

# **Bases de Datos**

## **Protocolos de bloqueo: 2 fases y Arbol**



**Dr. Diego R. Garcia**

**DEPARTAMENTO DE CIENCIAS E  
INGENIERÍA DE LA COMPUTACIÓN  
UNIVERSIDAD NACIONAL DEL SUR**



# Protocolos de bloqueo de 2 Fases

## Cada transacción:

- Antes de escribir  $W(X)$  o leer  $R(X)$  un dato  $X$  se debe solicitar y obtener el bloqueo correspondiente sobre  $X$ . Si no logra obtener un bloqueo la transacción espera hasta que ese dato es liberado por otra transacción.
- Debe respetar 2 fases:
  - **Fase de crecimiento**: la transacción solo puede solicitar o aumentar (upgrade) bloqueos
  - **Fase de Decrecimiento**: una vez que una transacción liberó o disminuyó (downgrade) un bloqueo no puede a solicitar o aumentar mas bloqueos, solo puede liberar o disminuir.

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

Analizaremos la planificación anterior puede obtenerse con un protocolo de bloqueo 2 fases utilizando distintas posibilidades de bloqueo.

Para verificar si es posible obtener la planificación agregaremos los bloqueos a la misma **siguiendo la siguiente política**:

- solicitar el bloqueo mínimo permitido para poder ejecutar la instrucción.
- solicitar los bloqueos lo mas tarde posible en la planificación.
- liberar los bloqueos lo antes posible (siempre respetando 2 fases!)

# Protocolos de bloqueo de 2 Fases

Utilizando solo bloqueos exclusivos Lock-X(A) (o abreviado L-X(A)) tanto para leer como para escribir un dato A.

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

T1	T2
L-X(C)	
R(C)	
	L-X(B)
	R(B)
	L-X(C)
...	

T2 no logra obtener Lock-X(C) porque el dato C lo tiene bloqueado T1. Por lo tanto T2 debe esperar y no podrá ejecutar R(C) antes que T1 ejecute R(A). Luego la planificación no se puede obtener utilizando solo Lock-X con un protocolo de 2 fases.

# Protocolos de bloqueo de 2 Fases

Utilizando bloqueos compartidos (Lock-S(A) o L-S(A)) y exclusivos (Lock-X(A) o L-X(A) )

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

T1	T2
L-S(C)	
R(C)	
	L-X(B)
	R(B)
	L-S(C)
	R(C)
L-X(A)	
R(A)	
W(A)	
	L-S(A)

T2 logra obtener el bloqueo porque T1 tiene bloqueado C en modo compartido.

T2 no obtiene Lock-S(A) porque A lo tiene bloqueado T1 en modo exclusivo. T2 debe esperar y no podrá ejecutar R(A) hasta que T1 obtenga L-S(B) para R(B) y pueda comenzar a liberar bloqueos.

T1	T2
L-S(C)	
R(C)	
	L-X(B)
	R(B)
	L-S(C)
	R(C)
L-X(A)	
R(A)	
W(A)	
L-S(B)	

Otra alternativa sería que T1 solicite L-S(B) antes, para poder liberar el bloqueo de A.

Tampoco podría obtener L-S(B) porque T lo tiene bloqueado en modo exclusivo

# Protocolos de bloqueo de 2 Fases

Utilizando bloqueos compartidos (L-S(A)), exclusivos (L-X(A)), upgrade (Up(A)) y downgrade (Do(A)). Utilizaremos Un(A) como abreviatura de Unlock(D).

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

T1 debe desbloquear A para que T2 pueda obtener L-S(A), pero antes debe solicitar L-S(B) que necesitará después. Recuerde que para respetar 2 fases, una vez liberado un bloqueo no se pueden solicitar mas

T1	T2
L-S(C)	
R(C)	
	L-S(B)
	R(B)
	L-S(C)
	R(C)
L-X(A)	
R(A)	
W(A)	
L-S(B)	
Un(A)	
	L-S(A)
	R(A)

T1	T2
...	...
	R(A)
R(B)	
Un(C)	
Un(B)	
	Up(B)
	W(B)
	Un(A)
	Un(B)
	Un(C)

