

## Θεματολόγιο

- ✦ Κλειδώματα – 2 Phase Locking
- ✦ Πώς γίνεται στην πράξη;
- ✦ Αδιέξοδα

1

## Πώς ελέγχουμε σειριοποιησιμότητα στην πράξη;

- ✦ Η σειριοποιησιμότητα όψεων είναι πολύ ακριβή για να ελεγχθεί, ούτως ή άλλως...
- ✦ Οι αλγόριθμοι για τον έλεγχο σειριοποιησιμότητας συγκρούσεων παίρνουν  $n^2$ [ενίοτε και καλύτερα] για  $n$  δοσοληψίες.
- ✦ Μόνο που ο έλεγχος αυτός γίνεται **αφού** φτιαχτεί το χρονοπρόγραμμα, ήτοι γίνεται λίγο αργά ...
- ✦ Γι' αυτό και έχουμε αναπτύξει on-line πρωτόκολλα ελέγχου του ταυτοχρονισμού των δοσοληψιών.

2

## Κλειδώματα και πρωτόκολλα

- **Πρωτόκολλο κλειδώματος** είναι ένα συγκεκριμένο είδος πρωτοκόλλων, που διασφαλίζουν την σειριοποιησιμότητα μιας σειράς ενεργειών του DBMS, στη βάση κανόνων για το τι μπορεί[ή δεν μπορεί] να προσπελάσει μια δοσοληψία.
- Η βασική έννοια γύρω από την οποία οργανώνεται ένα τέτοιο πρωτόκολλο, είναι η έννοια του «**κλειδώματος**» (η «κλειδιού» ή «κλειδαριάς» ...) *Στην αγγλική **lock**[ενίοτε, και **key**].*




3

## Κλείδωμα

- **Κλείδωμα** είναι μια μεταβλητή που σχετίζεται με ένα **στοιχειώδες δεδομένο** και **περιγράφει την κατάσταση του**, σε σχέση με πιθανές πράξεις που μπορούν να εφαρμοσθούν σε αυτό.

**Read@T1**

**Written@T2**

ID	Name	
12	XX	
13	YY	
14	ZZ	

*Elmasri &  
Navathe*

4

## 2 Phase Locking – 2PL

- ✦ Ο βασικός αλγόριθμος διαχείρισης του ταυτοχρονισμού δοσοληψιών είναι ο «**Αλγόριθμος Κλειδώματος Δύο Φάσεων**» -- στην αγγλική “2 Phase Locking” ή **2PL**

5

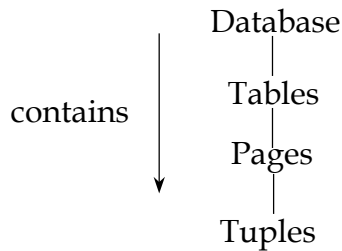
## 2PL

- ✦ Εξασφαλίζει τη σειριοποιησιμότητα επιτρέποντας στις δοσοληψίες να προσπελάσουν **αντικείμενα**, αν καταφέρουν να αποκτήσουν κάποιο κλείδωμα γι’ αυτά.
- ✦ Δύο είδη κλειδώματος:
  - ✦ **Ανάγνωσης**: λαμβάνεις κλείδωμα για να **διαβάσεις** ένα αντικείμενο
  - ✦ **Εγγραφής**: λαμβάνεις κλείδωμα για να **γράψεις ή/και για να διαβάσεις** ένα αντικείμενο

6

## Τι είναι «αντικείμενο»;

- Μπορούμε να κλειδώσουμε τη βάση σε πολλά «επίπεδα»
- Θεωρήστε προς το παρόν **εγγραφές**
- Στην πράξη, συνήθως μιλάμε για **σελίδες**



7

## Πρωτόλειος αλγόριθμος κλειδώματος ≠ 2PL

- Κάθε ενέργεια διαβάσματος/γραψίματος απαιτεί να (περιμένεις μέχρι να) έχεις πάρει το σωστό κλείδωμα
- Για να διαβάσεις, πρέπει να πάρεις ένα **read lock** – το οποίο και θα αποκαλούμε **Shared lock (S-lock)**. Είναι δυνατόν να υπάρχουν πολλές δοσοληψίες με S-lock για το ίδιο αντικείμενο
- Για να γράφεις, πρέπει να πάρεις ένα **write lock** – το οποίο και θα αποκαλούμε **eXclusive lock (X-lock)**. Αν μια δοσοληψία έχει X-lock για ένα αντικείμενο, απαγορεύεται οποιαδήποτε άλλη να έχει κάποιο κλείδωμα για το αντικείμενο αυτό
- Μπορείς να ξεκλειδώνεις (**Unlock**) αντικείμενα, αν δεν τα χρειάζεσαι.

