



HY360

Αρχεία και Βάσεις Δεδομένων

Διδάσκων: Δ. Πλεξουσάκης

Συναλλαγές
Διαχείριση Συναλλαγών

Τζικούλης Βασίλειος

Credits: Γιάννης Μακρυδάκης



Συναλλαγές

- Η ταυτόχρονες συναλλαγές (δοσοληψίες, **transactions**) με μια βάση δεδομένων από πολλές διεργασίες / χρήστες είναι απαραίτητη.
- Μια συναλλαγή αποτελεί μια αδιάσπαστη λογική ενότητα ενεργειών.
- Συνήθως μια συναλλαγή περιλαμβάνει μια ακολουθία/σειρά από αναγνώσεις(**reads**) και εγγραφές(**writes**), ενημερώσεις.



Συναλλαγές

- Οι ενέργειες πολλών συναλλαγών πρέπει να εναλλάσσονται για λόγους απόδοσης, και να εκτελούνται ενέργειες από όλες τις «ταυτόχρονες» συναλλαγές.
- Η «εικόνα» των δεδομένων της βάσης όπως αυτή παρουσιάζεται σε μια συναλλαγή είναι αναγκαίο να είναι **συνεπής**, στην διάρκεια της συναλλαγής.
- Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS) πρέπει να εξασφαλίζει τις ιδιότητες **ACID**.
 - **Atomicity** – Ατομικότητα
 - Όλες οι ενέργειες ή καμία
 - **Consistency** – Συνέπεια
 - Οι περιορισμοί πρέπει να ικανοποιούνται είτε εκτελείται μόνη μια συναλλαγή είτε μαζί με άλλες.
 - **Isolation** - Απομόνωση
 - Μια σειριακή εκτέλεση των συναλλαγών θα οδηγούσε στο ίδιο αποτέλεσμα.
 - **Durability** – Μονιμότητα ή Αντοχή
 - Οι αλλαγές που κάνει κάθε συναλλαγή παραμένουν στη βάση.



Συναλλαγές

- Η διαχείριση συναλλαγών ικανοποιεί τις ιδιότητες και θεωρεί κάθε συναλλαγή ως μια ακολουθία από τις ενέργειες:
 - **Read**
 - Ανάγνωση ενός «αντικειμένου» A από την βάση για την συναλλαγή T1. **R1(A)**
 - **Write**
 - Εγγραφή ενός «αντικειμένου» A στην βάση για την συναλλαγή T1. **W1(A)**
 - **Commit**
 - Επιτυχής ολοκλήρωση της συναλλαγής T1. **C1**
 - Οι ενέργειες της συναλλαγής πρέπει να φαίνονται.
 - **Abort**
 - Ανεπιτυχής προσπάθεια εκτέλεσης της συναλλαγής T1. **A1**
 - Οι όποιες ενέργειες της συναλλαγής έχουν ήδη εφαρμοστεί πρέπει να αναιρεθούν.



Διαχείριση Συναλλαγών

- Ένα **χρονοπρόγραμμα** ή **πρόγραμμα εκτέλεσης** (**schedule**) περιλαμβάνει εναλλασσόμενα ενέργειες από ένα σύνολο συναλλαγών.
 - Έστω οι συναλλαγές
 - T1: R(A) R(B) C
 - T2: R(A) W(A) R(B) W(B) C
 - Πιθανά χρονοπρογράμματα:
 - Ένα **πλήρες** schedule είναι:
 - S : R2(A) W2(A) R1(A) R1(B) R2(B) W2(B) C1 C2
 - Ένα **σειριακό** πλήρες schedule είναι:
 - S : R1(A) R1(B) C1 R2(A) W2(A) R2(B) W2(B) C2



Διαχείριση Συναλλαγών

- Όταν δύο χρονοπρογράμματα οδηγούν στα ίδια αποτελέσματα, ανεξάρτητα από την αρχική κατάσταση λέγονται **ισοδύναμα**.
- Ένα χρονοπρόγραμμα λέγεται **σειριακοποιήσιμο**, όταν είναι ισοδύναμο με ένα σειριακό χρονοπρόγραμμα.
- Αποδεκτά είναι μόνο τα σειριακοποιήσιμα!



