

# Ejercicio

- Considere la siguiente planificación suponiendo que:  $0 < ts(T1) < ts(T2) < ts(T3) < ts(T4)$
- Simule la aplicación del *protocolo de Multiversión*, indicando para cada instante de tiempo el valor de la estampilla de lectura (R-ts) y escritura (W-ts) de la versión correspondiente. Suponga que existen:
  - $\langle A0, R-ts=0, W-ts=0 \rangle$ ,
  - $\langle B0, R-ts=0, W-ts=0 \rangle$ ,
  - $\langle C0, R-ts=0, W-ts=0 \rangle$ .
- Si se produce un retroceso por la violación del protocolo indique en que puntos se produce y que acciones se realizan para recuperarse. Justifique.

# Esquemas Multiversión

- Los **esquemas multiversión**, cada **Write(Q)**, *si es exitoso*, crea una nueva copia de Q, mientras que en cada operación **Read(Q)**, el sistema selecciona que copia de Q será leída.
- Con cada **ítem de dato Q** se asocia una secuencia de **versiones**  $\langle Q_1, Q_2, \dots, Q_n \rangle$ .
- Para cada versión  $Q_k$  se tienen tres ítems de dato:
  - **Content**: es el valor de la versión de Q.
  - **R-ts( $Q_k$ )**: la mayor estampilla de tiempo de una transacción que leyó exitosamente la versión  $Q_k$ .
  - **W-ts( $Q_k$ )**: la estampilla de tiempo de la transacción que creó la versión de  $Q_k$ .
- Un **Write(Q)** *exitoso eventualmente crea una nueva versión de Q*. Los valores de **R-ts** y **W-ts** de la nueva versión se inicializan con  $ts(T_i)$ .
- El valor de **R-ts** se actualiza *si una transacción  $T_j$  lee el contenido de  $Q_k$  y  $R-ts(Q_k) < ts(T_j)$* .

# Multiversión

Supongamos que una transacción  $T_i$  realiza una operación Read o Write. Sea  $Q_k$  la versión de  $Q$  cuya **estampilla de tiempo de escritura** es la más grande estampilla menor o igual a  $ts(T_i)$ .

- 1) Si  $T_i$  realiza un **Read(Q)**, entonces el valor retornado es el contenido de la versión  $Q_k$ .
- 2) Si  $T_i$  realiza un **Write(Q)**, entonces tenemos tres casos:
  - ❑ Si  $ts(T_i) < R-ts(Q_k)$  entonces la transacción  $T_i$  **retrocede**.
  - ❑ Si  $ts(T_i) = W-ts(Q_k)$  entonces se **sobreescribe** el contenido de  $Q_k$ .
  - ❑ En otro caso, se crea una **nueva versión** de  $Q$ .

Tiempo	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	ACCION	Versión -Dato	R-Ts	w-Ts
1		read(B)			Lee con éxito	B0	TS(t2)	0
2	read(A)				Lee con éxito	A0	TS(t1)	0
3			read(A)		Lee con éxito	A0	TS(t3)	0
4		write(B)			Crea una versión con éxito	B1	TS(t2)	TS(t2)
5			write(B)		Crea una versión con éxito	<b>Desaparece la versión B2</b>	<b>Desaparece la versión B2</b>	<b>Desaparece la versión B2</b>
6	write(C)				Crea una versión con éxito	<b>Desaparece la versión C1</b>	<b>Desaparece la versión C1</b>	<b>Desaparece la versión C1</b>
7				write(C)	Crea una versión con éxito	<b>Desaparece la versión C2</b>	<b>Desaparece la versión C2</b>	<b>Desaparece la versión C2</b>
8				read(B)	Lee con éxito	B2	TS(t4)	TS(t3)
9		write(B)			Sobrescribe su versión	B1	TS(t2)	TS(t2)
10			read(C)		Lee con éxito	C1	TS(t3)	TS(t1)
11	write(A)				Falla T1 Retroce cascada T3 Retroce cascada T4			

- ***Falla T1 y retroceden en cascada T3 y T4***
- ***La única transacción que terminaría es T2***
- ***Desaparece la versión C1***
- ***Desaparece la versión B2***
- ***Desaparece la versión C2***