8

## Συμβολισμός

- ✦  $R_T(A)$  : η δοσοληψία  $T$  διαβάζει το αντικείμενο  $A$
- ✦  $W_T(A)$  : η δοσοληψία  $T$  γράφει το αντικείμενο  $A$
- ✦  $S_T(A)$  : η δοσοληψία  $T$  παίρνει S-lock για το αντικείμενο  $A$
- ✦  $X_T(A)$  : η δοσοληψία  $T$  παίρνει X-lock για το αντικείμενο  $A$
- ✦  $U_T(A)$  : η δοσοληψία  $T$  ξεκλειδώνει το αντικείμενο  $A$
- ✦  $COMMIT_T$ : η δοσοληψία  $T$  τερματίζει επιτυχώς
- ✦  $ABORT_T$ : η δοσοληψία  $T$  αποτυγχάνει

9

## Παράδειγμα

$T_7$   
**lock-X(B)**  
read(B)  
 $B := B - 50$   
write(B)  
**unlock(B)**  
**lock-S(A)**  
read(A)  
 $display(A+B)$   
**unlock(A).**

$R_7(B) ; W_7(B) ; R_7(A) .$

$X_7(B) ; R_7(B) ; W_7(B) ; U_7(B) ; S_7(A) ; R_7(A) U_7(A)$

10

## Παράδειγμα

$T_7$   
**lock-X(B)**  
read(B)  
 $B := B - 50$   
write(B)  
**unlock(B)**  
**lock-S(A)**  
read(A)  
display(A+B)  
**unlock(A)**

Βασική υπόθεση: ξέρουμε από πριν όλες τις ενέργειες της δοσοληψίας ...

11

## Συγκρούσεις

- Έστω δύο δοσοληψίες,  $T_1$  και  $T_2$ , εκ των οποίων, η  $T_1$  ζητά ένα κλειδίωμα και η  $T_2$  έχει ήδη ένα κλειδίωμα για το ίδιο αντικείμενο.
- Ο πίνακας συγκρούσεων πρέπει να σας θυμίζει κάτι...
- Άρνηση  $\Rightarrow$  «περίμενε μέχρι να γίνει unlock»

		Lock held	
		Read	Write
Lock requested	$T_1 \setminus T_2$	Read	Write
	Read	Write	☑
Write	Write	☒	☒

12

## 2 ειδών προβλήματα

- Αν προσπαθήσεις να **αυξήσεις** τις ταυτόχρονες δοσοληψίες, **ξεκλειδώνοντας αντικείμενα πολύ νωρίς**, μπορεί να έχεις προβλήματα ασυνέπειας
- Αν προσπαθήσεις να **καθυστερήσεις το ξεκλείδωμα**,
  - μπορεί να προκύψουν **αδιέξοδα** (όπου δύο δοσοληψίες περιμένουν –για πάντα– η μία την άλλη για το ξεκλείδωμα διαφορετικών αντικειμένων)
  - **καθυστερείς** (ενώ έχουμε πιο έξυπνους τρόπους)...

13

## Ξεκλειδώνοντας νωρίς...

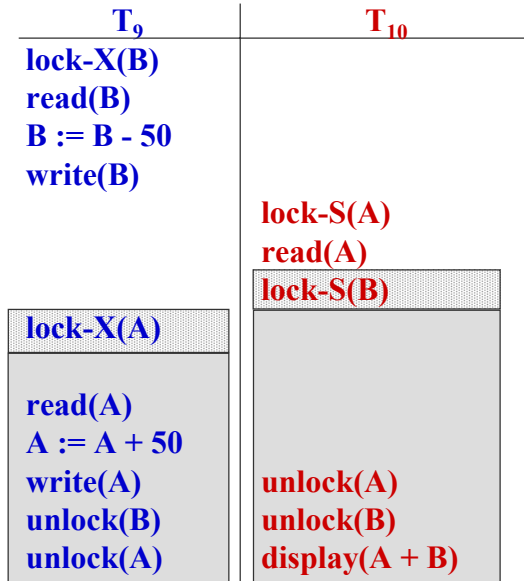
Επειδή η T7 ξεκλειδώνει το B νωρίς, το αποτέλεσμα της T8 είναι λάθος...