Διαχείριση Συναλλαγών

- Οι **αντικρουόμενες** ενέργειες είναι όσες
 - Ανήκουν σε διαφορετικές συναλλαγές ΚΑΙ
 - Λειτουργούν στα ίδια δεδομένα ΚΑΙ
 - Μία ενέργεια είναι Write.
- Παράδειγματα
 - αντικρουόμενες **R1(A) W2(A)**
 - Μη επαναλαμβανόμενη ανάγνωση
 - αντικρουόμενες **W1(A) W2(A)**
 - Τυφλή Εγγραφή
 - αντικρουόμενες **W1(A) R2(A)**
 - Ασυνεπής Ανάγνωση
 - μη αντικρουόμενες **R2(A) W2(A)**
 - μη αντικρουόμενες **R1(A) W2(B)**
 - μη αντικρουόμενες **R1(A) R2(A)**



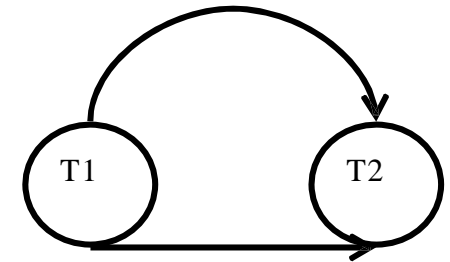
Διαχείριση Συναλλαγών

- Σε δύο ισοδύναμα χρονοπρογράμματα οι αντικρουόμενες ενέργειες εμφανίζονται με την ίδια σειρά.
 - Μη ισοδύναμα:
 - S1 : R2(A) W2(A) R1(A) R1(B) R2(B) W2(B) C1 C2
 - S2 : R1(A) R1(B) C1 R2(A) W2(A) R2(B) W2(B) C2
 - Εδώ
 - $W2(A) \ll_{S1} R1(A)$
 - $R1(A) \ll_{S2} W2(A)$

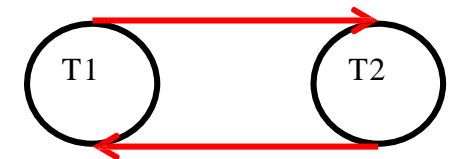


Διαχείριση Συναλλαγών

- Για ένα χρονοπρόγραμμα μπορούμε να κατασκευάσουμε τον γραφό προτεραιότητας.
- Ένας **γραφός προτεραιότητας** είναι ένας κατευθυνόμενος γραφός όπου:
 - Κόμβοι είναι οι δοσοληψίες T
 - Ακμές : μια ακμή $T_i \rightarrow T_j$, για κάθε ζεύγος αντικρουόμενων λειτουργιών για τις οποίες ισχύει ότι: $op_i(X) \ll_s op_j(X)$
- Αν περιέχει κύκλους δεν είναι σειριακοποιήσιμο.
- Παράδειγμα
 - $S1 : R2(A) W2(A) R1(A) R1(B) R2(B) W2(B)$
 - Αντικρουόμενες
 - $W2(A) R1(A)$
 - $R1(B) W2(B)$



Σειριακοποιήσιμο



Μη Σειριακοποιήσιμο



Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
- Για τα παρακάτω προγράμματα εκτέλεσης, κατασκευάσετε τους γράφους προτεραιότητας
 - a) R1(A) R2(A) W1(B) W2(B) R1(B) W2(C) W1(D)
 - b) R1(A) R2(A) R3(B) W1(A) R2(C) R2(B) W2(B) W1(C)
 - c) R1(A) W2(C) W1(B) R3(C) R2(B) W3(A)
 - d) W3(A)R1(A)W1(B)R2(B)W2(C)R3(C)R2(A)
 - e) R1(A)R2(A)R1(B)R2(B)R3(B)W1(A)W2(B)



Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
α) R1(A) R2(A) W1(B) W2(B) R1(B) W2(C) W1(D)



Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
α) $R1(A)$ $R2(A)$ $W1(B)$ $W2(B)$ $R1(B)$ $W2(C)$ $W1(D)$
Αντικρουόμενες:
 - $W1(B)$, $W2(B)$
 - $W2(B)$, $R1(B)$

Υπάρχει κύκλος!
Δεν είναι
σειριακοποιήσιμο





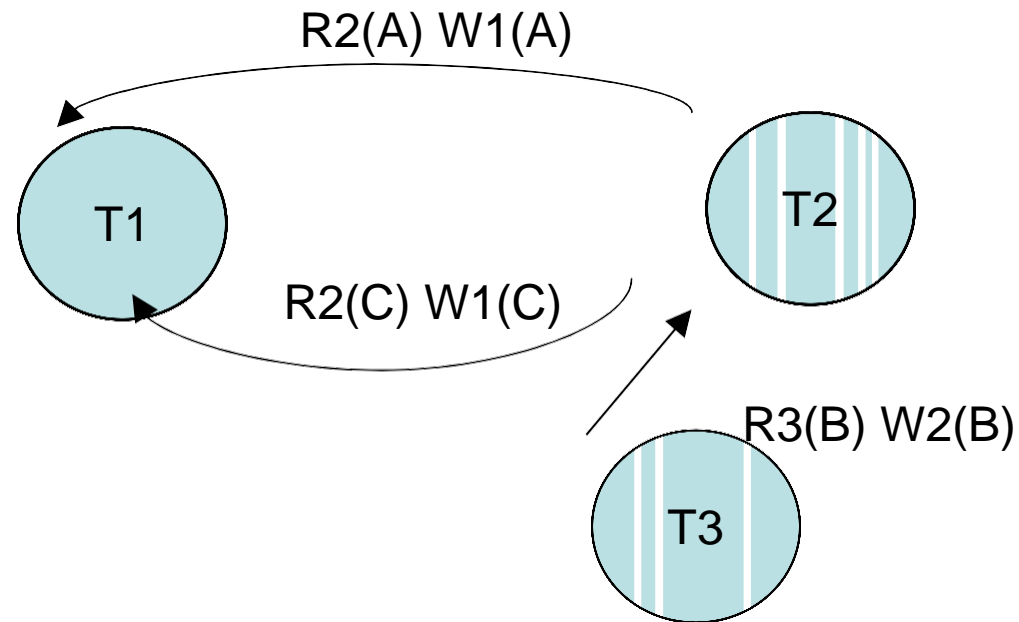
Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
b) R1(A) R2(A) R3(B) W1(A) R2(C) R2(B) W2(B) W1(C)



Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
b) R1(A) R2(A) R3(B) W1(A) R2(C) R2(B) W2(B) W1(C)
– Αντικρουόμενες:
 - R2(A), W1(A)
 - R3(B), W2(B)
 - R2(C), W1(C)





