

Διάλεξη 6: Εκλογή Προέδρου σε Σύγχρονους Δακτύλιους

ΕΠΛ 432: Κατανεμημένοι Αλγόριθμοι



Τι θα δούμε σήμερα

- Μη Ομοιόμορφος Αλγόριθμος Εκλογής Προέδρου σε Σύγχρονο Δακτύλιο
- Ομοιόμορφος Αλγόριθμος Εκλογής Προέδρου σε Σύγχρονο Δακτύλιο

Ομοιομορφία στους Μη-Ανώνυμους Αλγορίθμους

- **Ομοιόμορφος Αλγόριθμος:** δεν χρησιμοποιεί το μέγεθος του δακτυλίου
 - Υπάρχει μια μηχανή καταστάσεων για κάθε ταυτότητα, ανεξάρτητα από το μέγεθος του δακτυλίου
- **Μη-Ομοιόμορφος Αλγόριθμος:** χρησιμοποιεί το μέγεθος του δακτύλιου
 - Υπάρχει μια μηχανή καταστάσεων για κάθε ταυτότητα, και για κάθε μέγεθος δακτυλίου

Επισκόπηση Αλγορίθμων

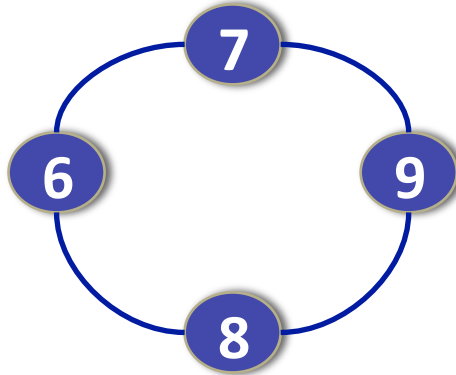
- Ασυμπτωτικά Βέλτιστα Φράγματα (με βάση την πολυπλοκότητα μηνυμάτων):
 - Ασύγχρονο Δακτύλιο
 - $\Theta(n \log n)$ μηνύματα
 - Σύγχρονο Δακτύλιο
 - $\Theta(n)$ κάτω από κάποιες προϋποθέσεις
 - αλλιώς, $\Theta(n \log n)$ μηνύματα

Μη Ομοιόμορφος Αλγόριθμος

- Αλγόριθμος επεξεργαστή με $AT=\chi$

- Προχώρα στην επόμενη φάση ϕ
- Αν $\phi=\chi$ ανακηρύξου πρόεδρος και στείλε τον AT σου χ προς τα αριστερά
- Αλλιώς εκτέλεσε η γύρους
 - Αν σε κάποιο από αυτούς τους γύρους λάβεις ένα $AT=\psi$ τότε ο ψ είναι πρόεδρος και εσύ δεν είσαι.

Παράδειγμα εκτέλεσης



- Φάσεις 0-5
 - Εκτελούνται 24 γύροι (4 γύροι σε κάθε φάση)
 - Κανένα μήνυμα δεν αποστέλλεται
- Φάση 6
 - Ο κόμβος με $AT=6$ στέλνει μήνυμα στον γύρο 25 το οποίο προωθείται σε όλο τον δακτύλιο.

Παρατηρήσεις

- Ποιος θα εκλεγεί Πρόεδρος;
 - Αυτός με τον μικρότερο AT
- Τι προϋποθέτει ο αλγόριθμος;
 - Γνωρίζουμε τον αριθμό των κόμβων n
 - Γνωρίζουμε τον συγχρονισμό των επεξεργασιών
 - Όλοι ξεκινούν τον ίδιο γύρο

Ανάλυση Αλγορίθμου

- **Ορθότητα:**
 - **Ικανοποιεί ΣΑ:** Εκλέγει ένα πρόεδρο, αυτόν με τον μικρότερο ΑΤ.
 - το μόνο μήνυμα που στέλνετε είναι αυτό που κουβαλά τον μικρότερο ΑΤ και κάνει το γύρο του δακτύλιου σε n γύρους
 - **Ικανοποιεί ΣΖ:** Κάθε επεξεργαστής μπαίνει σε κατάσταση προεδρίας η μη προεδρίας
- **Χρονική Πολυπλοκότητα:** $O(n * m)$ όπου m ο μικρότερος ΑΤ
 - **ΑΤ δεν είναι φραγμένος από συνάρτηση του n , ανεπιθύμητο**
- **Πολυπλοκότητα Μηνυμάτων:** $O(n)$ – Βέλτιστο
 - Κάθε κόμβος παραλαμβάνει ακριβώς ένα μήνυμα που καθορίζει τον πρόεδρο.