T <sub>7</sub>	T <sub>8</sub>
<b>lock-X(B)</b> <b>read(B)</b> <b>B := B - 50</b> <b>write(B)</b> <b>unlock(B)</b>	
<b>lock-X(A)</b> <b>read(A)</b> <b>A := A + 50</b> <b>write(A)</b> <b>unlock(A)</b>	<b>lock-S(A)</b> <b>read(A)</b> <b>unlock(A)</b> <b>lock-S(B)</b> <b>read(B)</b> <b>unlock(B)</b> <b>display(A + B)</b>

14

## Deadlock

Η μία δοσοληψία έχει κλειδώσει το αντικείμενο που χρειάζεται η άλλη. Σταματούν και οι δύο, αναμένοντας η μία την άλλη ...

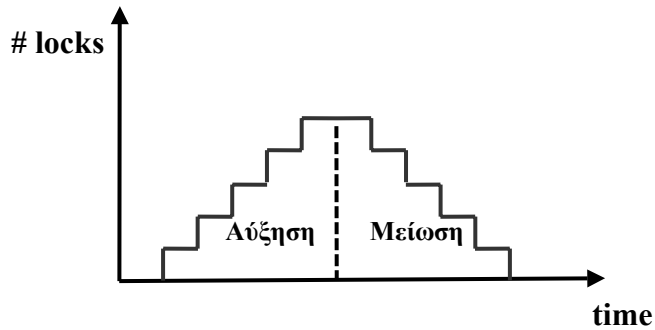


## 2PL

- **Ζήτα κλείδωμα πριν πράξεις:**
  - Για να **διαβάσεις**, πρέπει να πάρεις ένα **Shared lock (S-lock)**. Είναι δυνατόν να υπάρχουν πολλές δοσοληψίες με S-lock για το ίδιο αντικείμενο
  - Για να **γράψεις**, πρέπει να πάρεις ένα **eXclusive lock (X-lock)**. Αν μια δοσοληψία έχει X-lock για ένα αντικείμενο, απαγορεύεται οποιαδήποτε άλλη να έχει κάποιο κλείδωμα για το αντικείμενο αυτό
- Δεν μπορείς να ξεκλειδώσεις αντικείμενο μέχρι να έχεις πάρει και το τελευταίο κλείδωμα που χρειάζεσαι



## 2 Phase Locking



*Ξανά: έτσι και ξεκλειδώσεις αντικείμενο, δεν μπορείς να πάρεις άλλο κλείδωμα!!!*

17

## 2PL;

T <sub>9</sub>	T <sub>10</sub>	T <sub>7</sub>	T <sub>8</sub>
lock-X(B)	lock-S(A)	lock-X(B)	lock-S(A)
read(B)	read(A)	read(B)	read(A)
B := B - 50	lock-S(B)	B := B - 50	unlock(A)
write(B)	unlock(A)	write(B)	lock-S(B)
lock-X(A)	unlock(B)	unlock(B)	unlock(B)
read(A)	display(A + B)	lock-X(A)	display(A + B)
A := A + 50		read(A)	<b>NOT LEGAL</b>
write(A)		A := A + 50	
unlock(B)		write(A)	
unlock(A)		unlock(A)	

**LEGAL**

**LEGAL**

**NOT LEGAL**

KEY
GROWING
SHRINKING
VIOLATION

*Απλοϊκή περίπτωση που κάθε schedule έχει μόνο μία δοσοληψία*

18

## Διαχειριστής κλειδαριών

- Το τμήμα του DBMS που διαχειρίζεται τα κλειδώματα
- Αν το DBMS υποστηρίζει 2PL → **2PL scheduler**
- **Πίνακας κλειδαριών:**
  - ObjectID
    - TransactionID, LockType
    - Queue με δοσοληψίες εν αναμονή

	x	y	z
T1		S	
T2		S	
T3		W	

19

## Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;

S1: R1[x];W2[x];R1[x].

