

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός

Ενότητα # 4: Αμοιβαίος αποκλεισμός

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

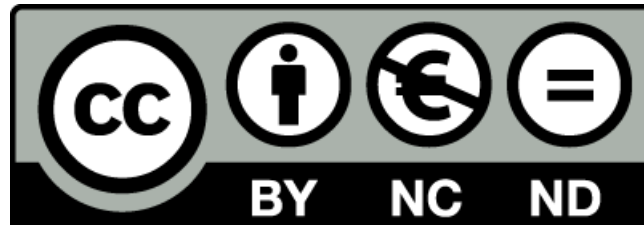
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

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 του προβλήματος του αμοιβαίου αποκλεισμού στα κατανεμημένα συστήματα.
- Εξοικείωση με τη συγκεντρωτική προσέγγιση στον αμοιβαίο αποκλεισμό.
- Κατανόηση των βασικών αλγορίθμων κατανεμημένου αμοιβαίου αποκλεισμού (Lamport, Ricart-Agrawala, LeLann, Chandy και Raymond).

Περιεχόμενα ενότητας

- Πρόβλημα αμοιβαίου αποκλεισμού
- Συγκεντρωτική προσέγγιση
- Αλγόριθμος Lamport
- Αλγόριθμος Ricart-Agrawala
- Αλγόριθμος LeLann
- Αλγόριθμος Chandy
- Αλγόριθμος Raymond

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

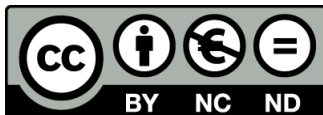


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Πρόβλημα αμοιβαίου αποκλεισμού

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Απαιτήσεις αμοιβαίου αποκλεισμού

- Ίδιο πρόβλημα με συγκεντρωτικά συστήματα
 - Αλλά χωρίς κοινό ρολόι και κοινή μνήμη
- Ασφάλεια: το πολύ μια διεργασία στην ΚΠ
- Βιωσιμότητα: κάθε αίτηση θα ικανοποιηθεί
 - Δε συμβαίνουν αδιέξοδα
 - Δε συμβαίνει υποσιτισμός
- Διάταξη: εξυπηρέτηση με τη σειρά
 - Σύμφωνα με την πρόδρομη σχέση

Γενικές υποθέσεις

- Κάθε διεργασία έχει άλλο αναγνωριστικό
- Μία μηχανή ανά διεργασία
- Επικοινωνία διεργασιών με μηνύματα
- Ανταγωνισμός για έναν μόνο πόρο
- Οι επεξεργαστές είναι πλήρως συνδεδεμένοι

Επίδοση συστήματος

- Κριτήρια επίδοσης
 - Πλήθος μηνυμάτων μέχρι την είσοδο στην ΚΠ
 - Πλήθος μηνυμάτων μέχρι την έξοδο από ΚΠ
- Επίδοση στην πράξη
 - Εξαρτάται από τις διεργασίες
 - Μικρές και αραιά εκτελούμενες ΚΠ
 - Κυριαρχεί το κόστος του αλγόριθμου
 - Μεγάλες και συχνά εκτελούμενες ΚΠ
 - Κυριαρχεί ο χρόνος αναμονής σε ουρές

Προσεγγίσεις

- Συγκεντρωτική προσέγγιση
 - Συντονιστής
- Κατανεμημένη προσέγγιση
 - Διάταξης γεγονότων
 - Αλγόριθμος Lamport (1978)
 - Αλγόριθμος Ricart-Agrawala (1981)
 - Μεταβίβασης σκυτάλης
 - Τοπολογία δακτυλίου (LeLann, 1977)
 - Τοπολογία δένδρου (Raymond, 1989)
 - Οποιαδήποτε τοπολογία (Chandy, 1982)

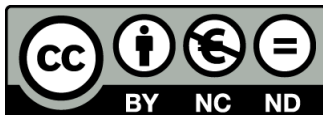
**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Συγκεντρωτική προσέγγιση

