

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



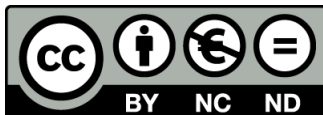
**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Κατανεμημένα Συστήματα

Ενότητα # 3: Ατομικές συναλλαγές

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Κατανεμημένα Συστήματα με Java», Ι. Κάβουρας, Ι. Μήλης, Γ. Ξυλωμένος, Α. Ρουκουνάκη, 3^η έκδοση, 2011, Εκδόσεις Κλειδάριθμος.



Σκοποί ενότητας

- Κατανόηση της έννοιας των συναλλαγών και των διαφόρων κατηγοριών τους.
- Εξοικείωση με τον έλεγχο ταυτοχρονισμού και τις διάφορες παραλλαγές του (κλείδωμα, αισιόδοξος έλεγχος και διάταξη χρονοσφραγίδων).
- Εισαγωγή στις κατανεμημένες συναλλαγές και τα πρωτόκολλα κατανεμημένης δέσμευσης.

Περιεχόμενα ενότητας

- Εισαγωγή
- Κατηγορίες συναλλαγών
- Έλεγχος ταυτοχρονισμού
- Μέθοδοι ελέγχου ταυτοχρονισμού
 - Κλείδωμα
 - Αισιόδοξος έλεγχος ταυτοχρονισμού
 - Διάταξη χρονοσφραγίδων
- Κατανεμημένες συναλλαγές

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Εισαγωγή

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές
Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Τι είναι οι συναλλαγές; (1 από 3)

- Σε τι χρειάζονται οι συναλλαγές;
 - Ταυτόχρονη προσπέλαση δεδομένων
 - Προβλήματα συγχρονισμού και συνέπειας
- Παράδειγμα: τραπεζικό σύστημα
 - `deposit(account, amount)`: κατάθεση
 - `withdraw(account, amount)`: ανάληψη
 - `getBalance(account)`: υπόλοιπο
 - `getBranchTotal()`: άθροισμα υπολοίπων

Τι είναι οι συναλλαγές; (2 από 3)

- Υλοποίηση μεταφοράς ποσού
 - `branch.withdraw (A, 100)`
 - `branch.deposit (B, 100)`
- Πρόβλημα: αν διακοπεί η σύνδεση;
 - Τα χρήματα εξαφανίζονται!
- Λύση: ατομική συναλλαγή
 - Ομαδοποίηση μιας σειράς πράξεων
 - Είτε εκτελούνται όλες
 - Είτε δεν εκτελείται καμία

Τι είναι οι συναλλαγές; (3 από 3)

- Ακύρωση: ακύρωση όλων των πράξεων
 - Αρκεί να αποτύχει έστω και μία πράξη
 - Αναίρεση όλων των ενδιάμεσων πράξεων
 - Επαναφορά αρχικής κατάστασης συστήματος
- Καταγωγή των συναλλαγών
 - Συστήματα με ταινίες (δεκαετία 1960!)
 - Ταινία 1: κατάσταση προηγούμενης ημέρας
 - Ταινία 2: συναλλαγές τρέχουσας ημέρας
 - Ταινία 3: κατάσταση τρέχουσας ημέρας

Χρήση συναλλαγών (1 από 2)

- Ατομικές συναλλαγές σήμερα
 - Μηχανισμός συγχρονισμού υψηλού επιπέδου
 - Κρύβει λεπτομέρειες από προγραμματιστές
 - Παρέχεται από ΛΣ ή περιβάλλον εκτέλεσης
- Θεμελιώδεις κλήσεις συναλλαγών
 - OpenTransaction: ξεκινά μία συναλλαγή
 - Ακολουθεί μία σειρά πράξεων
 - Η σειρά τους φαίνεται σαν μία μόνο ενέργεια

Χρήση συναλλαγών (2 από 2)

- Θεμελιώδεις κλήσεις συναλλαγών
 - CloseTransaction(t): ολοκληρώνει συναλλαγή
 - Στο τέλος των πράξεων
 - Προσπαθεί να κάνει Commit τη συναλλαγή
 - Αν δεν τα καταφέρει την κάνει Abort
 - AbortTransaction(t): ακυρώνει τη συναλλαγή
 - Επαναφορά σε αρχική κατάσταση
 - Δεν περιμένουμε να τελειώσει η σειρά

Ιδιότητες συναλλαγών (1 από 4)

- Ιδιότητες ατομικών συναλλαγών: ACID
 - Atomicity, Consistency, Independence, Durability
- Ατομικότητα (Atomicity)
 - Είτε εκτελούνται όλες οι πράξεις (commit)
 - Είτε ακυρώνονται όλες οι πράξεις (abort)
 - Μοναδική, αδιαίρετη, ακαριαία πράξη
 - Χωρίς ενδιάμεσες καταστάσεις ορατές εξωτερικά
 - Εξωτερικά είναι ορατή η κατάσταση πριν την έναρξη

Ιδιότητες συναλλαγών (2 από 4)

- Συνέπεια (Consistency)
 - Διατηρούν σταθερές του συστήματος
 - Παράδειγμα: έστω κινήσεις σε λογαριασμούς
 - Το υπόλοιπο της τράπεζας δεν αλλάζει
 - Προσωρινές εσωτερικές παραβιάσεις σταθερών
 - Αόρατες στον έξω κόσμο (άλλες συναλλαγές)
 - Μόνο το τελικό αποτέλεσμα είναι ορατό

Ιδιότητες συναλλαγών (3 από 4)

- Απομόνωση (Isolation)
 - Ή Σειριακοποίηση (Serializability)
 - Ταυτόχρονες συναλλαγές δεν αναμιγνύονται
 - Είναι σαν να εκτελέστηκαν ακολουθιακά
 - Δεν έχει σημασία ποια είναι η σειρά εκτέλεσης
 - Αρκεί να υπάρχει αντίστοιχη ακολουθιακή εκτέλεση
 - Απαιτεί κάποιας μορφής έλεγχο ταυτοχρονισμού
 - Για λόγους απόδοσης, μη ακολουθιακή υλοποίηση

Ιδιότητες συναλλαγών (4 από 4)

