

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός

Ενότητα # 2: Διάταξη συμβάντων, καθολικές καταστάσεις

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



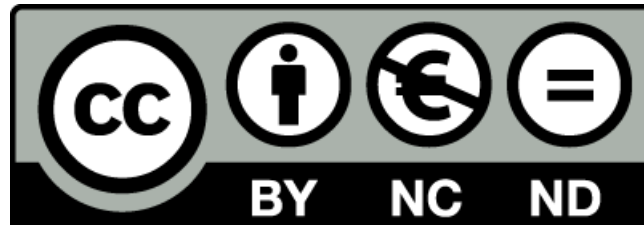
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

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 των φυσικών χρονιστών και των αλγορίθμων συγχρονισμού τους.
- Εισαγωγή στις μεθόδους διάταξης συμβάντων καθώς και στους λογικούς και διανυσματικούς χρονιστές.
- Εξοικείωση με την έννοια της καθολικής κατάστασης και κατανόηση της παθητικής και ενεργητικής στρατηγικής κατασκευής τους.

Περιεχόμενα ενότητας

- Συγχρονισμός φυσικών χρονιστών
- Διάταξη συμβάντων
- Καθολικές καταστάσεις
 - Παθητική στρατηγική: παρατηρήσεις
 - Ενεργητική στρατηγική: στιγμιότυπα

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

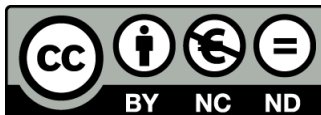


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Συγχρονισμός φυσικών χρονιστών

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 2: Διάταξη συμβάντων και καθολικές καταστάσεις

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



Συγχρονισμός (1 από 2)

- Συγχρονισμός διεργασιών
 - Απαραίτητος για τον αμοιβαίο αποκλεισμό
 - Βασίζεται σε καθολική διάταξη συμβάντων
 - Δεν έχει σημασία η ακρίβεια του χρονιστή
 - Αρκεί να διατάσσονται σωστά τα συμβάντα
- Κατανεμημένος συγχρονισμός
 - Δεν υπάρχει κοινός χρονιστής
 - Οι χρονιστές ολισθαίνουν μεταξύ τους
 - Δεν υπάρχει η έννοια του καθολικού χρόνου

Συγχρονισμός (2 από 2)

- Συγχρονισμός με μηνύματα
 - Τα μηνύματα έχουν άγνωστο χρόνο μετάδοσης
- Το πρόβλημα διάταξης συμβάντων
 - p_0 : λαμβάνει το m_1 από p_1 και το m_2 από p_2
 - Ποιο στάλθηκε πρώτο, το m_1 ή το m_2 ;
- Το πρόβλημα καθολικής κατάστασης
 - Ποια είναι η κατάσταση ενός υπολογισμού;
 - Ένωση τοπικών καταστάσεων όλων των διεργασιών
 - Πώς ξέρουμε ποιες καταστάσεις να ενώσουμε;

Φυσικοί χρονιστές (1 από 6)

- Φυσικός χρονιστής
 - Γνωστός και ως ρολόι
 - Κρύσταλλος, μετρητής και καταχωρητής
 - Σε κάθε ταλάντωση ο μετρητής μειώνεται
 - Στο 0 δημιουργεί διακοπή και φορτώνεται ξανά
- Χρονιστές λογισμικού
 - Αξιοποιούν φυσικό χρονιστή για μέτρηση χρόνου
 - Ενημερώνονται μέσω των διακοπών

Φυσικοί χρονιστές (2 από 6)

- Ολίσθηση (drift) χρονιστή
 - Η συχνότητα αποκλίνει από το ιδανικό
 - Απόκλιση από ένα ιδανικό χρονιστή αναφοράς
 - Έστω χρονιστής C και χρονικές στιγμές t και t'
 - Οι τιμές $t' > t$ ορίζονται από χρονιστή αναφοράς
 - Ο ρυθμός ολίσθησης r ορίζεται ως εξής
 - $r = [C(t') - C(t)] / (t' - t) - 1 = dC/dt - 1$
 - Ιδανικά $r = 0$, γιατί $dC/dt = 1$

Φυσικοί χρονιστές (3 από 6)

- Μέγιστος ρυθμός ολίσθησης
 - Ορίζεται από τον κατασκευαστή του χρονιστή
 - $|dC/dt-1| \leq \rho$
- Λόξωση (skew) χρονιστών
 - Οι διάφορες μηχανές αποκλίνουν μεταξύ τους
 - Η λόξωση εξαρτάται από τις ολισθήσεις των χρονιστών
 - Κάθε χρονιστής μπορεί να ολισθαίνει διαφορετικά

Φυσικοί χρονιστές (4 από 6)

- Εξωτερικός συγχρονισμός
 - Συγχρονισμός διεργασιών με εξωτερική πηγή
 - Έστω φράγμα συγχρονισμού $\delta > 0$
 - Έστω μια πηγή χρόνου αναφοράς R
 - Θέλουμε $|C_i(t) - R(t)| < \delta$
 - Για κάθε διεργασία p_i και για κάθε στιγμή t
 - Ο χρονιστής C_i είναι ακριβής μέσα στο φράγμα δ
 - Βάση: χρόνος UTC μέσω επίγειας μετάδοσης ή GPS

Φυσικοί χρονιστές (5 από 6)

- Εσωτερικός συγχρονισμός
 - Συγχρονισμός διεργασιών μεταξύ τους
 - Έστω φράγμα συγχρονισμού $\delta > 0$
 - Θέλουμε $|C_i(t) - C_j(t)| < \delta$
 - Για κάθε διεργασία p_i και για κάθε στιγμή t
 - Οι χρονιστές C_i συμφωνούν μέσα στο φράγμα δ
- Σχέση εξωτερικού - εσωτερικού συγχρονισμού
 - Έστω εξωτερικός με φράγμα δ
 - Τότε και εσωτερικός με φράγμα 2δ

Φυσικοί χρονιστές (6 από 6)

- Μονότονο ρολόι
 - Απλά αυξάνεται συνεχώς
 - $t' > t \Rightarrow C(t') > C(t)$
 - Περιέργως, αυτό αρκεί σε πολλές περιπτώσεις!
 - Πιο ισχυρή απαίτηση: όχι άλματα
 - Σταδιακή και όχι απότομη αλλαγή χρόνου
 - Απαιτείται μετά από διαδικασία συγχρονισμού

Αλγόριθμος του Cristian (1 από 2)

