

Διάλεξη 8: Πρόβλημα Αμοιβαίου Αποκλεισμού

ΕΠΛ 432: Κατανεμημένοι Αλγόριθμοι



Τι θα δούμε σήμερα

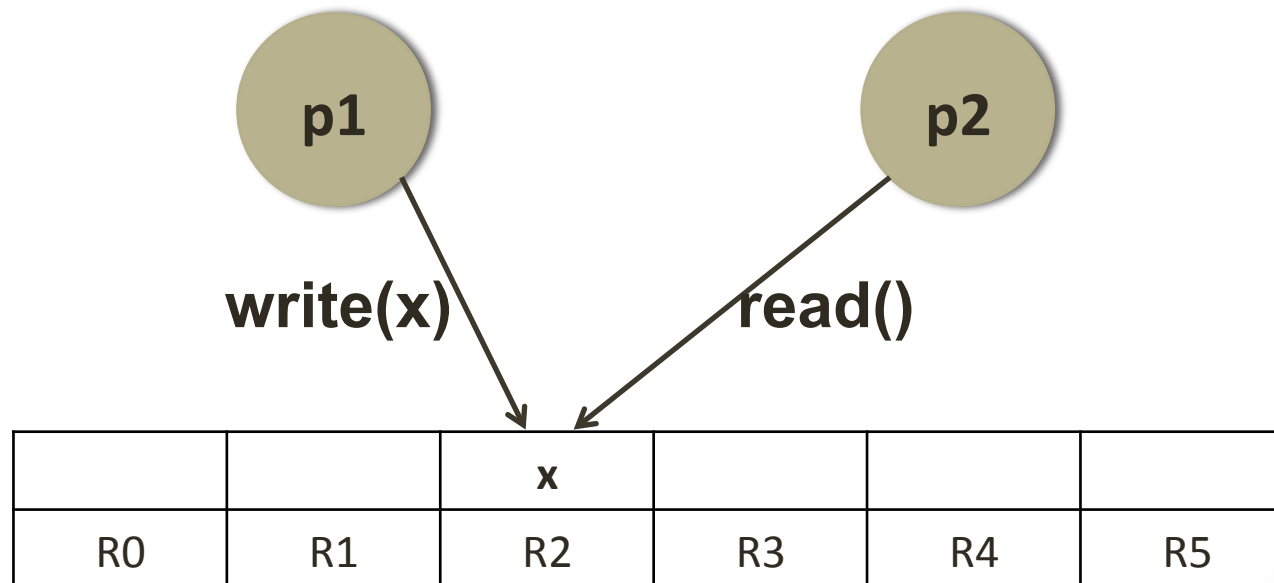
- Μοντέλο Κοινόχρηστης Μνήμης
- Αλγόριθμοι Αμοιβαίου Αποκλεισμού με Ισχυρούς Καταχωρητές

Μοντέλο Κοινόχρηστης Μνήμης

- Το σύστημα περιλαμβάνει
 - Σύνολο από **Επεξεργαστές**
 - Σύνολο από **Κοινόχρηστες Μεταβλητές**
- Οι επεξεργαστές **επικοινωνούν μέσω κοινόχρηστων μεταβλητών**
 - Και όχι μέσω ανταλλαγής μηνυμάτων
- Κάθε κοινόχρηστη μεταβλητή
 - **Χαρακτηρίζεται από ένα τύπο**, ο οποίος
 - προσδιορίζει ένα σύνολο από πράξεις, οι οποίες μπορούν να εκτελεστούν πάνω στην μεταβλητή **ατομικά**.