# Protocolos de bloqueo de 2 Fases

Deadlock: ejemplo usando bloqueos compartidos y exclusivos

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

T1 no obtiene el bloqueo debe esperar que T2 libere B

T1	T2
L-S(C)	
R(C)	
	L-X(B)
	R(B)
	L-S(C)
	R(C)
L-X(A)	
R(A)	
W(A)	
	L-S(A)
L-S(B)	

T2 no obtiene el bloqueo debe esperar que T1 libere A

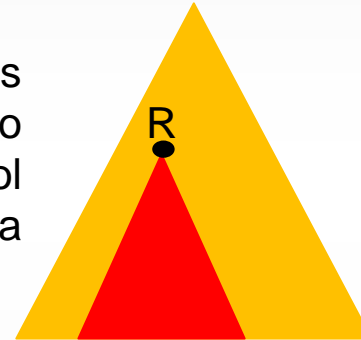
Para respetar 2 fases T2 no puede liberar el bloqueo de B hasta que obtenga el bloqueo de A (mantenido por T1). T1 tampoco puede liberar el bloqueo de A hasta que obtenga el bloqueo de B (mantenido por T1). Por lo tanto **caen en Deadlock**.

# Protocolo de Arbol

Un árbol define el ordenamiento de los bloqueos para todos los datos del sistema, siguiendo las siguientes reglas:

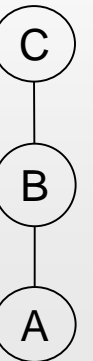
- El único bloqueo permitido es el exclusivo: Lock-X(A) o L-X(A)
- El primer bloqueo de cada transacción puede ser sobre cualquier dato.
- Después de su primer bloqueo una transacción puede bloquear un dato Q si tiene bloqueado al padre de Q en el árbol.

El 1er dato R que bloquea una transacción restringe los datos que podrá bloquear a lo largo de su ejecución. Solo podrá bloquear a los datos descendientes en el sub-árbol que tenga el dato R como raíz, avanzando de padres a hijos.



- Cuando una transacción desbloquea un dato Q ( $Un(Q)$ ) no puede volver a bloquear a Q.
- Los datos pueden desbloquearse en cualquier momento, es decir, puede no respetar 2 fases.

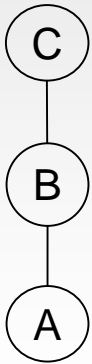
Analizaremos 2 planificaciones distintas con las misma transacciones T1 y T2 que utilizamos para el protocolo de 2 fases. Verificaremos si es posible obtener las planificaciones considerando el árbol:



# Protocolo de Arbol

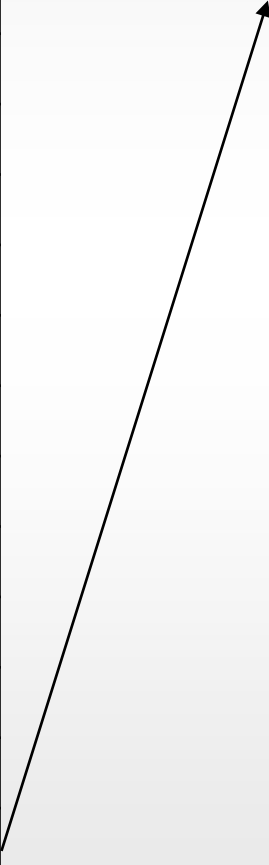
Consideremos la planificación:

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



T1	T2
	L-X(C)
	L-X(B)
	R(B)
	R(C)
	Un(C)
L-X(C)	
R(C)	
	L-X(A)
	R(A)
	Un(A)
	W(B)
	Un(B)
L-X(B)	
L-X(A)	

T1	T2
...	...
L-X(A)	
R(A)	
W(A)	
R(B)	
Un(C)	
Un(B)	
Un(A)	

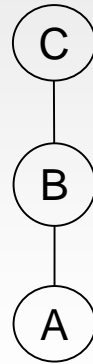




# Protocolo de Arbol

Consideremos la planificación:

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



T1 debe solicitar L-X(B) antes de liberar C (padre de B), dado que no puede bloquear a B si no tiene bloqueado a su padre, y una vez liberado C no se puede volver a bloquear.

T1	T2
L-X(C)	
R(C)	
L-X(B)	
Un(C)	
	L-X(C)
	L-X(B)

T2 no logra obtener el bloqueo porque T1 tiene bloqueado B que no puede liberar hasta el final.