- Εξωτερικός συγχρονισμός με εξυπηρετητή
 - Μηχανή με ρολόι UTC ή απλά αξιόπιστη πηγή
- Περιοδικά η διεργασία p στέλνει μήνυμα m_r
 - Ο εξυπηρετητής επιστρέφει m_t με την ώρα t
- Δεν αλλάζουμε απλά το χρονιστή σε t
 - Δεν ξέρουμε πόσο χρόνο απαιτήσε το m_t
 - Πρέπει να εκτιμήσουμε καθυστέρηση μετάδοσης

Αλγόριθμος του Cristian (2 από 2)

- Υπολογίζουμε την καθυστέρηση
 - Καταγράφουμε το χρόνο μετ' επιστροφής T_r
 - Θέτουμε το ρολόι σε $t+T_r/2$
- Υπόθεση: οι καθυστερήσεις είναι συμμετρικές
 - Συνήθως ισχύει σε κοντινές μηχανές
- Υπόθεση: δεν υπάρχει χρόνος επεξεργασίας
 - Αν υπάρχει, συνυπολογίζεται

Αλγόριθμος Berkeley (1 από 2)

- Εσωτερικός συγχρονισμός σε ένα δίκτυο
 - Προτάθηκε για μηχανές με Berkeley UNIX
- Λειτουργεί αντίστροφα από αλγόριθμο Cristian
 - Ένας υπολογιστής επιλέγεται ως κύριος
 - Περιοδικά ο κύριος ζητάει τις τιμές των υπηρετών
 - Εκτιμά τοπικές ώρες συνεκτιμώντας καθυστερήσεις
 - Υπολογισμός του μέσου όρου
 - Αποστολή διορθώσεων (+ ή -) στους υπηρέτες

Αλγόριθμος Berkeley (2 από 2)

- Αποφυγή προβληματικών χρονιστών
 - Οι ακραίες τιμές δεν υπολογίζονται
 - Ελαττωματικός χρονιστή αγνοείται
- Απαιτεί αλγόριθμο εκλογής αρχηγού (master)
- Ο χρονιστής του κυρίου δεν είναι «σωστός»
 - Μπορεί να είναι ένας απλός χρονιστής
 - Άλλωστε ο συγχρονισμός είναι εσωτερικός

Πρωτόκολλο ώρας δικτύου (1 από 4)

- Πρωτόκολλο ώρας δικτύου (NTP)
 - Αρχιτεκτονική υπηρεσίας ώρας
 - Πρωτόκολλο διανομής ωρας
 - Κατάλληλο για μεγάλα συστήματα (Διαδίκτυο)
- Ιεραρχικά στρώματα εξυπηρετητών (strata)
 - Στρώμα 1: συνδεδεμένοι με ρολόγια UTC
 - Πιθανόν μέσω ραδιοφώνου ή δορυφόρου
 - Ως στρώμα 0 υπονοούνται τα ρολόγια UTC

Πρωτόκολλο ώρας δικτύου (2 από 4)

- Στρώμα 2: συγχρονίζεται με το στρώμα 1
 - Στρώμα 3, 4, 5, ...: συγχρονίζεται με το παραπάνω
- Επικοινωνία εξυπηρετητών μέσω UDP
 - Μονόδρομη ή αμφίδρομη
 - Ανάλογα με τον τρόπο συγχρονισμού
- Τρόπος ευρείας/πολυεκπομπής
 - Κατάλληλος για τοπικά δίκτυα
 - Ο εξυπηρετητής περιοδικά πολυεκπέμπει ώρα
 - Οι παραλήπτες υπολογίζουν μικρή καθυστέρηση

Πρωτόκολλο ώρας δικτύου (3 από 4)

- Τρόπος πελάτη/εξυπηρετητή
 - Παρόμοια λειτουργία με αλγόριθμο Cristian
 - Πιο ακριβής από πολυεκπομπή
 - Κατάλληλος για μη κοντινές μηχανές
- Συμμετρικός ενεργητικός/παθητικός τρόπος
 - Συγχρονισμός μέσα σε σύνολο εξυπηρετητών
 - Διατηρεί ιστορία ανταλλαγών πληροφορίας
 - Σταδιακά βελτιώνεται η ακρίβεια των ρολογιών

Πρωτόκολλο ώρας δικτύου (4 από 4)

- Βελτιώσεις ακρίβειας
 - Κάθε εξυπηρετητής μιλάει με αρκετούς άλλους
 - Σε διάφορα strata
 - Φιλτράρισμα δεδομένων
 - Αλγόριθμος επιλογής ομοτίμων
 - Προτιμώνται εξυπηρετητές υψηλότερων στρωμάτων
 - Δεν είναι απαραίτητο να θέλουν να μας μιλήσουν

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Διάταξη συμβάντων

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 2: Διάταξη συμβάντων και καθολικές καταστάσεις

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



Πρόδρομη σχέση (1 από 2)

- Είναι απαραίτητος ο συγχρονισμός;
 - Αρκεί να συμφωνήσουμε σε συνεπή διάταξη
 - Κάθε συμβάν e έχει χρονοσφραγίδα $TS(e)$
 - Πρέπει να ισχύουν οι ακόλουθες σχέσεις
 - Τα συμβάντα μίας διεργασίας διατάσσονται κανονικά
 - Η αποστολή μηνύματος προηγείται της λήψης του
 - Πρόδρομη σχέση (happened-before relation)
 - Ή αιτιώδης διάταξη (causal ordering)

Πρόδρομη σχέση (2 από 2)

- Η πρόδρομη σχέση είναι μερική διάταξη
 - Καλύπτει τις παραπάνω σχέσεις
 - Επιπλέον ισχύει η μεταβατική ιδιότητα
- Απεικόνιση σε διάγραμμα χώρου – χρόνου
 - Αν $e \rightarrow e'$ τότε υπάρχει διαδρομή από e στο e'
 - Το e προηγείται αιτιωδώς του e'
 - Δύο μη σχετιζόμενα συμβάντα είναι σύνδρομα
 - Γράφουμε ότι $e \parallel e'$
 - Δεν ισχύει ούτε $e \rightarrow e'$ ούτε $e' \rightarrow e$