*Τρόπος:*

- Ας δοκιμάσω να βάλω locks & unlocks...
- Έχω μη επιτρεπτές συγκρούσεις;
- Έχουν όλες οι δοσοληψίες τις 2 φάσεις;

20

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

21

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

X1[y];R1[y];

	x	y	z
T1		X	
T2			
T3			

22

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y]; **R2[x]**; W1[y]; W3[y]; W1[z]; **R2[z]**; R3[z].

X1[y]; **R1[y]**; S2[x]; **R2[x]**;

	x	y	z
T1		X	
T2	S		
T3			

23

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y]; **R2[x]**; **W1[y]**; W3[y]; W1[z]; **R2[z]**; R3[z].

X1[y]; **R1[y]**; S2[x]; **R2[x]**; **W1[y]**;

Έχω ήδη το lock...



	x	y	z
T1		X	
T2	S		
T3			

24

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

Unlock κάνω; Το  
T3 θέλει το y

↑  
OXI!!

X1[y];R1[y];S2[x];R2[x];W1[y];

	x	y	z
T1		X	
T2	S		
T3		W	

25

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

Και τι κάνω;

\_\_\_\_\_

X1[y];R1[y];S2[x];R2[x];W1[y];X1[z];U1[y];

	x	y	z
T1			X
T2	S		
T3		W	

26

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

	x	y	z
T1			X
T2	S		
T3		X	

X1[y];R1[y];S2[x];R2[x];W1[y];X1[z];U1[y];X3[y];W3[y];

27

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

	x	y	z
T1			X
T2	S		
T3		X	

Έχω ήδη το lock...

X1[y];R1[y];S2[x];R2[x];W1[y];X1[z];U1[y];X3[y];W3[y];W1[z];

28

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

$R1[y]; R2[x]; W1[y]; W3[y]; W1[z]; R2[z]; R3[z].$

	x	y	z
T1			
T2	S		
T3		X	

$X1[y]; R1[y]; S2[x]; R2[x]; W1[y]; X1[z]; U1[y]; X3[y]; W3[y]; W1[z]; U1[z];$  ← Unlock AMEΣA

29

**Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;**

**S2:**

$R1[y]; R2[x]; W1[y]; W3[y]; W1[z]; R2[z]; R3[z].$

	x	y	z
T1			
T2	S		S
T3		X	

$X1[y]; R1[y]; S2[x]; R2[x]; W1[y]; X1[z]; U1[y]; X3[y]; W3[y]; W1[z]; U1[z]; S2[z]; R2[z];$

30

## Προκύπτει το παρακάτω χρονοπρόγραμμα από αλγόριθμο 2PL ;

S2:

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

ΔΕΝ υπάρχει  
σύγκρουση για S-locks!!

	x	y	z
T1			
T2	S		S
T3		X	S

X1[y];R1[y];S2[x];R2[x];W1[y];X1[z];U1[y];X3[y];W3[y];  
W1[z];U1[z];S2[z];R2[z];S3[z];R3[z];

31

... και ...

S2:

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].

✦ X1[y];R1[y];S2[x];R2[x];W1[y];X1[z];U1[y];X3[y];W3[y];W1[z];U1[z];S2[z];R2[z];S3[z];R3[z];U<sub>123</sub>[all] ...

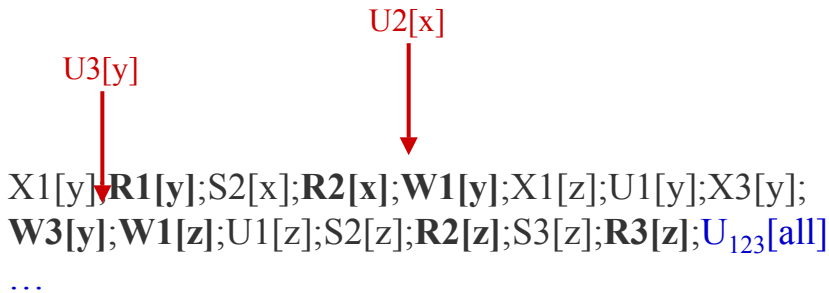
32



## Μπορούσα να είχα κάνει ;

S2:

R1[y];R2[x];W1[y];W3[y];W1[z];R2[z];R3[z].