- Σταθερότητα (Durability)
 - Ή Μονιμότητα (Permanence)
 - Η συναλλαγή επιφέρει μόνιμες αλλαγές
 - Αφού βέβαια δεσμευτεί
 - Καμία αποτυχία δεν μπορεί να τις αναιρέσει
 - Απαιτεί ανοχή βλαβών
 - Αντιμετώπιση καταστροφικών αποτυχιών

Μοντέλο συναλλαγών

- Εξυπηρετητές στοιχείων δεδομένων
 - Βάσεις δεδομένων, εγγραφές αρχείων, περιοχές μνήμης
- Δύο είδη πράξεων στα στοιχεία δεδομένων
 - Πρόσβαση (access) ή Ανάγνωση (read)
 - Ενημέρωση (update) ή Εγγραφή (write)
- Οι πράξεις είναι σειρά από read και write
 - Παράδειγμα: Branch.Deposit (Account, Amount)
 - Balance=account.Read()
 - Account.write(balance + amount)

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Κατηγορίες συναλλαγών

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές
Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Κατηγορίες συναλλαγών (1 από 4)

- Απλές ή επίπεδες συναλλαγές
 - Αυτές που ορίσαμε παραπάνω
 - Δεν επιτρέπουν μερικά αποτελέσματα
 - Αυτό μπορεί να είναι περιοριστικό
- Παράδειγμα: έστω ταξίδι με τρεις πτήσεις
 - Θέλουμε να κρατήσουμε τις δύο πρώτες
 - Δοκιμή εναλλακτικών επιλογών για την τρίτη
 - Ακύρωση μόνο αν δεν υπάρχει συνδυασμός

Κατηγορίες συναλλαγών (2 από 4)

- Φωλιασμένη συναλλαγή
 - Συναλλαγή που αποτελείται από υποσυναλλαγές
 - Εκτελούνται παράλληλα σε διαφορετικές μηχανές
 - Αυξάνουν την απόδοση του συστήματος
 - Απλουστεύουν τον προγραμματισμό
 - Δεσμεύονται ή ακυρώνονται ανεξάρτητα
 - Ο γονέας αποφασίζει αν θα δεσμευθεί
 - Ανάλογα με δέσμευση / ακύρωση των παιδιών
 - Μπορεί να δεσμευθεί παρά επί μέρους ακυρώσεις

Κατηγορίες συναλλαγών (3 από 4)

- Μονιμότητα σε φωλιασμένες συναλλαγές
 - Αν ακυρωθεί ο γονέας, ακυρώνονται και παιδιά
 - Ακόμη και αν είχαν δεσμευθεί νωρίτερα!
- Διαχείριση φωλιασμένων συναλλαγών
 - Χρήση ιδιωτικών αντιγράφων δεδομένων
 - Για κάθε υποσυναλλαγή που ξεκινά
 - Ακύρωση: τα αντίγραφα εξαφανίζονται
 - Δέσμευση: τα αντίγραφα περνούν στο γονέα
 - Οι υποσυναλλαγές βλέπουν ενδιάμεσα αποτελέσματα

Κατηγορίες συναλλαγών (4 από 4)

- Κατανεμημένες συναλλαγές
 - Εμπλέκονται πολλοί εξυπηρετητές
 - Οι εξυπηρετητές ασχολούνται με την ίδια συναλλαγή
 - Μπορεί να είναι επίπεδη ή φωλιασμένη
 - Απαιτούν κατανεμημένους αλγόριθμους
 - Έλεγχος ταυτοχρονισμού
 - Σειριακοποίηση συμβάντων

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Έλεγχος ταυτοχρονισμού

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Γιατί έλεγχος ταυτοχρονισμού;

- Ακολουθιακή υλοποίηση συναλλαγών
 - Εξασφαλίζει την απομόνωση εξ' ορισμού
- Παράλληλη υλοποίηση συναλλαγών
 - Απαιτεί εξασφάλιση της απομόνωσης!
- Σειριακά ισοδύναμη εκτέλεση
 - Ίδια αποτελέσματα με ακολουθιακή εκτέλεση
 - Δεν έχει σημασία με ποια ακριβώς σειρά
 - Απαιτεί έλεγχο ταυτοχρονισμού

Πρόβλημα 1: χαμένη ενημέρωση

T		U	
branch.withdraw(A, 50) branch.deposit(B, 50)		branch.withdraw(C, 70) branch.deposit(B, 70)	
balance = A.read()	100€		
A.write(balance - 50)	50€		
		balance = C.read()	300€
		C.write(balance - 70)	230€
balance = B.read()	200€		
		balance = B.read()	200€
		B.write(balance + 70)	270€
B.write(balance + 50)	250€		

- Αρχικά: A=100, B =200, C=300
 - Συναλλαγή T: μεταφορά 50 από A σε B
 - Συναλλαγή U: μεταφορά 70 από C σε B
 - Η T δεν βλέπει τις αλλαγές που έκανε η U
 - Βασίζεται στην παλιά τιμή του B

Πρόβλημα 2: ασυνεπής ανάκτηση

T		U	
branch.withdraw(A, 50) branch.deposit(B, 50)		branch.getBranchTotal()	
balance = A.read()	100€		
A.write(balance - 50)	50€		
		balance = A.read()	50€
		balance += B.read()	250€
		balance += C.read()	550€
balance = B.read()	200€		
B.write(balance + 50)	250€		

- Αρχικά: A=100, B =200, C=300
 - Συναλλαγή T: μεταφορά 50 από A σε B
 - Συναλλαγή U: υπολογισμός υπολοίπων
 - Η U δεν βλέπει όλες τις αλλαγές της T
 - Διαβάζει τη νέα τιμή του A
 - Διαβάζει την παλιά τιμή του B

Πρόβλημα 3: αλλοιωμένη ανάγνωση

T branch.deposit(A, 40)		U branch.deposit(A, 80)	
balance = A.read()	100€		
A.write(balance + 40)	140€		
		balance = A.read()	140€
		A.write(balance + 80)	180€
		Δεσμεύεται!	
Ακυρώνεται!			