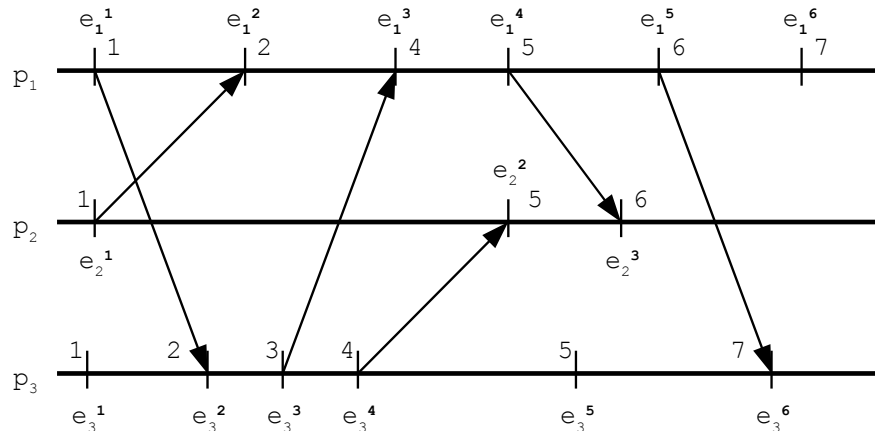
Λογικοί χρονιστές (1 από 4)

- Συνθήκη του χρονιστή
 - Έστω ότι έχουμε έναν καθολικό χρονιστή RC
 - Τότε έχουμε $e \rightarrow e' \Rightarrow RC(e) < RC(e')$
- Λογικοί χρονιστές: υλοποιούν πρόδρομη σχέση
 - Κάθε p_i έχει το δικό της λογικό χρονιστή LC_i
 - Μονότονα αυξανόμενος μετρητής στο λογισμικό
 - Δεν σχετίζεται με φυσικό χρονιστή
 - Η χρονοσφραγίδα $TS(e)$ του e είναι το $LC(e)$

Λογικοί χρονιστές (2 από 4)

- Λειτουργία λογικών χρονιστών
 - Αρχικά $LC_i = 0$ για όλα τα i
 - Για κάθε συμβάν e της διεργασίας p_i
 - Αν το e είναι εσωτερικό συμβάν ή αποστολή, $LC_i = LC_i + 1$
 - Αν το e είναι η αποστολή του m , τότε $TS(m) = LC_i$
 - Αν το e είναι η λήψη του m , τότε $LC_i = \max\{LC_i, TS(m)\} + 1$
 - Ο χρονιστής πάντα αυξάνεται, πιθανόν με άλματα
 - Τα συμβάντα της διεργασίας έχουν ολική διάταξη

Λογικοί χρονιστές (3 από 4)



- Παράδειγμα λογικών χρονιστών
 - Βασίζεται στον υπολογισμό προηγούμενου σχήματος
 - Τα σύνδρομα συμβάντα δεν είναι διατεταγμένα
 - Μπορούμε να επισυνάψουμε αναγνωριστικό διεργασίας
 - Αφού είναι σύνδρομα η σειρά τους δεν έχει σημασία

Λογικοί χρονιστές (4 από 4)

- Λογικοί χρονιστές και συνθήκη χρονιστή
 - Οι λογικοί χρονιστές ικανοποιούν τη συνθήκη
 - Αν $e \rightarrow e'$, τότε $LC(e) < LC(e')$
 - Το αντίστροφο όμως δεν ισχύει!
 - $e_3^1 \parallel e_1^2$ παρά το ότι $LC(e_3^1) < LC(e_1^2)$
 - Ιδανικά θα θέλαμε να ισχύει και το αντίστροφο
 - Η συνεπαγωγή να είναι ισοδυναμία
 - Ισχυρή συνθήκη χρονιστή για καθολικό ρολόι RC
 - $e \rightarrow e' \Leftrightarrow RC(e) < RC(e')$

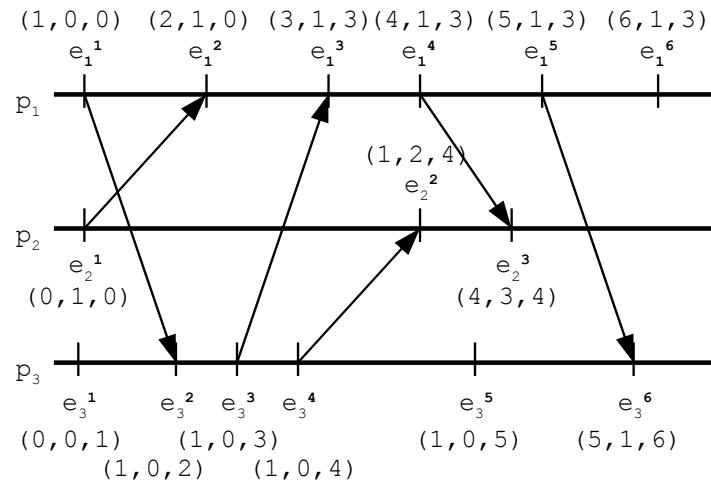
Διανυσματικοί χρονιστές (1 από 5)

- Διανυσματικοί χρονιστές
 - Για κάθε p_i το VC_i είναι διάνυσμα n στοιχείων
- Λειτουργία διανυσματικών χρονιστών
 - Αρχικά $VC_i[j]=0$ για όλα τα i, j
 - Για κάθε συμβάν e της διεργασίας p_i
 - Αν το e είναι εσωτερικό συμβάν ή αποστολή, $VC_i[i] = VC_i[i]+1$
 - Αν το e είναι η αποστολή του m , τότε $TS(m)=VC_i$
 - Αν το e είναι η λήψη του m , τότε:
 - $VC_i[j] = \max\{VC_i[j], TS(m[j])\}$
 - $VC_i[i]=VC_i[i]+1$

Διανυσματικοί χρονιστές (2 από 5)

- Λειτουργία διανυσματικών χρονιστών
 - Η διεργασία αριθμεί τα συμβάντα της ανεξάρτητα
 - Ο χρονιστής κωδικοποιεί τη γνώση της
 - Πόσα γεγονότα έχουν συμβεί στις άλλες διεργασίες
 - Η αποστολή ενημερώνει το ρολόι του παραλήπτη
 - Οτιδήποτε ξέρει ο αποστολέας ...
 - ... μπορεί να το έμαθε και ο παραλήπτης!
 - Άρα προχωράνε όλες οι τιμές του διανύσματος

Διανυσματικοί χρονιστές (3 από 5)



- Παράδειγμα διανυσματικών χρονιστών
 - Βασίζεται στον προηγούμενο υπολογισμό
 - Πώς συγκρίνονται οι χρονοσφραγίδες;
 - Πιθανές έξοδοι: διατεταγμένες ή σύνδρομες

Διανυσματικοί χρονιστές (4 από 5)