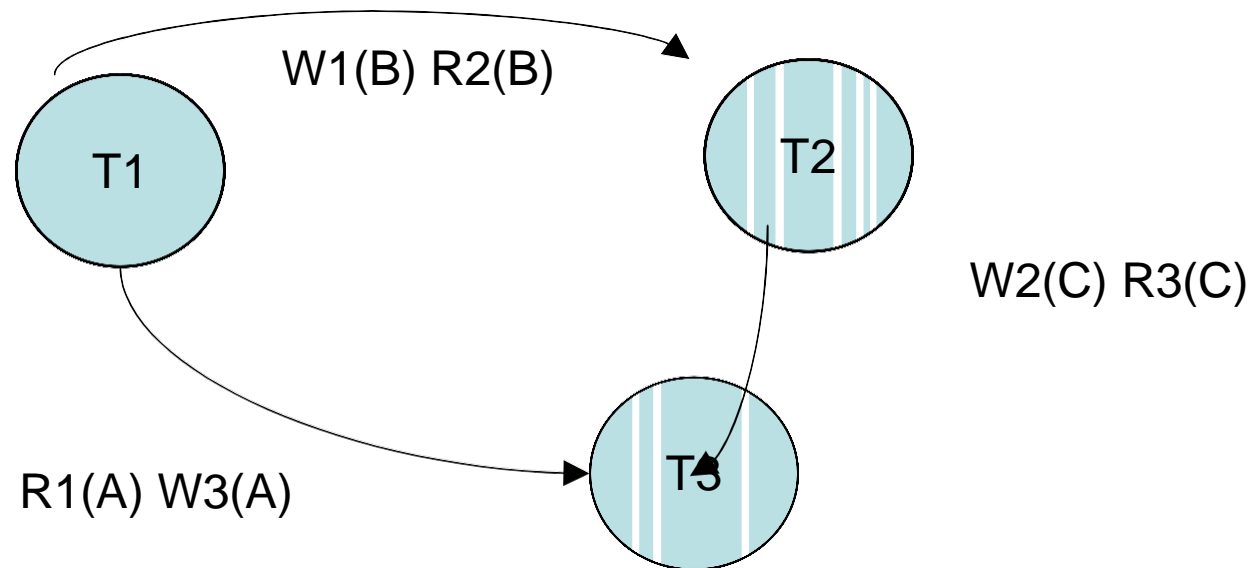
Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
c) R1(A) W2(C) W1(B) R3(C) R2(B) W3(A)



Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
c) R1(A) W2(C) W1(B) R3(C) R2(B) W3(A)
– Αντικρουόμενες:
 - R1(A), W3(A)
 - W2(C), R3(C)
 - W1(B), R2(B)





Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
D) W3(A)R1(A)W1(B)R2(B)W2(C)R3(C)R2(A)



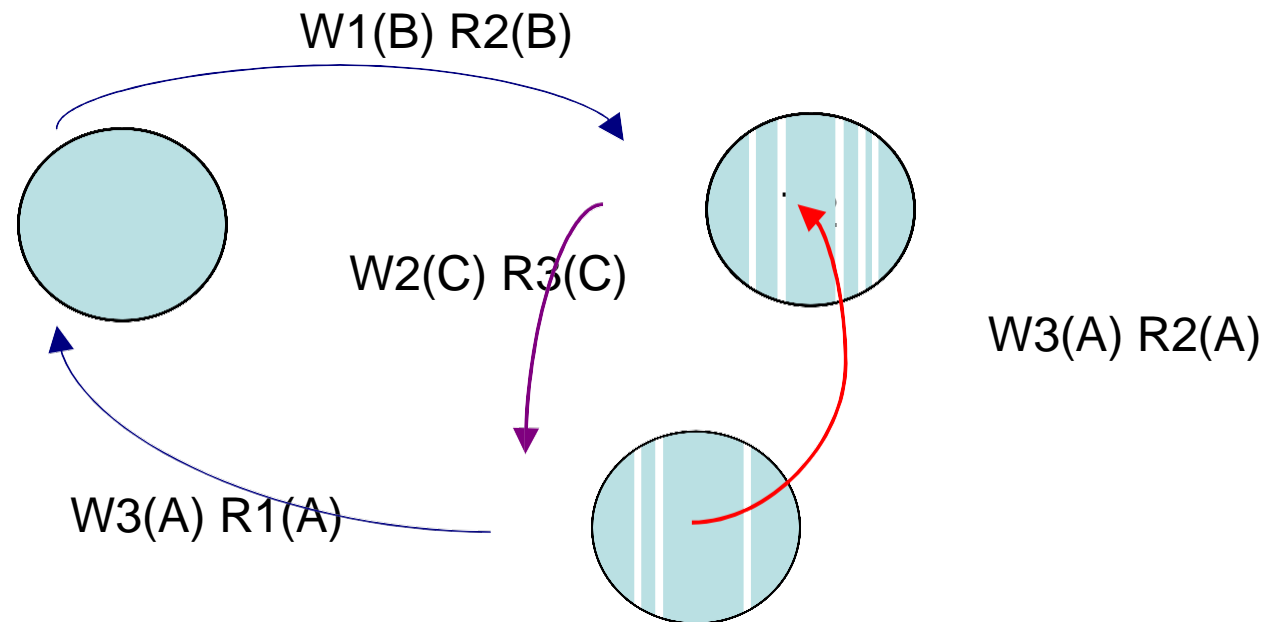
Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
D) $W3(A)R1(A)W1(B)R2(B)W2(C)R3(C)R2(A)$