- Αρχικά: $A=100$
 - Συναλλαγή T: κατάθεση 40 σε A
 - Συναλλαγή U: κατάθεση 80 σε A
 - Η U βλέπει τις αλλαγές της T και δεσμεύεται
 - Η T ακυρώνεται για κάποιο λόγο
 - Η U βασίστηκε σε ακυρωμένες αλλαγές

Πρόβλημα 4: πρόωρη εγγραφή

T		U	
branch.setBalance(A,300)		branch.setBalance(A,500)	
A.write(300)	300€		
		A.write(500)	500€

- Αρχικά: $A=100$
 - Έστω νέα πράξη `setBalance (account, amount)`
 - Δεσμεύεται πρώτα η U
 - Μετά ακυρώνεται η T
 - Επαναφέρουμε την κατάσταση πριν την T
 - Χάνεται εντελώς το αποτέλεσμα της U

Αντιμετώπιση προβλημάτων

- Ακύρωση συναλλαγών
 - Όποτε βασιζόμαστε σε λάθος δεδομένα
 - Κίνδυνος αλληπάλληλων ακυρώσεων
 - Κάθε ακύρωση οδηγεί σε άλλες
- Καθυστέρηση ανάγνωσης/εγγραφής
 - Όποτε βασιζόμαστε σε άλλες συναλλαγές
 - Περιμένουμε να δεσμευτούν ή να ακυρωθούν
 - Απαιτεί έλεγχο ταυτοχρονισμού

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Μέθοδοι ελέγχου ταυτοχρονισμού

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Έλεγχος ταυτοχρονισμού (1 από 3)

- Συγκρουόμενες (conflicting) πράξεις
 - Το αποτέλεσμα εξαρτάται από σειρά εκτέλεσης
 - Ζεύγη write και read στο ίδιο στοιχείο
 - Ναι: Αλλάζει το αποτέλεσμα της read
 - Ζεύγη write και write στο ίδιο στοιχείο
 - Ναι: Αλλάζει η τελική τιμή του στοιχείου
 - Ζεύγη read και read στο ίδιο στοιχείο
 - Όχι: Ίδιο αποτέλεσμα πάντα
 - Άρα, για σύγκρουση θέλουμε πάντα ένα write

Έλεγχος ταυτοχρονισμού (2 από 3)

- Αυστηρή εκτέλεση (strict execution)
 - Έστω ότι θέλουμε να κάνουμε read ή write
 - Περιμένουμε τις συναλλαγές που κάνουν write
 - Είτε να δεσμευθούν, είτε να ακυρωθούν
 - Αποφυγή αλλοιωμένης ανάγνωσης
 - Αποφυγή πρόωρης εγγραφής
 - Δεν χρειάζονται αλληπάλληλες ακυρώσεις

Έλεγχος ταυτοχρονισμού (3 από 3)

- Αλγόριθμοι ελέγχου ταυτοχρονισμού
 - Απαισιόδοξοι: εγγυώνται ότι όλα θα πάνε καλά
 - Κλείδωμα στοιχείων δεδομένων
 - Διάταξη με χρήση χρονοσφραγίδων
 - Εισάγουν καθυστερήσεις στο σύστημα
 - Αισιόδοξοι: ελέγχουν εκ των υστέρων τι έγινε
 - Αισιόδοξος έλεγχος ταυτοχρονισμού
 - Εισάγουν κίνδυνο πολλαπλών ακυρώσεων

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Κλείδωμα

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές
Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Αποκλειστική κλειδαριά (1 από 5)

- Αποκλειστικές κλειδαριές
 - Η συναλλαγή κλειδώνει το στοιχείο
 - Υποχρεωτικό κλείδωμα πριν κάθε προσπάθεια
 - Κάθε άλλη συναλλαγή περιμένει να ξεκλειδωθεί
 - Ξεκλείδωμα στο τέλος της προσπάθειας
 - Συγκρουόμενα ζεύγη: εκτέλεση με ίδια σειρά
 - Αλλιώς υπάρχει κίνδυνος αδιεξόδου!
 - Αποφεύγεται με μαζικό ξεκλείδωμα στο τέλος

Αποκλειστική κλειδαριά (2 από 5)

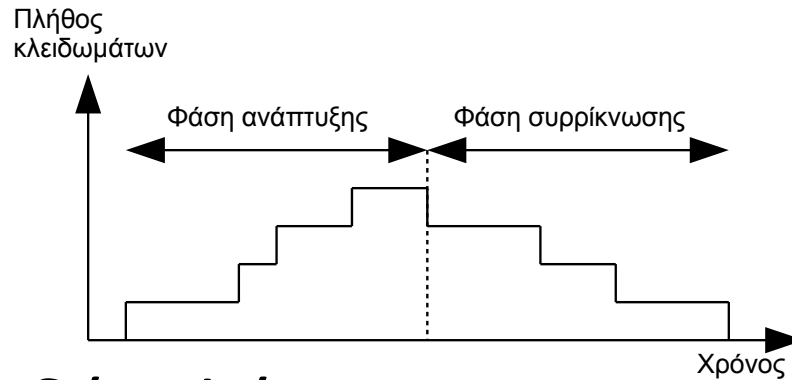
T branch.withdraw(A, 50) branch.deposit(B, 50)		U branch.withdraw(C, 70) branch.deposit(B, 70)	
Λειτουργίες	Κλειδώματα	Λειτουργίες	Κλειδώματα
balance = A.read()	κλειδώνει A		
A.write(balance - 50)			
		balance = C.read()	κλειδώνει C
		C.write(balance - 70)	
balance = B.read	κλειδώνει B		
		balance = B.read()	περιμένει B
		.	
B.write(balance + 50)		.	
	ξεκλειδώνει A,B	.	
			κλειδώνει B
		B.write(balance + 70)	
			ξεκλειδώνει B,C

- Παράδειγμα
 - Η U περιμένει να ξεκλειδωθεί το B
 - Η T ξεκλειδώνει το B (και το A) μόνο στο τέλος

Αποκλειστική κλειδαριά (3 από 5)