- Σύγκριση διανυσματικών χρονιστών
 - $VC(e) < VC(e') \Leftrightarrow VC(e)[k] \leq VC(e')[k]$ για κάθε k
 - Δύο χρονοσφραγίδες δεν είναι ποτέ ίσες!
 - Κάθε διεργασία αυξάνει το στοιχείο της, πλην λήψεων
 - Κάθε διανυσματικό ρολόι εξελίσσεται διαφορετικά
 - Σε λήψη, αλλάζουμε τουλάχιστον ένα στοιχείο
 - Άρα αποκλείεται να εξισωθούν οι χρονοσφραγίδες

Διανυσματικοί χρονιστές (5 από 5)

- Ισχυρή συνθήκη χρονιστή
 - Αν ισχύει ότι $VC(e) < VC(e')$, τότε $e \rightarrow e'$
 - Καλύπτεται και η ισχυρή συνθήκη χρονιστή
 - Ορισμένα συμβάντα δεν διατάσσονται
 - $e_3^1 \parallel e_1^2$ και οι χρονοσφραγίδες δεν συγκρίνονται
 - Δεν ισχύει ούτε $VC(e_3^1) < VC(e_1^2)$, ούτε $VC(e_1^2) < VC(e_3^1)$
 - Οι χρονοσφραγίδες δεν είναι ολικά διατεταγμένες
 - Είναι όμως σαφές πότε τα συμβάντα είναι σύνδρομα

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Καθολικές καταστάσεις

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 2: Διάταξη συμβάντων και καθολικές καταστάσεις

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



Τοπική και καθολική ιστορία

- Τοπική ιστορία διεργασίας p_i
 - Έστω ότι e_i^j είναι το γεγονός j της διεργασίας i
 - Τότε $h_i = e_1^1 e_1^2 e_1^3 e_1^4 e_1^5 e_1^6$
 - Η τοπική ιστορία είναι πλήρως διατεταγμένη
- Καθολική ιστορία υπολογισμού
 - Ένωση τοπικών ιστοριών διεργασιών
 - $H = h_1 \cup \dots \cup h_n$
 - Η καθολική ιστορία είναι μερικώς διατεταγμένη

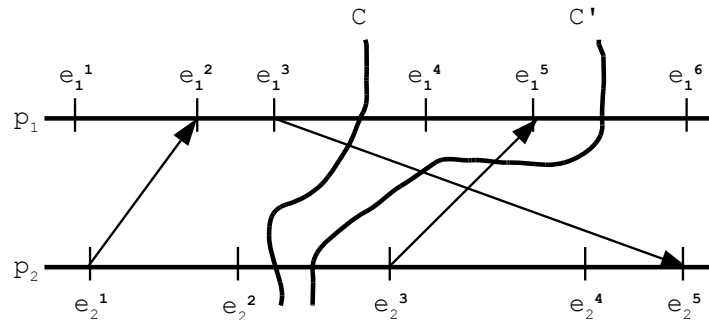
Τοπική και καθολική κατάσταση

- Τοπική κατάσταση διεργασίας p_i
 - Περιέχει τιμές όλων των τοπικών μεταβλητών
 - Μετά το e_i^k η κατάσταση είναι σ_i^k
 - Ορίζουμε ως σ_i^0 την αρχική κατάσταση
- Καθολική κατάσταση υπολογισμού
 - Ένωση τοπικών καταστάσεων διεργασιών
 - $\Sigma = (\sigma_1^{k1}, \dots, \sigma_n^{kn})$
 - Εξαρτάται από τα συμβάντα σε κάθε διεργασία

Τομή υπολογισμού (1 από 3)

- Τομή κατανεμημένου υπολογισμού
 - Υποσύνολο της καθολικής ιστορίας
 - Περιέχει k_i αρχικά γεγονότα από κάθε διεργασία p_i
 - $C = h_1^{k_1} \cup \dots \cup h_n^{k_n}$
 - Εναλλακτικά, αρκεί το (k_1, \dots, k_n)
 - Αντιστοιχεί στην καθολική κατάσταση $(\sigma_1^{k_1}, \dots, \sigma_n^{k_n})$
 - Τα τελευταία γεγονότα είναι το σύνορο της τομής
 - Μόνο ορισμένες καταστάσεις / τομές είναι εφικτές

Τομή υπολογισμού (2 από 3)



- Συνεπείς τομές: εφικτές καταστάσεις
 - Θα μπορούσαν να συμβούν στον υπολογισμό
 - Η τομή C είναι εφικτή
 - Η τομή C' είναι ανέφικτη
 - Περιέχει λήψη αλλά όχι αποστολή ενός μηνύματος

Τομή υπολογισμού (3 από 3)

- Πότε μία τομή C είναι συνεπής;
 - Για όλα τα συμβάντα e και e'
 - $(e \in C) \wedge (e' \rightarrow e) \Rightarrow e' \in C$
 - Αν το συμβάν e περιλαμβάνεται στην τομή C ...
 - ... πρέπει να περιλαμβάνεται και η αιτία του e'
 - Έστω ένα βέλος (μήνυμα) που «κόβει» την τομή
 - Πρέπει να ξεκινάει αριστερά και να καταλήγει δεξιά
 - Συνεπής καθολική κατάσταση
 - Αντιστοιχεί σε συνεπή τομή

Εκτέλεση υπολογισμού

- Εκτέλεση καταναεμημένου υπολογισμού
 - Πλήρης διάταξη γεγονότων καθολικής ιστορίας
 - Συνεπής με την πρόδρομη σχέση
 - $R = e_3^1 e_1^1 e_3^2 e_2^1 e_3^3 e_3^4 \dots$
 - Υπάρχουν κι άλλες δυνατές εκτελέσεις
 - Δίνει μια ακολουθία καθολικών καταστάσεων
 - $\Sigma = \Sigma^0 \Sigma^1 \Sigma^2 \Sigma^3 \dots$