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



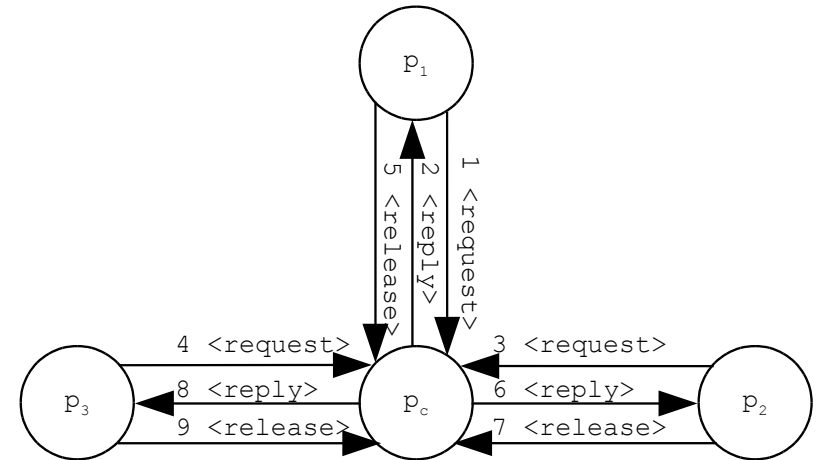
Συγκεντρωτικός αλγόριθμος (1 από 5)

- Χρήση συντονιστή για έλεγχο πρόσβασης
 - Εκλέγεται από διεργασίες συστήματος
 - Δίνει άδεια πρόσβασης μετά από αίτηση
 - Διατηρεί ουρά αναβληθεισών αιτήσεων
- Αλγόριθμος αμοιβαίου αποκλεισμού
 - Όταν η p_i θέλει πρόσβαση στην ΚΠ
 - Στέλνει $\langle \text{request}, i \rangle$ στον συντονιστή p_c

Συγκεντρωτικός αλγόριθμος (2 από 5)

- Αλγόριθμος αμοιβαίου αποκλεισμού
 - Όταν ο συντονιστής p_c λάβει $\langle \text{request}, i \rangle$
 - Αν η ΚΠ είναι ελεύθερη, στέλνει $\langle \text{reply} \rangle$
 - Αλλιώς εισάγει την αίτηση στην ουρά
 - Όταν η p_i λάβει $\langle \text{reply} \rangle$ εισέρχεται στην ΚΠ
 - Όταν η p_i βγει από την ΚΠ
 - Στέλνει $\langle \text{release}, i \rangle$ στον συντονιστή p_c
 - Αν μη άδεια ουρά, ο p_c στέλνει ένα $\langle \text{reply} \rangle$

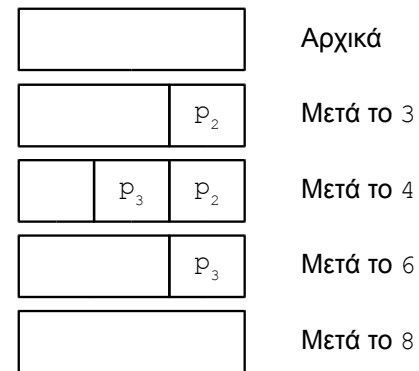
Συγκεντρωτικός αλγόριθμος (3 από 5)



(α)

- Παράδειγμα

- Συντονιστής (p_c)
- Τρεις διεργασίες (p_1, p_2, p_3)
- Πολιτική FIFO
- Σειρά αιτήσεων: 1, 2, 3



(β)

Συγκεντρωτικός αλγόριθμος (4 από 5)

- Απόδοση
 - Ανταλλαγή 3 μηνυμάτων ανά είσοδο
 - Ανταλλαγή 2 μηνυμάτων πριν την είσοδο
- Η απαίτηση διάταξης δεν καλύπτεται!
 - Εξαρτάται από καθυστερήσεις δικτύου
- Απλή υλοποίηση
 - Ακόμη και στο συντονιστή

Συγκεντρωτικός αλγόριθμος (5 από 5)