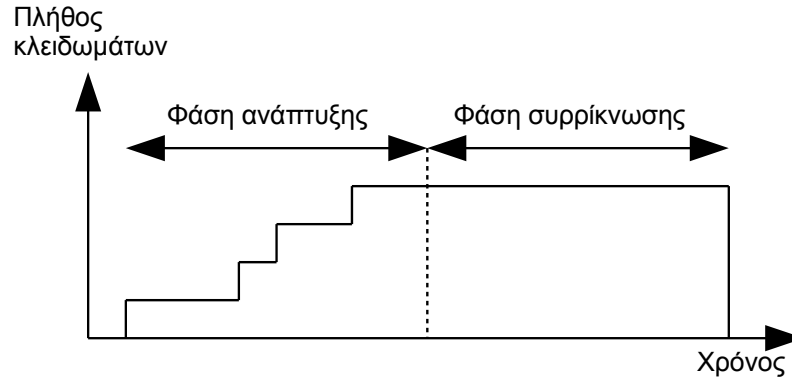
- Ξεκλείδωμα στοιχείων
 - Δύο δυνατές επιλογές
 - Όταν εκτελεστεί η πράξη του κλειδώματος
 - Κλείδωμα δύο φάσεων (2PL)
 - Δεν διασφαλίζει αυστηρή εκτέλεση
 - Όταν η συναλλαγή δεσμευθεί ή ακυρωθεί
 - Αυστηρό κλείδωμα δύο φάσεων (strict 2PL)
 - Διασφαλίζει αυστηρή εκτέλεση

Αποκλειστική κλειδαριά (4 από 5)



- Κλείδωμα δύο φάσεων
 - Φάση ανάπτυξης
 - Διαδοχικά κλειδώματα στοιχείων
 - Φάση συρρίκνωσης
 - Διαδοχικά ξεκλειδώματα στοιχείων
 - Προβλήματα αν η συναλλαγή δεν δεσμευθεί

Αποκλειστική κλειδαριά (5 από 5)



- Αυστηρό κλείδωμα δύο φάσεων
 - Φάση ανάπτυξης όπως στο απλό
 - Δεν υπάρχει φάση συρρίκνωσης
 - Όλα τα κλειδώματα απελευθερώνονται στο τέλος
 - Κανείς δεν βλέπει τις αλλαγμένες τιμές νωρίτερα
 - Διασφαλίζει την αυστηρή εκτέλεση

Καταμεριζόμενη κλειδαριά (1 από 3)

- Αποκλειστικά κλειδώματα
 - Κάθε πρόσβαση οδηγεί σε κλείδωμα
 - Τα ζεύγη read δεν συγκρούονται
 - Υπερβολικός περιορισμός παραλληλισμού
- Καταμεριζόμενες κλειδαριές
 - Κλειδαριές ανάγνωσης και κλειδαριές εγγραφής
 - Επιτρέπονται πολλά κλειδώματα ανάγνωσης
 - Επιτρέπεται μόνο ένα κλείδωμα εγγραφής

Καταμεριζόμενη κλειδαριά (2 από 3)

- Αν το στοιχείο είναι κλειδωμένο για εγγραφή
 - Δεν μπορεί να κλειδωθεί για τίποτα άλλο
- Αν είναι κλειδωμένο για ανάγνωση
 - Μπορεί να κλειδωθεί ξανά για ανάγνωση
- Προβιβασμός κλειδαριών
 - Μετατροπή κλειδαριάς σε πιο περιοριστική
 - Κλειδαριά ανάγνωσης σε εγγραφής
 - Επιτρέπεται αν η κλειδαριά δεν είναι μοιρασμένη

Καταμεριζόμενη κλειδαριά (3 από 3)

T branch.withdraw(A, 50) branch.deposit(B, 50)		U branch.withdraw(C, 70) branch.deposit(B, 70)	
Λειτουργίες	Κλειδώματα	Λειτουργίες	Κλειδώματα
balance = A.read()	κλειδώνει το A για ανάγνωση		
A.write(balance-50)	προάγει την κλειδαριά στο A	balance = C.read()	κλειδώνει το C για ανάγνωση
		C.write(balance-70)	προάγει την κλειδαριά στο C
balance = B.read()	κλειδώνει το B για ανάγνωση		
		balance = B.read()	μοιράζεται την κλειδαριά του B
B.write(balance+50)	περιμένει την U να ξεκλειδώσει το B		
...		B.write(balance+70)	περιμένει την T να ξεκλειδώσει το B
		...	

- Παράδειγμα αδιεξόδου
 - Μη αυστηρές κλειδαριές με προβιβασμό

Μειονεκτήματα κλειδαριών

- Εμφάνιση αδιεξόδων
 - Στο παράδειγμα με μη αποκλειστικό κλείδωμα
 - Κίνδυνος σε όλα τα είδη κλειδωμάτων
- Επιβάρυνση διαχείρισης κλειδαριών
 - Ακόμη και όταν δεν υπάρχει κίνδυνος
- Μείωση βαθμού παραλληλισμού
 - Ιδιαίτερα εμφανής στα αυστηρά κλειδώματα
 - Τα στοιχεία κλειδώνονται για πολύ ώρα

Παραλληλισμός κλειδαριών

- Βαθμός κοκκιότητας (granularity)
 - Πόσο μεγάλα είναι τα στοιχεία;
 - Όλη η τράπεζα, το κατάστημα, ο λογαριασμός;
 - Βάση, πίνακας, στήλη;
 - Μικρό: αυξάνεται ο παραλληλισμός
 - Περισσότερες παράλληλες κλειδαριές
 - Αυξημένος κίνδυνος αδιεξόδου
 - Μεγάλο: μειώνονται οι κλειδαριές

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αισιόδοξος έλεγχος ταυτοχρονισμού

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές
Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

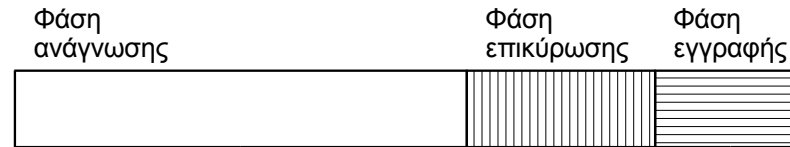
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Αισιόδοξος έλεγχος (1 από 5)