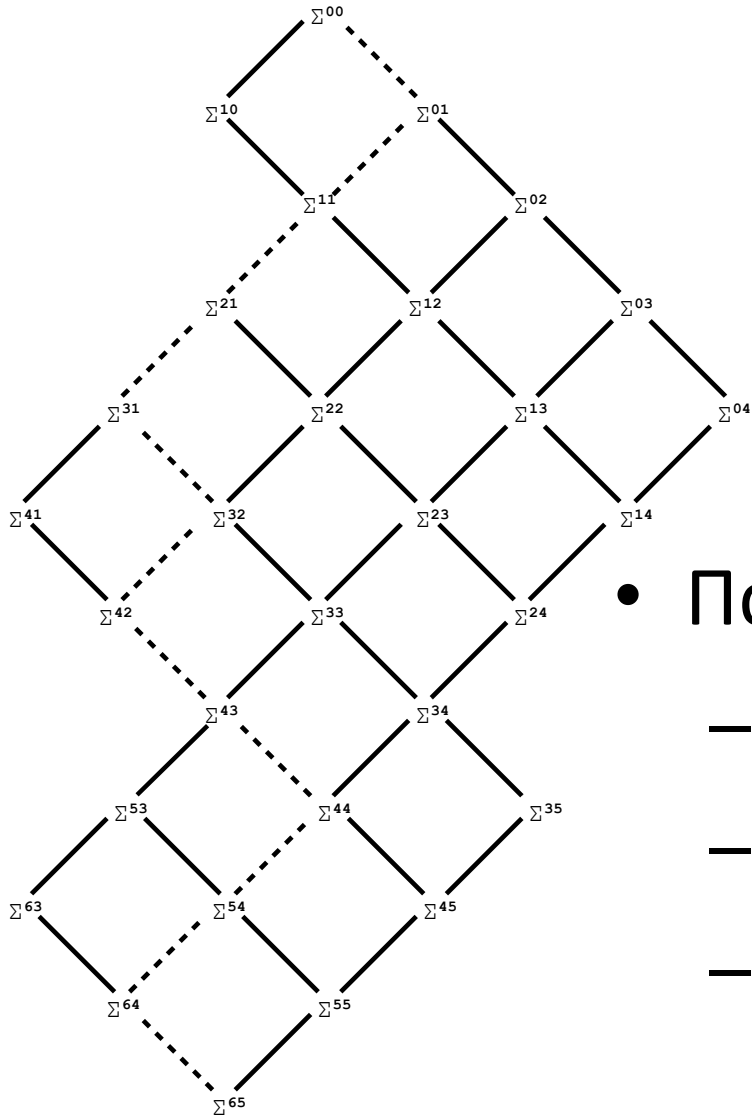
Προσπελάσιμες καταστάσεις

- Προσπελάσιμες καθολικές καταστάσεις
 - Έστω η ακολουθία καταστάσεων $\Sigma = \Sigma^0 \Sigma^1 \Sigma^2 \Sigma^3 \dots$
 - Η Σ^i προκύπτει από Σ^{i-1} με εκτέλεση συμβάντος
 - Η Σ^{i-1} οδηγεί στη Σ^i στην εκτέλεση R
 - Η Σ' είναι προσπελάσιμη από Σ στην R αν $\Sigma \rightarrow_R \Sigma'$
 - Η Σ' είναι προσπελάσιμη από Σ
 - Υπάρχει τουλάχιστον μία κατάλληλη R

Πλέγμα υπολογισμού (1 από 2)

- Αναπαράσταση υπολογισμού με πλέγμα
 - Περιέχει όλες τις συνεπείς καταστάσεις
 - Οι καταστάσεις συνδέονται με τις σχέσεις \rightarrow_R
 - Γενικά n ορθογώνιοι άξονες για n διεργασίες
 - Έστω η κατάσταση $(\sigma_1^{k_1}, \dots, \sigma_n^{k_n})$
 - Βρίσκεται στο επίπεδο $k_1 + \dots + k_n$ του πλέγματος
 - Κάθε διαδρομή είναι μία πιθανή εκτέλεση

Πλέγμα υπολογισμού (2 από 2)



- Παράδειγμα

- Υπολογισμός με δύο διεργασίες
- Διαδρομή: πιθανή εκτέλεση
- Σημείο: συνεπής τομή

Κατασκευή καταστάσεων (1 από 2)

- Πώς βρίσκουμε μια καθολική κατάσταση;
 - Χρειαζόμαστε καταστάσεις όλων των διεργασιών
 - Αναγκαστικά με ανταλλαγή μηνυμάτων
 - Δύσκολο πρόβλημα σε ασύγχρονο σύστημα
 - Καθυστέρηση ή / και απώλεια μηνυμάτων
 - Ξεπερασμένες, ατελείς, ασυνεπείς καταστάσεις
 - Κάθε διεργασία βγάζει διαφορετική εικόνα

Κατασκευή καταστάσεων (2 από 2)

- Κατασκευή καταστάσεων από παρατηρητή
 - Διεργασία που βρίσκει καθολικές καταστάσεις
 - Διεργασία παρακολουθητής
 - Δύο γενικές προσεγγίσεις
 - Παθητική προσέγγιση: παρατηρήσεις
 - Δεν εισάγουμε μηνύματα στο σύστημα
 - Ενεργητική προσέγγιση: στιγμιότυπα
 - Εισάγουμε μηνύματα στο σύστημα

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Παθητική στρατηγική: παρατηρήσεις

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 2: Διάταξη συμβάντων και καθολικές καταστάσεις

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



Παρατηρήσεις (1 από 3)

- Η διεργασία p_0 κατασκευάζει παρατήρηση O
 - Η p_i στέλνει μήνυμα στη p_0 για κάθε συμβάν e_i^k
 - Η παρατήρηση O εξαρτάται από σειρά άφιξης
- Ασύγχρονο σύστημα: καθυστερήσεις
 - Διαφορετικές διεργασίες φτιάχνουν διαφορετικά O
 - Οι παρατηρήσεις μπορεί να αντιστοιχούν σε
 - Μη εφικτές εκτελέσεις
 - Ασυνεπείς εκτελέσεις
 - Συνεπείς εκτελέσεις

Παρατηρήσεις (2 από 3)

- Συνεπείς παρατηρήσεις
 - Αντιστοιχούν σε συνεπείς εκτελέσεις
 - Διέρχονται μόνο από συνεπείς καταστάσεις
 - Σε κάθε στιγμή αντιστοιχούν σε συνεπείς τομές
- Κατασκευή συνεπών παρατηρήσεων
 - Διάκριση λήψης και παράδοσης μηνύματος
 - Ορισμός Κανόνων Παράδοσης (ΚΠ)
 - Πότε ένα ληφθέν μήνυμα μπορεί να παραδοθεί;

Παρατηρήσεις (3 από 3)

- Πρώτο βήμα: παράδοση FIFO
 - Μηνύματα που έχουν σταλεί από ίδια διεργασία i
 - $\text{send}_i(m) \rightarrow \text{send}_i(m') \Rightarrow \text{deliver}_j(m) \rightarrow \text{deliver}_j(m')$
 - Αρκεί η i να αριθμεί τα μηνύματά της
 - Οδηγεί σε εφικτές εκτελέσεις
 - Δεν παραβιάζεται η σειρά γεγονότων της i
 - Δεν οδηγεί απαραίτητα σε συνεπείς εκτελέσεις
 - Δεν ξέρουμε τι ισχύει ανάμεσα στις διεργασίες
 - Η FIFO συνδυάζεται με τους επόμενους ΚΠ