33

## Θεώρημα

- ✦ Όλα τα χρονοπρογράμματα που προκύπτουν από ένα διαχειριστή κλειδαριών 2PL, έχουν ακυκλικό γράφο σειριοποίησης
- ✦ Το αντίστροφο ΔΕΝ ισχύει.
- ✦ Λήμμα: αν δεν έχει ακυκλικό γράφο σειριοποίησης → δεν είναι 2PL.

34

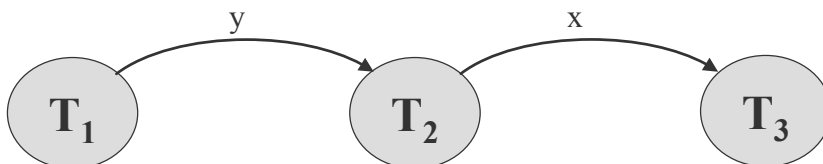
**Προκύπτει το παρακάτω χρονοπρόγραμμα από  
αλγόριθμο 2PL ;**

**S3:** W2[x];W3[x];W1[y];W2[y].

35

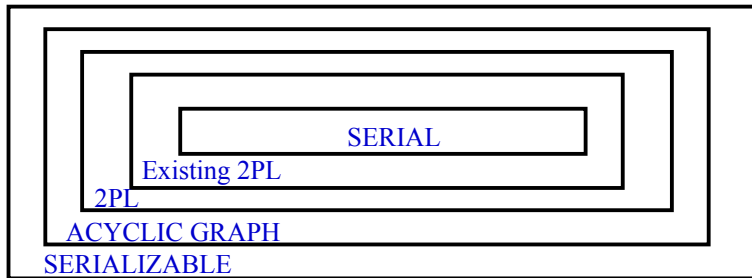
**Μπορώ να το σειριοποιήσω;**

✦ **S3:** W2[x];W3[x];W1[y];W2[y].



36

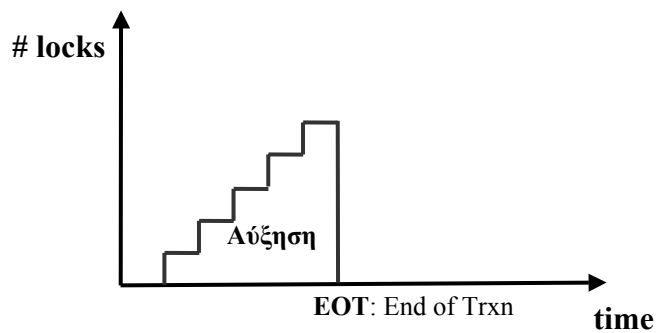
## Venn διάγραμμα



37

## Υπάρχων 2PL;

- ✦ **Αυστηρός 2PL:** όλα τα κλειδώματα αφήνονται στο τέλος της δοσοληψίας, μαζί.



38

## Αυστηρός 2PL

- ✦ Στην πράξη όλα τα συστήματα χρησιμοποιούν αυστηρό 2PL...

[Για το σπίτι] Προκύπτει από Strict 2PL;

$R1[y]; R2[x]; W1[y]; W3[y]; W1[z]; R2[z]; R3[z]$ .

39

## Θεματολόγιο

- ✦ Κλειδώματα – 2 Phase Locking
- ✦ Πώς γίνεται στην πράξη;
- ✦ Αδιέξοδα

40

## Πώς γίνεται στην πράξη;

- Κάθε δοσοληψία ζητά από το διαχειριστή κλειδωμάτων το αντίστοιχο κλειδί
- Αυτός ελέγχει τον πίνακα κλειδωμάτων και αν μπορεί, παραχωρεί το κλειδί και ενημερώνει τον πίνακα
- Αν δεν μπορεί, χρησιμοποιεί μια ουρά αναμονής ανά αντικείμενο, όπου και καταχωρείται η εν λόγω δοσοληψία. Η δοσοληψία περιμένει.