- Αισιόδοξος έλεγχος συνδρομής
 - Η πιθανότητα συγκρούσεων είναι μικρή
 - Κλειδαριές: υποθέτουν ότι έχουμε συγκρούσεις
 - Κλειδώνουμε τα στοιχεία για να τις αποφύγουμε
 - Στον αισιόδοξο έλεγχο υποθέτουμε το αντίθετο
 - Δεν κλειδώνουμε τίποτα προκαταβολικά
 - Ελέγχουμε για συγκρούσεις στο τέλος
 - Επίλυση συγκρούσεων: ακύρωση συναλλαγών

Αισιόδοξος έλεγχος (2 από 5)



- Τρεις φάσεις εκτέλεσης
 - Φάση ανάγνωσης: εκτέλεση λειτουργιών
 - Φάση επικύρωσης: έλεγχος για συγκρούσεις
 - Φάση εγγραφής: μονιμοποίηση ενεργειών
- Φάση ανάγνωσης
 - Δημιουργία προσωρινών εκδόσεων
 - Αλλαγές που έχει κάνει η συγκεκριμένη συναλλαγή

Αισιόδοξος έλεγχος (3 από 5)

- Φάση ανάγνωσης
 - Πράξεις read
 - Είτε με βάση τα στοιχεία του συστήματος
 - Είτε με βάση τις προσωρινές εκδόσεις
 - Πράξεις write
 - Καταγραφή αλλαγών στις προσωρινές εκδόσεις
 - Χρησιμοποιούνται στις επόμενες πράξεις read
 - Διατήρηση δύο συνόλων στοιχείων
 - Σύνολο αναγνώσεων: τι έχει διαβάσει η συναλλαγή
 - Σύνολο εγγραφών: τι έχει γράψει η συναλλαγή

Αισιόδοξος έλεγχος (4 από 5)

- Φάση επικύρωσης
 - Ξεκινά όταν κληθεί η `closeTransaction`
 - Έλεγχος συγκρούσεων
 - Ανάμεσα στις πράξεις αυτής και άλλων
 - Κανόνες για εντοπισμό και επίλυση συγκρούσεων
 - Συνήθως επιβολή διάταξης στις συναλλαγές
 - Επιτυχής: προχωράμε στη φάση εγγραφής
 - Ανεπιτυχής: κάποιες συναλλαγές ακυρώνονται

Αισιόδοξος έλεγχος (5 από 5)

- Φάση εγγραφής
 - Οι προσωρινές εκδόσεις γίνονται μόνιμες
 - Αντιγράφονται στον κύριο χώρο δεδομένων
 - Η συναλλαγή δεσμεύεται
- Πώς γίνεται η επικύρωση;
 - Ορίζουμε κανόνες επίλυσης συγκρούσεων
 - Οι κανόνες διασφαλίζουν τη σειριακή ισοδυναμία
 - Έστω ότι η συναλλαγή T άρχισε τη στιγμή t
 - Έλεγχος για συγκρούσεις που δεν είχαν δεσμευθεί την t

Επικύρωση συναλλαγών (1 από 8)

- Βασική ιδέα: αρίθμηση συναλλαγών
 - Η T αριθμείται με j όταν αρχίζει την επικύρωση
 - Οι αριθμοί ανατίθενται με αύξουσα σειρά
 - Αν $i < j$ τότε η i προηγείται της j
 - Αν περάσει την επικύρωση, κρατά τον αριθμό της
 - Αν δεν περάσει ή ακυρωθεί, χάνει τον αριθμό της
 - Αν έχει μόνο αναγνώσεις, χάνει τον αριθμό της
 - Δεσμεύεται αμέσως, χωρίς φάση εγγραφής

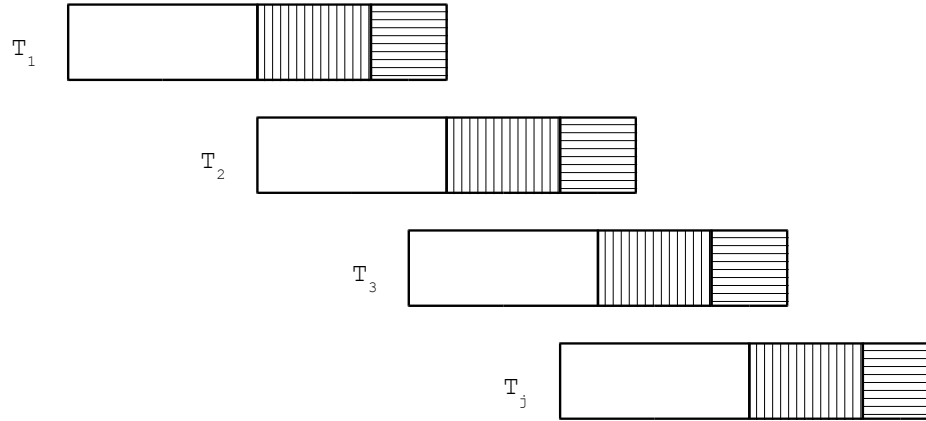
Επικύρωση συναλλαγών (2 από 8)

- Κανόνες επικύρωσης της T_j
 - Συγκρούσεις με άλλες συναλλαγές T_i
 - Για σειριακή ισοδυναμία δεν επιτρέπεται:
 1. Η T_i να έχει διαβάσει στοιχεία που έγραψε η T_j
 2. Η T_j να έχει διαβάσει στοιχεία που έγραψε η T_i
 3. Η T_i να έχει γράψει στοιχεία που έγραψε η T_j
 4. Η T_j να έχει γράψει στοιχεία που έγραψε η T_i

Επικύρωση συναλλαγών (3 από 8)

- Υπόθεση: η φάση ανάγνωσης κυριαρχεί
 - Οι άλλες δύο παίρνουν αναλογικά λίγο χρόνο
 - Άρα μία επικύρωση / εγγραφή σε κάθε στιγμή
 - Διασφαλίζεται εύκολα με αμοιβαίο αποκλεισμό
 - Έτσι ικανοποιούμε τις απαιτήσεις 3 και 4
 - Εγγραφή μόνο πάνω σε δεσμευμένες συναλλαγές
 - Μένουν οι απαιτήσεις 1 και 2
 - Συγκρούσεις ανάγνωσης και εγγραφής

Επικύρωση συναλλαγών (4 από 8)