Πραγματικός χρόνος (1 από 2)

- Παράδοση σε σύγχρονα συστήματα
 - Η καθυστέρηση μετάδοσης έχει άνω όριο δ
 - Υπάρχει χρονιστής πραγματικού χρόνου RC
 - Κάθε συμβάν έχει χρονοσφραγίδα $RC(e)$
 - Κανόνας Παράδοσης 1 (ΚΠ1)
 - Παράδωσε με αύξουσα σειρά χρονοσφραγίδων
 - Την t παράδωσε μηνύματα με χρονοσφραγίδες ως $t-\delta$

Πραγματικός χρόνος (2 από 2)

- Κανόνας παράδοσης 1
 - Διασφαλίζει ότι δεν έχουμε χάσει μηνύματα
 - Πρέπει να φτάσουν το πολύ δ μετά την αποστολή
 - Διασφαλίζει καθολική διάταξη μηνυμάτων
 - Οδηγεί πάντα σε συνεπείς παρατηρήσεις
 - Τα περισσότερα συστήματα είναι ασύγχρονα!
 - Δεν έχουν καθολικό ρολόι
 - Δεν έχουν άνω όριο καθυστέρησης

Λογικοί χρονιστές (1 από 3)

- Παράδοση σε ασύγχρονα συστήματα
 - Κάθε διεργασία p_i έχει λογικό ρολόι LC_i
 - Κάθε μήνυμα περιέχει χρονοσφραγίδα $LC(e)$
 - Παράδοση με αύξουσα σειρά χρονοσφραγίδων
- Πώς ξέρουμε ότι είδαμε όλα τα μηνύματα;
 - Οι λογικοί χρονιστές ανιχνεύουν χάσματα
 - Έστω δύο γεγονότα e και e' με $LC(e) < LC(e')$
 - Υπάρχει γεγονός e'' τέτοιο ώστε $LC(e) < LC(e'') < LC(e')$;

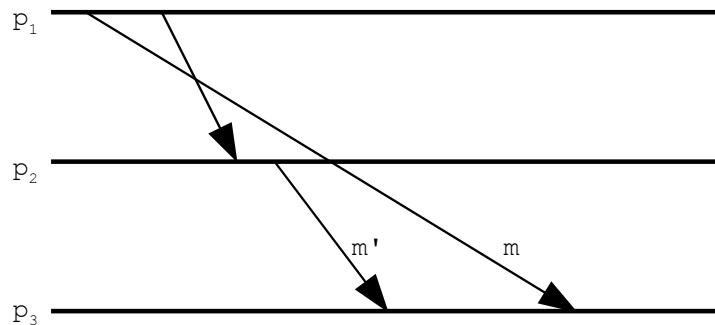
Λογικοί χρονιστές (2 από 3)

- Λογικοί χρονιστές και παράδοση FIFO
 - Αρίθμηση μηνυμάτων κάθε διεργασίας
 - Διασφάλιση FIFO ανάμεσα σε p_i και p_0
 - Έστω ότι έχουμε λάβει το m με $TS(m)$
 - Δεν θα λάβουμε αργότερα m' με $TS(m') < TS(m)$
 - Πότε ένα μήνυμα είναι σταθερό (stable);
 - Αν λάβουμε μήνυμα με μεγαλύτερη χρονοσφραγίδα
 - Από όλες τις άλλες διεργασίες

Λογικοί χρονιστές (3 από 3)

- Κανόνας παράδοσης 2 (ΚΠ2)
 - Παράδωσε όλα τα μηνύματα που είναι σταθερά
 - Ακολουθώντας τη σειρά των χρονοσφραγίδων
- Ανίχνευση χάσματος: γενικό πρόβλημα
 - Δεν λύνεται με χρονιστή πραγματικού χρόνου
 - Στα σύγχρονα συστήματα όμως δεν μας πειράζει
 - Το άνω όριο δ εντοπίζει τα σταθερά μηνύματα
 - Ουσιαστικά ο ΚΠ1 είναι απλούστευση του ΚΠ2

Αιτιώδης παράδοση (1 από 2)



- Αιτιώδης παράδοση
 - Η FIFO εφαρμόζεται σε μηνύματα μίας διεργασίας
 - Θέλουμε να την γενικεύσουμε σε πολλές διεργασίες
 - Αυτό εκφράζεται από την αιτιώδη παράδοση
 - $\text{send}_i(m) \rightarrow \text{send}_j(m') \Rightarrow \text{deliver}_k(m) \rightarrow \text{deliver}_k(m')$
 - Η παράδοση FIFO δεν αρκεί όμως
 - Αρκούν όμως οι ΚΠ1 και οι ΚΠ2

Αιτιώδης παράδοση (2 από 2)

- Οι ΚΠ1 και ΚΠ2 δεν είναι αποδοτικοί
 - Η σταθερότητα διασφαλίζεται με αναμονή
 - Στον ΚΠ1 περιμένουμε δ
 - Στον ΚΠ2 περιμένουμε επόμενα μηνύματα
 - Η αναμονή μπορεί να είναι περιττή
 - Έστω ένα μήνυμα με $LC(e)$
 - Πρέπει να πάρουμε e' με $LC(e') > LC(e)$ από όλους
 - Θα έπρεπε να περιμένουμε μόνο αν $e' \rightarrow e$

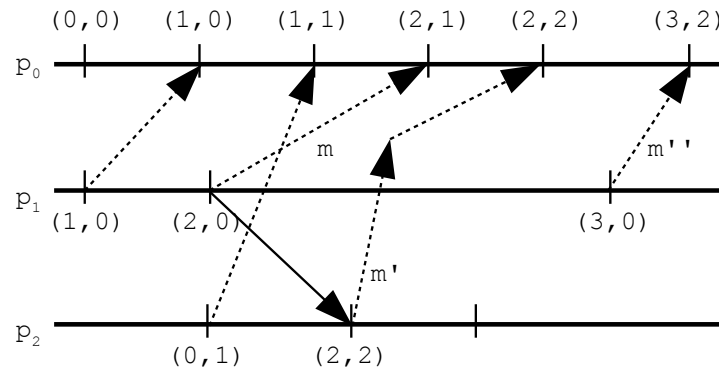
Διανυσματικοί χρονιστές (1 από 3)