Παράδειγμα

- Σύστημα: {p1, p2, R0,..., R5}
- Τύπος Μεταβλητών: **Ανάγνωσης/Εγγραφής**
 - Αλλαγή τιμής κατά την εγγραφή
 - Επιστροφή τιμής κατά την ανάγνωση
- Παράδειγμα: ο p1 στέλνει την τιμή “x” στον p2



Διατάξεις

- Μια διάταξη στο μοντέλο κοινόχρηστης μνήμης είναι ένα διάνυσμα της μορφής:

$$C = \{\sigma_0, \dots, \sigma_{n-1}, r_0, \dots, r_{m-1}\}$$

- σ_i : η κατάσταση του επεξεργαστή p_i
 - Δεν περιλαμβάνει εισερχόμενα, εξερχόμενα μηνύματα όπως το μοντέλο μηνυμάτων
- r_i : η τιμή της κοινόχρηστης μεταβλητής R_i
- Αρχική Διάταξη
 - Όλοι οι επεξεργαστές βρίσκονται στην αρχική τους κατάσταση
 - Όλες οι μεταβλητές περιέχουν την αρχική τους τιμή

Γεγονότα

- Τα μοναδικά γεγονότα είναι τα υπολογιστικά βήματα των επεξεργαστών
- Όταν ένας επεξεργαστής p_i εκτελεί ένα υπολογιστικό βήμα τότε τα ακόλουθα εκτελούνται ατομικά:
 - Ο p_i , βασισμένος στην τρέχουσα κατάσταση του, **επιλέγει την κοινόχρηστη μεταβλητή που θα προσπελάσει και την πράξη που θα εκτελέσει** πάνω σε αυτή.
 - **Η τιμή της μεταβλητής αλλάζει** στη νέα διάταξη σύμφωνα με την σημασιολογία της πράξης
 - Η κατάσταση του p_i στην νέα διάταξη καθορίζεται από την **συνάρτηση μετάβασής του**, με βάση την **προηγούμενη του κατάσταση** και το **αποτέλεσμα της πράξης** πάνω στην κοινόχρηστη μεταβλητή.

Εκτελέσεις

- Τμήμα Εκτέλεσης
 - Μια ακολουθία της μορφής

$$\Delta_0, \gamma_1, \Delta_1, \gamma_2, \Delta_2, \gamma_3, \dots$$

όπου Δ_k διατάξεις και γ_k γεγονότα

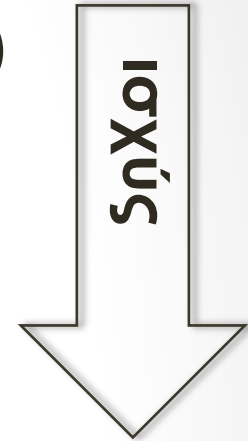
- Νόμιμη Εκτέλεση είναι ένα τμήμα εκτέλεσης το οποίο
 - Ξεκινά με την αρχική διάταξη
 - Εκτελεί ένα άπειρο αριθμό υπολογιστικών βημάτων

Μέτρα Πολυπλοκότητας

- Δεν επικεντρωνόμαστε στα μηνύματα
- Μας ενδιαφέρει περισσότερο η **χωρική πολυπλοκότητα**
 - **Αριθμός Κοινόχρηστων Μεταβλητών** που χρειάζονται
- Παράγοντες που επηρεάζουν την πολυπλοκότητα
 - Πόσο ισχυρός είναι ο τύπος των κοινόχρηστων μεταβλητών που χρησιμοποιούνται;
 - Πόσο ισχυρή είναι η συνθήκη ζωτικότητας που πρέπει να ικανοποιηθεί;

Τύποι μεταβλητών

- Θα δούμε 3 τύπους μεταβλητών:
 - Ανάγνωσης/Μεταβολής/Εγγραφής (Read/Modify/Write)
 - Ελέγχου/Ενημέρωσης (Test&Set)
 - Ανάγνωσης/Εγγραφής (Read/Write)



Μεταβλητή Read/Modify/Write

- Η πράξη read/modify/write επιτρέπει στον επεξεργαστή να κάνει τα ακόλουθα βήματα **ατομικά**
 - Διαβάζει την τιμή κάποιας κοινόχρηστης μεταβλητής
 - Υπολογίζει μια νέα τιμή συναρτήση της τιμής που διάβασε
 - Γράφει τη νέα τιμή στη κοινόχρηστη μεταβλητή
- Υποστηρίζει την ακόλουθη πράξη (ατομικά):
 - $\text{rmw}(V, f)$, where f is any function
 - `temp := V`
 - `V := f(V)`
 - `return temp`