41

## Πώς γίνεται στην πράξη;

- Όταν μια δοσοληψία τελειώνει [επιτυχώς, ή μη], όλα τα κλειδώματα αποδεσμεύονται.
- Όσες δοσοληψίες περιμένουν στην ουρά αναμονής για κάθε κλειδώμα, ενεργοποιούνται και λαμβάνουν τα κλειδώματα με τη σειρά
- Π.χ., αν η ουρά είναι S1 [x] S2 [x] X3 [x] για το x, όταν θα ενεργοποιηθεί η ουρά, η T1 και T2 θα πάρουν το S-lock και η ουρά θα γίνει X3 [x]

42

## Λιμοκτονία -- starvation

- ✦ Έστω ότι η T1 έχει S-lock στο x
- ✦ Έστω ότι η T2 ζητά X-lock στο x → θα μπει στην ουρά αναμονής Q : X2 [x]
- ✦ Έστω ότι, ακολούθως, έρχεται η T3 και ζητά S-lock στο x. Θα το λάβει;
- ✦ **OXI!** Αν ακολουθήσουμε αυτή την πολιτική, είναι πιθανό η T2 να περιμένει για πάντα...
- ✦ Η T3 θα μπει στην ουρά Q : X2 [x] S3 [x]

43

## Ατομικότητα του κλειδώματος

- ✦ Για να διασφαλίζεται η ατομικότητα του κλειδώματος και ξεκλειδώματος:
  - ✦ χρησιμοποιείται σηματοφορέας στην μνήμη για τον πίνακα κλειδωμάτων
  - ✦ χρησιμοποιούνται μικρής διάρκειας **μάνταλα** (latches) για τις **σελίδες** που διαβάζονται ή ενημερώνονται, αν το κλείδωμα γίνεται ανά **εγγραφή** (ουσιαστικά κλειδώνοντας την αντίστοιχη I/O ενέργεια read/write, για να μην υπάρχει σύγκρουση σε επίπεδο σελίδας)

44

## Αναβάθμιση κλειδώματος

- Μέχρι τώρα υποθέταμε ότι κάθε δοσοληψία γνωρίζει εκ προοιμίου όλα τα κλειδώματα που θα ζητήσει.
- Π.χ.  $T: R(A) R(B) W(A)$  θα κλειδώσει με  $X$  το  $A$  στην αρχή, λόγω του  $W(A) \rightarrow$

$$T: X_T(A) R(A) S_T(B) R(B) W(A) U(A) U(B) .$$

- Τι γίνεται όμως αν δεν τα γνωρίζει;
- Η λύση έγκειται στην **αναβάθμιση κλειδώματος** (αντίστοιχα, υποβάθμιση)

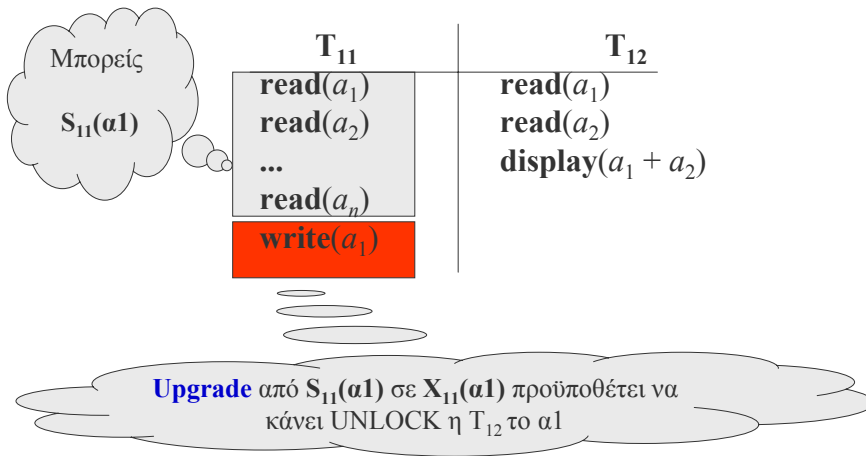
45

## Κανόνες αναβάθμισης

- Αναβαθμίσεις επιτρέπονται μόνο στη φάση αύξησης
- Υποβαθμίσεις επιτρέπονται μόνο στη φάση μείωσης
- Αν υπάρχει και άλλη δοσοληψία που έχει S-lock ένα αντικείμενο, και θες να αναβαθμίσεις από S- σε X-lock, πρέπει να περιμένεις...