- Πώς μπορούμε να αποφύγουμε την αναμονή;
 - Χρειαζόμαστε την ισχυρή συνθήκη χρονιστή
 - $TC(e) < TC(e') \Leftrightarrow e \rightarrow e'$
 - Άρα χρειαζόμαστε διανυσματικούς χρονιστές
 - Κάθε p_i έχει ένα διανυσματικό ρολόι VC_i
 - Τα VC_i ικανοποιούν την ισχυρή συνθήκη
 - Αν συγκρίνονται, τα γεγονότα έχουν αιτιώδη σχέση
 - Αν δεν συγκρίνονται, τότε είναι σύνδρομα

Διανυσματικοί χρονιστές (2 από 3)

- Πότε το m της p_j είναι σταθερό;
 - Έχουν παραδοθεί τα προηγούμενα μηνύματα της p_j
 - m' : τελευταίο μήνυμα της p_k που παραδόθηκε ($k \neq j$)
 - Δεν υπάρχει m'' της p_k που να μην παραδόθηκε ώστε:
 - $\text{send}_k(m') \rightarrow \text{send}_k(m'') \rightarrow \text{send}_j(m)$
 - Η p_0 διατηρεί πίνακα μετρητών $D[1 \dots n]$
 - $D[i] = \text{TS}(m_i)[i]$
 - m_i : τελευταίο μήνυμα της p_i που παραδόθηκε
 - Δείχνει πόσα μηνύματα παραδόθηκαν από την p_i

Διανυσματικοί χρονιστές (3 από 3)



- Κανόνας παράδοσης 3 (ΚΠ3)
 - Παράδωσε το μήνυμα m από την p_j όταν:
 - $D[j] = TS(m)[j] - 1$
 - Δηλαδή, είναι το επόμενο μήνυμα της p_j
 - $D[k] \geq TS(m)[k]$ για κάθε $k \neq j$
 - Δηλαδή, δεν υπάρχουν προηγούμενα μηνύματα της p_k
 - Μόλις παραδόσεις το m , θέσε $D[j] = TS(m)[j]$

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Ενεργητική στρατηγική: στιγμιότυπα

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 2: Διάταξη συμβάντων και καθολικές καταστάσεις
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



Στιγμιότυπα (1 από 3)

- Ενεργητική κατασκευή κατάστασης
 - Η p_0 ζητάει και συλλέγει τοπικές καταστάσεις
 - Η p_0 δημιουργεί μία καθολική κατάσταση
 - Λέμε ότι η p_0 δημιουργεί στιγμιότυπα
- Κατάσταση καναλιού
 - Σε ένα κανάλι γενικά βρίσκονται μηνύματα
 - $\chi_{i,j}$: μηνύματα από p_i που δεν έλαβε ακόμη η p_j
 - Έχουν επηρεάσει ήδη την σ_i
 - Δεν έχουν επηρεάσει ακόμη την σ_j

Στιγμιότυπα (2 από 3)

- Σύνολα καναλιών
 - IN_i = εισερχόμενα κανάλια της p_i
 - OUT_i = εξερχόμενα κανάλια της p_i
 - Διεργασίες άμεσα συνδεδεμένες με την p_i
- Υποθέσεις πρωτοκόλλων στιγμιοτύπων
 - Κανάλια FIFO και μίας κατεύθυνσης
 - Ισχυρά συνδεδεμένα, όχι πλήρης τοπολογία

Στιγμιότυπα (3 από 3)

- Καθολική κατάσταση
 - Τοπικές καταστάσεις + καταστάσεις καναλιών
 - Μας ενδιαφέρουν τα εισερχόμενα κανάλια
 - Στόχος: κατασκευή συνεπούς κατάστασης
 - Όταν ληφθεί το μήνυμα από την p_0
 - Καταγράφεται η σ_i
 - Καταγράφονται τα εισερχόμενα μηνύματα
 - Συνεχίζουμε μέχρι κάποια συνθήκη τερματισμού
 - Επιστροφή τοπικής κατάστασης + καταστάσεις καναλιών

Πρώτο πρωτόκολλο (1 από 4)

- Εφαρμόζεται σε σύγχρονο σύστημα
 - Υπάρχει καθολικός φυσικός χρονιστής RC
 - Κάθε μήνυμα έχει χρονοσφραγίδα αποστολής
 - Φραγμένη καθυστέρηση μηνυμάτων
- Λειτουργία πρώτου πρωτοκόλλου
 - Η p_0 επιλέγει μία χρονική στιγμή t_{ss}
 - Αρκετά μακρινή ώστε να ειδοποιηθούν όλοι
 - Η p_0 στέλνει παντού <λάβε στιγμιότυπο την t_{ss} >

Πρώτο πρωτόκολλο (2 από 4)

- Λειτουργία πρώτου πρωτοκόλλου
 - Όταν $RC = t_{ss}$ κάθε διεργασία p_i
 - Καταγράφει την τοπική της κατάσταση σ_i
 - Στέλνει ένα κενό μήνυμα στα OUT_i
 - Θέτει κάθε σύνολο $\chi_{j,i}$ ίσο με το κενό
 - Αρχίζει να καταγράφει τα μηνύματα από τα IN_i
 - Όταν η p_i δεχθεί μήνυμα από p_j με $TS(M) \geq t_{ss}$
 - Σταματάει την καταγραφή μηνυμάτων από την p_j
 - Δηλώνει στην p_0 την κατάσταση $\chi_{j,i}$

Πρώτο πρωτόκολλο (3 από 4)

- Αδιαίρετη εκτέλεση καταγραφής + αποστολής
 - Τα μηνύματα αυτά είναι τα πρώτα με $TS(m) \geq t_{ss}$
 - Δείχνουν πότε πρέπει να τελειώσει η καταγραφή
 - Τα μηνύματα που καταγράφονται στην $\chi_{i,j}$
 - Έχουν σταλεί από την χ_i πριν τη στιγμή t_{ss}
 - Θα ληφθούν από την p_j μετά τη στιγμή t_{ss}
 - Άρα είναι τα εκκρεμή μηνύματα τη στιγμή t_{ss}

Πρώτο πρωτόκολλο (4 από 4)

- Συνέπεια καθολικών καταστάσεων
 - Έστω τομή C_{ss} που αντιστοιχεί στην κατάσταση
 - Το e ανήκει στην C_{ss} αν και μόνο αν $RC(e) < t_{ss}$
 - Η συνέπεια απαιτεί $(e \in C_{ss}) \wedge (e' \rightarrow e) \Rightarrow (e' \in C_{ss})$
 - Το RC διασφαλίζει ότι $e' \rightarrow e \Rightarrow RC(e') < RC(e)$
 - Άρα $(e' \in C_{ss})$ λόγω του ότι $RC(e') < RC(e) < t_{ss}$