Μεταβλητή Test & Set

- Ειδική περίπτωση της read/modify/write όπου $f(V) = 1$.
 - Διαβάζει την τιμή κάποιας κοινόχρηστης μεταβλητής
 - Γράφει την τιμή 1 στη κοινόχρηστη μεταβλητή
- Υποστηρίζει δύο ατομικές πράξεις
 - `test&set(V)`:

```
temp := V
V := 1
return temp
```
 - `reset(V)`:

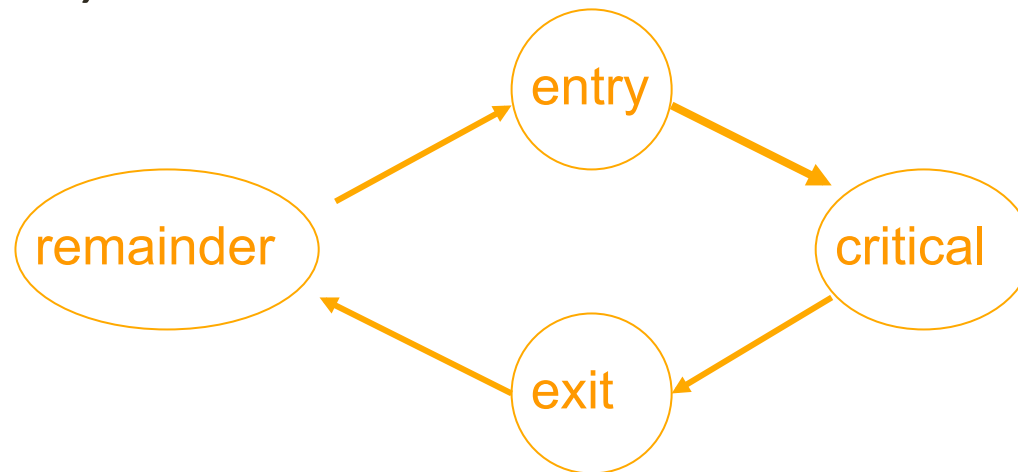
```
V := 0
```

Μεταβλητή Read / Write

- Απλές λειτουργίες Εγγραφής και Ανάγνωσης
- Ανάγνωση: Επιστρέφει την τιμή της κοινόχρηστης μεταβλητής
- Εγγραφή: Αλλάζει την τιμή της κοινόχρηστης μεταβλητής

Το Πρόβλημα του Αμοιβαίου Αποκλεισμού

- Ο κώδικας κάθε επεξεργαστή χωρίζεται σε τέσσερις τομείς:



- **entry**: συγχρονίσου με τους άλλους για να διασφαλιστεί αποκλειστική πρόσβαση στο ...
- **critical**: χρησιμοποίησε τον πόρο και ακολούθως...
- **exit**: καθάρισε τα ίχνη σου και μπες σε κατάσταση όπου...
- **remainder**: δεν σε ενδιαφέρει η χρήση του πόρου

Αλγόριθμοι Αμοιβαίου Αποκλεισμού

- Ένας αλγόριθμος αμοιβαίου αποκλεισμού προσδιορίζει τον κώδικα για τους τομείς εισόδου (entry) και εξόδου (exit) και διασφαλίζει:
 - **Αμοιβαίος Αποκλεισμός** (Συνθήκη Ασφαλείας): Σε κάθε εκτέλεση του αλγορίθμου το πολύ ένας επεξεργαστής βρίσκεται στο κρίσιμο τμήμα σε κάθε διάταξη.
 - Κάποιο είδος συνθήκης ζωτικότητας ή προόδου.