– Αντικρουόμενες:

- $W3(A), R1(A)$
- $W3(A), R2(A)$
- $W1(B), R2(B)$
- $W2(C), R3(C)$

Υπάρχουν κύκλοι!
Δεν είναι
σειριακοποιήσιμο

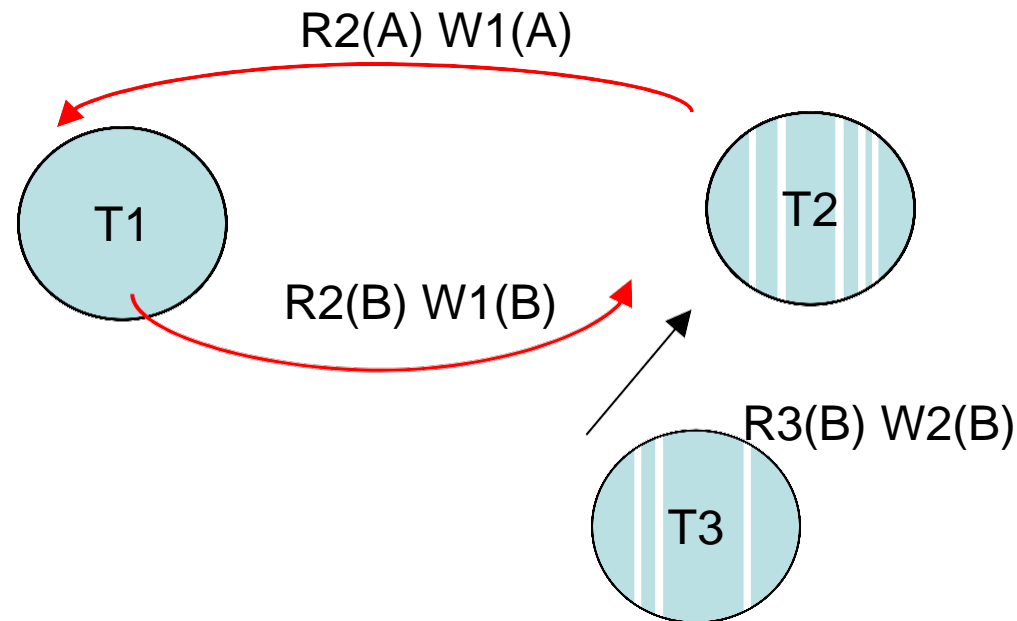




Διαχείριση Συναλλαγών

- Παράδειγμα – γράφος προτεραιότητας
Ε) $R1(A)R2(A)R1(B)R2(B)R3(B)W1(A)W2(B)$
 - Αντικρουόμενες:
 - $R2(A), W1(A)$
 - $R1(B), W2(B)$
 - $R3(B), W2(B)$