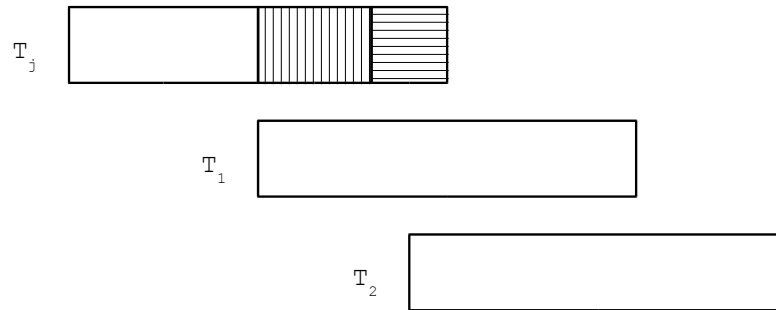


- Αντίστροφη επικύρωση
 - Έλεγχος συγκρούσεων με προηγούμενες
- Κανόνας 1: ισχύει πάντα
 - Η T_i δεν πρέπει να διαβάσει ό,τι έγραψε η T_j
 - Όλες οι αναγνώσεις των T_i έγιναν πριν την T_j

Επικύρωση συναλλαγών (5 από 8)

- Κανόνας 2
 - Η T_j δεν πρέπει να διαβάσει ό,τι έγραψε η T_i
 - Έλεγχος συνόλου ανάγνωσης T_j
 - Σύγκριση με σύνολα εγγραφής T_i
 - Εάν υπάρχουν επικαλύψεις η T_j ακυρώνεται
 - Πρέπει να διατηρούμε σύνολα εγγραφών
 - Ως την επικύρωση των επικαλυπτόμενων συναλλαγών
 - Αυτό μπορεί να έχει σημαντικό κόστος
 - Αν η T_j δεν έχει αναγνώσεις ο κανόνας ισχύει πάντα

Επικύρωση συναλλαγών (6 από 8)



- Ευθεία επικύρωση
 - Έλεγχος συγκρούσεων με επόμενες
 - Δεν ξεκίνησαν ακόμη την επικύρωση, δεν έχουν i
- Κανόνας 2: ισχύει πάντα
 - Η T_j δεν μπορεί να διαβάσει λάθος στοιχεία
 - Οι άλλες ακολουθούν την T_j

Επικύρωση συναλλαγών (7 από 8)

- Κανόνας 1
 - Έλεγχος συνόλου εγγραφής T_j
 - Σύγκριση με σύνολα ανάγνωσης άλλων
 - Εάν υπάρχουν επικαλύψεις έχουμε πρόβλημα
 - Αν η T_j δεν έχει εγγραφές, ο κανόνας ισχύει πάντα
- Επίλυση συγκρούσεων
 - Ακύρωση των επόμενων συναλλαγών
 - Μπορεί να ακυρώνονται πολλές άλλες

Επικύρωση συναλλαγών (8 από 8)

- Επίλυση συγκρούσεων
 - Ακύρωση της T_j
 - Δεν ακυρώνονται πολλές άλλες
 - Μπορεί όμως να ακυρωθούν για άλλους λόγους
 - Οπότε η T_j ακυρώθηκε χωρίς λόγο
 - Αναμονή της T_j για τις άλλες συναλλαγές
 - Περιμένουμε μήπως ακυρωθούν οι άλλες
 - Ελπίζουμε ότι δεν θα χρειαστεί ακύρωση

Ευθεία ή αντίστροφη επικύρωση;

- Αντίστροφη επικύρωση
 - Περιοριστική στην επίλυση συγκρούσεων
 - Ένα σύνολο ανάγνωσης, πολλά σύνολα εγγραφής
 - Απαιτεί διατήρηση πολλών συνόλων εγγραφής
- Ευθεία επικύρωση
 - Ευέλικτη επίλυση συγκρούσεων
 - Δεν είναι προφανής η καλύτερη λύση όμως
 - Ένα σύνολο εγγραφής, πολλά σύνολα ανάγνωσης

Υποσιτισμός και αδιέξοδα

- Κίνδυνος υποσιτισμού
 - Συνεχής ακύρωση των ίδιων συναλλαγών
 - Μπορούμε να τις βοηθήσουμε
 - Αν έχουν ακυρωθεί πολλές φορές
 - Εκτέλεση με αμοιβαίο αποκλεισμό
- Δεν εμφανίζονται αδιέξοδα
 - Χωρίς να περιορίζεται ο παραλληλισμός
 - Με κόστος πιθανή επανάληψη συναλλαγών

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Διάταξη χρονοσφραγίδων

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Χρονοσφραγίδες (1 από 9)

- Διάταξη χρονοσφραγίδων
 - Η συναλλαγή T λαμβάνει χρονοσφραγίδα $ts(T)$
 - Η αρίθμηση γίνεται στην εκκίνηση της συναλλαγής
 - Στον αισιόδοξο έλεγχο δίνεται στην επικύρωση
 - Η $ts(T)$ συνοδεύει όλες τις πράξεις της T
 - Αναγνώσεις και εγγραφές
 - Αναγνώσεις από τις κανονικές εκδόσεις στοιχείων
 - Εγγραφές σε προσωρινές εκδόσεις στοιχείων

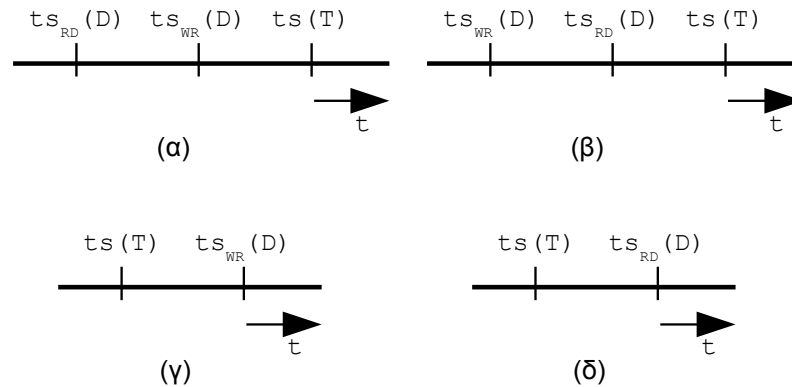
Χρονοσφραγίδες (2 από 9)