Συνθήκες Ζωτικότητας

- **Όχι Αδιέξοδο**: Αν κάποιος επεξεργαστής βρίσκεται στο τμήμα εισόδου (entry) σε κάποια διάταξη, τότε κάποιος επεξεργαστής θα βρεθεί μέσα στο κρίσιμο τμήμα (critical) σε κάποια μεταγενέστερη διάταξη
- **Όχι Παρατεταμένη Στέρηση**: Αν κάποιος επεξεργαστής βρίσκεται στο τμήμα εισόδου (entry) σε κάποια διάταξη, τότε ο ίδιος επεξεργαστής θα βρεθεί μέσα στο κρίσιμο τμήμα (critical) σε κάποια μεταγενέστερη διάταξη

Αλγόριθμος για μεταβλητή Test & Set

- Code for **entry section**:

```
wait until test&set(V) = 0
```

- Code for **exit section**:

```
reset(V)
```

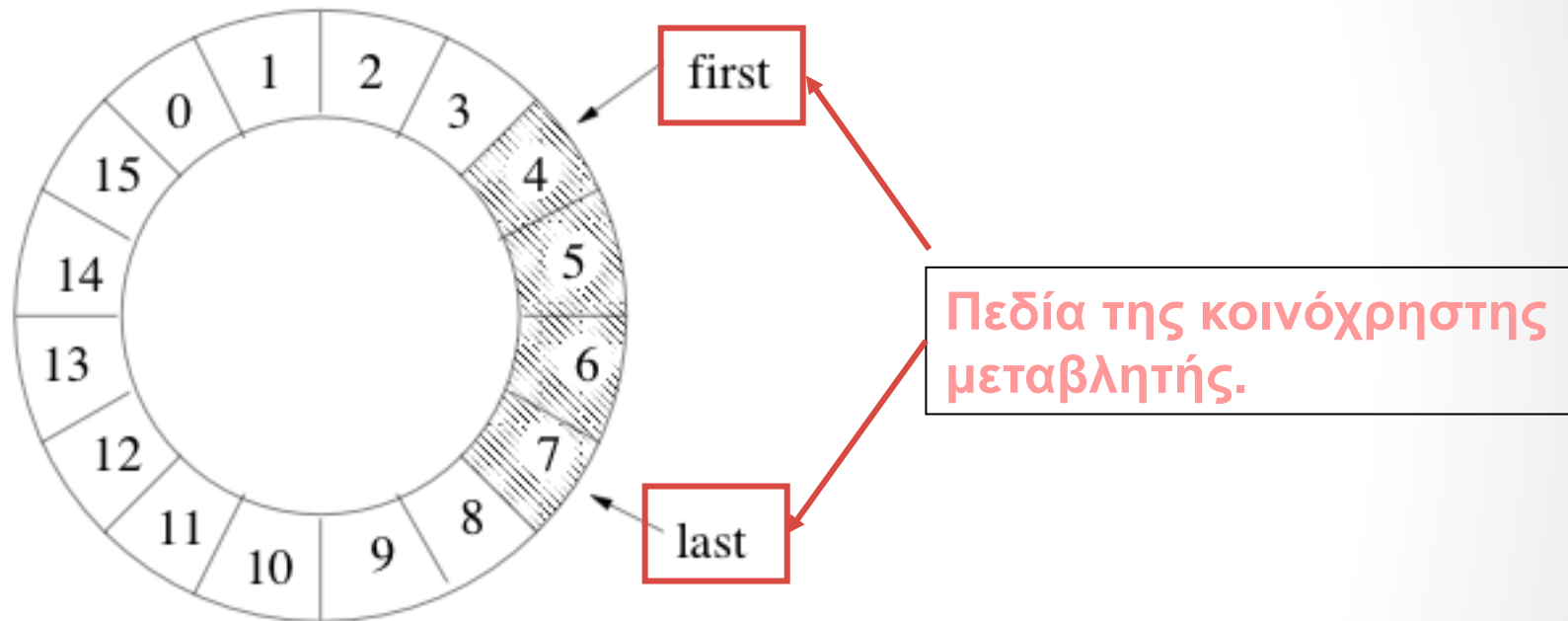

Ορθότητα Αλγορίθμου

- **Αμοιβαίος Αποκλεισμός: Ισχύει**
 - Σκεφτείτε τι θα έπρεπε να ισχύει αν δυο επεξεργαστές βρίσκονταν στο κρίσιμο τμήμα
- **Αποφυγή Αδιεξόδου: Ισχύει**
 - $V=0$ εαν και μόνο εαν κανένας επεξεργαστής δεν βρίσκεται μέσα στο κρίσιμο τμήμα
- **Αποφυγή Παρατεταμένης Στέρησης: Δεν Ισχύει**
 - Κάποιος επεξεργαστής μπορεί να μην καταφέρει να μπει στο κρίσιμο τμήμα έστο κι αν περιμένει στο τμήμα εισόδου.