Υπάρχει κύκλος!
Δεν είναι
σειριακοποιήσιμο





Διαχείριση Συναλλαγών - locks

- Για να επιτύχουμε σειριακοποιησιμότητα χρησιμοποιούμε **locks**, δικαιώματα πρόσβασης σε κάποιο αντικείμενο μιας βάσης δεδομένων
 - **Shared Lock**: όταν το έχει μια δοσοληψία, μπορούν να αποκτήσουν shared locks και οι υπόλοιπες (συνήθως χρησιμοποιείται για reads)
 - **Exclusive Lock**: όταν το έχει μια δοσοληψία, καμία άλλη δε μπορεί να διαβάσει ή να γράψει το συγκεκριμένο αντικείμενο (συνήθως χρησιμοποιείται για writes)
- Κάθε δοσοληψία πρέπει να αποκτήσει lock για ένα αντικείμενο πριν το προσπελάσει
- Όλα τα αντικείμενα που κλειδώνονται από μια δοσοληψία, πρέπει να ξεκλειδώνονται από αυτή, αλλιώς καμία άλλη δοσοληψία δε μπορεί να αποκτήσει πρόσβαση σε αυτά
- Μια δοσοληψία πρέπει να περιμένει εάν ζητήσει lock σε ένα αντικείμενο που είναι κλειδωμένο από άλλη δοσοληψία



Διαχείριση Συναλλαγών - locks

- Με τη χρήση locks, αποφεύγονται τα προβλήματα που δημιουργούνται μεταξύ αντικρουόμενων λειτουργιών
- Εμφανίζονται όμως νέα προβλήματα
 - **Livelock**: μια δοσοληψία προσπαθεί να αποκτήσει lock σε ένα αντικείμενο, στο οποίο διαδοχικά αποκτούν locks άλλες δοσοληψίες, έτσι περιμένει επ' αόριστον
 - **Deadlock**: για ένα σύνολο δοσοληψιών, κάθε μία περιμένει μία από τις άλλες να κάνει unlock ένα αντικείμενο ώστε να αποκτήσει lock σε αυτό.



Διαχείριση Συναλλαγών – 2PL

- Ένα πρόγραμμα εκτέλεσης ακολουθεί το πρωτόκολλο **Two Phase Locking (2PL)**, αν ισχύουν τα παρακάτω:
 - Όταν μια δοσοληψία θέλει να διαβάσει ή να γράψει κάποιο αντικείμενο, πρέπει να αποκτήσει ένα lock (read RL ή write WL) σε αυτό
 - Εάν ένα αντικείμενο είναι κλειδωμένο με write lock, κάθε δοσοληψία που θέλει να αποκτήσει πρόσβαση σε αυτό πρέπει να περιμένει, δηλαδή το lock είναι αποκλειστικό. (WL = XL)
 - Όλα τα locks που χρειάζεται μια δοσοληψία αποκτώνται κατά τη διάρκεια του **Growing phase (Φάση Ανάπτυξης)**.
 - Όλα τα locks ελευθερώνονται από τη δοσοληψία στο **Shrinking phase. Φάση Συρρίκνωσης**
 - Κανένα lock δε μπορεί να αποκτηθεί κατά τη διάρκεια του shrinking phase. *Συνεπώς, για κάθε δοσοληψία, όλα τα locks πρέπει να προηγούνται του πρώτου unlock.*
- Ένα πρόγραμμα εκτέλεσης που ακολουθεί το 2PL είναι πάντα σειριακοποιήσιμο



Διαχείριση Συναλλαγών – 2PL

- Παράδειγμα
 - Για τα παρακάτω προγράμματα εκτέλεσης, δείξτε πως θα εκτελεστούν με βάση το πρωτόκολλο 2PL. Υποθέστε ότι
 - υπάρχει μόνο αποκλειστικό lock
 - κάθε lock αποκτάται αμέσως πριν από το αντίστοιχο read/write
 - ελευθερώνονται όλα μαζί αμέσως μετά το τελευταίο read/write.
- a) R1(A) R2(A) W1(B) W2(B) R1(B) W2(C) W1(D)
- b) R1(A) R2(A) R3(B) W1(A) R2(C) R2(B) W2(B) W1(C)
- c) R1(A) W2(C) W1(B) R3(C) R2(B) W3(A)
- d) W3(A)R1(A)W1(B)R2(B)W2(C)R3(C)R2(A)
- e) R1(A)R2(A)R1(B)R2(B)R3(B)W1(A)W2(B)



Διαχείριση Συναλλαγών – 2PL

- Παράδειγμα – 2PL

a) R1(A) R2(A) W1(B) W2(B) R1(B) W2(C) W1(D)