Ομοιόμορφος Αλγόριθμος

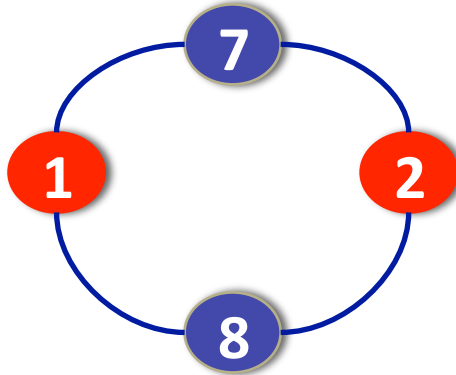
- Πιο χαλαρό υπολογιστικό μοντέλο
 - Οι επεξεργαστές **δεν ξεκινούν όλοι στον ίδιο γύρο**
 - Ξυπνούν τυχαία, ή
 - Ξυπνούν όταν λάβουν μήνυμα
 - Ομοιόμορφος: **δεν βασίζεται στη γνώση του n**
- Τεχνική: Ταχύτητες Μηνυμάτων
 - **Γρήγορα Μηνύματα**
 - Προχωρούν μια ακμή κάθε γύρο
 - **Αργά Μηνύματα**
 - Χρειάζεται περισσότερους από ένα γύρους να προχωρήσουν σε μια ακμή

Ομοιόμορφος Αλγόριθμος

- Αλγόριθμος επεξεργαστή με $AT=\chi$

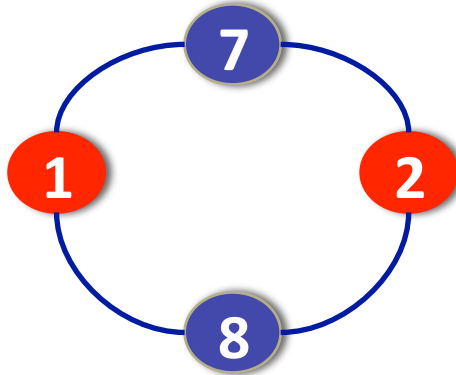
- Αν ξυπνήσεις τυχαία θέσε τον εαυτό σου ως **ενεργό** και στείλε τον AT σου στα δεξιά σου, μέσα σε ένα **γρήγορο** μήνυμα (μια ακμή ανα γύρο).
- Αν ξυπνήσεις όταν λάβεις μήνυμα τότε θέσε τον εαυτό σου ως **προωθητή** και μπες σε κατάσταση μη προεδρίας– **δεν θα γίνεις ποτέ πρόεδρος**
- Αν λάβεις μήνυμα με $AT=\psi$
 - **Αν είσαι ενεργός** κανε το μήνυμα **αργό**
 - Αν ψ **μικρότερο** από όποιο AT παρέλαβες μέχρι στιγμής προώθησε το μήνυμα
 - Αν είναι **αργό** περιμένε 2^ψ γύρους πριν το προωθήσεις.
 - Αν $\chi=\psi$ αυτοανακυρήξου πρόεδρος

Παράδειγμα εκτέλεσης



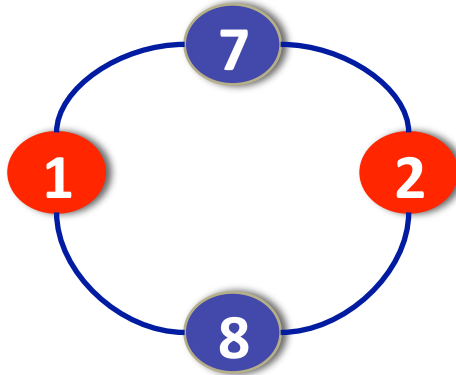
- Υποθέτουμε ότι 1 και 2 είναι ενεργοί και στέλνουν μηνύματα στον πρώτο γύρο
- Γύρος 1:
 - Παραλαβή μηνύματος 1 στον 8
 - Παραλαβή μηνύματος 2 στον 7
 - 8 και 7 προωθούν τα μηνύματα

Παράδειγμα εκτέλεσης



- Γύρος 2:
 - Παραλαβή μηνύματος 1 στον 2 (καθυστέρηση 2^1 γύρους)
 - Παραλαβή μηνύματος 2 στον 1 (δεν θα το προωθήσει)
- Γύρος 3: Τίποτα
- Γύρος 4:
 - Προώθηση μηνύματος 1 από τον 2

Παράδειγμα εκτέλεσης



- Γύρος 5:
 - Παραλαβή μηνύματος 1 στον 7
 - Προώθηση μηνύματος 1 από 7
- Γύρος 6:
 - Παραλαβή μηνύματος 1 από 1 – Ανακύρρηση Προέδρου

Παρατήρηση

- Τι επιτυγχάνω με τις καθυστερήσεις των μηνυμάτων;

Προσπαθώ να μειώσω τα μηνύματα που στέλνονται στον δακτύλιο. Το μήνυμα του Προέδρου είναι το πιο γρήγορο.

