

## Concurrencia

Para el siguiente esquema:

$T_1$	$T_2$	$T_3$	$T_4$
read(A)			
		read(A)	
	read(A)		
write(A)			
			read(B)
write(B)			
	write(C)		
		read(C)	
			read (D)
		read(D)	
			write(D)
		write(D)	

- Genere el grafo de precedencia utilizando el criterio *Reads antes de Writes* y determine si el esquema de ejecución es serializable.
  - Si  $T_j$  lee el valor de un ítem  $A$ , que fue escrito por  $T_i$ , entonces se crea un arco  $T_i \longrightarrow T_j$ .
  - Si  $T_i$  lee el valor de un ítem  $A$  y a continuación  $T_j$  lo modifica entonces existe un arco  $T_i \longrightarrow T_j$ .
- Determine si hay o no transacciones que abortan cuando se aplica el protocolo de *timestamp* y los puntos donde abortan. Considere que a partir del punto donde una transacción aborta, todas las demás operaciones de la transacción desaparecen del esquema de ejecución. Los timestamp de las transacciones son:  $T_1=1$ ,  $T_2=3$ ,  $T_3=2$  y  $T_4=5$ .

El protocolo es:

- Si la transacción  $T_i$  intenta ejecutar un **read**(Q)
  - Si  $T_s(i) < W_{TS}(Q)$  no se ejecuta la operación y  $T_i$  se re-ejecuta.
  - Si  $T_s(i) > W_{TS}(Q)$  se ejecuta la operación y  $R_{TS}(Q) = \max\{R_{TS}(Q), T_s(i)\}$ .
- Si la transacción  $T_i$  intenta ejecutar un **write**(Q)
  - Si  $T_s(i) < R_{TS}(Q)$  no se ejecuta la operación y  $T_i$  se re-ejecuta.
  - Si  $T_s(i) < W_{TS}(Q)$  no se ejecuta la operación y  $T_i$  continúa.
  - Si no, sí se ejecuta la operación y  $W_{TS}(Q) = T_s(i)$ .

Responder la pregunta 2, en la siguiente tabla.

Wts(A)	0								
Rts(A)	0								
Wts(B)	0								
Rts(B)	0								
Wts(C)	0								
Rts(C)	0								
Wts(D)	0								
Rts(D)	0								

Para el esquema de ejecución del protocolo de 2 fases mostrado en la parte inferior de la Figura 1, completar el esquema de ejecución sin sincronización de transacciones. Asuma que en un quantum de CPU solo se puede ejecutar dos instrucciones write o read.

Complete este esquema de ejecución

T1	T2	T3

Protocolo 2 Fases

T1	T2	T3
lock(A;B;C)		
write( A )		
write( B )		
	lock( A )	
		lock( A )
read( C )		
write( C )		
unlock(A;B;C)		
	lock(B;C)	
	write( C )	
	read( A )	
	read( B )	
	write( B )	
	unlock(A;B;C)	
		lock( B )
		read( B )
		write( A )
		unlock(A;B)

Figura 1: Tablas para la pregunta 3.

## Solucion problema anterior

T1	T2	T3
write( A )		
write( B )		
	write( C )	
	read( A )	
		read( B )
		write( A )
**read( C )		
write( C )		
	read( B )	
	write( B )	

TS(1)=1 TS(2)= 2 TS(3)= 3

WTS(Q)

A	0	1	3			4		
B	0	1			2	4		
C	0	2				4		

RTS(Q)

A	0	2						
B	0	3		2				
C	0		T1			4		

### Protocolo 2 Fases

T1	T2	T3
lock(A;B;C)		
write( A )		
write( B )		
	lock( A )	
		lock( A )
read( C )		
write( C )		
unlock(A;B;C)		
	lock(B,C)	
	write( C )	
	read( A )	
	read( B )	
	write( B )	
	unlock(A;B;C)	
		lock( B )
		read( B )
		write( A )
		unlock(A;B)