46

## Αναβάθμιση



47

## Θεματολόγιο

- ✦ Κλειδώματα – 2 Phase Locking
- ✦ Πώς γίνεται στην πράξη;
- ✦ Αδιέξοδα

48




## Αδιέξοδο

- Η κατάσταση κατά την οποία δύο δοσοληψίες  $T$  και  $T'$  αναμένουν η μία την άλλη για την απελευθέρωση κάποιων κλειδωμάτων
- Εναλλακτικοί όροι: «Λειτουργική παύση» ή “**deadlock**”
- Προφανώς, ο παραπάνω ορισμός γενικεύεται για περισσότερες από δύο δοσοληψίες

49

## Αδιέξοδο

- Θεωρήστε το παρακάτω χρονοπρόγραμμα  
 $S1[x], R1[x], X2[y], W2[y], X2[x], W2[x], X1[y], W1[y]$   

- Η  $T1$  έχει S-lock στο  $x$
- Η  $T2$  έχει X-lock στο  $y$ . Όταν ζητά X-lock στο  $x$ , υποχρεωτικά περιμένει την  $T1$
- Η  $T1$  πάει να ζητήσει X-lock στο  $y$ , οπότε υποχρεωτικά περιμένει την  $T2$  ...

50

## Γράφος αναμονής (blocking graph)

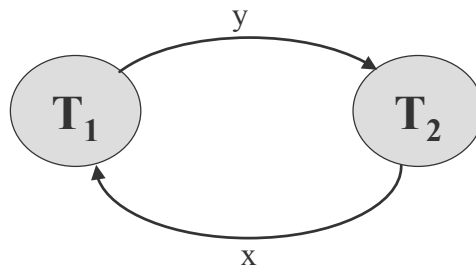
- Για κάθε δοσοληψία και ένας κόμβος
- Μία κατευθυνόμενη ακμή από την δοσοληψία T1 στην δοσοληψία T2, αν η T1 **περιμένει** την T2 να **απελευθερώσει** κάποιο κλείδωμα
- Επίσης γνωστός και ως γράφος **waits-for**

**ΠΡΟΣΟΧΗ:** Δεν είναι ίδιος με το γράφο  
σειριοποιησιμότητας!!!

51

## Αδιέξοδο

- Θεωρήστε το παρακάτω χρονοπρόγραμμα  
S1[x],R1[x],X2[y],W2[y],X2[x],W2[x],X1[y],W1[y]



52

## Θεώρημα

- **Θεώρημα:** Υπάρχει αδιέξοδο αν και μόνο αν ο γράφος αναμονής έχει κύκλο
- Στην πράξη το σύστημα ελέγχει τον γράφο αναμονής περιοδικά
- Εναλλακτικά, αντί για γράφο, μπορεί να μετρά πόση ώρα μια δοσοληψία αναμένει για ένα κλείδωμα και να την τερματίζει αν περάσει κάποιο όριο...

53

## Ανίχνευση ή αποτροπή;

- Αντί να περιμένουμε να συμβεί το αδιέξοδο, μπορούμε να το αποτρέψουμε προληπτικά
- **Συντηρητικός 2PL:** μια δοσοληψία αποκτά όλα τα κλειδώματα που χρειάζεται στο ξεκίνημα της. Αν δεν μπορεί να τα πάρει **ΟΛΑ**, δεν ξεκινά, αλλά αναμένει!

*Εν γένει, προτιμώνται τα αδιέξοδα...*

54

## Ποια δοσοληψία θα «τερματιστεί»;

- Αυτή με τα λιγότερα κλειδώματα ...
- Αυτή με τα περισσότερα κλειδώματα ...
- Αυτή που είναι η πιο πρόσφατη ...
- Αυτή που εκτιμάται ότι χρειάζεται την πιο πολλή δουλειά...