T1	T2
L1(A), R1(A)	
	L2(A) NO
L1(B), W1(B), R1(B)	...
L1(D), W1(D)	...
U1(A), U1(B), U1(D)	
	L2(A), R2(A)
	L2(B), W2(B)
	L2(C), W2(C)
	U2(A), U2(B), U2(C)



Διαχείριση Συναλλαγών – 2PL

- Παράδειγμα – 2PL

b) R1(A) R2(A) R3(B) W1(A) R2(C) R2(B) W2(B) W1(C)

T1	T2	T3
L1(A), R1(A)		
	L2(A), NO	
		L3(B), R3(B), U3(B)
W1(A)		
L1(C), W1(C)		
U1(A), U1(C)		
	L2(A), R2(A)	
	L2(C), R2(C)	
	L2(B), R2(B), W2(B)	
	U2(A), U2(C), U2(B)	



Διαχείριση Συναλλαγών – 2PL

- Παράδειγμα – 2PL
c) R1(A) W2(C) W1(B) R3(C) R2(B) W3(A)

T1	T2	T3
L1(A), R1(A)		
	L2(C), W2(C)	
L1(B), W1(B)		
U1(A), U1(B)		
		L3(C), NO
	L2(B), R2(B)	
	U2(C), U2(B)	
		L3(C), R3(C)
		L3(A), W3(A)
		U3(C), U3(A)



Διαχείριση Συναλλαγών – 2PL

- Παράδειγμα – 2PL
d) W3(A)R1(A)W1(B)R2(B)W2(C)R3(C)R2(A)

T1	T2	T3
		L3(A), W3(A)
L1(A), NO		
	L2(B), R2(B)	
	L2(C), W2(C)	
		L3(C), NO
	L2(A), NO	





Διαχείριση Συναλλαγών – 2PL

- Παράδειγμα – 2PL

e) R1(A)R2(A)R1(B)R2(B)R3(B)W1(A)W2(B)

T1	T2	T3
L1(A), R1(A)		
	L2(A), NO	
L1(B), R1(B)		
		L3(B), NO
W1(A)		
U1(A), U2(B)		
	L2(A), R2(A)	
	L2(B), R2(B)	
		L3(B), NO
	W2(B)	
	U2(A), U2(B)	
		L3(B), R3(B), U3(B)



Διαχείριση Συναλλαγών – 2PL

- Πόσα νόμιμα σύγχρονα προγράμματα εκτέλεσης υπάρχουν για τις παρακάτω δύο δοσοληψίες;
 - T1 : L1(A) R1(A) W1(A) L1(B) R1(B) W1(B) U1(A) U1(B)
 - T2 : L2(B) R2(B) W2(B) L2(A) R2(A) W2(A) U2(B) U2(A)

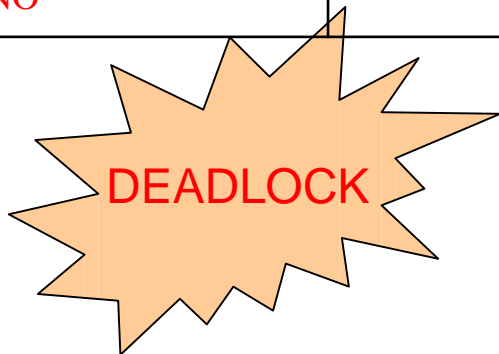


Διαχείριση Συναλλαγών – 2PL

- T1 : L1(A) R1(A) W1(A) L1(B) R1(B) W1(B) U1(A) U1(B)
- T2 : L2(B) R2(B) W2(B) L2(A) R2(A) W2(A) U2(B) U2(A)

T1	T2
L1(A), R1(A) W1(A)	
	L2(B) R2(B) W2(B)
	L2(A) NO
L1(B), NO	

T1	T2
	L2(B) R2(B) W2(B)
L1(A), R1(A) W1(A)	
	L2(A) NO
L1(B), NO	



Άρα μόνο σειριακές εκτελέσεις.

T1, T2

Ή

T2, T1

