



# Dvanaesta Nedelja

## – Transakcije: Obezbeđenje serijalizovanosti –

**Autori: Miloš Cvetanović**

- **Mehanizam Zaključavanja**
  - Binarno zaključavanje – L(X), UN(X)
  - Kompleksno zaključavanje – LS(X), LX(X), UN(X) , (opciono: DN(X), UP(X))
- **Matrica kompatibilnosti za tipove zaključavanja**
- **Pravilno formirana transakcija**
- **Legalno izvršenje transakcija**
  
- **Dvofazni protokol zaključavanja**
  - Osnovni Dvofazni protokol zaključavanja (binarno zaključavanje)
  - Osnovni Dvofazni protokol zaključavanja (kompleksno zaključavanje)
  - Striktni Dvofazni protokol zaključavanja
- **Protokol u obliku stabla**
- **Mehanizam Vremenskog Markiranja**

## Striktни protokol Vremenskog markiranja

### If T requests to read $x$ :

- **R1:** If  $TS(T) < WTS(x)$ , then T is too old; abort T; rollback T
- **R2:** If  $TS(T) \geq WTS(x)$ , then
  - If the value of  $x$  is committed, grant T's read and if  $TS(T) > RTS(x)$  assign  $TS(T)$  to  $RTS(x)$
  - If the value of  $x$  is not committed ( $C(x)$  is false), T waits (to avoid a dirty read)

### If T requests to write $x$ :

- **W1:** If  $TS(T) < RTS(x)$ , then T is too old; abort T; rollback T
- **W2:** If  $RTS(x) \leq TS(T) < WTS(x)$ , then no transaction that read  $x$  should have read the value T is attempting to write and no transaction will read that value (See R1)
  - If  $x$  is committed ( $C(x)$  is true), grant the request but do not do the write
    - This is called the Thomas Write Rule
  - If  $x$  is not committed, T waits to see if newer value will commit. If it does, discard T's write, else perform it
- **W3:** If  $WTS(x), RTS(x) \leq TS(T)$ , then
  - If  $x$  is committed, grant the request and assign  $TS(T)$  to  $WTS(x)$ , set  $C(x)$  to false
  - If the value of  $x$  is not committed ( $C(x)$  is false), T waits

### If T requests to commit :

- **C1:** Set  $C(x)$  to true for all elements  $x$  written by T, and all transactions that are waiting for  $x$  to be committed are allowed to proceed

### If T requests to abort:

- **A1:** All T's writes has to be cancelled, and all transactions that are waiting for T, to commit or abort, are allowed to proceed

## Primer 6 – Protokol Vremenskog markiranja (1)

- Na slici je prikazan redosled izvršavanja skupa transakcija  $\{T_1, T_2, T_3\}$
- Upotrebom *striktnog protokola Vremenskog markiranja* proveriti da li se dobija isti redosled. Ako vremenske marke transakcija  $T_1$ ,  $T_2$  i  $T_3$  imaju vrednosti 200, 150 i 175, respektivno.

Vreme	$T_1$	$T_2$	$T_3$
$t_1$	Read(B)		
$t_2$		Read(A)	
$t_3$			Read(C)
$t_4$	Write(B)		
$t_5$	Write(A)		
$t_6$	Commit		
$t_7$		Write(C)	
$t_8$		Commit	
$t_9$			Write(A)
$t_{10}$			Commit

# Primer 6 – Protokol Vremenskog markiranja (2)

T	Op	S	RA	WA	CA	RB	WB	CB	RC	WC	CC	T1	T2	T3
			0	0	1	0	0	1	0	0	1	200	150	175
T1	READ (B)	OK				200								
T2	READ (A)	OK	150											
T3	READ (C)	OK							175					
T1	WRITE (B)	OK					200	0						
T1	WRITE(A)	OK		200	0									
T1	COMMIT	OK			1			1						
T2	WRITE (C)	ROLLBACK											225	
T3	WRITE (A)	IGNORE												
T3	COMMIT	OK												
T2	READ (A)	OK	225											
T2	WRITE (C)	OK								225	0			
T2	COMMIT	OK									1			

### If T requests to read x:

- **R1:** if  $TS(T) < WTS(x)$ , then T is too old; abort T; rollback T
- **R2:** if  $TS(T) \geq WTS(x)$ , then
  - **If  $C(x)$  is committed ( $C(x)$  is true), grant T's read and if  $TS(T) > RTS(x)$  assign  $TS(T)$  to  $RTS(x)$**
  - **Else T waits**

### If T requests to write x:

- **W1:** If  $TS(T) < RTS(x)$ , then T is too old; abort T; rollback T
- **W2:** If  $RTS(x) \leq TS(T) < WTS(x)$ 
  - **If  $C(x)$  is committed ( $C(x)$  is true), grant the request but do not do the write (TWR)**
  - **Else T waits**
- **W3:** If  $WTS(x), RTS(x) \leq TS(T)$ , then
  - **If  $C(x)$  is committed, grant the request and assign  $TS(T)$  to  $WTS(x)$  and set  $C(x)$  to false**
  - **Else T waits**