- Μειονέκτημα: αποτυχία συντονιστή
 - Ανίχνευση αποτυχίας
 - Εκλογή νέου συντονιστή
 - Ανακατασκευή της ουράς
- Καθυστέρηση <reply>
 - Μήπως απέτυχε ο συντονιστής;
 - Λύση: αποστολή <permission_denied>
 - Ο συντονιστής είναι ζωντανός

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αλγόριθμος Lamport

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



Αλγόριθμος Lamport (1 από 6)

- Η διεργασία ρωτά τις άλλες πριν μπει στην ΚΠ
 - Μήνυμα request
- Περιμένει να πάρει απαντήσεις από όλες
 - Μήνυμα reply
- Όταν βγει από ΚΠ στέλνει μηνύματα σε όλες
 - Μήνυμα release
- Τα μηνύματα διατηρούνται σε ουρά αιτήσεων
 - Ταξινόμηση με βάση λογικές χρονοσφραγίδες

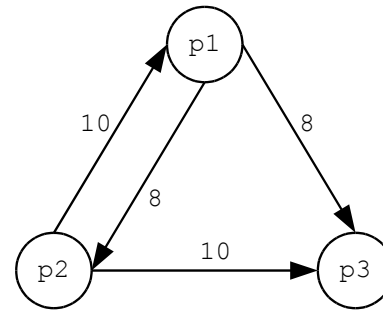
Αλγόριθμος Lamport (2 από 6)

- Όταν p_i θέλει να εισέλθει στην ΚΠ
 - Αποστολή $\langle \text{request}, t_i \rangle$ σε όλες τις διεργασίες
 - Εισάγει την αίτησή της στην ουρά της
- Λήψη $\langle \text{request}, t_i \rangle$ από την p_j
 - Εισάγει την αίτηση στην ουρά με βάση το t_i
 - Αποστολή $\langle \text{reply} \rangle$ στην p_i
- Όταν η p_i βγαίνει από την ΚΠ
 - Διαγράφει την αίτησή της από την ουρά της
 - Στέλνει $\langle \text{release} \rangle$ σε όλες τις διεργασίες

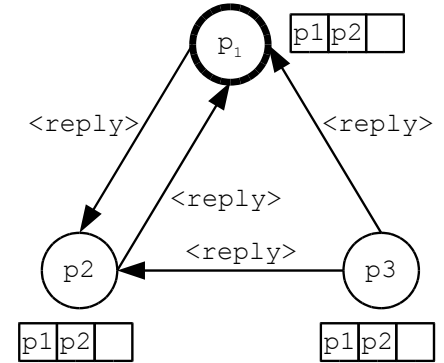
Αλγόριθμος Lamport (3 από 6)

- Λήψη <release> από την p_j
 - Αφαιρεί αίτηση της p_i από την ουρά της
- Η p_i εισέρχεται στην ΚΠ αν και μόνο αν
 - Έχει λάβει <reply> από όλες τις άλλες διεργασίες
 - Η αίτησή της είναι στην κορυφή της ουράς της

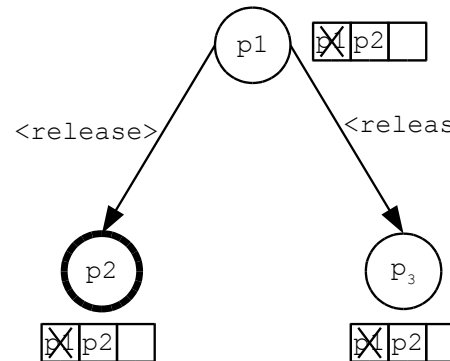
Αλγόριθμος Lamport (4 από 6)



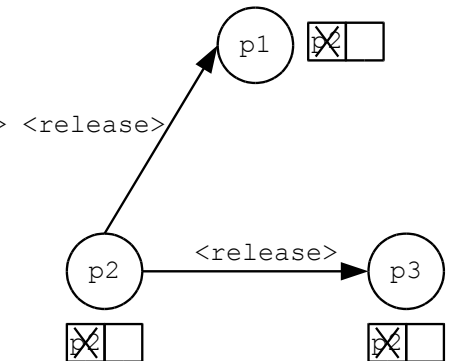
(α)



(β)



(γ)