- Χρονοσφραγίδες στοιχείου D
 - Μία χρονοσφραγίδα εγγραφής $ts_{WR}(D)$
 - Αντιστοιχεί στην δεσμευμένη τιμή
 - Προσωρινές χρονοσφραγίδες εγγραφής $ts(T)$
 - Μία για κάθε προσωρινή έκδοση
 - Πάντα μεγαλύτερες από την $ts_{WR}(D)$
 - Μία χρονοσφραγίδα ανάγνωσης $ts_{RD}(D)$
 - Αντιστοιχεί στην πιο πρόσφατη ανάγνωση

Χρονοσφραγίδες (3 από 9)

- Κανόνες επικύρωσης: εγγραφή
 - Η T με $ts(T)$ θέλει να γράψει το στοιχείο D
 - Πρέπει να ισχύει $ts(T) > ts_{WR}(D)$
 - Πρέπει να έπεται της δεσμευμένης έκδοσης
 - Πρέπει να ισχύει $ts(T) > ts_{RD}(D)$
 - Πρέπει να έπεται όλων των αναγνώσεων
 - Αν δεν ισχύουν οι περιορισμοί, η T ακυρώνεται
 - Αλλιώς έχουμε νέα προσωρινή έκδοση του D

Χρονοσφραγίδες (4 από 9)



- Παραδείγματα εγγραφής
 - (α) και (β): η εγγραφή είναι έγκυρη
 - (γ): συγκρούεται με μεταγενέστερη εγγραφή
 - (δ): συγκρούεται με μεταγενέστερη ανάγνωση
 - Στις περιπτώσεις (γ) και (δ) η T ακυρώνεται

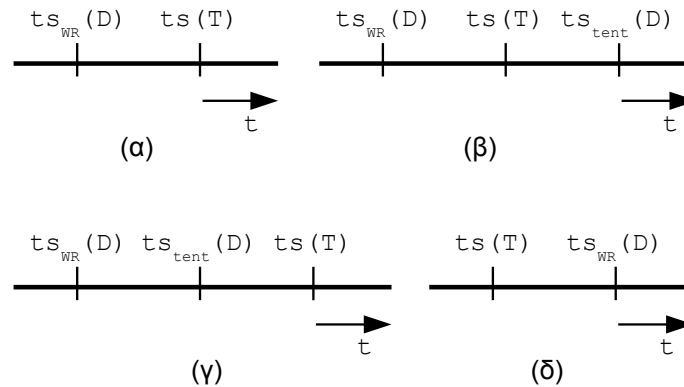
Χρονοσφραγίδες (5 από 9)

- Κανόνες επικύρωσης: ανάγνωση
 - Η T με $ts(T)$ θέλει να διαβάσει το στοιχείο D
 - Πρέπει να ισχύει $ts(T) > ts_{WR}(D)$
 - Αλλιώς η συναλλαγή T ακυρώνεται
 - Αν ναι, βρίσκουμε πιο πρόσφατη έκδοση του D
 - Η έκδοση με το μέγιστο $ts_{tent}(D)$
 - Με την προϋπόθεση ότι $ts_{tent}(D) \leq ts(T)$
 - Νεότερες προσωρινές εκδόσεις δεν επηρεάζουν την T

Χρονοσφραγίδες (6 από 9)

- Κανόνες επικύρωσης: ανάγνωση
 - Αν η έκδοση είναι δεσμευμένη, τη διαβάζουμε
 - Σε αυτή την περίπτωση, $ts_{WR}(D)=ts_{tent}(D)$
 - Αν δεν είναι δεσμευμένη, περιμένουμε
 - Αν η συναλλαγή δεσμευθεί, διαβάζουμε
 - Ουσιαστικά πάμε στην προηγούμενη περίπτωση
 - Αν η συναλλαγή ακυρωθεί, δοκιμάζουμε ξανά
 - Ουσιαστικά επαναλαμβάνουμε την επιλογή έκδοσης

Χρονοσφραγίδες (7 από 9)



- Παραδείγματα ανάγνωσης
 - (α): η εγγραφή είναι έγκυρη
 - (β): μεταγενέστερη προσωρινή, έγκυρη
 - (γ): προγενέστερη προσωρινή, περιμένουμε
 - (δ): μεταγενέστερη εγγραφή, ακυρώνουμε

Χρονοσφραγίδες (8 από 9)

T	U	A (r)	A (w)	B (r)	B (w)	C (r)	C (w)
		{}	{S}	{}	{S}	{}	{S}
openTransaction b = A.read()		{T}					
	openTransaction b = C.read()					{U}	
A.write(b - 50) b = B.read()			{S,T}				
	C.write(b - 70) b = B.read()			{T}			{S,U}
B.write(b + 50) Ακυρώνεται!				{U}			
	B.write(b + 70)				{S,U}		

- Ίδιο παράδειγμα με τα κλειδώματα
 - Χρονοσφραγίδες ανάγνωσης / εγγραφής
 - Η S έχει γράψει A, B, C και έχει δεσμευθεί
 - Η T ακυρώνεται όταν πάει να γράψει το B
 - Το B το έχει διαβάσει η U και $ts(T) < ts(U)$

Χρονοσφραγίδες (9 από 9)

- Κανόνας επικύρωσης εγγραφής
 - Οι πρόωρες εγγραφές ακυρώνονται
- Κανόνας επικύρωσης ανάγνωσης
 - Οι πρόωρες αναγνώσεις αναμένουν
 - Οι καθυστερημένες αναγνώσεις ακυρώνονται
- Απαισιόδοξος έλεγχος συνδρομής
 - Δεν υπάρχει περίπτωση ακύρωσης στο τέλος
 - Δεν έχουμε αδιέξοδα, αντίθετα με τα κλειδώματα
 - Έχουμε ακυρώσεις, όπως στον αισιόδοξο έλεγχο

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Κατανεμημένες συναλλαγές

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Κατανεμημένος έλεγχος

- Τι γίνεται με πολλούς εξυπηρετητές;
 - Έλεγχος ταυτοχρονισμού σε κάθε εξυπηρετητή
 - Επιπλέον έλεγχος ταυτοχρονισμού μεταξύ τους
 - Διάταξη χρονοσφραγίδων
 - Απαιτείται κοινά αποδεκτή διάταξη στο χρόνο
 - Αισιόδοξος έλεγχος
 - Απαιτείται καθολική επικύρωση
 - Εναλλακτικά, διάταξη χρονοσφραγίδων στη δέσμευση
 - Κλειδώματα: απαιτείται έλεγχος αδιεξόδων

