13. Comente los problemas de concurrencia que pueden presentar los siguientes escenarios y discuta posibles soluciones desde la perspectiva del control concurrente y el modelo de datos:

- a) Votación electrónica
- b) Un catálogo de productos de un almacén
- c) Campañas políticas
- d) Una base de datos de noticias
- e) Un registro de ventas de años anteriores
- f) Cajeros automáticos (ej. Redbank)
- g) Gestión de inventarios
- h) Coordinación de equipos de fútbol
- i) Control de flotas (ej. camiones, barcos)
- j) Repositorios de software (ej. CVS, SVN)
- k) Operaciones policiales
- l) Emergencias en caso de catástrofes naturales (ej. erupciones volcánicas, maremotos)
- m) La recaudación de impuestos
- n) Un servicio de correo electrónico (ej. gmail)
- o) La actualización de índices en una base de datos
- p) Sistemas conmutados (ej. telefonía)
- q) Registros de patentes industriales

14. Indique si cada uno de los siguientes planes es recuperable:

a)

T1	T2
R(A)	
	R(B)
	W(A)
W(B)	
	commit
commit	

b)

T1	T2	T3
		W(B)
	R(A)	
R(B)		
		abort
	W(B)	
W(A)		
commit		
	commit	

c)

T1	T2	T3	T4
R(A)			
		R(B)	
	R(C)		
		W(C)	
R(B)			
			R(B)
W(C)			
	R(D)		
			R(D)
		W(C)	
			W(B)
	W(A)		
commit			
		commit	
	abort		
			commit