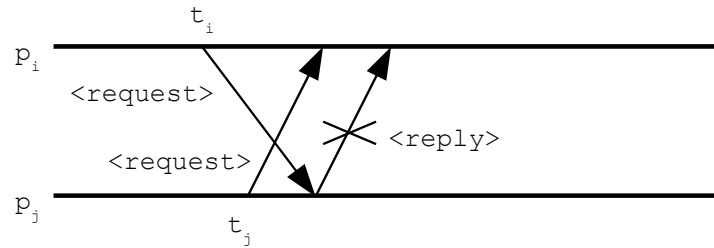
(δ)

- Παράδειγμα

- p_1 : χρονοσφραγίδα 8

- p_2 : χρονοσφραγίδα 10

Αλγόριθμος Lamport (5 από 6)



- Βελτίωση του αλγορίθμου
 - Η p_i στέλνει μήνυμα με TS t_i
 - Η p_j στέλνει μήνυμα με TS t_j
 - Η p_j δεν χρειάζεται να στείλει reply αν $t_i < t_j$
 - Η p_i ξέρει ότι η p_j ακολουθεί
 - Έστω ότι η p_i έλαβε m αιτήσεις με πιο πρόσφατο TS
 - Αρκεί να λάβει reply από τις άλλες $n-m-1$ διεργασίες

Αλγόριθμος Lamport (6 από 6)

- Ασφάλεια: όλοι έχουν ίδια σειρά στις ουρές
 - Εκεί χρησιμεύουν τα reply
- Βιωσιμότητα: κάθε αίτηση εξυπηρετείται
 - Η τρέχουσα διεργασία εξέρχεται από ΚΠ
 - Η επόμενη στην ουρά εισέρχεται στην ΚΠ
- Διάταξη: εξυπηρέτηση με τη σειρά
 - Χρήση των χρονοσφραγίδων Lamport
- Πολυπλοκότητα: $3(n-1)$ ανά είσοδο
 - Καθυστέρηση $2(n-1)$ πριν την είσοδο

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αλγόριθμος Ricart-Agrawala

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



Αλγόριθμος Ricart-Agrawala (1 από 6)

- Παραλλαγή αλγόριθμου Lamport
 - Απάντηση και αποδέσμευση με μήνυμα <OK>
 - Αντί για <reply> και <release>
 - Κάθε διεργασία διατηρεί μία ουρά
 - Ταξινομημένη με βάση λογικές χρονοσφραγίδες
 - Αποστολή αιτήσεων σε όλους πριν από ΚΠ
 - Όποιος δεν ενδιαφέρεται απαντά άμεσα
 - Όποιος ενδιαφέρεται απαντά ανάλογα με διάταξη

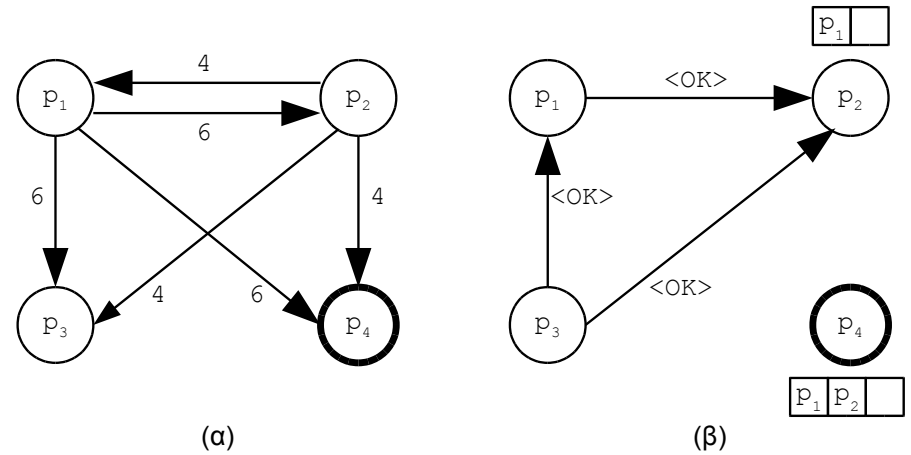
Αλγόριθμος Ricart-Agrawala (2 από 6)