Αλγόριθμος για μεταβλητή RMW

- Διαισθητικά: Βάλε τους επεξεργαστές που θέλουν να μπουν στο κρίσιμο τμήμα σε μια κυκλική λίστα.
- Η κοινόχρηστη μεταβλητή έχει δύο πεδία
 - **Κορυφή (first)**: δείχνει τον αριθμό του επόμενου κόμβου που θα μπει στο ΚΤ
 - **Ουρά (last)**: δείχνει τον επόμενο διαθέσιμο αριθμό που δεν δεσμεύτηκε από κάποιο κόμβο

Διαισθητική Δομή



Αλγόριθμος για μεταβλητή RMW

- Code for **entry section**:

```
// increment last to enqueue self
position := rmw(V, (V.first, V.last+1))
// wait until first equals this value
repeat
    queue := rmw(V, V)
until (queue.first = position.last)
```

- Code for **exit section**:

```
// dequeue self
rmw(V, (V.first+1, V.last))
```

Παράδειγμα Εκτέλεσης

- Ας υποθέσουμε ότι ο επεξεργαστής $p1$ θέλει να μπει στο ΚΤ
- Η κατάσταση της μεταβλητής:
 - $V.first = 1$
 - $V.last = 3$

0	1	2	3	4	5
---	---	---	---	---	---

- Εκτέλεση: `position := rmw(V, (V.first, V.last+1))`
 - `position.last = 4`
 - $V.first = 1$
 - $V.last = 4$

0	1	2	3	4	5
---	---	---	---	---	---

Παράδειγμα Εκτέλεσης

- Η τοπική μεταβλητή `queue` αλλάζει ως εξής: `queue := rmw(V, V)`
 - `queue.first = 1`
 - `V.first = 1`
 - `V.last = 3`



- Τελειώνει ο πρώτος επεξεργαστής στη σειρά
 - `position.last = 4`
 - `Queue.first = 2`
 - `V.first = 2`
 - `V.last = 4`



Παράδειγμα Εκτέλεσης

- Τελειώνει ο δεύτερος επεξεργαστής στη σειρά
 - `position.last = 4`
 - `queue.first = 3`
 - `V.first = 3`
 - `V.last = 4`

0	1	2	3	4	5
---	---	---	---	---	---

- Τελειώνει ο τρίτος επεξεργαστής στη σειρά
 - `position.last = 4`
 - `queue.first = 4`
 - `V.first = 4`
 - `V.last = 4`

0	1	2	3	4	5
---	---	---	---	---	---

Παράδειγμα Εκτέλεσης

- Τώρα ισχύει: `queue.first = position.last`
 - Ο p1 μπαίνει στο ΚΤ
- Μόλις τελειώσει εκτελεί: `rmw(V, (V.first+1, V.last))`
 - `V.first = 5`
 - `V.last = 4`

0	1	2	3	4	5
---	---	---	---	---	---

Ορθότητα Αλγορίθμου

- **Αμοιβαίος Αποκλεισμός: Ισχύει**
 - Μόνο ένας επεξεργαστής μπαίνει στο κρίσιμο τμήμα
- **Αποφυγή Αδιεξόδου: Ισχύει**
 - Πάντα κάποιος θα μπαίνει στο κρίσιμο τμήμα αφού η διαφορά του $V.last$ και του $V.first$ είναι το πολύ 1 και $V.last \leq V.first$ όταν δεν υπάρχει κανένας στο ΚΤ.
- **Αποφυγή Παρατεταμένης Στέρησης: Ισχύει**
 - Ακολουθούμε σειρά τύπου FIFO

Πολυπλοκότητα

- Η μεταβλητή κρατά δύο πεδία $[0 \dots n-1]$
- Η μεταβλητή μπορεί να βρεθεί σε n^2 διαφορετικές τιμές
- Το μέγεθος της μεταβλητής είναι $2 \log n$ bits

Ερωτήσεις;