Δεύτερο πρωτόκολλο (1 από 4)

- Ασύγχρονο σύστημα
 - Υπάρχουν μόνο λογικοί χρονιστές LC_i
 - Μη φραγμένη καθυστέρηση μηνυμάτων
 - Πρόβλημα: η συνθήκη $LC_i = t_{ss}$ δεν έχει νόημα
 - Ο LC_i μπορεί να μην λάβει ποτέ αυτή την τιμή
 - Ο έλεγχος της τιμής αυξάνει το t_{ss} κατά 1
 - Πρόβλημα: πόσο μεγάλο πρέπει να είναι το t_{ss} ;
 - Διαλέγουμε μια τιμή ω που είναι πολύ μακριά

Δεύτερο πρωτόκολλο (2 από 4)

- Αλγόριθμος εκκίνησης στιγμιότυπων
 - Αν (e εσωτερικό συμβάν ή αποστολή) και ($LC = t_{ss} - 2$)
 - Εκτέλεσε το συμβάν e
 - Ξεκίνα τον αλγόριθμο στιγμιότυπων
 - Αν (e λήψη) και ($TS(m) \geq t_{ss}$) και ($LC \leq t_{ss} - 1$)
 - Τοποθέτησε το μήνυμα πίσω στο κανάλι
 - Ενεργοποίησε το συμβάν e για εκτέλεση
 - Θέσε $LC = t_{ss} - 1$
 - Ξεκίνα τον αλγόριθμο στιγμιότυπων

Δεύτερο πρωτόκολλο (3 από 4)

- Λειτουργία δευτέρου πρωτοκόλλου
 - Η p_0 στέλνει παντού <λάβε στιγμιότυπο την ω >
 - Η p_0 θέτει $LC_0 = \omega$
 - Όταν $LC_i = \omega$ κάθε διεργασία p_i
 - Καταγράφει την τοπική της κατάσταση σ_i
 - Στέλνει ένα κενό μήνυμα στα OUT_i
 - Θέτει κάθε σύνολο $\chi_{j,i}$ ίσο με το κενό
 - Αρχίζει να καταγράφει τα μηνύματα από τα IN_i

Δεύτερο πρωτόκολλο (4 από 4)

- Λειτουργία δεύτερου πρωτοκόλλου
 - Όταν η p_i δεχθεί μήνυμα από p_j με $TS(M) \geq \omega$
 - Σταματάει καταγραφή μηνυμάτων από την p_j
 - Δηλώνει στην p_0 την κατάσταση $\chi_{j,i}$
- Η p_0 φτάνει άμεσα στο ω
 - Άρα στέλνει κενά μηνύματα με χρονοσφραγίδα ω
 - Όλες οι p_i θα φτάσουν το ω μόλις τα λάβουν
 - Άρα ο αλγόριθμος θα ξεκινήσει παντού στο ω

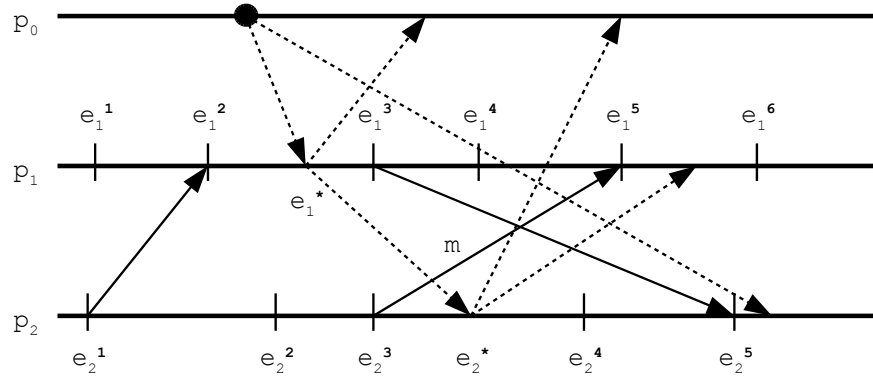
Τρίτο πρωτόκολλο (1 από 3)

- Τρίτο πρωτόκολλο (Chandy και Lamport)
 - Απλοποίηση δευτέρου πρωτοκόλλου
 - Το <λάβε στιγμιότυπο την ω > είναι περιττό
 - Αρκεί να ξεκινάμε όταν λάβουμε το κενό μήνυμα
 - Αν όμως υπάρχουν και άλλα κενά μηνύματα;
 - Αντί για κενό, στέλνουμε <πάρε στιγμιότυπο>
 - Οι λογικοί χρονιστές είναι περιττοί
 - Το ειδικό μήνυμα ξεκινάει τον αλγόριθμο

Τρίτο πρωτόκολλο (2 από 3)

- Λειτουργία τρίτου πρωτοκόλλου
 - Η p_0 στέλνει <πάρε στιγμιότυπο> στον εαυτό της
 - Όταν η p_i δέχεται <πάρε στιγμιότυπο> πρώτη φορά από p_f
 - Καταγράφει την τοπική της κατάσταση s_i
 - Στέλνει <πάρε στιγμιότυπο> στα OUT_i
 - Θέτει το σύνολο $\chi_{f,i}$ ίσο με το κενό
 - Αρχίζει να καταγράφει τα μηνύματα από IN_i εκτός από p_f
 - Όταν η p_i δέχεται ξανά <πάρε στιγμιότυπο> από $p_s \langle \rangle p_f$
 - Σταματάει την καταγραφή των μηνυμάτων από p_s
 - Δηλώνει στην p_0 την κατάστασή $\chi_{s,i}$

Τρίτο πρωτόκολλο (3 από 3)



- Παράδειγμα τρίτου πρωτοκόλλου
 - Το $\chi_{1,2}$ είναι κενό: στο γεγονός e_1^* δεν εκκρεμεί τίποτα
 - Το $\chi_{2,1}$ περιέχει το m : στο γεγονός e_2^* εκκρεμεί
- Ορθότητα πρωτοκόλλου
 - Αρχική κατάσταση Σ^α , τελική Σ^τ , κατασκευάζει την Σ^σ
 - Τότε υπάρχει εκτέλεση R τέτοια ώστε $\Sigma^\alpha \rightarrow_R \Sigma^\sigma \rightarrow_R \Sigma^\tau$

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Τέλος Ενότητας # 2

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 2: Διάταξη συμβάντων και καθολικές καταστάσεις

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