- Όταν η p_i θέλει να εισέλθει στην ΚΠ
 - Αποστολή $\langle \text{request}, t_i \rangle$ σε όλες τις διεργασίες
- Όταν η p_j λαμβάνει $\langle \text{request}, t_i \rangle$ από την p_i
 - Αν είναι στην ΚΠ εισάγει την αίτηση στην ουρά της
 - Διαφορετικά
 - Αν δεν ενδιαφέρεται για την ΚΠ στέλνει $\langle \text{OK} \rangle$
 - Αν έχει εκδώσει αίτηση με t_j
 - Αν $t_i < t_j$ στέλνει $\langle \text{OK} \rangle$
 - Αλλιώς εισάγει αίτηση στην ουρά της

Αλγόριθμος Ricart-Agrawala (3 από 6)

- Όταν η p_i λάβει <OK> από όλες τρεις διεργασίες
 - Εισέρχεται στην ΚΠ
- Όταν η p_i βγει από την ΚΠ
 - Στέλνει <OK> στις διεργασίες στην ουρά της
 - Διαγράφει όλες τις διεργασίες από την ουρά της

Αλγόριθμος Ricart-Agrawala (4 από 6)

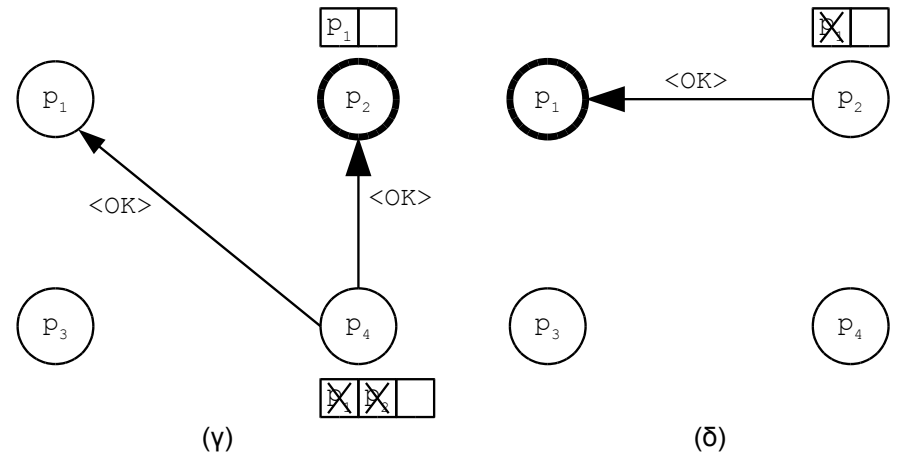


(α)

(β)

- Παράδειγμα

- p_4 : είναι στην ΚΠ
- p_1 : χρονοσφραγίδα 6
- p_2 : χρονοσφραγίδα 4



(γ)

(δ)

Αλγόριθμος Ricart-Agrawala (5 από 6)

- Ισχύουν οι απαιτήσεις;
 - Ασφάλεια: η διεργασία με $\min TS$ έχει τα <OK>
 - Βιωσιμότητα: αλυσίδα αναμονής με βάση τα TS
 - Διάταξη: η είσοδος στην ΚΠ γίνεται με βάση τα TS
- Κόστος αλγορίθμου
 - $2(n-1)$ μηνύματα για κάθε είσοδο στην ΚΠ
 - Ίδιο ακριβώς πλήθος πριν την είσοδο στην ΚΠ

Αλγόριθμος Ricart-Agrawala (6 από 6)

- Κατανεμημένοι ή συγκεντρωτικοί αλγόριθμοι;
 - Ο κατανεμημένος είναι χειρότερος!
 - Έχει πολλά σημεία αποτυχίας (κάθε διεργασία)
 - Όλοι έχουν τόσο φόρτο όσο ο συντονιστής
 - Κάθε διεργασία πρέπει να γνωρίζει όλες τις άλλες
- Γιατί ασχολούμαστε λοιπόν;
 - Υπάρχει κατανεμημένος αλγόριθμος
 - Υπάρχουν και πιο γρήγορες λύσεις