- Πολυπλοκότητα αν έστειλα όλα τα μηνύματα: $O(n^2)$

Ανάλυση Αλγορίθμου

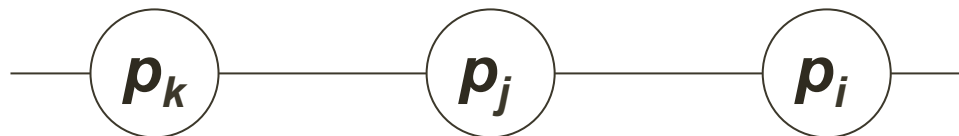
- **Ορθότητα:**
 - **Ικανοποιεί ΣΑ:** Εκλέγει ένα πρόεδρο, αυτόν με τον μικρότερο ΑΤ.
 - **Ικανοποιεί ΣΖ:** Κάθε επεξεργαστής μπαίνει σε κατάσταση προεδρίας η μη προεδρίας
- **Χρονική Πολυπλοκότητα:** $O(n \cdot 2^m)$ όπου m ο μικρότερος ΑΤ
 - Χειρότερος από τον πρώτο αλγόριθμο
 - **ΑΤ δεν είναι φραγμένος από συνάρτηση του n , ανεπιθύμητο**
- **Πολυπλοκότητα Μηνυμάτων:** $O(n)$ – Βέλτιστο

Ανάλυση Μηνυμάτων

- Χωρίζουμε τα μηνύματα σε 3 τύπους:
 - Γρήγορα
 - Αργά που στέλνονται ενόσω το μήνυμα του προέδρου είναι γρήγορο
 - Αργά που στέλνονται ενόσω το μήνυμα του προέδρου είναι αργό
- Πρέπει να μετρήσουμε τα μηνύματα κάθε τύπου

Αριθμός Μηνυμάτων Τύπου 1

- Λήμμα: Κανένας επεξεργαστής δεν προωθεί περισσότερα από ένα γρήγορα μηνύματα.



- Υπόθεσε ότι ο p_i προωθεί τα δύο γρήγορα μηνύματα από τον p_k και p_j .
- Όταν το γρήγορο μήνυμα του p_k φτάνει στον p_j τότε:
 - αν ο p_j είναι ενεργός (έστειλε γρήγορο μήνυμα) τότε καθυστερεί το μήνυμα του p_k
 - αν ο p_j δεν είναι ενεργός τότε προωθεί το γρήγορο μήνυμα από το p_k αλλά ο ίδιος δεν στέλνει γρήγορο μήνυμα
- => Κανένας δεν προωθεί 2 γρήγορα μηνύματα
- => Τύπος 1 μηνύματα το πολύ n

Αριθμός Μηνυμάτων Τύπου 2

- Αργά μηνύματα όταν το μήνυμα του προέδρου είναι γρήγορο
- Χειρότερη περίπτωση:
 - Πρόεδρος AT=0: πάντα γρήγορος
 - Υπόλοιποι AT είναι μικροί: 1 με (n-1)
- Το μήνυμα του προέδρου θα είναι γρήγορο για πολύ n γύρους
 - ως τότε το μήνυμα θα φτάσει πίσω στον πρόεδρο
- Μήνυμα με AT=ψ προωθείται $n/2^\psi$ φορές στους n γύρους

- => Μηνύματα Τύπου 2: $\sum_{i=1}^{n-1} \frac{n}{2^i} \leq n$

Αριθμός Μηνυμάτων Τύπου 3

- Αργά μηνύματα όταν το μήνυμα του προέδρου είναι αργό
- Χειρότερη περίπτωση:
 - Πρόεδρος AT=L: πάντα αργός
 - Υπόλοιποι AT είναι μικροί: L+1 με L+(n-1)
- Το μήνυμα του προέδρου θα είναι αργό για πολύ $n \cdot 2^L$ γύρους
 - ως τότε το μήνυμα θα φτάσει πίσω στον πρόεδρο
 - Δεν στέλνονται μηνύματα μετά από αυτό
- Μήνυμα με AT=ψ προωθείται $n \cdot 2^L / 2^\psi$ φορές στους $n \cdot 2^L$ γύρους

- => Μηνύματα Τύπου 3:
$$\sum_{i=L+1}^{L+n-1} \frac{2^L n}{2^i} \leq 2n$$

Συνολικός Αριθμός Μηνυμάτων

- Τύπος 1 μηνύματα: το πολύ n
- Τύπος 2 μηνύματα: το πολύ n
- Τύπος 3 μηνύματα: το πολύ $2n$
- => Συνολικά μηνύματα το πολύ $4n = O(n)$

Ερωτήσεις;