Κατανεμημένη δέσμευση (1 από 9)

- Πρόβλημα κατανεμημένης δέσμευσης
 - Έστω συναλλαγή σε πολλούς εξυπηρετητές
 - Πρέπει να δεσμευθούν όλοι ή κανένας
 - Χρήση συντονιστή για τη λήψη της απόφασης
 - Οι συμμετέχοντες ακολουθούν τον συντονιστή
- Πρωτόκολλο δέσμευσης μίας φάσης
 - Συντονιστής: διατάζει συμμετέχοντες να δεσμευτούν
 - Περιμένει απάντηση από όλους
 - Αν κάποιος δεν τα καταφέρουν όμως;
 - Δεν διασφαλίζεται η κατανεμημένη δέσμευση!

Κατανεμημένη δέσμευση (2 από 9)

- Πρωτόκολλο δέσμευσης δύο φάσεων
 - Πρώτα ρωτάμε και μετά αποφασίζουμε
 - Φάση ψηφοφορίας και φάση ολοκλήρωσης
- Φάση ψηφοφορίας
 - Ο συντονιστής ρωτάει τους συμμετέχοντες
 - Αποστολή μηνύματος <vote request>
 - Οι συμμετέχοντες απαντούν με τη γνώμη τους
 - Αποστολή μηνύματος <vote commit> ή <vote abort>

Κατανεμημένη δέσμευση (3 από 9)

- Φάση ολοκλήρωσης
 - Ο συντονιστής λαμβάνει μία απόφαση
 - Αν όλοι ψήφισαν δέσμευση, στέλνει <vote commit>
 - Αν κάποιοι ψήφισαν ακύρωση, στέλνει <vote abort>
 - Οι συμμετέχοντες πράττουν ανάλογα
 - Όποιος ψήφισε ακύρωση, δεν περιμένει
 - Η συναλλαγή θα ακυρωθεί σίγουρα

Κατανεμημένη δέσμευση (4 από 9)

- Αντιμετώπιση προβλημάτων
 - Απώλεια μηνυμάτων ή συμμετεχόντων
 - Χρήση διαστημάτων εκπνοής
 - Αναμετάδοση αν δεν έχουμε απάντηση
 - Αν πέσει ο συντονιστής όμως;
 - Οι συμμετέχοντες εμποδίζονται κατά την ψηφοφορία
 - Δεν υπάρχει τρόπος να λάβουν απόφαση
 - Το πρόβλημα είναι το σημείο εμποδισμού
 - Από την ψηφοφορία έχουμε δύο επιλογές

Κατανεμημένη δέσμευση (5 από 9)

- Συνθήκες μη εμποδισμού του Skeen
 - Απαγορεύονται δύο είδη καταστάσεων
 - Με μετάβαση είτε σε δέσμευση είτε σε ακύρωση
 - Χωρίς βέβαιη απόφαση και μετάβαση σε δέσμευση
- Πρωτόκολλο δέσμευσης τριών φάσεων
 - Εισαγωγή νέας φάσης
 - Ανάμεσα στη φάση αναμονής και τη δέσμευση
 - Ονομάζεται φάση προδέσμευσης

Κατανεμημένη δέσμευση (6 από 9)

- Πρωτόκολλο δέσμευσης τριών φάσεων
 - Ο συντονιστής ρωτάει τους συμμετέχοντες
 - Αποστολή μηνύματος <vote request>
 - Οι συμμετέχοντες απαντούν με τη γνώμη τους
 - Αποστολή μηνύματος <vote commit> ή <vote abort>
 - Ο συντονιστής λαμβάνει μία απόφαση
 - Αν όλοι ψήφισαν δέσμευση, στέλνει <pre commit>
 - Αν κάποιιοι ψήφισαν ακύρωση, στέλνει <global abort>

Κατανεμημένη δέσμευση (7 από 9)

- Πρωτόκολλο δέσμευσης τριών φάσεων
 - Οι συμμετέχοντες πράττουν ανάλογα
 - Αν πάρουν <pre commit>, στέλνουν <ready commit>
 - Αν πάρουν <global abort>, ακυρώνουν τη συναλλαγή
 - Ο συντονιστής τελικά στέλνει <global commit>
 - Αφού λάβει όλα τα μηνύματα <ready commit>
 - Οι συμμετέχοντες δεσμεύουν τη συναλλαγή
 - Μόλις λάβουν το μήνυμα <global commit>

Κατανεμημένη δέσμευση (8 από 9)

- Αντιμετώπιση προβλημάτων
 - Έστω ότι ο συντονιστής καταρρέει
 - Περιμένουμε ένα χρονικό διάστημα
 - Αν έχει στείλει <pre commit>, προχωράμε
 - Αν δεν έχει στείλει, ακυρώνουμε
 - Αν κάποιος συμμετέχων δεν λάβει μηνύματα
 - Ρωτάει τους άλλους τι έλαβαν
 - Αντιμετώπιση χαμένων μηνυμάτων

Κατανεμημένη δέσμευση (9 από 9)

- Μπορούμε να αποφύγουμε την τρίτη φάση;
 - Το πρόβλημα που λύνει είναι η αβεβαιότητα
 - Αν χάσουμε ένα μήνυμα, πώς ξέρουμε τι έγινε;
 - Μπορεί οι άλλοι συμμετέχοντες να καταρρεύσουν
 - Άρα δεν έχουμε κάποιον να ρωτήσουμε
 - Εναλλακτικές μέθοδοι αντιμετώπισης
 - Κάθε συμμετέχων αναμεταδίδει το <vote commit>
 - Μετά δεσμεύει το δικό του τμήμα της συναλλαγής
 - Δύο φάσεις, αλλά προσθήκη πολλών μηνυμάτων

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Τέλος Ενότητας # 3

Μάθημα: Κατανεμημένα Συστήματα, **Ενότητα # 3:** Ατομικές συναλλαγές

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