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αλγόριθμος LeLann

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



Αλγόριθμος LeLann (1 από 4)

- Χρήση ειδικού μηνύματος: σκυτάλη (token)
 - Είσοδος στην ΚΠ μόνο αν έχουμε τη σκυτάλη
- Αλγόριθμος LeLann
 - Λογική διάταξη διεργασιών σε δακτύλιο
 - Η σκυτάλη μεταδίδεται στο δακτύλιο
 - Μετάδοση πάντα στην επόμενη στο δακτύλιο
- Όταν η p_i λαμβάνει τη σκυτάλη
 - Αν θέλει να μπει στην ΚΠ δεσμεύει τη σκυτάλη
 - Μόλις βγει από την ΚΠ μεταβιβάζει τη σκυτάλη
 - Αλλιώς τη μεταβιβάζει άμεσα στην επόμενη

Αλγόριθμος LeLann (2 από 4)

- Απαιτήσεις αμοιβαίου αποκλεισμού
 - Ασφάλεια: μόνο ο κάτοχος μπαίνει στην ΚΠ
 - Βιωσιμότητα: όλοι παίρνουν τη σκυτάλη
 - Σε κάθε λήψη η διεργασία μπαίνει μία φορά στην ΚΠ
 - Διάταξη: με όποια σειρά επιβάλλει ο δακτύλιος
- Απόδοση
 - $1-\infty$ μηνύματα ανά είσοδο στην ΚΠ
 - $0-(n-1)$ μηνύματα καθυστέρηση μέχρι την είσοδο

Αλγόριθμος LeLann (3 από 4)

- Αποτυχία διεργασίας
 - Αποτυχία ολόκληρου του συστήματος
 - Χρήση μηνύματος <acknowledgement>
 - Στέλνεται σε προηγούμενη στη λήψη της σκυτάλης
 - Αν δεν λάβει επιβεβαίωση
 - Στέλνει τη σκυτάλη παραπέρα
 - Πρέπει να γνωρίζει μερικές επόμενες

Αλγόριθμος LeLann (4 από 4)

- Απώλεια σκυτάλης
 - Εκλογή αρχηγού για παρακολούθηση σκυτάλης
 - Στέλνει περιοδικά μήνυμα <who has the token>
 - Όποιος την έχει θέτει ένα πεδίο στη μήνυμα
 - Αν επιστρέψει χωρίς απάντηση
 - Δημιουργία νέας σκυτάλης
 - Αποστολή σκυτάλης σε επόμενο

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αλγόριθμος Chandy

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Αλγόριθμος Chandy (1 από 3)

- Δεν εξαρτάται από τοπολογία
 - Υποθέτει ότι γνωρίζουμε όλες τις διεργασίες!
- Η σκυτάλη μεταβιβάζεται μόνο όταν πρέπει
- Μορφή σκυτάλης
 - Η σκυτάλη είναι ένα διάνυσμα (f_1, f_2, \dots, f_N)
 - f_i = πόσες φορές η p_i μπήκε στην ΚΠ
- Κάθε διεργασία διατηρεί ουρά αιτήσεων

Αλγόριθμος Chandy (2 από 3)

- Όταν η p_i θέλει να μπει στην ΚΠ
 - Στέλνει $\langle \text{request } p_i, m_i \rangle$ στις άλλες διεργασίες
 - Περιμένει μέχρι να λάβει τη σκυτάλη
- Όταν η p_j λαμβάνει $\langle \text{request } p_i, m_i \rangle$
 - Αν δεν έχει τη σκυτάλη
 - Εισάγει την αίτηση στην ουρά της
 - Αν έχει τη σκυτάλη
 - Αν δεν είναι σε ΚΠ στέλνει τη σκυτάλη στην p_i
 - Αν είναι σε ΚΠ εισάγει την αίτηση στην ουρά

Αλγόριθμος Chandy (3 από 3)

- Όταν η p_i λάβει τη σκυτάλη
 - Θέτει $f_i = m_i$ στη σκυτάλη και μπαίνει στην ΚΠ
- Όταν η p_i βγει από την ΚΠ
 - Αν η ουρά αιτήσεων είναι κενή
 - Κρατάει τη σκυτάλη μέχρι να λάβει αίτηση
 - Αν η ουρά αιτήσεων δεν είναι κενή
 - Εξάγει την πρώτη αίτηση $\langle \text{request } p_j, m_j \rangle$
 - Εάν $m_j > f_j$ στέλνει τη σκυτάλη στην p_j
 - Αλλιώς αγνοεί την αίτηση και επαναλαμβάνει

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**

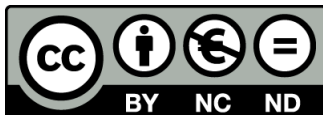


**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Αλγόριθμος Raymond

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: 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



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Αλγόριθμος Raymond (1 από 8)

- Υποθέτει διεργασίες οργανωμένες σε δένδρο
 - Μπορεί να είναι επικαλυπτικό δένδρο
 - Δεν συμμετέχουν όλες οι διεργασίες
 - Ορισμένες εκπροσωπούνται από άλλες
 - Η σκυτάλη βρίσκεται πάντα στη ρίζα
 - Οι ακμές δείχνουν πάντα προς τη ρίζα
 - Η σκυτάλη αντιστρέφει ακμές όταν κινείται
 - Κάθε διεργασία διατηρεί ουρά αιτήσεων

Αλγόριθμος Raymond (2 από 8)

- Όταν η p_i θέλει να μπει στη ΚΠ
 - Εισάγει την αίτησή της στην ουρά της
 - Αν ήταν κενή στέλνει $\langle \text{request } p_i \rangle$ σε γονέα
- Όταν η p_j λάβει $\langle \text{request } p_i \rangle$ και δεν είναι ρίζα
 - Εισάγει την αίτηση στην ουρά
 - Αν ήταν κενή στέλνει $\langle \text{request } p_j \rangle$ σε γονέα

Αλγόριθμος Raymond (3 από 8)

- Όταν η p_j λάβει $\langle \text{request } p_i \rangle$ και είναι ρίζα
 - Αν είναι σε ΚΠ εισάγει το μήνυμα στην ουρά
 - Αν δεν είναι σε ΚΠ
 - Αντιστρέφει την ακμή λήψης
 - Στέλνει τη σκυτάλη στην ακμή

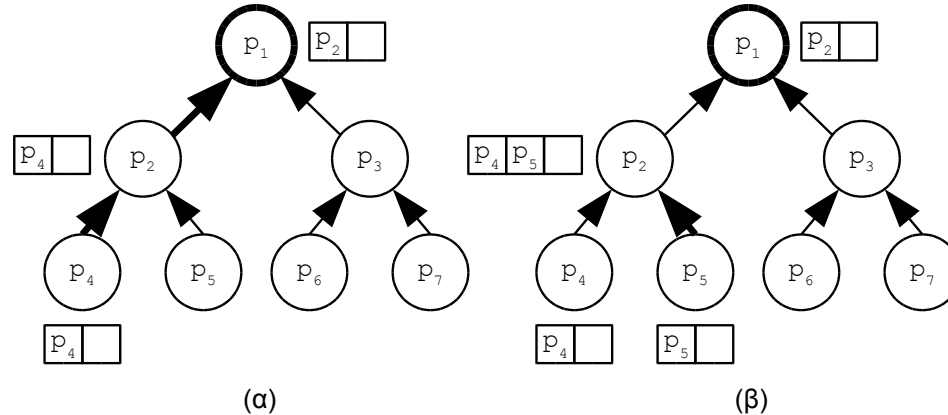
Αλγόριθμος Raymond (4 από 8)

- Όταν η p_j λαμβάνει τη σκυτάλη
 - Εξάγει αίτηση p_k από την κεφαλή της ουράς
 - Αν η p_k είναι η p_j εισέρχεται στην ΚΠ
 - Αν η p_k είναι άλλη διεργασία
 - Αντιστρέφει την ακμή προς την p_k
 - Στέλνει τη σκυτάλη στην ακμή
 - Αν η ουρά της δεν είναι άδεια, στέλνει $\langle \text{request } p_j \rangle$ στην p_k

Αλγόριθμος Raymond (5 από 8)

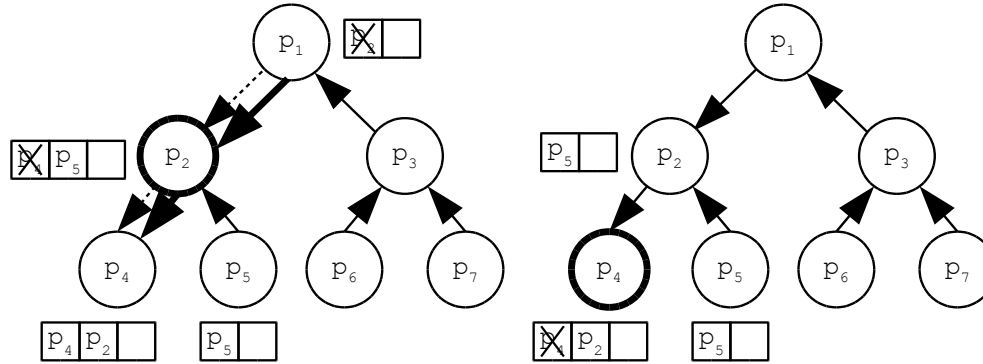
- Όταν η p_j βγαίνει από την ΚΠ
 - Αν η ουρά της δεν είναι άδεια
 - Εξάγει την p_k από την κεφαλή της ουράς
 - Αντιστρέφει την ακμή προς την p_k
 - Στέλνει τη σκυτάλη στην ακμή
 - Αν η ουρά της δεν είναι άδεια, στέλνει `<request p_j >` στην p_k

Αλγόριθμος Raymond (6 από 8)



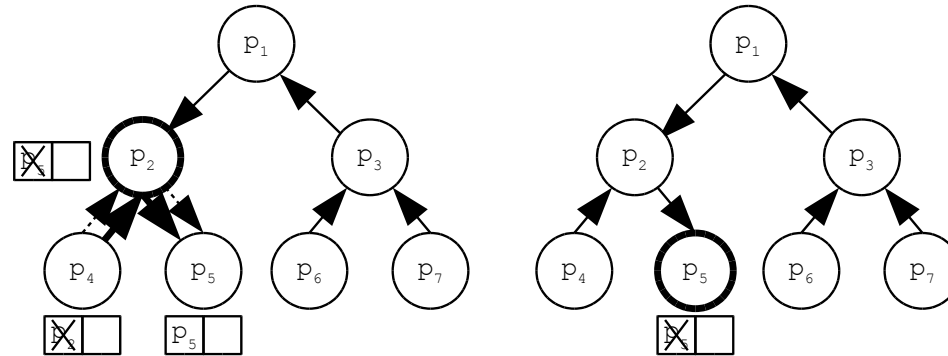
- Παράδειγμα
 - Αρχικά σκυτάλη στην p_1
 - Η p_4 θέλει να μπει στην ΚΠ
 - Η p_5 θέλει να μπει στην ΚΠ
 - Η p_2 τις εκπροσωπεί

Αλγόριθμος Raymond (7 από 8)



(γ)

(δ)



(ε)

(στ)

- Παράδειγμα: συνεχίζει η p_2 και μετά η p_5

Αλγόριθμος Raymond (8 από 8)

- Απαιτήσεις αμοιβαίου αποκλεισμού
 - Ασφάλεια: το κουπόνι βρίσκεται στην ρίζα
 - Βιωσιμότητα: ακυκλική δομή δένδρου
 - Διάταξη: επιβάλλεται από δομή δένδρου
- Πολυπλοκότητα: ανάλογη του $\log n$
 - Υποθέτουμε τυχαία κατασκευή δένδρου
 - Τόση είναι η μέση απόσταση δύο κόμβων

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Τέλος Ενότητας # 4

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 4: Αμοιβαίος αποκλεισμός

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

